Approximation Algorithm for Connected Submodular Function Maximization Problems

Wenzheng Xu[a], He Xue[a,b], Jing Li[c], Weifa Liang[d], Zichuan Xu[e], Pan Zhou[f], Xiaohua Jia[d], and Sajal K. Das[g]
[a]College of Computer Science, Sichuan University, Chengdu, P. R. China
[b]Sichuan Fire Science and Technology Research Institute of MEM, Chengdu, P. R. China
[c]Department of Computing, The Hong Kong Polytechnic University, Hong Kong, P. R. China
[d]Department of Computer Science, City University of Hong Kong, Hong Kong, P. R. China
[e]School of Software, Dalian University of Technology, Dalian, P. R. China
[f]School of Cyber Science and Engineering, Huazhong University of Science & Technology, P. R. China
[g]Department of Computer Science, Missouri University of Science and Technology, Rolla, USA
Emails: wenzheng.xu3@gmail.com, he.xue@scfri.cn, jing5li@polyu.edu.hk, weifa.liang@cityu.edu.hk
z.xu@dlut.edu.cn, panzhou@hust.edu.cn, csjia@cityu.edu.hk, sdas@mst.edu
Corresponding author: Zichuan Xu

***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 while 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 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 $K \leq 21$, and is no less than $\frac{1-1/e}{6}$ when $K \geq 22$, compared with the current best approximation ratio $\frac{1-1/e}{7}(= \frac{1-1/e}{2h+3})$ for the problem. We finally evaluate the algorithm performance in the application of deploying a UAV network. Experimental results demonstrate 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 its empirical approximation ratio 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**

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 [4], [5], [6], [9], [18], [19], [20], [28], [31]. 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 [1], [3], [21], 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 the UAVs [4], [5], [6], [20], [31] or the sum of data rates of the users served by the UAVs [28].

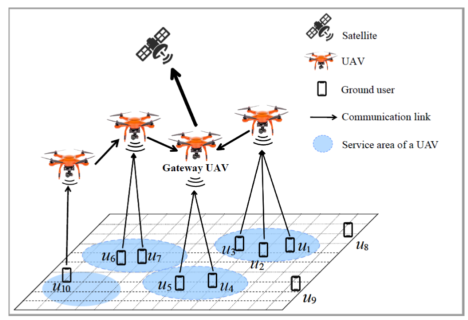


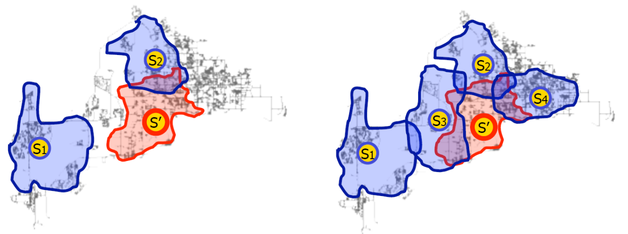
Fig. 1. A UAV network with four UAVs provisions emergent communication services to people trapped in a disaster zone, where the network is connected to the Internet via a satellite.

There are many other important applications of the connected submodular function maximization problem, such as the placement of sensors to monitor PoIs [10], [15], the scheduling of sensors to cover targets in a duty-cycled sensor network [13], the identification of users to influence others in social networks [2], [13], the deployment of wireless power chargers to charge sensors [30], [32], etc.

2575-8411/24/$31.00 ©2024 IEEE
DOI 10.1109/ICDCS60910.2024.00106

2024 IEEE 44th International Conference on Distributed Computing Systems (ICDCS) | 979-8-3503-8605-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICDCS60910.2024.00106

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.



(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.

We note that the objective functions f in many applications of the problem usually can be cast as monotone submodular functions [15] 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 [15], 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 [28], different users have different extents of importance [5], each UAV has a limited service capacity (i.e., the maximum number of users served by the UAV) [28], [31], etc. On the other hand, in the application of placing sensors [7], [26], 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 [5], [6], [10], [24], 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.

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 $\frac{1-1/e}{2h+3}$ [27], [29] 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 [5], [6], [10], [24]), the algorithm approximation ratio is at least $\frac{1-1/e}{5}$ and may be as large as 1 in some special cases when $K \leq 21$, and is no less than $\frac{1-1/e}{6}$ when $K \geq 22$, compared with the current best approximation ratio $\frac{1-1/e}{7} (= \frac{1-1/e}{2h+3})$ for the problem [27], [29].

The techniques used in the proposed approximation algorithm are very different from the one in [27], [29]. Specifically, we estimate a new upper bound on the optimal solution to the problem in Section III, which helps us to find a $\frac{1-1/e}{2h+2}$-approximate solution if the diameter D of the spanning tree induced by the optimal solution is small (i.e., $D \leq 4h + 4$). Otherwise (i.e., the diameter D is large with $D > 4h + 4$), we design a novel graph decomposition technique in Section IV, which is different from the tree decomposition technique in [27], [29]. By leveraging the graph decomposition technique, we can still find a $\frac{1-1/e}{2h+2}$-approximate solution.

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 $\frac{1-1/e}{2h+3}$ [27], [29] 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. 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 peoples are served by the deployed UAV network in a disaster area, and thus more peoples 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 [22]: (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 [5], [28], the number of covered PoIs in IoT networks [4], the number of influenced users in social networks [2], 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.

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 [27], [29]. Note that the typical value of h in the various applications is 2, 3, or 4 [27], [29]. 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., $R_{user} \leq R_{uav}$, 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 $R'_{user} = \sqrt{R^2_{user} - H^2_{uav}}$, where H_{uav} is the height of the UAV and $H_{uav} \leq R_{user}$. It can be seen that $0 \leq R'_{user} \leq R_{user} \leq R_{uav}$. Following the work in [27], [29], the value of h is defined as:

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

Note that the value of R'_{user} in many applications usually is no more than half of R_{uav}, i.e., $R'_{user} \leq \frac{R_{uav}}{2}$ [5], [6], [10], [24]. Then, by Eq. (1), h usually is equal to 2.

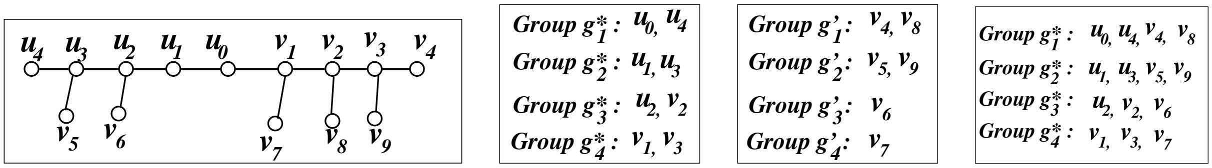
D. Problem definition

We formally define the *connected submodular function maximization problem* as follows. Specifically, given an undirected graph $G(V, E)$, a monotone submodular function $f : 2^V \mapsto \mathbb{Z}^{\geq 0}$ 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., $\mathcal{L}(A, B) \geq h$), 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* [11]. 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 [8], [11], which will be used as a subroutine of the proposed algorithm for the problem studied in this paper.



(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.

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 basic idea of the proposed algorithm is as follows. 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 but more involved, 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 $\sum_{v \in g_j^*} \mathcal{L}(v, u_0) \le D/2$ with $1 \le j \le D/2$. 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., $\mathcal{L}(v, u_0) \le D/2$, 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., $|g'_j| \le \lceil \frac{K-D}{D/2} \rceil$ with $1 \le j \le D/2$, 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 $1 \le j \le D/2$, see Fig. 3(d).

We 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., $f(g_2^*) \ge \frac{f(S^*)}{D/2} = \frac{OPT}{D/2}$, where S^* is the set of nodes in tree T^*. Note that the number of nodes in group g_2^* may be as large as $\lceil \frac{K}{D/2} \rceil < \frac{K}{D/2} + 1$, as each node in g_2^* may be $D/2$ hops away from 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, $f(V') \ge (1-1/e) f(g_2^*) \ge \frac{1-1/e}{D/2} OPT$. 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 $v_j \ne v_k$, if the sum of their shortest distances to node v_i is no more than $D/2$ (i.e., $\mathcal{L}(v_i, v_j) + \mathcal{L}(v_i, v_k) \le D/2$), 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 set such that $f(V')$ is maximized, i.e., $V' = \arg\max_{S_{ijk}}\{f(S_{ijk})\}$.

Note that the inducted 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

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 $\frac{1-1/e}{\lceil D/2 \rceil} \geq \frac{1-1/e}{2h+2}$.

Proof: Denote by S^* an optimal solution to the problem and by OPT its value, 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.

Note 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 $|g_2'| \leq \lceil \frac{K-D}{D/2} \rceil + 3 = \lfloor \frac{K-1}{D/2} \rfloor + 2 = \lfloor \frac{K-1-D/2}{D/2} \rfloor + 3$. 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. 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. 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 [22], $S_{ijk} \setminus \{v_i, v_j, v_k\}$ is a $(1-1/e)$-approximate solution to $g_2' \setminus \{v_i, v_j, v_k\}$, i.e.,

$$f(S_{ijk}) - f(\{v_i, v_j, v_k\}) \geq (1-1/e)(f(g_2') - f(\{v_i, v_j, v_k\})). \tag{2}$$

We thus have

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

The theorem then follows. ■

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.

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, where e is the base of the natural logarithm. 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 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 IV-B). 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 [8], [11] 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 [8], [11] 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 [8], [11] 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 [8], [11] 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 $Q_{ub} = q$.

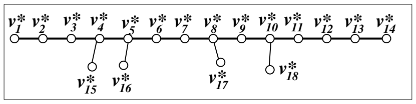
In summary, in the binary search, if the approximation algorithm [8], [11] for the QST problem with quota q finds a tree T_q with no more than K nodes (i.e., $|V(T_q)| \leq K$), we set $Q_{lb} = q$. Otherwise ($|V(T_q)| > K$), we set $Q_{ub} = q$. The binary search continues until $Q_{lb} + 1 = Q_{ub}$. 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 [8], [11] 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

Algorithm 2 Algorithm `ApproAlgLarge` for the connected submodular function maximization problem with a large diameter

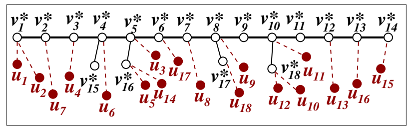
Input: An undirected graph $G(V,E)$, a positive integer K, a positive integer h(=2, 3, or 4), and a monotone submodular function $f: 2^V \mapsto \mathbb{Z}^{\geq 0}$ such that $f(A) + f(B) = f(A \cup B)$ for all subsets A and B of V with $\mathcal{L}(A,B) \geq h$.

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

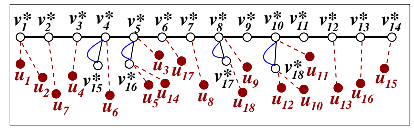
/* Assign profits to nodes in V */
$U \leftarrow \emptyset$;
for $1 \leq i \leq n$ **do**
  Find a node $v_i \in V \setminus U$ such that $f(U \cup \{v_i\})$ is maximized;
  $p(v_i) \leftarrow f(U \cup \{v_i\}) - f(U)$;
  $U \leftarrow U \cup \{v_i\}$;
end for
$S \leftarrow \emptyset$;
$Q_{lb} \leftarrow 0$, $Q_{ub} \leftarrow \sum_{i=1}^{n} p(v_i)$; /* Q_{lb} and Q_{ub} are the lower and upper bounds on Q, respectively */
while $Q_{lb} + 1 < Q_{ub}$ **do**
  $q \leftarrow \lfloor \frac{Q_{lb}+Q_{ub}}{2} \rfloor$;
  Find a tree T_q for the QST problem with quota q by invoking the approximation algorithm in [8], [11];
  if the number of nodes in T_q is no more than K **then**
    $Q_{lb} \leftarrow q$; // update the lower bound
    $S \leftarrow V(T_q)$; // $V(T_q)$ is the set of nodes in T_q
  else
    $Q_{ub} \leftarrow q$; // update the upper bound
  end if
end while
return the set S.



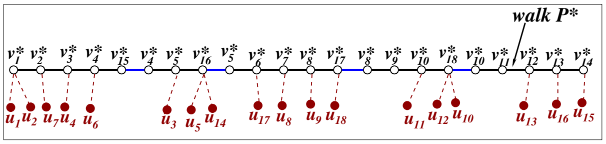
(a) A spanning tree T^* of the optimal solution, where $K = 18$, $D = 13$, and $h = 2$



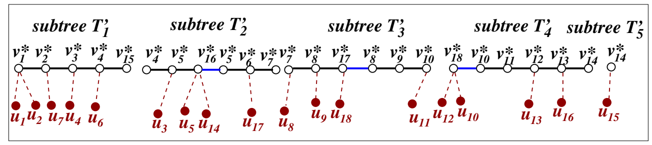
(b) Obtain a tree T by connecting the K nodes in U_{h-1} to tree T^*, where each node in U_{h-1} is $h-1$ hops away from tree T^* and $h-1 = 2-1 = 1$



(c) Obtain a graph G' by duplicating the edges in tree T^* but not in the longest path P' in T^*



(d) Find a walk P^* that visits the edges in $E(T^*) \cup (E(T^*) \setminus P')$



(e) Decompose the graph in Fig. 4(c) into subtrees such that the number of edges in each subtree is no more than $\lceil \frac{K-1}{2} \rceil$

Fig. 4. An illustration of the tree decomposition with a large diameter in the optimal solution.

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

In this subsection, we show that `Algorithm` 2 delivers a $\frac{1-1/e}{2h+2}$-approximate algorithm. Recall that we assumed that S^* is an optimal solution to the problem and T^* is a spanning tree of the induced subgraph $G[S^*]$. Let $S^* = \{v_1^*, v_2^*, \ldots, v_K^*\}$, see Fig. 4(a) with $K = 18$. Notice that in this section we assumed that the diameter D is larger than $4h+4$. For example, Fig. 4(a) shows that the diameter D in tree T^* is $13 > 4h+4 = 4*2+4 = 12$, where $h = 2$.

Note that `Algorithm` 2 first assigns profits to nodes in V in a greedy way. Consider the first K nodes $u_1, u_2, \ldots, u_K$ that are assigned profits by the algorithm and the K nodes are within $h-1$ hops away from the nodes in tree T^*. Let U_{h-1} be the set of the K nodes, i.e., $U_{h-1} = \{u_1, u_2, \ldots, u_K\}$. Following the existing study in [13], the sum of profits of nodes in U_{h-1} is no less than $(1-1/e)OPT$, where OPT is the value of the optimal solution, i.e., $OPT = f(S^*)$. Since each node u_i in U_{h-1} is no more than $h-1$ hops away from tree T^*, we can obtain a tree T by connecting node u_i to its nearest node in T^* with a path no more than $h-1$ edges, see Fig. 4(b) with $h = 2$. It can be seen that the number of edges in tree T is no more than $K-1+K(h-1)$, where $K = |V(T^*)|$.

The study in [27], [29] shows that tree T can be decomposed into $2h+3$ subtrees such that the number of edges in each subtree is no more than $\lceil \frac{K-1}{2} \rceil$. **Different from the study in [27], [29]**, we show the tree T can be decomposed into only $2h+2$ subtrees so that the number of edges in each subtree is no more than $\lceil \frac{K-1}{2} \rceil$, through a novel graph decomposition technique different from the one in [27], [29].

Lemma 1: Assume that the diameter D is larger than $4h+4$. Tree T can be decomposed into $2h+2$ subtrees and the number of edges in each subtree is no more than $\lceil \frac{K-1}{2} \rceil$.

Proof: Let P' be the longest path in tree T^*, e.g., the path between v_1^* and v_{14}^* in Fig. 4(a). We can obtain a graph G' from tree T (see Fig. 4(b)) by duplicating the edges in tree T^* but not in the longest path P' in T^*. Fig. 4(c) shows that we duplicate four edges $(v_4^*, v_{15}^*), (v_5^*, v_{16}^*), (v_8^*, v_{17}^*)$, and (v_{10}^*, v_{18}^*). The set of duplicated edges can be represented as $E(T^*) \setminus P'$. Since the diameter D (i.e., the number of edges in the longest path P' in T^*) is larger than $4h+4$, then $D \geq 4h+5$. The number of edges in G' is upper bounded as

$$\begin{aligned}
|E(G')| &= |E(T)| + |E(T^*) \setminus P'| \\
&\leq K-1+K(h-1)+|E(T^*)|-|E(P')| \\
&= K-1+K(h-1)+K-1-D \\
&\leq (h+1)K-2-(4h+5), \text{ as } D \geq 4h+5 \\
&= (h+1)K-4h-7. \qquad (4)
\end{aligned}$$

We then find a walk P^* that visits the edges in $E(T^*) \cup (E(T^*) \setminus P')$ once and only once, see Fig. 4(d). It can be seen that a node in walk P^* may be attached with some subpaths that connect some nodes in U_{h-1}, where the number of edges in each subpath is no more than $h-1$. Assume that there are J nodes in walk P^* and let $P^* = v'_1 - v'_2 - \cdots - v'_J$, where v'_j is a node in T^* and a node v_i^* in T^* may appear more than once in P^*. For example, each of the four nodes $v_4^*, v_5^*, v_8^*, v_{10}^*$ appears twice in walk P^* of Fig. 4(d).

We now decompose graph G' into $2h+2$ subtrees, while ensuring that the number of edges in each subtree be no more than $\lceil \frac{K-1}{2} \rceil$. We decompose graph G' along walk P^*. Assume that the edges in the left of node v'_j have already been decomposed into some subtrees, and $j = 1$ initially. Let T'_l be a partially decomposed subtree and there are no edges in T'_l initially. Denote by m_l the number of edges already in T'_l. Assume that there are s_j subpaths attached to node v'_j and there are no more than $h-1$ edges in each subpath. If the sum of the number m_l of edges already in T'_l and the number of the edges in the s_j subpaths is no more than $\lceil \frac{K-1}{2} \rceil$, i.e., $m_l + s_j(h-1) \leq \lceil \frac{K-1}{2} \rceil$, we add all edges in the s_j subpaths to T'_l. Otherwise ($m_l + s_j(h-1) > \lceil \frac{K-1}{2} \rceil \geq m_l$), we add $\lfloor \frac{\lceil \frac{K-1}{2} \rceil - m_l}{h-1} \rfloor$ of the s_j subpaths to T'_l, and create the next subtree $T'_{l+1} = \emptyset$. It can be seen that the number of edges in T'_l is at least $\lceil \frac{K-1}{2} \rceil - (h-1) + 1 = \lceil \frac{K-1}{2} \rceil - h + 2$, but no more than $\lceil \frac{K-1}{2} \rceil$. Notice that when creating the next subtree T'_{l+1}, if the number of edges in the residual of G' is no more than $\lceil \frac{K-1}{2} \rceil$, we add all rest edges to T'_{l+1}. Fig. 4(e) illustrates the graph decomposition procedure and five subtrees are decomposed from G'.

We assume that L subtrees $T'_1, T'_2, \ldots, T'_L$ are obtained in the end, where L is a positive integer. We show that L is no more than $2h+2$ as follows. Notice that the number of edges in each of the first $L-1$ subtrees $T'_1, T'_2, \ldots, T'_{L-1}$ is at least $\lceil \frac{K-1}{2} \rceil - h + 2$,

It can be seen that $\lceil \frac{K-1}{2} \rceil = \frac{K-1}{2}$ if K is an odd number, while $\lceil \frac{K-1}{2} \rceil = \frac{K}{2}$ if K is an even number. Since $\frac{K-1}{2} < \frac{K}{2}$, we focus on the case where K is odd. Then, $\lceil \frac{K-1}{2} \rceil - h + 2 = \frac{K-1}{2} - h + 2$. We now upper bound the number of remaining edges in G' when we have decomposed $2h+1$ subtrees $T'_1, T'_2, \ldots, T'_{2h+1}$.

$$\begin{aligned} & (h+1)K - 4h - 7 - \sum_{l=1}^{2h+1} |E(T'_l)|, \text{ by Ineq. (4)} \\ \leq \quad & (h+1)K - 4h - 7 - (2h+1)(\frac{K-1}{2} - h + 2) \\ = \quad & \frac{K-1}{2} + 2h^2 - 6h - 8. \end{aligned} \tag{5}$$

We show that the value of $\frac{K-1}{2} + 2h^2 - 6h - 8$ is no more than $\lceil \frac{K-1}{2} \rceil = \frac{K-1}{2}$ when h =2, 3 or 4. Case (i) with $h=2$, $\frac{K-1}{2} + 2h^2 - 6h - 8 = \frac{K-1}{2} - 12 < \frac{K-1}{2}$; Case (ii) with $h=3$, $\frac{K-1}{2} + 2h^2 - 6h - 8 = \frac{K-1}{2} - 8 < \frac{K-1}{2}$; and Case (iii) with $h=4$, $\frac{K-1}{2} + 2h^2 - 6h - 8 = \frac{K-1}{2}$. This indicates that no more than $2h+2$ subtrees are decomposed. The lemma then follows. ■

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 $\frac{1-1/e}{2h+2}$, where e is the base of the natural logarithm.

Proof: Following existing studies in [13], the sum of profits of nodes in U_{h-1} is no less than $(1-1/e)OPT$, where OPT is the value of the optimal solution. Since the nodes in U_{h-1} are contained in graph G' (e.g., Fig. 4(c)), the sum of profits of nodes in G' also is no less than $(1-1/e)OPT$.

Following Lemma 1, G' can be decomposed into no more than $2h+2$ subtrees so that the number of edges in each subtree is no more than $\lceil \frac{K-1}{2} \rceil$. Then, there is one of the $2h+2$ subtrees, e.g., T'_l, with no more than $\lceil \frac{K-1}{2} \rceil$ edges, and the sum of profits of nodes in the subtree is no less than $\frac{1-1/e}{2h+2}OPT$. Following the discussion in Section IV-A, `Algorithm` 2 finds a tree T with its profit sum no less than $\frac{1-1/e}{2h+2}OPT$ and the number of edges in T is no more than $2\lceil \frac{K-1}{2} \rceil - 1 (= K-1)$. The theorem then follows. ■

V. Better approximation ratio analysis when $h = 2$

Notice that the value of h is equal to 2 in most cases [5], [6], [10], [24], 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 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. (6).

$$Approximation\ ratio = \begin{cases} 1, & 1 \leq K \leq 3 \\ \dfrac{1-1/e}{2}, & 4 \leq K \leq 5 \\ \dfrac{1-1/e}{3}, & 6 \leq K \leq 7 \\ \dfrac{1-1/e}{4}, & 8 \leq K \leq 9 \\ \dfrac{1-1/e}{5}, & 10 \leq K \leq 21 \\ \dfrac{1-1/e}{6}, & K \geq 22. \end{cases} \tag{6}$$

Proof: When $K \leq 3$, `Algorithm` 1 finds the optimal solution by an enumeration, and its approximation ratio thus is one. On the other hand, $K \geq 4$, note that the diameter D of the spanning tree T^* induced by an optimal solution is no more than $K-1$, i.e., $D \leq K-1$, as there are K nodes in tree T^*. Following Theorem 1, the approximation ratio of `Algorithm` 1 is $\frac{1-1/e}{\lceil D/2 \rceil} \geq \frac{1-1/e}{\lceil (K-1)/2 \rceil}$. Therefore, the approximation ratio in Eq. (6) with $4 \leq K \leq 9$ follows.

We consider the case with $10 \leq K \leq 21$. If the diameter D is no more than 10, then the algorithm in Section III delivers a solution with an approximation ratio $\frac{1-1/e}{\lceil D/2 \rceil} \geq \frac{1-1/e}{5}$. On the other hand, when $D \geq 11$, we analyze the approximation ratio of the solution delivered by the algorithm in Section IV as follows. Similar to Ineq. (4), the number of edges in G' with $h=2$ is upper bounded as $|E(G')| \leq K-1+K+K-1-D \leq 3K-2-11 = 3K-13$, where $D \geq 11$. Notice that in the tree decomposition, the number of edges in each of the first four decomposed subtrees $T'_1, T'_2, \ldots, T'_4$ is at least $\lceil \frac{K-1}{2} \rceil - h + 2 = \frac{K-1}{2}$, when K is odd. The number of rest edges after the decompositions of the first four subtrees is no more than $3K - 13 - 4\frac{K-1}{2} = K - 11$, which is no more than $\frac{K-1}{2} = \lceil \frac{K-1}{2} \rceil$ when $K \leq 21$. The case that K is even can be analyzed similarly, omitted. This indicates that no more than 5 subtrees are decomposed. Then, the solution delivered by the algorithm in Section IV has an approximation ratio of $\frac{1-1/e}{5}$. Finally, the approximation ratio is $\frac{1-1/e}{6}$ when $K \geq 22$, 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 [31]. 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 [31]. 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 [16], [23].

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 [31], assuming that the UAV hovering height H_{uav} is 300 m [1]. 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 [3], [21], [28].

We adopt real wireless channel models and parameters in the models are similar to the ones in [1], [28], [31], which are omitted here due to space limitation.

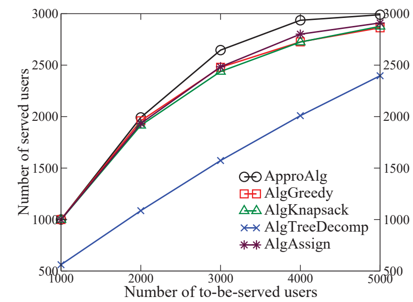
Recall that the proposed algorithms ApproAlgSmall and ApproAlgLarge solve the problem in the two cases where the diameter D in the optimal solution is no more than $4h + 4$ or larger than $4h + 4$, respectively. Denote by ApproAlg the better solution between the two solutions delivered by the two algorithms,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

In addition to the proposed algorithm ApproAlg, we also considered the following four benchmark algorithms. (i) Algorithm AlgGreedy [4] delivers a greedy heuristic solution. (ii) Algorithm AlgKnapsack [28] 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 [5], [13] 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 [27], [29].

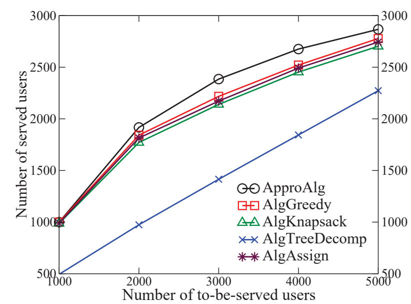
B. Algorithm Performance

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 and Fig. 5(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. 5(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. 5(b) plots the algorithm performance in the strip area, which demonstrates similar curves as those in Fig. 5(a). However, we can see from Fig. 5(a) and Fig. 5(b) that the number of served users by each of the five algorithms in the strip 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 are used to serve users.

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. 6(a) shows that the number of users served in the solution delivered by algorithm

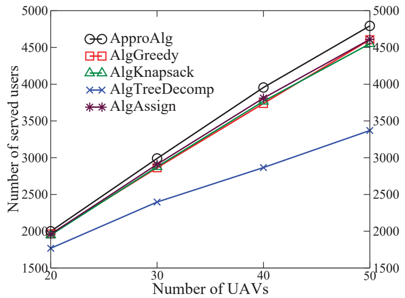


(a) The number of served users by different algorithms in the *square* area

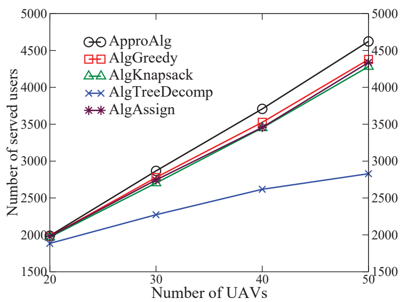


(b) The number of served users by different algorithms in the *strip* area

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

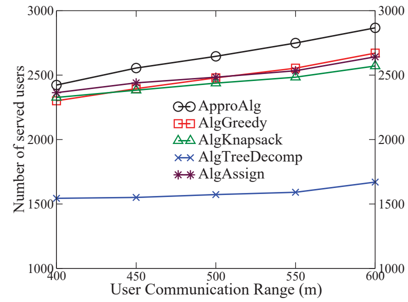


(a) The number of served users by different algorithms in the *square* area

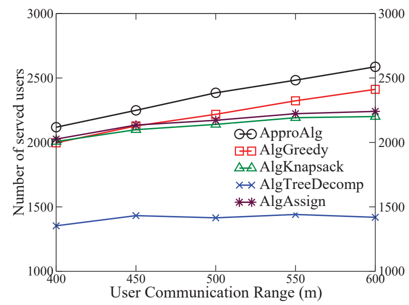


(b) The number of served users by different algorithms in the *strip* area

Fig. 6. 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

Fig. 7. The algorithm performance by varying the communication range R_{user} between a ground user and a UAV in the air from 400 m to 600 m, when there are $K = 30$ UAVs and $m = 3,000$ to-be-served users.

`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.

We finally investigate the algorithm performance by varying the communication range R_{user} between a ground user and a UAV in the air from 400 m to 600 m, when there are $K = 30$ UAVs and $m = 3,000$ to-be-served users in the network. Fig. 7(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. 7 shows that the number of served users by algorithm `ApproAlg` is from 2.5% to 7.5% larger than those by the other four algorithms. In addition, Fig. 7 plots that the empirical approximation ratio of the proposed algorithm `ApproAlg` is between 0.7 and 0.99.

VII. Related Work

The connected submodular function maximization problem studied in this paper has drawn many attentions recently [10], [12], [13], [17], [27], [29], [30]. For example, Khuller *et al.* [12], [13] 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.* [17] improved the approximation ratio $\frac{1-1/e}{12}(\approx 0.0527)$ in [12], [13] to $\frac{1-(1/e)^{\frac{7}{8}}}{11}(\approx 0.053)$. Huang *et al.* [10] 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.* [30] 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 [10] and [30] 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.* [27], [29] 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 [27], [29], 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.* [14] 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.* [28] recently studied the similar problem [14], 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 [25].

However, the time complexity of the algorithm in [28] 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 [4], [5], [6], [9], [20], [31]. Coletta *et al.* [4] proposed a greedy heuristic for deploying UAVs to cover the maximum number of targets. Danilchenko *et al.* [5], [6] adopted the algorithm in [13] 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.* [9] 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.* [31] devised a motion control algorithm for deploying UAVs to serve as many as users under the network connectivity constraint. Liu *et al.* [20] studied the similar problem and devised an algorithm based on deep reinforcement learning.

VIII. Conclusions

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 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 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.

Acknowledgement

The work by Wenzheng Xu was supported by the National Natural Science Foundation of China (NSFC) with grant number 62272328 and Sichuan Science and Technology Program with grant number 24NSFJQ0152.

References

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

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

[3] X. Cao, P. Yang, M. Alzenad, X. Xi, D. Wu, and H. Yanikomeroglu, "Airborne communication networks: a survey," *IEEE J. Sel. Areas Commun. (JSAC)*, vol. 36, no. 9, pp. 1907–1926, 2018.

[4] A. Coletta *et al.*, "A^2-UAV: application-aware content and network optimization of edge-assisted UAV systems," in *Proc. 42nd IEEE Int. Conf. Comput. Commun. (INFOCOM)*, 2023, pp. 1–10.

[5] K. Danilchenko *et al.*, "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, 2023.

[6] K. Danilchenko *et al.*, "Doing their best: How to provide service by limited number of drones?" *Wireless Netw.*, vol. 29, pp. 209–220, 2023.

[7] 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, 2019.

[8] N. Garg, "Saving an epsilon: a 2-approximation for the k-MST problem in graphs," in *Proc. 37th Anuu. ACM Symp. Theory Comput. (STOC)*, 2005, pp. 396–402.

[9] 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, 2022.

[10] L. Huang *et al.*, "Approximation algorithms for the connected sensor cover problem," *Theor. Comput. Sci.*, vol. 809, pp. 563–574, 2020.

[11] 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 (SODA)*, 2000, pp. 760–769.

[12] 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)*, 2014, pp. 1702–1713.

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

[14] T. Kuo, K. C. Lin, and M. Tsai, "Maximizing submodular set function with connectivity constraint: theory and application to networks," *IEEE/ACM Trans. Netw. (ToN)*, vol. 23, no. 2, pp. 533–546, 2015.

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

[16] C. C. Lai *et al.*, "Adaptive and fair deployment approach to balance offload traffic in multi-UAV cellular networks," *IEEE Trans. Veh. Techn. (TVT)*, vol. 72, no. 3, pp. 3724–3738, 2023.

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

[18] S. Li, C. Xiang, W. Xu, J. Peng, Z. Xu, J. Li, W. Liang, and X. Jia, "Coverage maximization of heterogeneous UAV networks," in *Proc. 43rd IEEE Int. Conf. Distrib. Comput. Syst.*, 2023, pp. 120–130.

[19] S. Li, J. Li, C. Xiang, W. Xu, J. Peng, Z. Wang, W. Liang, X. Yao, X. Jia, and S. K. Das, "Maximizing network throughput in heterogeneous UAV networks," *IEEE/ACM Trans. Netw.*, early access, 2024.

[20] C. H. Liu, Z. Chen, J. Tang, J. Xu, and C. Piao, "Energy-efficient UAV control for effective and fair communication coverage: a deep reinforcement learning approach," *IEEE J. Sel. Areas Commun. (JSAC)*, vol. 36, no. 9, pp. 2059–2070, 2018.

[21] 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. 24th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, 2018, pp. 35–49.

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

[23] C. Song *et al.*, "Modelling the scaling properties of human mobility," *Nature Physics*, vol. 6, no. 10, pp. 818–823, 2010.

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

[25] M. Sviridenko, "A note on maximizing a submodular set function subject to a knapsack constraint," *Operations Research Lett.*, vol. 32, pp, 41–43, 2004.

[26] B. Wang, "Coverage problems in sensor networks: a survey", *ACM Comput. Surv.*, vol. 43, no. 4, pp. 32:1–32:53, 2011.

[27] W. Xu *et al.*, "Maximizing h-hop independently submodular functions under connectivity constraint," in *Proc. 41st IEEE Int. Conf. Comput. Commun. (INFOCOM)*, 2022, pp. 1099–1108.

[28] W. Xu, Y. Sun, R. Zou, W. Liang, Q. Xia, F. Shan, T. Wang, X. Jia, and Z. Li, "Throughput maximization of UAV networks", *IEEE/ACM Trans. Netw. (ToN)*, vol. 30, no. 2, pp. 881–895, 2022.

[29] 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, 2023.

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

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

[32] P. Zhou, C. Wang, and Y. Yang, "Design of self-sustainable wireless sensor networks with energy harvesting and wireless charging," *ACM Trans. Sensor Netw. (TOSN)*, vol. 17, no. 4, pp. 45:1–45:38, 2021.