Approximation Algorithm and Applications for Connected Submodular Function Maximization Problems

Ziming Wang, Jing Li, He Xue, Wenzheng Xu, *Member, IEEE*, Weifa Liang, *Senior Member, IEEE*, Zichuan Xu, *Member, IEEE*, Jian Peng, Pan Zhou, *Member, IEEE*, Xiaohua Jia, *Fellow, IEEE*, and Sajal K. Das, *Fellow, IEEE*

***Abstract*— In this paper, we study a connected submodular function maximization problem, which arises from many applications including deploying UAV networks to serve users and placing sensors to cover Points of Interest (PoIs). Specifically, given a budget K, the problem is to find a subset S with K nodes from a graph G, so that a given submodular function $f(S)$ on S is maximized and the induced subgraph $G[S]$ by the nodes in S is connected, where the submodular function f can be used to model many practical application problems, such as the number of users within different service areas of the deployed UAVs in S, the sum of data rates of users served by the UAVs, the number of covered PoIs by placed sensors, etc. We then propose a novel $\frac{1-1/e}{2h+2}$-approximation algorithm for the problem, improving the best approximation ratio $\frac{1-1/e}{2h+3}$ for the problem so far, through estimating a novel upper bound on the problem and designing a smart graph decomposition technique, where e is the base of the natural logarithm, h is a parameter that depends on the problem and its typical value is 2. In addition, when $h = 2$, the algorithm approximation ratio is at least $\frac{1-1/e}{5}$ and may be as large as 1 in some special cases when $K \leq 23$, and is no less than $\frac{1-1/e}{6}$ when $K \geq 24$, compared with the current best approximation ratio $\frac{1-1/e}{7}(=\frac{1-1/e}{2h+3})$ for the problem. Finally, experimental results in the application of deploying a UAV network demonstrate that, the number of users within the service area of the deployed UAV network by the proposed algorithm is up to 7.5% larger than those by existing algorithms, and the throughput of the deployed UAV network by the proposed algorithm is up to 9.7% larger than those by the algorithms. Furthermore, the empirical approximation ratio of the proposed algorithm is between 0.7 and 0.99, which is close to the theoretical maximum value one.**

***Index Terms*— UAV deployment, sensor placement, submodular function maximization, approximation algorithms.**

Received 26 April 2024; revised 30 August 2024; accepted 5 October 2024; approved by IEEE TRANSACTIONS ON NETWORKING Editor S. Magnusson. Date of publication 17 October 2024; date of current version 14 February 2025. The work of Wenzheng Xu was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62272328 and in part by Sichuan Science and Technology Program under Grant 2024NSFJQ0026. The work of Jian Peng was supported in part by the Cooperative Program of Sichuan University and Yibin under Grant 2020CDYB-30, in part by the Cooperative Program of Sichuan University and Zigong under Grant 2022CDZG-6, in part by the Key Research and Development Program of Sichuan Province of China under Grant 22ZDYF3599, and in part by Sichuan Science and Technology Program under Grant 2022ZDZX0011. The work of Sajal K. Das was supported by U.S. NSF under Award ECCS-2319995, Award OAC-2104078, and Award CNS-2030624. *(Ziming Wang and He Xue are co-first authors.) (Corresponding author: Wenzheng Xu.)*

Ziming Wang is with the College of Computer Science and West China Second Hospital, Sichuan University, Chengdu 610065, China (e-mail: wangziming@motherchildren.com).

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

He Xue is with the College of Computer Science, Sichuan University, Chengdu 610065, China, and also with Sichuan Fire Science and Technology Research Institute of MEM, Chengdu 610065, China (e-mail: he.xue@scfri.cn).

Wenzheng Xu and Jian Peng are with the College of Computer Science, Sichuan University, Chengdu 610065, China (e-mail: wenzheng.xu3@gmail.com; jianpeng@scu.edu.cn).

Zichuan Xu is with the School of Software, Dalian University of Technology, Dalian 116621, China (e-mail: z.xu@dlut.edu.cn).

Pan Zhou is with the School of Cyber Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: panzhou@hust.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.2024.3477532, provided by the authors.

Digital Object Identifier 10.1109/TNET.2024.3477532

I. INTRODUCTION

IN THIS paper, we study a connected submodular function maximization problem, which arises from many applications. Before defining the problem, we introduce one of its potential applications: the deployment of a UAV (Unmanned Aerial Vehicle) network in a disaster zone [1], [2], [3], [4], [5], [6], [7], [8], [9]. When a natural disaster (e.g., an earthquake, a flooding, or a mudslide) occurs, communication infrastructures may have been damaged and thus may not work. It is important to provision emergent communication services to rescue teams and people trapped in the disaster area. Multiple UAVs with each equipped with an LTE base station can quickly fly to the disaster area and act as aerial base stations in the air [10], [11], [12], see Fig. 1. Assume that there are only K available UAVs immediately after the disaster, e.g., $K = 10$. Since the disaster area may be very large and there are many people trapped in the area, the limited number K of UAVs may not be able to serve all people. An important problem is how to deploy a connected UAV network that consists of the K UAVs in the air of the disaster area, so as to maximize an important objective function of the deployed UAVs, e.g., the number of users within the service areas of

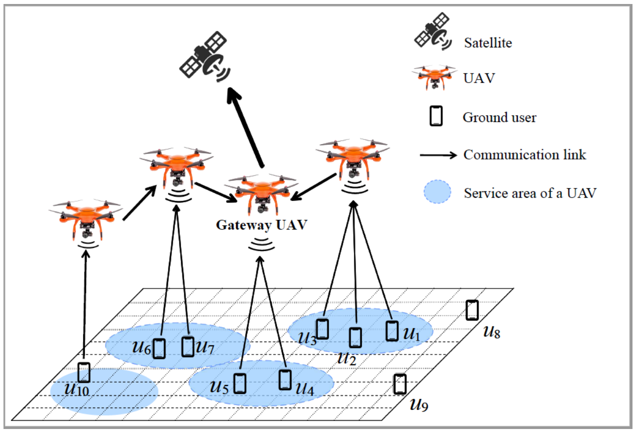
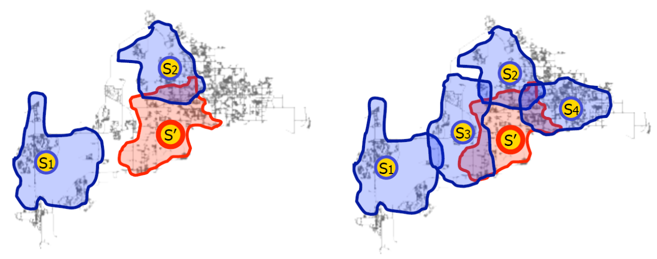


Fig. 1. A UAV network with four UAVs provisions emergent communication services to people trapped in a disaster zone, where the network is connected to the Internet via a satellite.



(a) The additional coverage area by placing sensor s' when sensors s_1 and s_2 have been deployed

(b) The additional coverage area by placing sensor s' when sensors s_1, s_2, s_3, and s_4 have been deployed

Fig. 2. An illustration of the submodular property in the application of placing sensors to detect water contaminations.

the UAVs [1], [2], [3], [7], [9] or the sum of data rates of the users served by the UAVs [8].

There are many other important applications of the connected submodular function maximization problem, such as the placement of sensors to monitor PoIs [13], [14], the scheduling of sensors to cover targets in a duty-cycled sensor network [15], the identification of users to influence others in social networks [15], [16], [17], the deployment of wireless power chargers to charge sensors [18], [19], etc.

Motivated by many important applications of the connected submodular function maximization problem in different fields, in this paper we design a performance-guaranteed approximation algorithm for it. We briefly define the problem as follows. Consider a given undirected graph $G(V, E)$ and a given positive integer K with $K \leq |V|$. For example, in the application of the deployment of a UAV network in a disaster zone, V is the set of potential aerial deployment locations of UAVs, and there is an edge (v_i, v_j) between two locations v_i and v_j in E if their Euclidean distance is no greater than the communication range between the two UAVs in the air. The problem is to find a subset S of V with K nodes so that the value of a given objective function $f(S)$, e.g., the number of users within the service areas of the deployed UAVs in S, is maximized while the induced subgraph $G[S]$ by the nodes in S is connected.

We note that the objective functions f in many applications of the problem usually can be cast as monotone submodular functions [14] that satisfy two important properties: (i) f is monotone. For example, more PoIs will be covered if more sensors are deployed. (ii) f is submodular, which means that the marginal gain of f is decreasing. For example, consider the placement application of sensors to detect water contamination in a drinking water distribution network [14], where each sensor can monitor water contamination within its coverage area. Fig. 2(a) shows the coverage areas of three sensors s_1, s_2 and s', respectively. Fig. 2 illustrates the submodular property, that is, the additional coverage area by placing sensor s' when sensors s_1, s_2, s_3, and s_4 have been deployed (see Fig. 2(b)) is no more than its additional coverage area when only sensors s_1 and s_2 have been deployed (see Fig. 2(a)).

Monotone submodular functions are very powerful tools, because they can be used to model many real problems, even if there are many restrictions in real applications. For example, in the application of deploying a UAV network in a disaster zone, a user may have his/her minimum data rates requirement [8], different users have different extents of importance [2], each UAV has a limited service capacity (i.e., the maximum number of users served by the UAV) [8], [9], etc. On the other hand, in the application of placing sensors [20], [21], the sensing range of a sensor may be directional, the sensing area may be irregular, the quality of sensing decreases with increasing distance away from the sensor, etc.

Submodular functions usually do not have the additivity property, i.e., $f(S) \neq \sum_{v \in S} f(v)$. For example, consider the application of deploying a UAV network. If a user is within the service areas of two UAVs v_1 and v_2 simultaneously, assume that $f(S)$ represents the number of users served by the deployed UAVs in a set S. It can be seen that $f(\{v_1, v_2\}) < f(\{v_1\}) + f(\{v_2\})$. However, if the two UAVs are far away from each other, no users will be within their service areas at the same time. Under this scenario, we have $f(\{v_1, v_2\}) = f(\{v_1\}) + f(\{v_2\})$. Then, submodular functions have a *partial additivity property* in many applications of the problem studied in this paper. Specifically, we assume that $f(A \cup B) = f(A) + f(B)$ for any two subsets A and B of V, if the minimum hops between nodes in sets A and B of graph G is no less than a given positive integer h. The typical value of h is 2, 3, or 4, and $h = 2$ in most cases [2], [3], [13], [22], which will be introduced in Section II. Note that the case with $h \geq 2$ indicates that the Euclidean distance between any UAV in A and any UAV in B is larger than the communication range of the two UAVs.

A. Novelty

The novelty of this paper is that a novel $\frac{1-1/e}{2h+2}$-approximation algorithm for the connected submodular function maximization problem is devised, which improves its best approximation ratio $\frac{1-1/e}{2h+3}$ [23], [24] so far, where e is the base of the natural logarithm, and $h = 2$, 3, or 4. In addition, when $h = 2$ ($h = 2$ in most cases [2], [3], [13], [22]), the algorithm approximation ratio is at least $\frac{1-1/e}{5}$ and may be as large as 1 in some special cases when $K \leq 23$, and is no less than $\frac{1-1/e}{6}$ when $K \geq 24$, compared with the current best approximation ratio $\frac{1-1/e}{7}(=\frac{1-1/e}{2h+3})$ for the problem [23], [24]. For example, when $K \leq 8$, the approximation ratio of the proposed algorithm is at least $\frac{1-1/e}{2}$, which is much larger than the approximation ratio $\frac{1-1/e}{7}$ in [23] and [24].

The techniques used in the proposed approximation algorithm are very different from the one in [23] and [24]. On one hand, if the diameter D of the spanning tree T^* induced by the optimal solution is small (i.e., $D \leq 4h+4$), we estimate a new upper bound on the optimal solution in Section III, which helps us find a $\frac{1-1/e}{2h+2}$-approximate solution when $D \leq 4h+4$. On the other hand, when the diameter D is large with $D > 4h+4$, we design a novel graph decomposition technique in Section IV. Specifically, the existing graph decomposition technique in [23] and [24] considered a tree T that consists of the optimal tree T^* and $K-1$ paths connected to tree T^* (the number of edges in each of the $K-1$ paths is no more than $h-1$), and decomposed tree T into $2h+3$ subtrees by a DFS (Depth-First-Search), such that the number of edges in each subtree is no greater than $\lfloor \frac{K}{2} \rfloor$, where there are some overlapped edges in two of the $2h+3$ subtrees, and the sum of overlapped edges is no greater than $K-1$. **In contrast**, the novel graph decomposition technique proposed in this paper considers a tree T' that consists of the optimal tree T^* and K paths, rather than $K-1$ paths in [23] and [24], that are connected to tree T^* (the number of edges in each of the K paths is still no more than $h-1$), and decomposes tree T' into only $2h+2$ ($< 2h+3$) subtrees while ensuring that the number of edges in each decomposed subtree is no greater than $\lfloor \frac{K}{2} \rfloor$, by first obtaining a graph G' from T' by duplicating edges in the optimal tree T^* but not in the longest path P' in T^*, then finding a path P^* that visits the edges in tree T^* and the duplicated edges, and finally decomposing graph G' along path P^*, where the number of edges in the path P^* is no greater than $2(K-1) - D \leq 2(K-1) - (2h+5)$, as the diameter D is larger than $4h+4$.

B. Contributions

The main contributions of this paper are summarized as follows. We propose an improved $\frac{1-1/e}{2h+2}$-approximation algorithm for the connected submodular function maximization problem, which is larger than its best approximation ratio $\frac{1-1/e}{2h+3}$ [23], [24] so far. We evaluate the performance of the proposed algorithm in the application of deploying a UAV network. Experimental results demonstrate that the number of served users in the solution delivered by the proposed algorithm is up to 7.5% larger than those by existing algorithms, and the network throughput by the proposed algorithm is up to 9.7% larger than those by the algorithms. In addition, its empirical approximation ratio is between 0.7 and 0.99, which is close to the theoretical maximum value one.

Since there are many potential applications of the problem, we think the improvement of the approximation ratio from $\frac{1-1/e}{2h+3}$ to $\frac{1-1/e}{2h+2}$ may have a significant impact in the different applications, e.g., more trapped people are served by the deployed UAV network in a disaster area, and thus more people may be rescued in the end.

The rest of the paper is organized as follows. We introduce preliminaries in Section II. We propose two algorithms for the problem in two distinct cases in Sections III and IV, respectively. We provide a better approximation ratio analysis when $h = 2$ in Section V. We evaluate the algorithm performance in Section VI. We review related studies in Section VII, and conclude the paper in Section VIII.

II. Preliminaries

A. System Model

Consider an undirected graph $G(V, E)$ with node set V and edge set E. G may represent a UAV network, an IoT network, a social network, etc. Assume that G is connected.

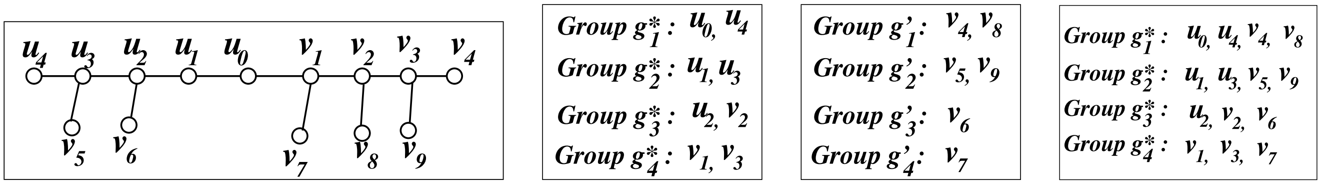
Consider a monotone submodular function $f : 2^V \mapsto \mathbb{Z}^{\geq 0}$ defined on the subsets of V that satisfies three properties [25]: (i) $f(\emptyset) = 0$; (ii) $f(A) \leq f(B)$ for all $A \subseteq B \subseteq V$; and (iii) submodularity: $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$ for all $A, B \subseteq V$. Note that function f may represent the sum of data rates of users served by deployed UAVs [2], [8], the number of covered PoIs in IoT networks [1], the number of influenced users in social networks [16], etc.

Denote by $\mathcal{L}(u, v)$ the minimum number of hops (i.e., the number of edges) in a shortest path between any two nodes u and v in G. Also, denote by $\mathcal{L}(A, B)$ the minimum number of hops in the shortest path between nodes in a set A and nodes in another set B with $\emptyset \neq A, B \subseteq V$, i.e., $\mathcal{L}(A, B) = \min_{u \in A, v \in B}\{\mathcal{L}(u, v)\}$. It can be seen that $\mathcal{L}(A, B) \geq 0$.

B. Partial Additivity Property of Function f

For any two disjoint subsets A and B of V (i.e., $A \cap B = \emptyset$), it can be seen that $f(A \cap B) = 0$ since $f(\emptyset) = 0$. Then, $\mathcal{L}(A, B) \geq 1$. Also, following the submodularity property of function f, we have $f(A) + f(B) \geq f(A \cup B)$. On one hand, the sum of $f(A)$ and $f(B)$ usually is strictly larger than $f(A \cup B)$, i.e., $f(A) + f(B) > f(A \cup B)$, if the minimum number of hops between nodes in sets A and B is small, e.g., $\mathcal{L}(A, B) = 1$. For example, a ground user may be within the coverage areas of two UAVs u and v simultaneously if the two UAVs can communicate with each other (i.e., $\mathcal{L}(u, v) = 1$), where u and v are contained in sets A and B, respectively.

On the other hand, the sum of $f(A)$ and $f(B)$ is equal to $f(A \cup B)$, i.e., $f(A) + f(B) = f(A \cup B)$, if the minimum number of hops between nodes in sets A and B is large. For example, no ground users will be within the coverage areas of UAVs in set A and UAVs in set B at the same time, if the UAVs in A and B are far away from each other, due to the limited coverage range of each UAV. Under this scenario, the number of users served by the UAVs in $A \cup B$ is equal to the sum of the numbers of users served by the UAVs in sets A and B, respectively, i.e., $f(A \cup B) = f(A) + f(B)$. Without loss of generality, assume that $f(A \cup B) = f(A) + f(B)$ for any two subsets A and B of V with the minimum hops between sets A and B being no less than a given positive integer h (e.g., $h = 2$), i.e., $\mathcal{L}(A, B) \geq h$. Note that the case with $h \geq 2$ indicates that the Euclidean distance between any UAV in A and any UAV in B is larger than the communication range of the two UAVs.



(a) A spanning tree T^* of the optimal solution, where $K = 14$ and $D = 8$

(b) Partition the nodes in the longest path except the end node v_4 into $D/2$ groups $g_1^*, g_2^*, \ldots, g_{D/2}^*$

(c) Partition the rest nodes in tree T^* into $D/2$ groups $g_1', g_2', \ldots, g_{D/2}'$

(d) Add nodes in group g_j' to group g_j^* with $1 \le j \le D/2$

Fig. 3. The basic idea of the approximation algorithm with a small diameter in the optimal solution.

C. The Value of h

We briefly discuss the choice of the value h in the application of deploying a UAV network, and the choices of the value of h in other applications were introduced in [23] and [24]. Note that the typical value of h in the various applications is 2, 3, or 4 [23], [24]. Denote by R_{uav} the communication range between two UAVs in the air, and by R_{user} the communication range between a UAV in the air and a ground user. The value of R_{user} usually is no greater than the value of R_{uav}, i.e., $R_{user} \le R_{uav}$, as there are usually much less obstacles between two UAVs in the air, compared with the communication between a UAV in the air and a ground user. Then, the horizontal communication range R'_{user} between the UAV and the ground user is $R'_{user} = \sqrt{R_{user}^2 - H_{uav}^2}$, where H_{uav} is the height of the UAV and $H_{uav} \le R_{user}$. It can be seen that $0 \le R'_{user} \le R_{user} \le R_{uav}$. Following the work in [23] and [24], the value of h is defined as:

$$h = \begin{cases} 2, & \text{if } 0 < R'_{user} \le \dfrac{R_{uav}}{2}, \\ 3, & \text{if } \dfrac{R_{uav}}{2} < R'_{user} \le \dfrac{\sqrt{2}}{2} R_{uav}, \\ 4, & \text{if } \dfrac{\sqrt{2}}{2} R_{uav} < R'_{user} \le R_{uav}. \end{cases} \quad (1)$$

Note that the value of R'_{user} in many applications usually is no more than half of R_{uav}, i.e., $R'_{user} \le \frac{R_{uav}}{2}$ [2], [3], [13], [22]. Then, by Eq. (1), h usually is equal to 2.

D. Problem Definition

We formally define the *connected submodular function maximization problem* as follows. Specifically, given an undirected graph $G(V, E)$, a monotone submodular function $f : 2^V \mapsto \mathbb{Z}^{\ge 0}$ with $f(A) + f(B) = f(A \cup B)$ if the minimum hops between any two subsets A and B of V are no less than a given positive integer h (i.e., $\mathcal{L}(A, B) \ge h$), and a positive integer K (i.e., the budget), the problem is to find a subset S of V with K nodes so that the value of $f(S)$ is maximized, while ensuring that the induced subgraph $G[S]$ is connected, where the value of h may be 2, 3, or 4.

E. A Related Problem

We introduce a related problem – *Quota Steiner Tree (QST) problem* [27]. Given an undirected graph $G(V, E)$ and a positive integer q (i.e., the quota), each node v in V is associated with a profit $p(v)$, the QST problem is to find a subtree T in graph G such that the sum of profits of nodes in T, i.e., $\sum_{v \in T} p(v)$, is no less than the quota q, and the number of edges in T is minimized, where $p(v)$ is a non-negative integer.

Note that there is an approximation algorithm for the QST problem and its approximation ratio is strictly less than 2 [26], [27], which will be used as a subroutine of the proposed algorithm for the problem studied in this paper.

III. Approximation Algorithm With a Small Diameter in the Optimal Solution

Consider an optimal solution S^* to the connected submodular function maximization problem with $|S^*| = K$. Following the definition of the problem, the induced subgraph $G[S^*]$ is connected. Since the cost of each edge in G is one, consider any spanning tree T^* of $G[S^*]$, e.g., Fig. 3(a). Denote by D the diameter of tree T^*, which is the number of edges in the longest shortest path in tree T^* between any two nodes. For example, Fig. 3(a) shows that the diameter D is 8.

In this section, we assume that the diameter D is no more than $4h+4$, i.e., $D \le 4h+4$, where $h = 2$, 3, or 4. Under this assumption, we propose an approximation algorithm for the problem and its approximation ratio is at least $\frac{1-1/e}{\lceil D/2 \rceil} \ge \frac{1-1/e}{2h+2}$, where e is the base of the natural logarithm. On the other hand, if the diameter D is larger than $4h + 4$ (i.e., $D > 4h + 4$), in the next section we will devise a $\frac{1-1/e}{2h+2}$-approximation algorithm for the problem. However, we do not know whether the diameter D is smaller or larger than $4h + 4$. But we do know that the better solution between the two solutions delivered by the algorithms proposed in this and next sections enjoys a performance guarantee of $\frac{1-1/e}{2h+2}$.

A. Basic Idea of the Algorithm

The intuition behind the proposed algorithm is as follows. When the diameter D is small (i.e., $D \le 4h + 4$), we can partition the K nodes in the optimal spanning tree T^* into $D/2$ groups, and each group has roughly $\frac{K}{D/2}$ nodes. Then, the value of one group, say g_j^*, is no less than $\frac{1}{D/2}$ of the optimal solution T^*, i.e., $f(g_j^*) \ge \frac{f(T^*)}{D/2} = \frac{OPT}{D/2} \ge \frac{OPT}{2h+2}$. In addition, we are able to find a set V' with no more than $\frac{K}{D/2}$ nodes, such that the value of V' is no less than $1-1/e$ the value of group g_j^*, by using a greedy strategy, i.e., $f(V') \ge (1-1/e)f(g_j^*) \ge \frac{1-1/e}{2h+2} OPT$. Finally, since each node in V'

is no more than $D/2$ hops from the middle node u_0 in the longest path of tree T^*, we can obtain a spanning tree T with no more than K nodes, by connecting each node in V' to u_0. Therefore, $f(T) \geq f(V') \geq \frac{1-1/e}{2h+2}OPT$, since nodes in V' are contained in T and function f is monotone.

Specifically, denote by P^* the longest path in tree T^*, e.g., the path from nodes u_4 to v_4 in Fig. 3(a). It can be seen that the number of edges in P^* is D, which is the diameter of tree T^*. For the sake of convenience, we assume that D is an even number. Otherwise (D is odd), the discussion is similar by simply replacing $D/2$ with $\lceil D/2 \rceil$ in the following discussion for the case that D is even, omitted. Let u_0 be the middle node in path P^*, see Fig. 3(a). We partition the nodes in tree T^* into $D/2$ groups as follows.

We first partition the nodes in path P^* except its one end node into $D/2$ groups $g_1^*, g_2^*, \ldots, g_{D/2}^*$, such that there are exactly two nodes in each of the $D/2$ groups and the sum of their shortest distances to node u_0 is no greater than $D/2$, i.e., $|g_j^*| = 2$ and $\sum_{v \in g_j^*} \mathcal{L}(v, u_0) \leq D/2$ with $1 \leq j \leq D/2$. For example, Fig. 3(b) shows that the 8 nodes $u_4, u_3, u_2, u_1, u_0, v_1, v_2, v_3$ are partitioned into $4 (= D/2)$ groups, and for nodes in each group, the sum of their shortest distances to node u_0 is no greater than $D/2 = 4$. Notice that we can find the nodes in any group g_j^* by enumerating all combinations, and there are no more than $\binom{n}{3}$ such combinations, where $\binom{n}{3}$ is the number of different ways of choosing 3 nodes from set V with n nodes.

Consider the rest $K - D$ nodes in T^*. We can see that the shortest distance $\mathcal{L}(v, u_0)$ between any of the rest $K-D$ nodes and node u_0 is no more than $D/2$, i.e., $\mathcal{L}(v, u_0) \leq D/2$, where v is in $T \setminus \cup_{j=1}^{D/2} g_j^*$. We partition the $K - D$ rest nodes into $D/2$ groups $g_1', g_2', \ldots, g_{D/2}'$ such that the number of nodes in each of the $D/2$ groups is roughly equal, i.e., $|g_j'| \leq \lceil \frac{K-D}{D/2} \rceil$ with $1 \leq j \leq D/2$, see Fig. 3(c). We finally add the nodes in group g_j' to group g_j^*, i.e., $g_j^* \leftarrow g_j^* \cup g_j'$, where $1 \leq j \leq D/2$, see Fig. 3(d).

We now **estimate a novel upper bound** on the optimal solution to the problem. That is, there is one of the $D/2$ groups, e.g., g_2^* in Fig. 3(d), such that the value of $f(g_2^*)$ is no less than $\frac{1}{D/2}$ the value $f(S^*)$ of tree T^*, i.e., $f(g_2^*) \geq \frac{f(S^*)}{D/2} = \frac{OPT}{D/2}$, where S^* is the set of nodes in tree T^*. Notice that the number of nodes in group g_2^* may be as large as $\lceil \frac{K}{D/2} \rceil < \frac{K}{D/2} + 1$, and each node in g_2^* may be $D/2$ hops away from the middle node u_0.

There is an important property in group g_2^*. That is, there are two nodes in g_2^* such that the sum of their shortest distances to the middle node u_0 is no more than $D/2$, and the shortest distance between each of the rest nodes in g_2^* and node u_0 is no greater than $D/2$. Then, we can find a $(1 - 1/e)$-approximate solution V' for group g_2^* by partial enumerations and greedy searches, while ensuring that set V' still satisfy the property. Then, $f(V') \geq (1 - 1/e)f(g_2^*) \geq \frac{1-1/e}{D/2}OPT$. On the other hand, we can construct a connected subgraph such that the nodes in V' are contained in the subgraph and the number of nodes in the subgraph is no more than the budget K. In the next subsection, we show how to find the set V' and the connected subgraph.

B. Approximation Algorithm With a Small Diameter

The algorithm proceeds as follows. Recall that the diameter D of tree T^* is no more than $4h + 4$, where $h = 2$, 3, or 4. Consider each fixed value of $D = 2, 4, 6, \ldots$, or $4h + 4$. Assume that node v_i is the middle node in path P^*, which can be found by enumerations. Let V_i be the set of nodes within $D/2$ hops from v_i. Note that node v_i itself is contained in V_i. For any two nodes v_j and v_k in V_i with $v_j \neq v_k$, if the sum of their shortest distances to node v_i is no more than $D/2$ (i.e., $\mathcal{L}(v_i, v_j) + \mathcal{L}(v_i, v_k) \leq D/2$), we find a subset S_{ijk} of V_i as follows. Initially, let $S_{ijk} = \{v_i, v_j, v_k\}$. We add nodes to S_{ijk} in a greedy way. Specifically, in each iteration, we add a node $v_l \in V_i \setminus S_{ijk}$ such that the value of $f(S_{ijk} \cup \{v_l\})$ is maximized. Notice that the iterations continue if the sum of shortest distances of nodes in S_{ijk} to v_i is no larger than $K - 1$, i.e., $\sum_{v_l \in S_{ijk}} \mathcal{L}(v_i, v_l) \leq K - 1$. It can be seen that the number of added nodes is at least $\lfloor \frac{K-1-D/2}{D/2} \rfloor$, as each added node may be as far as $D/2$ hops away from v_i. Let V' be the best solution among the $O(n^3)$ solutions S_{ijk}s, i.e., $V' = \arg\max_{S_{ijk}, 1 \leq i,j,k \leq n, j \neq k} \{f(S_{ijk})\}$.

Note that the induced subgraph $G[V']$ may not connected. We construct a connected subgraph such that nodes in V' are contained in the subgraph and the number of nodes in the subgraph is no greater than K as follows.

First, we construct a weighted complete graph $G' = (V', E'; w' : E' \mapsto \mathbb{Z}^{\geq 0})$, where the weight $w'(u, v)$ of each edge (u, v) is the shortest distance between nodes u and v in graph G. We then find a minimum spanning tree T' in G'. For each edge (u, v) in tree T', denote by $P(u, v)$ the corresponding shortest path in G between nodes u and v. We finally construct a graph $G_S = (S, E_S)$ from tree T' and graph G, which is the union of the shortest paths $P(u, v)$s. It can be seen that the number of nodes in G_S is no greater than K, as the sum of shortest distances of nodes in S_{ijk} to v_i is no larger than $K - 1$. The algorithm is described in `Algorithm` 1.

C. Algorithm Analysis

We first show that `Algorithm` 1 delivers a feasible solution. We then analyze its approximation ratio.

Lemma 1: `Algorithm` 1 delivers a feasible solution to the connected submodular function maximization problem.

Proof: The proof is contained in Section I of the supplementary file. ■

We analyze the approximation ratio of `Algorithm` 1.

Theorem 1: Given an undirected graph $G(V, E)$, a positive integer K, a positive integer h(=2, 3, or 4), and a monotone submodular function $f : 2^V \mapsto \mathbb{Z}^{\geq 0}$ such that $f(A) + f(B) = f(A \cup B)$ for all subsets A and B of V with $\mathcal{L}(A, B) \geq h$, assume that the diameter D of the spanning tree T^* induced by the optimal solution is no more than $4h + 4$. There is an approximation algorithm, i.e., `Algorithm` 1, for the connected submodular function maximization problem. Its approximation ratio is $\frac{1-1/e}{\lceil D/2 \rceil} \geq \frac{1-1/e}{2h+2}$, and its time complexity is $O(Kn^4 T_C(f(V)))$, where e is the base of the natural logarithm, and $T_C(f(V))$ is the time complexity of the calculation of $f(V)$.

Algorithm 1 Algorithm `ApproAlgSmall` for the Problem With a Small Diameter

Input: An undirected graph $G(V, E)$, a positive integer K, a positive integer h(=2, 3, or 4), and a monotone submodular function $f : 2^V \mapsto \mathbb{Z}^{\geq 0}$ such that $f(A) + f(B) = f(A \cup B)$ for all subsets A and B of V with $\mathcal{L}(A, B) \geq h$.

Output: A subset S of V with K nodes so that $f(S)$ is maximized, while ensuring that $G[S]$ is connected.

$V' \leftarrow \emptyset$;
for $D = 2, 4, 6, \ldots, 4h + 4$ **do**
  for each node v_i in V **do**
    // Assume that v_i is the middle node in path P^*;
    Let V_i be the set of nodes within $D/2$ hops from v_i;
    for any two nodes v_j and v_k in V_i with $v_j \neq v_k$ **do**
      if $\mathcal{L}(v_i, v_j) + \mathcal{L}(v_i, v_k) \leq D/2$ **then**
        // Ensure that the sum of their shortest distances to v_i is no more than $D/2$;
        Let $S_{ijk} \leftarrow \{v_i, v_j, v_k\}$;
        Let $d_{sum} \leftarrow \mathcal{L}(v_i, v_j) + \mathcal{L}(v_i, v_k)$;
        for $d_{sum} \leq K - 1$ **do**
          Find a node $v_l \in V_i \setminus S_{ijk}$ such that $f(S_{ijk} \cup \{v_l\})$ is maximized;
          if $d_{sum} + \mathcal{L}(v_i, v_l) > K - 1$ **then**
            Break the inner **for** loop;
          end if
          Let $S_{ijk} \leftarrow S_{ijk} \cup \{v_l\}$;
          Let $d_{sum} \leftarrow d_{sum} + \mathcal{L}(v_i, v_l)$;
        end for
        if $f(S_{ijk}) > f(V')$ **then**
          Let $V' \leftarrow S_{ijk}$; // Find a better solution
        end if
      end if
    end for
  end for
end for
Construct a weighted complete graph $G' = (V', E'; w')$, where the weight $w'(u, v)$ of each edge (u, v) is the shortest distance between nodes u and v in graph G;
Find a minimum spanning tree (MST) T' in G';
Construct a graph $G_S = (S, E_S)$ from tree T' and graph G, which is the union of shortest paths $P(u, v)$s, where $P(u, v)$ is the corresponding shortest path in G between nodes u and v for each edge (u, v) in tree T';
return the set S.

Proof: Recall that S^* is the optimal solution to the problem. Denote by OPT the value of S^*, i.e., $OPT = f(S^*)$. Consider a spanning tree T^* in the induced connected subgraph $G[S]$. Following the discussion in Section III-A, we can partition the nodes in tree T^* into $D/2$ groups $g_1^*, g_2^*, \ldots, g_{D/2}^*$, see Fig. 3(c). Then, the value of one group, e.g., g_2^* in Fig. 3(c), is at least $\frac{1}{D/2}$ the value of S^*, i.e., $f(g_2^*) \geq \frac{f(S^*)}{D/2} = \frac{OPT}{D/2}$.

Let v_i be the middle node of the longest path in tree T^*. Following the construction of group g_2^*, there are two nodes in g_2^*, say v_j and v_k, such that the sum of their shortest distances to v_i is no greater than $D/2$. Note that node v_i may or may not be contained in g_2^*. Let $g_2' = g_2^* \cup \{v_i\}$. Then, $f(g_2') \geq f(g_2^*) \geq \frac{OPT}{D/2}$, as f is monotone.

Notice that the number of nodes in g_2' is no more than $\lceil \frac{K-D}{D/2} \rceil + 3$ as $|g_2^*| \leq \lceil \frac{K-D}{D/2} \rceil + 2$. Since $\lceil \frac{K}{D/2} \rceil = \lfloor \frac{K-1}{D/2} \rfloor + 1$ ($D/2$ is a positive integer), we have $|g_2'| \leq \lceil \frac{K-D}{D/2} \rceil + 3 = \lceil \frac{K}{D/2} \rceil + 1 = \lfloor \frac{K-1}{D/2} \rfloor + 2 = \lfloor \frac{K-1-D/2}{D/2} \rfloor + 3$. In addition, each node v_l in $g_2' \setminus \{v_i, v_j, v_k\}$ is no more than $D/2$ hops away from v_i.

On the other hand, consider the subset S_{ijk} found by `Algorithm 1`, where nodes v_i, v_j, v_k are contained in S_{ijk} and the other nodes in S_{ijk} are found in a greedy way. We define function $\phi(S) = f(S \cup \{v_i, v_j, v_k\}) - f(\{v_i, v_j, v_k\})$, for any subset S of V. It can be seen that $\phi(\emptyset) = 0$ and function $\phi(S)$ is monotone and submodular, since function f is. Let $S_1 = S_{ijk} \setminus \{v_i, v_j, v_k\}$ and $S_2 = g_2' \setminus \{v_i, v_j, v_k\}$. Since the number of nodes S_{ijk} is at least $\lfloor \frac{K-1-D/2}{D/2} \rfloor + 3$ and each node v_l in $S_{ijk} \setminus \{v_i, v_j, v_k\}$ is no greater than $D/2$ hops away from v_i, following the study in [25], S_1 is a $(1 - 1/e)$-approximate solution to S_2 in terms of function $\phi(.)$, i.e., $\phi(S_1) \geq (1 - 1/e)\phi(S_2)$. Then,

$$\begin{aligned} f(S_{ijk}) - f(\{v_i, v_j, v_k\}) &= f(S_1 \cup \{v_i, v_j, v_k\}) \\ &\quad - f(\{v_i, v_j, v_k\}) \\ &= \phi(S_1) \\ &\geq (1 - 1/e)\phi(S_2) \\ &= (1 - 1/e)(f(S_2 \cup \{v_i, v_j, v_k\}) \\ &\quad - f(\{v_i, v_j, v_k\})) \\ &= (1 - 1/e)(f(g_2') \\ &\quad - f(\{v_i, v_j, v_k\})). \end{aligned} \tag{2}$$

We thus have

$$\begin{aligned} f(S) &\geq f(V'), \text{ as } V' \subseteq S \text{ and } f \text{ is monotone} \\ &\geq f(S_{ijk}), \quad \text{as } V' = \arg\max_{S_{ijk}} \{f(S_{ijk})\} \\ &= f(S_{ijk}) - f(\{v_i, v_j, v_k\}) + f(\{v_i, v_j, v_k\}) \\ &\geq (1 - 1/e)(f(g_2') - f(\{v_i, v_j, v_k\})) \\ &\quad + f(\{v_i, v_j, v_k\}) \\ &\geq (1 - 1/e) \cdot f(g_2'), \text{ as } 1 - 1/e \leq 1 \\ &\geq (1 - 1/e) \cdot f(g_2^*), \text{ as } f(g_2') \geq f(g_2^*) \\ &\geq \frac{1 - 1/e}{D/2} OPT, \text{ as } f(g_2^*) \geq \frac{OPT}{D/2}. \end{aligned} \tag{3}$$

The analysis of the time complexity is trivial, omitted. The theorem then follows. ■

IV. Approximation Algorithm With a Large Diameter

In the previous section, we assumed that the diameter D of the spanning tree T^* of the induced subgraph by the optimal solution is no more than $4h + 4$, i.e., $D \leq 4h + 4$, and we proposed a $\frac{1-1/e}{\lceil D/2 \rceil}$-approximation algorithm for the connected submodular function maximization problem. In the following, we deal with the case where $D > 4h + 4$, by devising a $\frac{1-1/e}{2h+2}$-approximation algorithm for the case.

A. Algorithm With a Large Diameter

The intuition of the algorithm is as follows. When the diameter is large (i.e., $D > 4h + 4$), an existing study [15] showed that, there is a tree T in G that consists of the optimal spanning tree T^* and K paths that are connected to tree T^* (the number of edges in each of the K paths is no more than

$h-1$), such that the value of T is no less than $1-1/e$ of the optimal tree T^*, i.e., $f(T) \geq (1-1/e)f(T^*) = (1-1/e)OPT$. The best algorithm for the problem so far in [23], [24] decomposed tree T into $2h+3$ subtrees such that the number of edges in each subtree is no greater than $\lfloor \frac{K}{2} \rfloor$. In this paper, we propose a novel tree decomposition technique, which is able to decompose tree T into only $2h+2$ subtrees when $D > 4h+4$, while ensuring that number of edges in each subtree is still no more than $\lfloor \frac{K}{2} \rfloor$. Then, the value of one of the $2h+2$ subtrees, say T', is no less than $\frac{1}{2h+2}$ the value of tree T, i.e., $f(T') \geq \frac{f(T)}{2h+2} \geq \frac{1-1/e}{2h+2}OPT$. In addition, we can find a tree T'' such that the number of edges in T'' is less than twice the number of edges in T', while ensuring that the value of T'' is no less than the value of T', by invoking the 2-approximation algorithm for the Quota Steiner Tree problem [26], [27].

The algorithm proceeds as follows. It first assigns a profit $p(v_i)$ to each node $v_i \in V$ in a greedy way. Specifically, let U be the set of nodes that have been assigned profits and $U = \emptyset$ initially. In the ith iteration, it finds a node v_i in $V \setminus U$ such that the value $f(U \cup \{v_i\})$ is maximized, i.e., $v_i = \arg\max_{v_j \in V \setminus U}\{f(U \cup \{v_i\})\}$. The profit $p(v_i)$ of v_i is assigned as $f(U \cup \{v_i\}) - f(U)$, i.e., $p(v_i) = f(U \cup \{v_i\}) - f(U)$. Since function f is monotone, we know that $f(U \cup \{v_i\}) \geq f(U)$ and $p(v_i) \geq 0$. The iteration continues until each node in V is assigned a profit. Let $v_1, v_2, \ldots, v_n$ be the order of the nodes in V that are assigned profits by the algorithm. Due to the submodularity of function f, we know that $p(v_1) \geq p(v_2) \geq \cdots \geq p(v_n)$.

We will show an important property (see Lemma 1 in Section II of the supplementary file). That is, there is a subtree T' in G such that the sum of profits of nodes in T' is at least $\lceil \frac{1-1/e}{2h+2}OPT \rceil$, and there are no more than $\lceil \frac{K-1}{2} \rceil$ edges in T', where OPT is the value of the optimal solution. Let $Q = \lceil \frac{1-1/e}{2h+2}OPT \rceil$. Following the definition of the QST problem (see Section II-E for its definition), if the quota q in the QST problem is set as Q, the approximation algorithm for the QST problem [26], [27] will find a tree T in G such that the sum of profits of nodes in T is at least $Q(\geq \frac{1-1/e}{2h+2}OPT)$, and the number of edges in tree T is strictly less than $2|E(T')| \leq 2\lceil \frac{K-1}{2} \rceil$, i.e., $|E(T)| < 2\lceil \frac{K-1}{2} \rceil$. Then, $|E(T)| \leq 2\frac{K-1}{2} = K-1$. The number of nodes in T thus is no more than the budget K. Therefore, tree T is a $\frac{1-1/e}{2h+2}$-approximate solution.

We however do not know the value of Q. We can find the value of Q by a binary search as follows. Denote by Q_{lb} and Q_{ub} the lower and upper bounds on Q, respectively. Initially, let $Q_{lb} = 0$ and $Q_{ub} = \sum_{v_i \in V} p(v_i)$. Let $q = \lfloor \frac{Q_{lb}+Q_{ub}}{2} \rfloor$. Denote by T_q^* the optimal tree for the QST problem with quota q. On one hand, if $q \leq Q$, it can be seen that tree T' is a feasible solution to the QST problem with quota q, since the sum of profits in tree T' is no less than the quota q. Then, the number of edges in T' is no less than the number of edges in tree T_q^*, i.e., $|E(T')| \geq |E(T_q^*)|$, as T_q^* is the optimal solution. If we invoke the approximation algorithm [26], [27] for the QST problem with quota q, the algorithm will find a tree T_q in G such that the sum of profits of nodes in T is no less than q, and the number of edges in tree T_q is strictly less than $2|E(T_q^*)| \leq 2|E(T')| \leq 2\lceil \frac{K-1}{2} \rceil$. Then, the number of edges in T_q is no more than $K-1$, i.e., $|E(T_q)| \leq K-1$, and the number of nodes in T_q thus is no more than K. Then, we set the updated lower bound Q_{lb} on Q as q, i.e., $Q_{lb} = q$.

On the other hand, if $q > Q$, consider the tree T_q found by the approximation algorithm [26], [27] for the QST problem with quota q. The number of nodes in T_q may or may not less than K. If the number of nodes in T_q is no more than K, i.e., $|V(T_q)| \leq K$, this indicates that we find a better solution with its value q larger than $Q(\geq \frac{1-1/e}{2h+2}OPT)$. In contrast ($|V(T_q)| > K$), we conclude that $q > Q$. Otherwise ($q \leq Q$), the approximation algorithm [26], [27] for the QST problem with quota q will find a tree with no more than K nodes, while ensuring that the sum of node profits in the tree be at least q. Therefore, when the number of nodes in tree T_q is larger than K (i.e., $|V(T_q)| > K$), we know that q is larger than Q and we set the updated upper bound Q_{ub} on Q as $Q_{ub} = q$.

In summary, in the binary search, if the approximation algorithm [26], [27] for the QST problem with quota q finds a tree T_q with no more than K nodes (i.e., $|V(T_q)| \leq K$), we set $Q_{lb} = q$. Otherwise ($|V(T_q)| > K$), we set $Q_{ub} = q$. The binary search continues until $Q_{lb}+1 = Q_{ub}$. It can be seen that when the binary search stops, the value of Q_{lb} is no less than Q, since $Q_{ub}(= Q_{lb}+1)$ is strictly larger than Q. That is, $Q_{lb} \geq Q \geq \frac{1-1/e}{2h+2}OPT$. In addition, the approximation algorithm [26], [27] for the QST problem with quota Q_{lb} finds a tree with no more than K nodes and the sum of node profits in the tree is at least $Q_{lb} \geq \frac{1-1/e}{2h+2}OPT$, which indicates that the set of nodes in the tree is a $\frac{1-1/e}{2h+2}$-approximate solution.

The algorithm is described in `Algorithm` 2.

B. Algorithm Analysis

We analyze the performance of `Algorithm` 2.

Theorem 2: Given an undirected graph $G(V,E)$, positive integers K and h(=2, 3, or 4), and a monotone submodular function $f: 2^V \mapsto \mathbb{Z}^{\geq 0}$ such that $f(A)+f(B) = f(A \cup B)$ for all subsets A and B of V with $\mathcal{L}(A,B) \geq h$, assume that the diameter D of the spanning tree T^* induced by the optimal solution S^* is larger than $4h+4$. There is an approximation algorithm, i.e., `Algorithm` 2, for the connected submodular function maximization problem, its approximation ratio is $\frac{1-1/e}{2h+2}$, and its time complexity is $O(n^2 T_C(f(V)) + n^3 \log n \log f(V))$, where e is the base of the natural logarithm, and $T_C(f(V))$ is the time complexity of the calculation of $f(V)$.

Proof: The proof is contained in Section II of the supplementary file. ■

V. Better Approximation Ratio Analysis When $h = 2$

Notice that the value of h is equal to 2 in most cases [2], [3], [13], [22], which indicates that the horizontal communication range R'_{user} between a ground user and an aerial UAV is no more than half the communication range R_{uav} between two UAVs in the air in the application of the deployment of UAV networks, the sensing range of a sensor is no more than half the communication range of two sensors in the application of the placement of sensor networks, etc. In this section, we show

Algorithm 2 Algorithm `ApproAlgLarge` for the Connected Submodular Function Maximization Problem With a Large Diameter

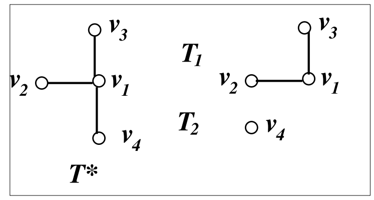
Input: An undirected graph $G(V, E)$, a positive integer K, a positive integer h(=2, 3, or 4), and a monotone submodular function $f : 2^V \mapsto \mathbb{Z}^{\geq 0}$ such that $f(A) + f(B) = f(A \cup B)$ for all subsets A and B of V with $\mathcal{L}(A, B) \geq h$.
Output: A subset S of V with K nodes so that $f(S)$ is maximized, while ensuring that $G[S]$ be connected.

```
/* Assign profits to nodes in V */
U ← ∅;
for 1 ≤ i ≤ n do
    Find a node v_i ∈ V \ U such that f(U ∪ {v_i}) is maximized;
    p(v_i) ← f(U ∪ {v_i}) − f(U);
    U ← U ∪ {v_i};
end for
S ← ∅;
Q_lb ← 0, Q_ub ← Σ_{i=1}^{n} p(v_i); /* Q_lb and Q_ub are the lower and upper bounds on Q, respectively */
while Q_lb + 1 < Q_ub do
    q ← ⌊(Q_lb + Q_ub)/2⌋;
    Find a tree T_q for the QST problem with quota q by invoking the approximation algorithm in [26] and [27];
    if the number of nodes in T_q is no more than K then
        Q_lb ← q; // update the lower bound
        S ← V(T_q); // V(T_q) is the set of nodes in T_q
    else
        Q_ub ← q; // update the upper bound
    end if
end while
return the set S.
```

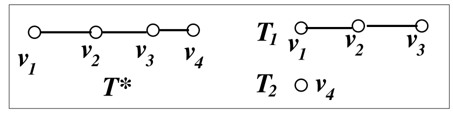
that the better solution between the two solutions delivered by the algorithms in Sections III and IV, respectively, has an approximation ratio better than $\frac{1-1/e}{2h+2} = \frac{1-1/e}{6}$ in some special cases.

Lemma 2: When $h = 2$, the approximation ratio of the better solution between the two solutions delivered by the algorithms in Sections III and IV, respectively, is shown in Eq. (4).

$$Approximation\ ratio = \begin{cases} 1, & 1 \leq K \leq 3 \\ \frac{1}{2}, & 4 \leq K \leq 6 \\ \frac{1-1/e}{2}, & 7 \leq K \leq 8 \\ \frac{1-1/e}{3}, & 9 \leq K \leq 11 \\ \frac{1-1/e}{4}, & 12 \leq K \leq 19 \\ \frac{1-1/e}{5}, & 20 \leq K \leq 23 \\ \frac{1-1/e}{6}, & K \geq 24. \end{cases} \quad (4)$$



(a) The topology of T^* when $D = 2$ and the decomposed subtrees T_1 and T_2



(b) The topology of T^* when $D = 3$ and the decomposed subtrees T_1 and T_2

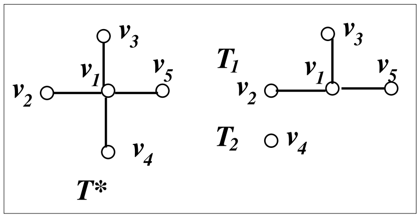
Fig. 4. The two different topologies of tree T^* with $K = 4$.

Proof: The intuition behind algorithm analysis is that, for the special case that $h = 2$ and $K \leq 23$, since the number of nodes K in the optimal tree T^* is not too much, we can consider possible topologies of T^* and decompose T^* into less than $2h + 2$ subtrees.

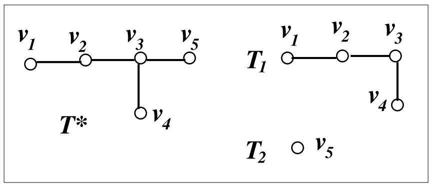
When $K \leq 3$, `Algorithm` 1 finds the optimal solution by an enumeration, and its approximation ratio thus is one.

When $K = 4$, we show that the spanning tree T^* induced by an optimal solution can be decomposed into 2 subtrees T_1 and T_2, and there are no more than 3 nodes in each of the two subtrees. Then, the value of one subtree, e.g., the value $f(T_1)$ of subtree T_1, is no less than $\frac{1}{2}$ the value $f(T^*)$ of tree T^*. In addition, `Algorithm` 1 finds the subtree T_1 by an enumeration. Therefore, the approximation ratio with $K = 4$ is $\frac{1}{2}$. Since $K = 4$, the diameter D of tree T^* may be 2 or 3. Fig. 4 shows the two different topologies of T^* and the decomposed subtrees T_1 and T_2 for each of the two topologies.

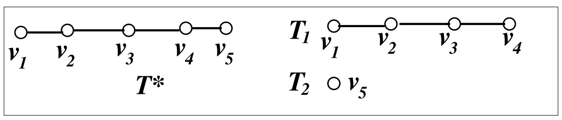
When $K = 5$ or 6, we show that the tree T^* can also be decomposed into 2 subtrees T_1 and T_2, and there are no more than 4 nodes in each of the two subtrees, e.g., Fig. 5 shows the three different topologies of T^* and the decomposed subtrees T_1 and T_2 for each of the three topologies when $K = 5$. Then, the value of one subtree, e.g., the value $f(T_1)$ of subtree T_1, is no less than $\frac{1}{2}$ the value $f(T^*)$ of tree T^*, i.e., $f(T_1) \geq \frac{f(T^*)}{2} = \frac{OPT}{2}$. Since there are only four nodes v_i, v_j, v_k, v_l in T_1, assume that the sum of shortest distances $\mathcal{L}(v_i, v_j)$ and $\mathcal{L}(v_i, v_k)$ is no more than 2, and the shortest distance $\mathcal{L}(v_i, v_l)$ is no more than $\lceil D/2 \rceil$. In addition, `Algorithm` 1 finds a subset S_{ijk} by first enumerating the three nodes v_i, v_j, v_k, i.e., $S_{ijk} = \{v_i, v_j, v_k\}$ initially. The algorithm then finds a node v in $V \setminus S_{ijk}$ and v is no more than $\lceil D/2 \rceil$ hops away from v_i, such that the value of $f(S_{ijk} \cup \{v\})$ is maximized. Since node v_l in tree T_1 is also no more than $\lceil D/2 \rceil$ hops from v_i, $f(T_1) = f(\{v_i, v_j, v_k, v_l\}) = f(S_{ijk} \cup \{v_l\}) \leq f(S_{ijk} \cup \{v\})$. For example, in Fig. 5(b), node v_4 in tree T_1 is no more than $\lceil D/2 \rceil = 2$ hops away from $v_2(= v_i)$. Finally, the algorithm adds node v to S_{ijk}, i.e., $S_{ijk} = S_{ijk} \cup \{v\}$. Then, $f(T_1) \leq f(S_{ijk})$. In addition, the number of nodes in the spanning tree of nodes in S_{ijk} is no greater than $2 + \lceil D/2 \rceil + 1 \leq 2 + \lceil (K-1)/2 \rceil + 1$, which is no more than K when $K = 5$ or



(a) The topology of T^* when $D = 2$ and the decomposed subtrees T_1 and T_2



(b) The topology of T^* when $D = 3$ and the decomposed subtrees T_1 and T_2



(c) The topology of T^* when $D = 4$ and the decomposed subtrees T_1 and T_2

Fig. 5. The three different topologies of tree T^* with $K = 5$.

6. On the other hand, the tree decomposition when $K = 6$ is similar, omitted, and there are 6 different topologies of T^*.

The proof for the approximation ratio with $7 \leq K \leq 23$ is contained in Section III of the supplementary file.

Finally, the approximation ratio is $\frac{1-1/e}{6}$ when $K \geq 24$, by Theorems 1 and 2. ∎

VI. Performance Evaluation

A. Experimental Environment

We consider an important application of the connected submodular function maximization problem, which is to deploy a connected UAV network with K UAVs above an area to serve as many ground users as possible [9]. We consider two different shapes of the area: square and strip. A square area may be a disaster area, e.g., an earthquake occurred in the area, and the UAV network is deployed for providing emergent communication to trapped people and rescue teams. The length of the square is 3 km [9]. On the other hand, a strip area may be the boundary of a country and a UAV network is used for military communications. The length and width of the strip area are 10 km and 1 km, respectively. The number of users in the area varies from 1,000 to 5,000, and the user density in the area subjects to the fat-tailed distribution, which implies that most users are clustered at a few places whereas a few users are scattered at other places [28], [29].

There are from 20 to 50 UAVs deployed in the area. The UAV communication range R_{uav} is 600 m, while the communication range R_{user} between a ground user and a UAV in the air is 500 m [9], assuming that the UAV hovering height H_{uav} is 300 m [10]. The number of served users by each UAV is no more than 100 users, due to the constraints on the size, weight, and power supply of the UAV [8], [11], [12].

We adopt real wireless channel models and parameters in the models are similar to the ones in [8], [9], and [10], which are omitted here due to space limitation.

Recall that the proposed algorithms `ApproAlgSmall` and `ApproAlgLarge` solve the problem in the two cases where the diameter D in the optimal solution is no more than $4h + 4$ or larger than $4h + 4$, respectively. Denote by `ApproAlg` the better solution between the two solutions delivered by the two algorithms, respectively.

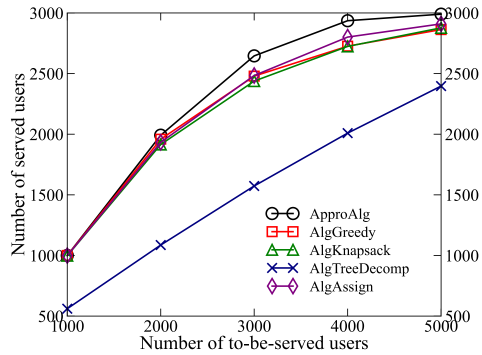
We consider two widely used monotone submodular functions f. The first is the number of served ground users by deployed UAVs [5], [6], and the other is the network throughput, i.e., the sum of user data rates, in the UAV network [8].

In addition to the proposed algorithm `ApproAlg`, we also consider the following four benchmark algorithms. (i) Algorithm `AlgGreedy` [1] delivers a greedy heuristic solution. (ii) Algorithm `AlgKnapsack` [8] finds a $\frac{1-1/e}{\sqrt{K}}$-approximate solution, by reducing the problem to the submodular maximization problem with a knapsack constraint. (iii) Algorithm `AlgTreeDecomp` [2], [15] finds a $\frac{1-1/e}{4h}$-approximate solution, by decomposing a tree with $2hK$ nodes so that the profit sum of nodes in the tree is no less than $(1-1/e)OPT$, into $4h$ subtrees with the number of nodes in each subtree no greater than K, and choosing the best subtree among the $4h$ subtrees. (iv) Algorithm `AlgAssign` finds a $\frac{1-1/e}{2h+3}$-approximate solution [23], [24].

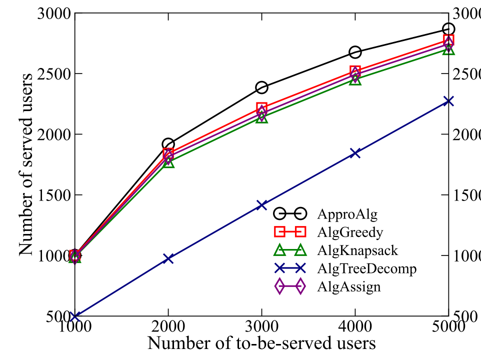
All experiments were performed on a server with an Intel(R) Core(TM) i9-9900K CPU (3.6 GHz) and 32 GB RAM.

B. Algorithm Performance

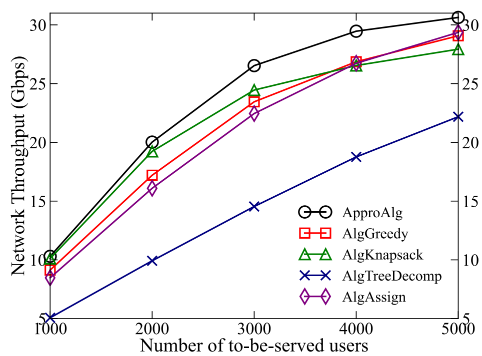
We first study the algorithm performance, by varying the number m of to-be-served ground users from 1,000 to 5,000, when there are $K = 30$ UAVs. Fig. 6(a) shows that the numbers of served users by the fours algorithms `ApproAlg`, `AlgGreedy`, `AlgKnapsack` and `AlgAssign` are close to the number of to-be-served users when $m = 1,000$, while the number of served users by the proposed algorithm `ApproAlg` is from 2.7% to 6.5% larger than those by the other four algorithms. For example, the average numbers of served users by the five algorithms `ApproAlg`, `AlgGreedy`, `AlgKnapsack`, `AlgTreeDecomp` and `AlgAssign` are 2,646, 2,479, 2,438, 1,573, 2,484, respectively, when there are $m = 3,000$ to-be-served users in the network. Notice that the maximum number of served users by any algorithm is no more than $OPT_{ub} = \min\{m,\ 100K\}$, where m is the number of to-be-served users, K is the number of UAVs, and each UAV can serve no more than 100 users. The empirical approximation ratio of an algorithm then is the ratio of the number of users served by the algorithm to the upper bound OPT_{ub}. Fig. 6(a) demonstrates that the empirical approximation ratio of the proposed algorithm `ApproAlg` is between 0.88 and 0.99, which is close to the theoretical maximum value one. Fig. 6(b) plots the algorithm performance in the strip area, which demonstrates similar curves as those in Fig. 6(a). However, it can be seen from Fig. 6(a) and Fig. 6(b) that the number of served users by each of the five algorithms in the strip



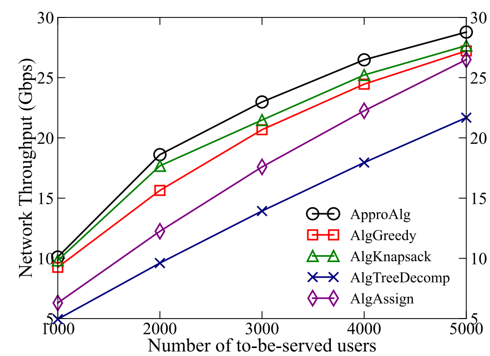
(a) The *number of served users* by different algorithms in the *square* area



(b) The *number of served users* by different algorithms in the *strip* area



(c) The *network throughput* by different algorithms in the *square* area



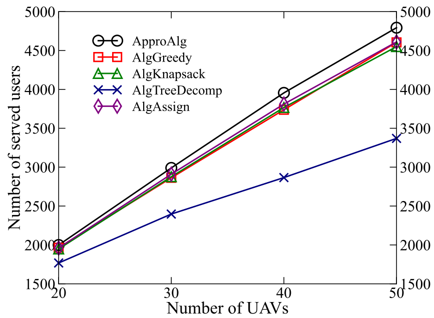
(d) The *network throughput* by different algorithms in the *strip* area

Fig. 6. The algorithm performance by varying the number of to-be-served users from 1,000 to 5,000, when there are $K = 30$ UAVs.

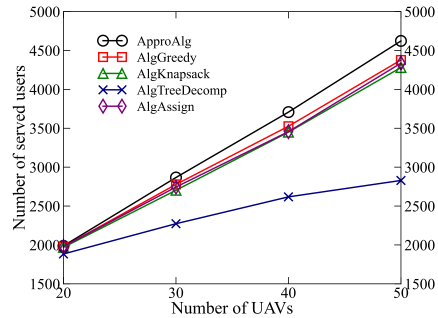
area is less than the number in the square area. For example, the number of served users by algorithm `ApproAlg` in the strip area is 2,867, while the number in the square area is 2,990, when there are 5,000 to-be-served users in the network. The rationale behind the phenomenon is the average distance between different users in the strip area is larger than that in the square area. For example, the largest distance between users in the strip area is $\sqrt{10^2 + 1^2} = 10.04$ km, while the largest distance between users in the square area is only $\sqrt{3^2 + 3^2} = 4.24$ km. Then, more UAVs need to act as relays in the strip area, and less UAVs thus can be used to serve users. Similarly, Fig. 6(c) shows that the network throughput in the solution delivered by algorithm `ApproAlg` is up to 9.7% larger than those by the other four algorithms in the square area, while Fig. 6(d) demonstrates that the throughput by algorithm `ApproAlg` is from 2.8% to 7% larger than those by the other four algorithms in the strip area.

We then evaluate the performance of different algorithms by varying the number K of UAVs from 20 to 50, when there are $m = 5,000$ to-be-served users. Fig. 7(a) shows that the number of users served in the solution delivered by algorithm `ApproAlg` is from 2% to 3.7% larger than those by the other four algorithms in the square area, while the number by algorithm `ApproAlg` is up to 7.2% larger than those by the other four algorithms in the strip area. In addition, Fig. 7(c) and (d) plot that the network throughput of the solution delivered by algorithm `ApproAlg` is up to 7.3% larger than those by the other four algorithms in the square area, whereas the throughput by algorithm `ApproAlg` is up to 6.9% larger than those by the other four algorithms in the strip area.

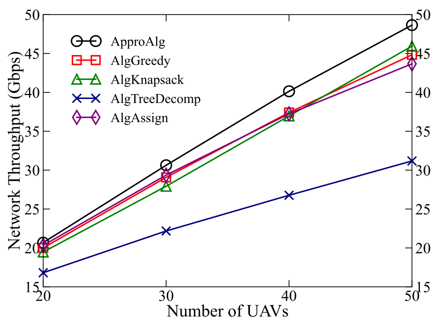
We also investigate the algorithm performance by varying the communication range R_{user} between a ground user and an aerial UAV from 400 m to 800 m, when there are $K = 30$ UAVs, $m = 3,000$ to-be-served users in the network, and R_{uav} =800 m. Notice that when the user communication range R_{user} increases from 400 m to 800 m, the horizontal communication range R'_{user} between a ground user and a UAV increases from 264.5 m to 741.6 m, where $R'_{user} = \sqrt{R^2_{user} - H^2_{uav}}$ and $H_{uav} = 300$ m. Following Eq. (1), the value of h increases from 2 to 4, when R'_{user} increases from 264.5 m to 741.6 m. Specifically, $h = 2$ if R'_{user} is between 264.5 m and $\frac{R_{uav}}{2} = 400\ m$, $h = 3$ if R'_{user} is larger than 400 m but no more than $\frac{\sqrt{2}}{2}R_{uav} = 565.7\ m$, and $h = 4$ if R'_{user} is larger than 565.7 m but no more than 741.6 m. Fig. 8(a) demonstrates that the number of served users by each of the five algorithms increases with the growth of the user communication range R_{user}, as a UAV is able to serve more users with a larger communication range R_{user}. Fig. 8(a) and (b) show that the number of served users by algorithm `ApproAlg` is from 2% to 7.5% larger than those by the other four algorithms. In addition, Fig. 8 (a) and (b) demonstrate that the empirical approximation ratio of the proposed algorithm `ApproAlg` is between 0.7 and



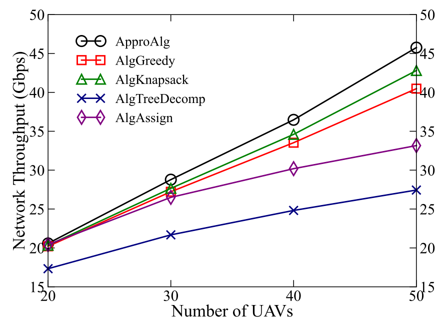
(a) The *number of served users* by different algorithms in the *square* area



(b) The *number of served users* by different algorithms in the *strip* area

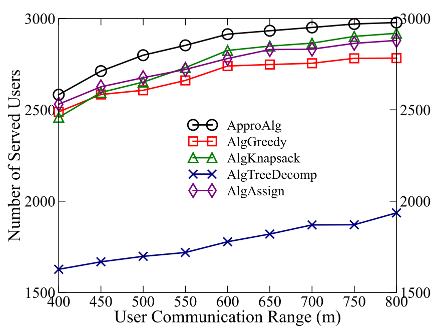


(c) The *network throughput* by different algorithms in the *square* area

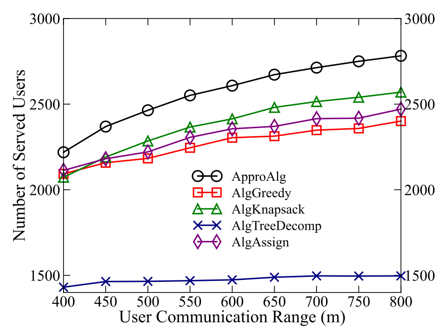


(d) The *network throughput* by different algorithms in the *strip* area

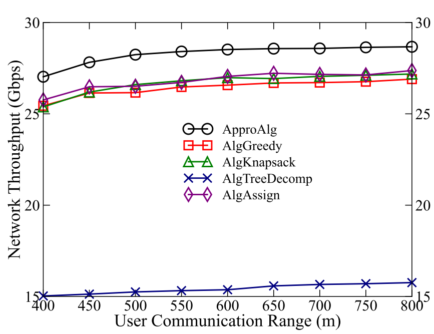
Fig. 7. The algorithm performance by varying the number of UAVs from 20 to 50, when there are $m = 5,000$ to-be-served users.



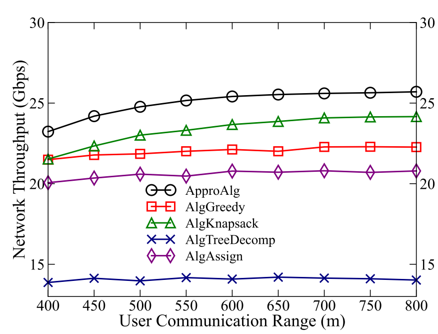
(a) The *number of served users* by different algorithms in the *square* area



(b) The *number of served users* by different algorithms in the *strip* area



(c) The *network throughput* by different algorithms in the *square* area

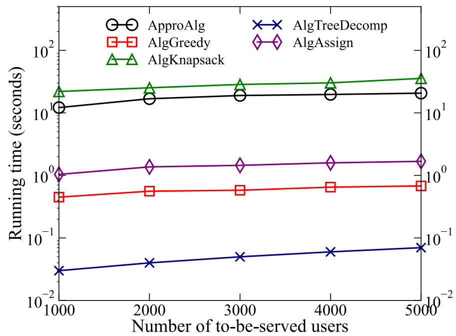


(d) The *network throughput* by different algorithms in the *strip* area

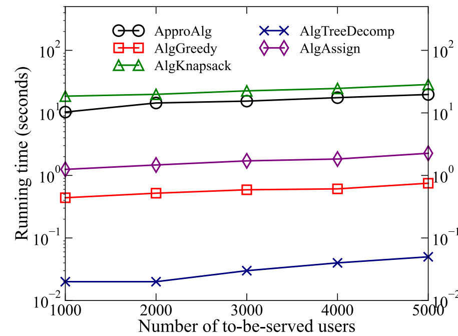
Fig. 8. The algorithm performance by varying the communication range R_{user} between a ground user and an aerial UAV from 400 m to 800 m, when there are $K = 30$ UAVs and $m = 3,000$ to-be-served users.

0.99. Furthermore, Fig. 8(c) and (d) show that the network throughput by algorithm `ApproAlg` is from 4.7% to 8% larger than those by the other four algorithms.

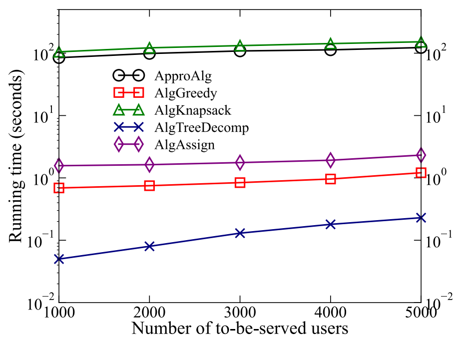
We finally evaluate the running time of different algorithms by varying the number m of to-be-served ground users from 1,000 to 5,000, when there are $K = 30$ UAVs and the R_{user}



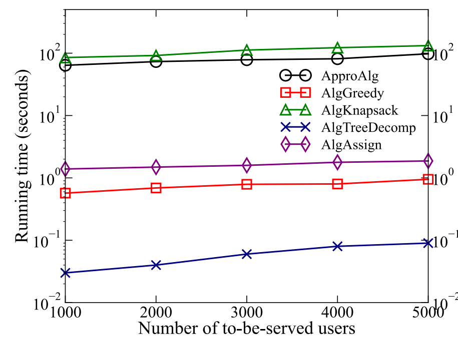
(a) The running time of different algorithms in the *square* area, and function f is the *number of served users*



(b) The running time of different algorithms in the *strip* area, and function f is the *number of served users*



(c) The running time of different algorithms in the *square* area, and function f is the *network throughput*



(d) The running time of different algorithms in the *strip* area, and function f is the *network throughput*

Fig. 9. The algorithm running time by varying the number m of to-be-served ground users from 1,000 to 5,000, when there are $K = 30$ UAVs and R_{user} is 500 m.

is 500 m. Fig. 9 (a) and (b) shows that the running time of the proposed algorithm ApproAlg is no greater than 20 seconds when the objective function f is to maximize the number of served users, and the running time is no more than 2 min when the objective function f is to maximize the network size. Such a computation delay is acceptable in real applications of deploying UAV networks.

VII. Related Work

The connected submodular function maximization problem studied in this paper has drawn many attentions recently [13], [15], [18], [23], [24], [30], [31]. For example, Khuller et al. [15], [30] conducted pioneering studies on the problem by proposing a $\frac{1-1/e}{12}$-approximation algorithm when $h = 3$. Specifically, they investigated a problem of finding a set S with K nodes in a graph to maximize the number of nodes dominated by the node set S, while ensuring that $G[S]$ is a connected subgraph, where a node v is dominated by S if it is contained in S or one of its neighbors is contained in S. Lamprou et al. [31] improved the approximation ratio $\frac{1-1/e}{12}(\approx 0.0527)$ in [15] and [30] to $\frac{1-(1/e)^{\frac{7}{8}}}{11}(\approx 0.053)$. Huang et al. [13] studied a problem of choosing K sensors in a sensor network to cover the maximum number of targets by the chosen sensors and the communication graph of the chosen sensors is connected. Yu et al. [18] considered a problem of placing K wireless chargers to maximize the overall charging utility, under the connectivity constraint for wireless chargers. Notice that both the approximation ratios of the algorithms proposed in [13] and [18] decrease from $\frac{1-1/e}{32}$ to $\frac{1-1/e}{128}$ when the value of h increases from 2 to 4. Xu et al. [23], [24] recently proposed a $\frac{1-1/e}{2h+3}$-approximation algorithm, and it can be seen that the approximation ratio $\frac{1-1/e}{2h+3}$ decreases from $\frac{1-1/e}{7}$ to $\frac{1-1/e}{11}$ when h increases from 2 to 4. In contrast, the approximation ratio $\frac{1-1/e}{2h+2}$ of the algorithm proposed in this paper is larger than the ratio $\frac{1-1/e}{2h+3}$ in [23] and [24], by finding a novel upper bound on the optimal solution of the problem and devising a new graph decomposition technique.

Other researchers paid attentions to the case where the value of h may be very large as well. For example, Kuo et al. [32] proposed a $\frac{1-1/e}{5(\sqrt{K}+1)}$-approximation algorithm for the problem, under the application scenario of the deployment of K wireless routers in a wireless network. Xu et al. [8] recently studied the similar problem [32], and improved the approximation ratio to $\frac{1-1/e}{\sqrt{K}}$, by reducing the problem to a submodular function maximization problem (without the connectivity constraint) subject to a knapsack constraint [33]. However, the time complexity of the algorithm in [8] is very high, which indicates that the algorithm is applicable to only small- or medium-scale graphs. In addition, the approximation ratio $\frac{1-1/e}{\sqrt{K}}$ is small when K is large.

UAV network deployments have gained lots of attentions [1], [2], [3], [4], [7], [9]. Coletta et al. [1] proposed a greedy heuristic for deploying UAVs to cover the maximum number of targets. Danilchenko et al. [2], [3] adopted the algorithm in [15] to deploy a connected UAV network in the air to cover the maximum number of ground users, where each user has his/her minimum SNR requirement. Huang et al. [4] studied a problem of minimizing the average

UAV-user distance under LoS scenario, while maintaining the connectivity of the deployed UAV network. Zhao et al. [9] devised a motion control algorithm for deploying UAVs to serve as many as users under the network connectivity constraint. Liu et al. [7] studied the similar problem and devised an algorithm based on deep reinforcement learning.

VIII. Conclusion

In this paper, we studied the connected submodular function maximization problem, which has many applications, such as deploying a UAV network to serve users and placing sensors to monitor PoIs. We proposed a novel $\frac{1-1/e}{2h+2}$-approximation algorithm for the problem, improving its best approximation ratio $\frac{1-1/e}{2h+3}$ so far, through estimating a novel upper bound on the problem and designing a smart graph decomposition technique, where h is a parameter that depends on the problem and its typical value is 2. We finally evaluated the algorithm performance in the application of deploying a UAV network, and experimental results demonstrate that the number of users within the service area of the deployed UAV network by the proposed algorithm is up to 7.5% larger than those by existing algorithms, and the network throughput by the proposed algorithm is up to 9.7% larger than those by the algorithms.

References

[1] A. Coletta et al., "A^2-UAV: Application-aware content and network optimization of edge-assisted UAV systems," in *Proc. IEEE Conf. Comput. Commun. (IEEE INFOCOM)*, May 2023, pp. 1–10.

[2] K. Danilchenko, Z. Nutov, and M. Segal, "Covering users with QoS by a connected swarm of drones: Graph theoretical approach and experiments," *IEEE/ACM Trans. Netw.*, vol. 31, no. 6, pp. 2483–2498, Dec. 2023.

[3] K. Danilchenko, Z. Nutov, and M. Segal, "Doing their best: How to provide service by limited number of drones?" *Wireless Netw.*, vol. 29, no. 1, pp. 209–220, Jan. 2023.

[4] H. Huang and A. V. Savkin, "Deployment of heterogeneous UAV base stations for optimal quality of coverage," *IEEE Internet Things J.*, vol. 9, no. 17, pp. 16429–16437, Sep. 2022.

[5] S. Li et al., "Coverage maximization of heterogeneous UAV networks," in *Proc. IEEE 43rd Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2023, pp. 120–130.

[6] S. Li et al., "Maximizing network throughput in heterogeneous UAV networks," *IEEE/ACM Trans. Netw.*, vol. 32, no. 3, pp. 2128–2142, Jun. 2024.

[7] C. H. Liu et al., "Energy-efficient UAV control for effective and fair communication coverage: A deep reinforcement learning approach," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2059–2070, Sep. 2018.

[8] W. Xu et al., "Throughput maximization of UAV networks," *IEEE/ACM Trans. Netw.*, vol. 30, no. 2, pp. 881–895, Apr. 2022.

[9] 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.

[10] 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.

[11] X. Cao, P. Yang, M. Alzenad, X. Xi, D. Wu, and H. Yanikomeroglu, "Airborne communication networks: A survey," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 10, pp. 1907–1926, Sep. 2018.

[12] M. Moradi, K. Sundaresan, E. Chai, S. Rangarajan, and Z. M. Mao, "SkyCore: Moving core to the edge for untethered and reliable UAV-based LTE networks," in *Proc. Annu. Int. Conf. Mobile Comput. Netw.*, 2018, pp. 35–49.

[13] L. Huang, J. Li, and Q. Shi, "Approximation algorithms for the connected sensor cover problem," *Theor. Comput. Sci.*, vol. 809, pp. 563–574, Feb. 2020.

[14] A. Krause and D. Golovin, "Submodular function maximization," *Tractability*, vol. 3, pp. 71–104, Feb. 2014.

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

[16] K. Avrachenkov, P. Basu, G. Neglia, B. Ribeiro, and D. Towsley, "Online myopic network covering," 2012, *arXiv:1212.5035*.

[17] W. Xu, M. Rezvani, W. Liang, J. X. Yu, and C. Liu, "Efficient algorithms for the identification of top-k structural hole spanners in large social networks," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 5, pp. 1017–1030, May 2017.

[18] N. Yu, H. Dai, G. Chen, A. X. Liu, B. Tian, and T. He, "Connectivity-constrained placement of wireless chargers," *IEEE Trans. Mobile Comput.*, vol. 20, no. 3, pp. 909–927, Mar. 2021.

[19] P. Zhou, C. Wang, and Y. Yang, "Design of self-sustainable wireless sensor networks with energy harvesting and wireless charging," *ACM Trans. Sensor Netw.*, vol. 17, no. 4, pp. 1–38, Nov. 2021.

[20] R. Elhabyan, W. Shi, and M. St-Hilaire, "Coverage protocols for wireless sensor networks: Review and future directions," *J. Commun. Netw.*, vol. 21, no. 1, pp. 45–60, Feb. 2019.

[21] B. Wang, "Coverage problems in sensor networks: A survey," *ACM Comput. Surv.*, vol. 43, no. 4, p. 32, 2011.

[22] A. Srinivas, G. Zussman, and E. Modiano, "Construction and maintenance of wireless mobile backbone networks," *IEEE/ACM Trans. Netw.*, vol. 17, no. 1, pp. 239–252, Feb. 2009.

[23] W. Xu et al., "Maximizing h-hop independently submodular functions under connectivity constraint," in *Proc. IEEE Conf. Comput. Commun. (IEEE INFOCOM)*, May 2022, pp. 1099–1108.

[24] W. Xu et al., "An approximation algorithm for the h-hop independently submodular maximization problem and its applications," *IEEE/ACM Trans. Netw.*, vol. 31, no. 3, pp. 1216–1229, Mar. 2023.

[25] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions-I," *Math. Program.*, vol. 4, no. 1, pp. 265–294, 1978.

[26] N. Garg, "Saving an epsilon: A 2-approximation for the k-MST problem in graphs," in *Proc. 37th Annu. ACM Symp. Theory Comput.*, May 2005, pp. 396–402.

[27] D. S. Johnson, M. Minkoff, and S. Phillips, "The prize collecting steiner tree problem: Theory and practice," in *Proc. 11th Annu. ACM-SIAM Symp. Discrete Algorithms*, 2000, pp. 760–769.

[28] C.-C. L. Bhola, A.-H. Tsai, and L.-C. Wang, "Adaptive and fair deployment approach to balance offload traffic in multi-UAV cellular networks," *IEEE Trans. Veh. Technol.*, vol. 72, no. 3, pp. 3724–3738, Mar. 2023.

[29] C. Song, T. Koren, P. Wang, and A. L. Barabasi, "Modelling the scaling properties of human mobility," *Nat. Phys.*, vol. 6, no. 10, pp. 818–823, 2010.

[30] S. Khuller, M. Purohit, and K. K. Sarpatwar, "Analyzing the optimal neighborhood: Algorithms for budgeted and partial connected dominating set problems," in *Proc. 25th Annu. ACM-SIAM Symp. Discrete Algorithms (SODA)*, Jan. 2014, pp. 1702–1713.

[31] I. Lamprou, I. Sigalas, and V. Zissimopoulos, "Improved budgeted connected domination and budgeted edge-vertex domination," *Theor. Comput. Sci.*, vol. 858, pp. 1–12, Feb. 2021.

[32] T.-W. Kuo, K. C. Lin, and M.-J. Tsai, "Maximizing submodular set function with connectivity constraint: Theory and application to networks," *IEEE/ACM Trans. Netw.*, vol. 23, no. 2, pp. 533–546, Apr. 2015.

[33] M. Sviridenko, "A note on maximizing a submodular set function subject to a knapsack constraint," *Oper. Res. Lett.*, vol. 32, no. 1, pp. 41–43, Jan. 2004.

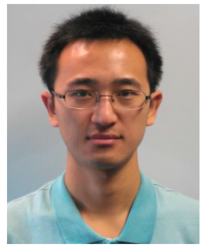
Ziming Wang received the M.Eng. degree in computer science from Sichuan University in 2013, where he is currently pursuing the Ph.D. degree with the College of Computer Science. He is also with the Information Management Department, West China Second Hospital, Sichuan University. His research interests include UAV networking, medical artificial intelligence, and hospital information management.

Jing Li received the B.Sc. and Ph.D. degrees (Hons.) from The Australian National University in 2018 and 2022, respectively. He is currently a Post-Doctoral Fellow with the City University of Hong Kong. His research interests include edge computing, the Internet of Things, digital twin, network function virtualization, and combinatorial optimization.

He Xue received the B.Sc. and M.E. degrees from the University of Electronic Science and Technology of China in 2015 and 2018, respectively. She is currently pursuing the joint Ph.D. degree with the College of Computer Science, Sichuan University, and also with Sichuan Fire Science and Technology Research Institute of MEM, Chengdu, China. Her current research interests include UAV networking, algorithm design, and analysis.

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 is currently an Associate Professor with Sichuan University. Also, he was a Visitor with The Australian National University and The Chinese University of Hong Kong. His research interests include the Internet of Things, UAV networks, mobile computing, approximation algorithms, combinatorial optimization, online social networks, and graph theory. He has published more than 100 papers on prestigious journals and conferences, such as IEEE/ACM TRANSACTIONS ON NETWORKING (IEEE ToN), IEEE TRANSACTIONS ON COMPUTERS (IEEE TC), IEEE TRANSACTIONS ON MOBILE COMPUTING (IEEE TMC), IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS (IEEE TPDS), IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (IEEE TKDE), INFOCOM, and ICDCS.

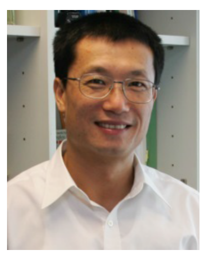
Weifa Liang (Senior Member, IEEE) received the B.Sc. degree in computer science from Wuhan University, China, in 1984, the M.E. degree in computer science from the University of Science and Technology of China in 1989, and the Ph.D. degree in computer science from The Australian National University in 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 with 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 IEEE TRANSACTIONS ON COMMUNICATIONS.



Zichuan Xu (Member, IEEE) received the B.Sc. and M.E. degrees in computer science from Dalian University of Technology, China, in 2008 and 2011, respectively, and the Ph.D. degree in computer science from The Australian National University in 2016. He was a Research Associate with University College London. He is currently a 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.

Jian Peng received the B.A. and Ph.D. degrees from the University of Electronic Science and Technology of China (UESTC) in 1992 and 2004, respectively. He is currently a Professor with the College of Computer Science, Sichuan University. His recent research interests include wireless sensor networks, big data, and cloud computing.

Pan Zhou (Member, IEEE) received the B.S. degree in the advanced class from Huazhong University of Science and Technology (HUST), Wuhan, China, in 2006, and the Ph.D. degree from the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA, in 2011. He is currently a Professor with the School of Electronic Information and Communications, HUST. He was a Senior Technical Member with Oracle Inc., American Company, Boston, MA, USA, from 2011 to 2013. His current research interests include security and privacy, machine learning and big data analytics, and information networks.



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 an Editor of IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS (2006–2009), *Journal of World Wide Web*, *Wireless Networks*, and *Journal of Combinatorial Optimization*. 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, the Ad Hoc and Sensor Networking Symposium, and the Panel Co-Chair of IEEE INFOCOM 2011.

Sajal K. Das (Fellow, IEEE) is currently the Chair of the Computer Science Department and the Daniel St. Clair Endowed Chair of Missouri University of Science and Technology. 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 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. 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*. He is the Co-Founder of the IEEE PerCom, IEEE WoWMoM, and ICDCN Conferences; and served on numerous conference committees as the general chair, the program chair, or a program committee member.