Maximizing h-hop Independently Submodular Functions Under Connectivity Constraint

Wenzheng Xu[a], Dezhong Peng[a,b], Weifa Liang[c], Xiaohua Jia[c], Zichuan Xu*[d], Pan Zhou[e], Weigang Wu[f], Xiang Chen[f]
[a]College of Computer Science, Sichuan University, Chengdu, P. R. China
[b]Shenzhen Peng Cheng Laboratory, Shenzhen, 518052, P. R. China
[c]Department of Computer Science, City University of Hong Kong, Hong Kong, P. R. China
[d]School of Software, Dalian University of Technology, Dalian, P. R. China
[e]School of Cyber Science and Engineering, Huazhong University of Science and Technology, Wuhan, China
[f]Sun Yat-sen University, Guangzhou, 510006, China.
Emails: wenzheng.xu3@gmail.com, pengdz@scu.edu.cn, weifa.liang@cityu.edu.hk, csjia@cityu.edu.hk, z.xu@dlut.edu.cn, panzhou@hust.edu.cn, wuweig@mail.sysu.edu.cn, chenxiang@mail.sysu.edu.cn
*Corresponding author: Zichuan Xu

IEEE INFOCOM 2022 - IEEE Conference on Computer Communications | 978-1-6654-5822-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/INFOCOM48880.2022.9796957

***Abstract*—This study is motivated by the maximum connected coverage problem (MCCP), which is to deploy a connected UAV network with given K UAVs in the top of a disaster area such that the number of users served by the UAVs is maximized. The deployed UAV network must be connected, since the received data by a UAV from its served users need to be sent to the Internet through relays of other UAVs. Motivated by this application, in this paper we study a more generalized problem – the h-hop independently submodular maximization problem, where the MCCP problem is one of its special cases with $h = 4$. We propose a $\frac{1-1/e}{2h+3}$-approximation algorithm for the h-hop independently submodular maximization problem, where e is the base of the natural logarithm. Then, one direct result is a $\frac{1-1/e}{11}$-approximate solution to the MCCP problem with $h = 4$, which significantly improves its currently best $\frac{1-1/e}{32}$-approximate solution. We finally evaluate the performance of the proposed algorithm for the MCCP problem in the application of deploying UAV networks, and experimental results show that the number of users served by deployed UAVs delivered by the proposed algorithm is up to 12.5% larger than those by existing algorithms.**

***Index Terms*—UAV communication networks; maximum connected coverage problem; submodular function maximization; approximation algorithms.**

I. Introduction

In this paper, we study an h-hop independently submodular maximization problem, which is defined later. The problem has many potential applications. One important application arises in the context of Unmanned Aerial Vehicles (UAV) networks. Wireless communication by leveraging the use of UAVs has attracted lots of attentions recently [13], [22], [42]. Unlike terrestrial communication systems, low-altitude UAV systems are more cost-effective by enabling on-demand operations, more swift and flexible for deployment and configuration [9], [14], [32], [35], [36], [40], [41]. Due to its maneuverability and flexibility, a UAV can act as an aerial base station (BS) by equipping with a lightweight base station device [8], [24]. It is expected that UAV networks consists of multiple UAVs are perfectly suitable for unexpected and temporary communication demands, such as natural disasters, traffic congestion, and concerts [3]. In addition, because of their high flying height, UAVs usually have higher Line-of-Sight (LoS) link opportunities with ground users, compared to terrestrial BSs [1]. Fig. 1 shows a UAV network in which four UAVs serve as aerial base stations to provide communication services to the trapped people in a disaster zone. With the help of the UAV network, the trapped people can send and receive critical voices, videos, and data to/from the rescue team, thereby saving their lives and reducing injuries.

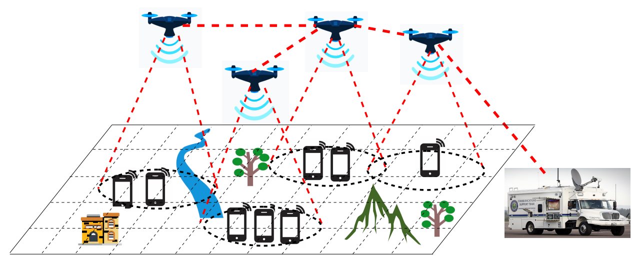


Fig. 1. A UAV network that provides communication services for ground users in a disaster area, where the network is connected to the Internet via an emergency communication vehicle.

Assume that all UAVs hover at the same altitude H_{uav}, which is the optimal altitude for the maximum coverage from the sky [1], [42], e.g., $H_{uav} = 300\ m$. Denote by R the communication range between any two UAVs at altitude H_{uav}, and denote by r' the communication range between a ground user and a UAV at altitude H_{uav}. Notice that r' is no greater than R [15]. Let $r = \sqrt{r'^2 - H_{uav}^2}$. Assume that a UAV hovers at a location with its coordinate $(x_i,\ y_i,\ H_{uav})$. Then, its coverage area is a disk that centers at location $(x_i, y_i, 0)$ with radius r, i.e., the set of points with coordinates $(x, y, 0)$ such that $(x-x_i)^2+(y-y_i)^2 \leq r^2$. Thus, the ground users in the disk can communicate with the UAV directly. Let $\alpha = \frac{r}{R}$. Then, $0 < \alpha \leq 1$, as $r \leq r' \leq R$.

Our study is motivated by a fundamental *maximum connected coverage problem (MCCP)* [42] in a UAV network, which is to deploy K UAVs for serving people in a disaster zone, such that the number of users served is maximized, subject to the constraint that the communication subnetwork induced by the K UAVs is connected. The rationale behind the connectivity constraint is that, the received data by a UAV from its served users need to be sent to a gateway UAV in the UAV network, where the gateway UAV is connected to the Internet, with the help of an emergency communication vehicle or satellites, see Fig. 1.

In this paper, we study a more generalized problem – the h-hop independently submodular maximization problem, which is briefly defined as follows. Notice that the MCCP problem is a special case of the problem studied in this paper with $h = 4$.

Given an undirected, connected graph $G = (V, E)$, let $f : 2^V \mapsto \mathbb{Z}^{\geq 0}$ be a monotone function on the subsets of V, i.e., $f(A) \leq f(B)$ for any subsets A and B of V with $A \subseteq B$. In addition, given a positive integer $h \geq 1$, we say that f is *h-hop independently submodular* based on G if it meets the following two properties:

(i) *Submodularity*: $f(A \cup \{v\}) - f(A) \geq f(B \cup \{v\}) + f(B)$ for any two subsets A and B of V with $A \subseteq B$, and any node $v \in V \setminus B$. The submodularity captures the property of *diminishing returns* in economics and many fields [6].

(ii) *h-hop independence*: $f(A) + f(B) = f(A \cup B)$ for any two non-empty subsets A and B of V with the minimum number of hops in G between any node in A and any node in B being at least h.

In this paper, we consider an *h-hop independently submodular maximization problem* in $G(V, E)$, which is to find a subset S of K nodes in V such that the value of $f(S)$ is maximized, subject to the constraint that the induced subgraph $G[S]$ of G by the nodes in S is *connected*, where K is a given positive integer with $1 \leq K \leq |V|$ and f is h-hop independently submodular.

In addition to the aforementioned application of deploying UAV networks, there are many other potential applications of the h-hop independently submodular maximization problem. For example, consider the problem of placing K sensors at some strategic locations to monitor PoIs (Points of Interest) in an IoT network such that the number of PoIs monitored by the K sensors is maximized, subject to the constraint that the communication subnetwork induced by the K sensors is connected [15]. Other applications include deploying wireless power chargers in wireless sensor networks [38], [39], placing wireless routers in wireless networks [20], choosing influential connected users in social networks [2], [17], [18], [23], [29], [31], [33].

There are several studies on special cases of the h-hop independently submodular maximization problem. For example, Garg [12] proposed a $\frac{1}{3+\epsilon}$-approximation algorithm for the problem when $h = 1$, where ϵ is a given constant with $0 < \epsilon \leq 1$. Notice that the submodular function f meets the additive property when $h = 1$, i.e., for any subset S of V, $f(S) = \sum_{v \in S} f(\{v\})$. Khuller *et al.* [18] proposed a $\frac{1-1/e}{12}$-approximation algorithm for the problem when $h = 3$, where e is the base of the natural logarithm. Yu *et al.* [38], [39] proposed a $\frac{1-1/e}{8(\lceil \frac{4}{\sqrt{3}}\alpha \rceil + 1)^2}$-approximation algorithm for the MCCP problem, where $\alpha = \frac{r}{R}$ with $0 < r \leq R$. It can be seen that the approximation ratio is a value between $\frac{1-1/e}{128}$ and $\frac{1-1/e}{32}$, as $0 < \alpha \leq 1$.

A. Main contributions

The main contributions of this paper are as follows.

(i) To the best of our knowledge, we are the first to introduce the h-hop independently submodular maximization problem, which generalizes many optimization problems arisen in different domains, such as the MCCP problem of deploying a UAV network to serve as many users as possible.

(ii) In this paper, we propose a $\frac{1-1/e}{2h+3}$-approximation algorithm for the problem when $h \geq 2$. Consequently, the proposed algorithm delivers $\frac{1-1/e}{9}$ and $\frac{1-1/e}{11}$ approximate solutions to the problem, when $h = 3$ and $h = 4$, respectively, while the best approximation ratios so far for these two special cases with $h = 3$ and $h = 4$ are $\frac{1-1/e}{12}$ [18] and $\frac{1-1/e}{32}$ [38], [39], respectively.

(iii) We finally evaluate the performance of the proposed algorithm for the MCCP problem in the application of deploying UAV networks, and experimental results show that the number of users served by deployed UAVs in the solution delivered by the proposed algorithm is up to 12.5% larger than those by existing algorithms.

B. Technical novelties

We are motivated by the study in [18]. There are two major technical differences between our work and the work in [18]. The first one is that, unlike the algorithm in [18] that assigns profits to nodes in only one way, we assign profits to nodes in multiple ways. We show that there is a tree T in G with the profit sum of the nodes in T being no less than $(1-1/e)OPT$ among one of the multiple profit assignments, and the number of the edges in T is no greater than $(K-1)h$, which is less than the number $(Kh-1)$ in [18] when $h \geq 1$.

The other difference is that the traditional tree decomposition technique adopted in [18] decomposes a tree T with $(K-1)h$ edges into $4h$ subtrees so that the number of nodes in each subtree is no more than $\frac{K}{2}$. We here propose a novel tree decomposition technique that decomposes a tree T with $(K-1)h$ edges into $2h+3$ subtrees, such that the number of nodes in each subtree is no more than $\frac{K}{2}$, by exploring important structure properties of the tree T. Note that $2h + 3 < 4h$ for any integer h if $h \geq 2$. By utilizing the proposed tree decomposition technique, we devise a novel approximation algorithm for the h-hop independently submodular maximization problem, and its approximation ratio is $\frac{1}{4}$ and $\frac{1-1/e}{2h+3}$ when $h = 1$ and $h \geq 2$, respectively.

The rest of the paper is organized as follows. Section II introduces preliminaries and defines the problem. Section III proposes a $\frac{1-1/e}{2h+3}$-approximation algorithm for the h-hop independently submodular maximization problem, while Section IV shows the approximation ratio. Section V evaluates the performance of the proposed algorithms. Section VI reviews related work, and Section VII concludes the paper.

II. Preliminaries

We consider an undirected, connected graph $G = (V, E)$, where V is the set of nodes and E is the set of edges. For any two nodes u and v in V, denote by $l(u, v)$ the minimum number of hops (i.e., edges) in G between nodes u and v. Also, for any two non-empty subsets A and B of V, denote by $l(A, B)$ the minimum number of hops between nodes in A and B, i.e., $l(A, B) = \min_{u \in A, v \in B}\{l(u, v)\}$.

We consider a nondecreasing submodular function $f : 2^V \mapsto \mathbb{Z}^{\geq 0}$, which meets the following three properties:

(i) $f(\emptyset) = 0$;

(ii) Monotonicity: $f(A) \leq f(B)$ for any two subsets A and B of V with $A \subseteq B$; and

(iii) Submodularity: $f(A \cup \{v\}) - f(A) \geq f(B \cup \{v\}) + f(B)$ for any two subsets A and B of V with $A \subseteq B$, and any node $v \in V \setminus B$.

A function $f : 2^V \mapsto \mathbb{Z}^{\geq 0}$ is an *h-hop independently submodular function* in a graph $G = (V, E)$ if and only if (i) f is nondecreasing and submodular; and (ii) for any two non-empty subsets A and B of V, if the minimum number of hops between the nodes in A and the nodes in B is no less than h (i.e., $l(A, B) \geq h$), then $f(A) + f(B) = f(A \cup B)$, where $h \geq 1$ is a given positive integer.

In this paper, we consider *an h-hop independently submodular maximization problem*, which is defined as follows. Given an undirected, connected graph $G = (V, E)$, an h-hop independently submodular function $f : 2^V \mapsto \mathbb{Z}^{\geq 0}$, and a positive integer K, the problem is to find a set S of K nodes in V such that the value of $f(S)$ is maximized, subject to the constraint that the induced subgraph $G[S]$ by the nodes in S is connected.

We assume that the values of h and K satisfy the following relationship: $2h + 3 \leq \sqrt{K}$. The rationale behind the assumption is as follows. Xu *et al.* [34] devised a $\frac{1-1/e}{\sqrt{K}}$-approximation algorithm for finding a set S with K nodes in G such that a submodular function $f(S)$ is maximized, subject to that $G[S]$ is connected, where e is the base of the natural logarithm. This implies that the algorithm also delivers a $\frac{1-1/e}{\sqrt{K}}$-approximate solution to the problem considered in this paper. However, the approximation ratio $\frac{1-1/e}{\sqrt{K}}$ is small when K is large. Under the assumption that $2h + 3 \leq \sqrt{K}$, we will propose an improved algorithm with an approximation ratio $\frac{1-1/e}{2h+3}$ for the problem in this paper, which is no less than $\frac{1-1/e}{\sqrt{K}}$.

We define a *Quota Steiner Tree (QST) problem* [16]. Given an undirected graph $G = (V, E)$, a profit function $p : V \mapsto \mathbb{Z}^{\geq 0}$, a cost function $c : E \mapsto \mathbb{Z}^{\geq 0}$, and a positive integer (quota) q, the problem is to find a subtree T in G such that the cost of the T, i.e., $\sum_{e \in E(T)} c(e)$, is minimized, subject to the constraint that the profit sum of nodes in T is no less than q, i.e., $\sum_{v \in V(T)} p(v) \geq q$. Notice that there is a 2-approximation algorithm for the QST problem [12], [16], and the algorithm will be part of the solution to the problem in this paper.

III. Approximation Algorithm

In this section, we propose a $\frac{1-1/e}{2h+3}$-approximation algorithm for the h-hop independently submodular maximization problem.

A. Basic idea

The basic idea behind the proposed algorithm is that we assign profits to nodes in graph G in n different ways with $n = |V|$. We find a tree T_i in G with no more than K nodes so that the profit sum of nodes in T_i is maximized in each of the n profit assignments, by invoking the 2-approximation algorithm for the QST problem, where the QST problem here is to find a tree in G such that the number of nodes in the tree is minimized, subject to the constraint that the profit sum of nodes in the tree is at least a given quota q. The solution to the problem then is the set of nodes in one of the n found trees $T_1, T_2, \ldots, T_n$ such that the profit sum of nodes in the tree is maximized.

B. Approximation algorithm

Given an undirected, connected graph $G = (V, E)$, an h-hop independently submodular function $f : 2^V \mapsto \mathbb{Z}^{\geq 0}$, and a positive integer K, let $V = \{v_1, v_2, \ldots, v_n\}$, where $n = |V|$. We assign profits to nodes in G with n different ways.

Denote by $p_i(v)$ the profit assigned to node $v \in V$ in G in the ith way with $1 \leq i \leq n$. This profit assignment proceeds as follows.

We start by assigning a profit $f(\{v_i\})$ to node v_i, i.e., $p_i(v_i) = f(\{v_i\})$. We then choose a node v in $V \setminus \{v_i\}$ with the maximum marginal profit $f(\{v, v_i\}) - f(\{v_i\})$ and assign node v the profit $p_i(v) = f(\{v, v_i\}) - f(\{v_i\})$, where ties are broken arbitrarily. The profit assignment procedure continues until each node in G is assigned a profit. The detailed profit assignment procedure is given in `Algorithm` 1.

Algorithm 1 Profit assignment procedure

Input: An undirected, connected graph $G = (V, E)$, an h-hop independently submodular function $f : 2^V \mapsto \mathbb{Z}^{\geq 0}$, and a starting node v_i

Output: the assigned profit $p_i(v)$ of each node $v \in V$ in the ith way

```
Assign profit f({v_i}) to the starting node v_i, i.e., p_i(v_i) = f({v_i});
Let D ← {v_i}; /* the set of nodes assigned profits already*/
Let U ← V \ D;
while U ≠ ∅ do
    Choose a node v in U with the maximum marginal profit f({v} ∪ D) − f(D), i.e., v = arg max_{v_j ∈ U} {f({v_j} ∪ D) − f(D)};
    Let p_i(v) = f({v} ∪ D) − f(D);
    Let D ← D ∪ {v};
    Let U ← U \ {v};
end while
return the assigned profit p_i(v) of each node v in V.
```

Having assigned a profit $p_i(v)$ to each node $v \in V$ in the ith way, we find a tree T_i with no more than K nodes such that the profit sum of the nodes in T_i is maximized, based on the profit assignment. Denote by q_{opt} the optimal profit sum.

Since the total profit q_{opt} must be in the interval of $[f(\{v_i\}), f(V)]$, the value of q_{opt} can be calculated by a binary search. Specially, let lb and ub be the lower and upper bounds on q_{opt}, respectively. Initially, let $lb = f(\{v_i\})$ and $ub = f(V)$. Let $q = \lfloor \frac{lb+ub}{2} \rfloor$. We can find a tree T_q in G based on the profit assignment of the ith way so that the number of nodes in T_q is minimized, subject to the constraint that the profit sum of nodes in T_q is no less than q, by invoking the 2-approximation algorithm for the QST problem. Consider the number of nodes $|V(T_q)|$ in tree T_q. If $|V(T_q)| \le K$, this implies that $q(= \lfloor \frac{lb+ub}{2} \rfloor)$ is no more than the optimal profit sum q_{opt}. In this case, let q become the updated lower bound on q_{opt}, i.e., $lb = q$. Otherwise ($|V(T_q)| > K$), this indicates that the value q is larger than q_{opt}, i.e., $q > q_{opt}$. Let q become the updated upper bound on q_{opt}, i.e., $ub = q$. The binary search will terminate when $ub = lb + 1$. Finally, the tree T_i can be found, by invoking the 2-approximation algorithm for the QST problem with a quota of $lb(= ub - 1)$.

The algorithm for the h-hop independently submodular maximization problem is presented in `Algorithm` 2.

Algorithm 2 Approximation algorithm for the h-hop independently submodular maximization problem

Input: An undirected, connected graph $G = (V, E)$, an h-hop independently submodular function $f : 2^V \mapsto \mathbb{Z}^{\ge 0}$, and a positive integer K.

Output: A set S of K nodes in G such that the value of $f(S)$ is maximized, subject to the constraint that the induced subgraph $G[S]$ is connected.

Let $S \leftarrow \emptyset$;
for $1 \le i \le n$ **do**
  Assign profits to nodes in V starting from node v_i by invoking `Algorithm` 1;
  Let $lb \leftarrow f(\{v_i\})$ and $ub \leftarrow f(V)$; /* lb and ub are the lower and upper bounds on the value of q_{opt}, respectively */
  while $lb + 1 < ub$ **do**
    Let $q \leftarrow \lfloor \frac{lb+ub}{2} \rfloor$; /* q is the quota in the QST problem */
    Find a tree T_q in G with the minimum number of nodes, subject to the constraint that the profit sum of nodes in T_q, i.e., $\sum_{v \in V(T_q)} p_i(v)$, is no less than quota q, by invoking the 2-approximation algorithm for the QST problem;
    if the number of nodes in T_q is no greater than K **then**
      Let $lb \leftarrow q$; /* the quota q is no more than q_{opt} */
    else
      Let $ub \leftarrow q$; /* the quota q is larger than q_{opt} */
    end if
  end while
  Let $q \leftarrow lb$, where $lb = ub - 1$;
  Find a tree T_i in G with the minimum number of nodes, subject to the constraint that the profit sum of nodes in T_i is no less than quota q, by invoking the 2-approximation algorithm for the QST problem. Notice that the number of nodes in T_i must be no greater than K.
  if $f(V(T_i)) > f(S)$ **then**
    Let $S \leftarrow V(T_i)$; /* find a better set of nodes */
  end if
end for
return set S.

IV. Analysis of the Approximation Algorithm

Denote by L_0 the set of nodes in an optimal solution to the problem. Then, $OPT = f(L_0)$. Also, denote by L_{h-1} the set of nodes such that the minimum number of hops in G between any node v in L_{h-1} and any node in L_0 is no more than $h-1$, but node v is not contained in L_0, i.e., $L_{h-1} = \{v \mid v \in V \setminus L_0,\ l(v, L_0) \le h-1\}$, where $h \ge 1$, and $l(v, L_0)$ is the minimum number of hops between node v and nodes in L_0 in G. Let $L_h = V \setminus (L_0 \cup L_{h-1})$. It can be seen that the minimum number of hops between nodes in L_0 and nodes in L_h is no less than h, i.e., $l(L_0,\ L_h) \ge h$.

Consider a node v_i with the *maximum profit* in the optimal solution L_0, i.e., $v_i = \arg\max_{v \in L_0}\{f(v)\}$. Recall that in the ith **'for'** loop of `Algorithm` 2, we first assign a profit $p_i(v_i) = f(v_i)$ to node v_i, then assign profits to the other nodes in G greedily. Denote by D' the first K nodes in set $L_0 \cup L_{h-1}$ that have been assigned profits by the profit assignment procedure. Let $D' = \{v_i, v_1, v_2, \ldots, v_{K-1}\}$ with $i \notin \{1, 2, \ldots, K-1\}$. Denote by $p_i(D')$ the profit sum of nodes in D', i.e., $p_i(D') = \sum_{v \in D'} p_i(v)$.

Proof roadmap: In the rest, we first show that the profit sum of nodes in D' is no less than $(1 - 1/e) \cdot OPT$, i.e., $p_i(D') \ge (1 - 1/e)OPT$. We then prove that there is a tree T in G spanning the nodes in D', such that the number of nodes in T is no more than $(K-1)h + 1$. The profit sum of nodes in T thus is no less than $(1 - 1/e)OPT$. We also show that tree T can be decomposed into no more than $2h+3$ subtrees such that the number of nodes in each subtree is no more than $\frac{K}{2}$. Then, there must have a subtree T' among the $2h+3$ subtrees such that the profit sum of nodes in T' is no less than $\frac{1}{2h+3}$ of the profit sum of nodes in T, i.e., $\sum_{v \in T'} p_i(v) \ge \frac{\sum_{v \in T} p_i(v)}{2h+3} \ge \frac{1-1/e}{2h+3} OPT$. Finally, a tree in G with no more than $2\frac{K}{2} = K$ nodes can be found such that the profit sum of nodes in the tree is no less than $\frac{1-1/e}{2h+3} \cdot OPT$, by invoking the 2-approximation algorithm for the QST problem in [12], [16].

We start by showing that the profit sum of nodes in D' is no less than $(1 - 1/e) \cdot OPT$.

Lemma 1: Consider node v_i in the optimal solution L_0 with the maximum profit and the profit assignment procedure starting with node v_i. Let D' be the first K nodes in set $L_0 \cup L_{h-1}$ with the assigned profits. Then, $p_i(D') \ge (1 - 1/e) \cdot OPT$.

Proof: The proof is omitted, since it is similar to the one in [18]. ■

A. The existence of a tree T with $(K-1)h+1$ nodes that spans all nodes in D'

We then show that there is a tree T in G spanning the nodes in D' such that the number of nodes in T is no more than $(K-1)h + 1$, which is less than Kh in [18].

Lemma 2: Given node $v_i \in L_0$ with the maximum profit and the profit function $p_i : V \mapsto \mathbb{Z}^{\ge 0}$, there is a tree T in G

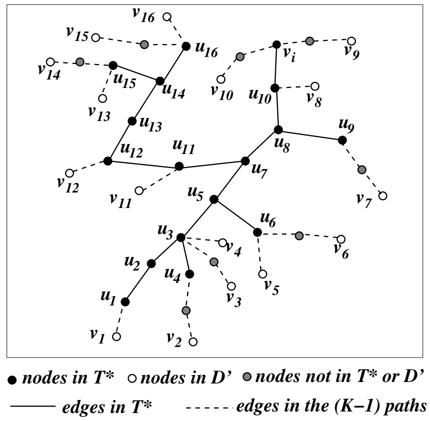


Fig. 2. The constructed tree T spanning the nodes in D', where $K = 17$ and $h = 3$. Notice that node v_i is contained by both T^* and D'.

with no more than $(K-1)h+1$ nodes such that the profit sum of nodes in T, i.e., $\sum_{v \in V(T)} p_i(v)$, is no less than $(1 - 1/e) \cdot OPT$, where e is the base of the natural logarithm.

Proof: We construct a tree T in G spanning all nodes in D' such that T contains no more than $(K-1)h+1$ nodes, based on profit function $p_i(\cdot)$.

Since L_0 is the optimal solution, the induced subgraph $G[L_0]$ by the nodes in L_0 is connected. Denote by T^* a spanning tree in $G[L_0]$, assuming that the cost of each edge is one. Notice that v_i is in L_0 and each node in D' is contained in $L_0 \cup L_{h-1}$, where L_{h-1} is the set of nodes such that the minimum number of hops in G between any node v in L_{h-1} and any node in L_0 is no greater than $h-1$, but node v is not contained in L_0. Then, it can be seen that there is a path P_k in G between any node v_k in $D' \setminus \{v_i\}$ and a node u_k in L_0 such that the number of edges in P_k is no more than $h-1$.

A tree T can be constructed, which is the union of T^* and the $K-1$ found paths, i.e., $T = T^* \bigcup (\bigcup_{k=1}^{K-1} P_k)$. Fig. 2 illustrates such a tree, where $K = 17$, $h = 3$, $L_0 = V(T^*) = \{v_i, u_1, u_2, \ldots, u_{16}\}$, $D' = \{v_i, v_1, v_2, \ldots, v_{16}\}$, and the number of edges in each path P_k is no more than $h - 1 = 2$.

The number of edges in T is no more than $|E(T)| = |E(T^*)| + \sum_{k=1}^{K-1} |E(P_k)| \leq K - 1 + (K-1) \cdot (h-1) = (K-1)h$. The number of nodes in T thus is no greater than $(K-1)h+1$. The lemma then follows. ■

B. A novel tree decomposition

We now show that there is a subtree T' in G with no more than $\lfloor \frac{K}{2} \rfloor$ nodes such that the profit sum of nodes in T' is no less than $\frac{1-1/e}{2h+3} \cdot OPT$. Following Lemma 2, there is a tree T in G with no more than $(K-1)h+1$ nodes such that the profit sum $\sum_{v \in V(T)} p_i(v)$ of the nodes in T is no less than $(1 - 1/e) \cdot OPT$. We here propose a novel tree decomposition technique that decomposes T into no more than $2h+3$ subtrees such that the number of nodes in each subtree is no greater than $\lfloor \frac{K}{2} \rfloor$, where $2h+3 < 4h$ for any integer h with $h \geq 2$. Then, there must be a subtree T' among the $2h+3$ subtrees such that the profit sum of nodes in T' is no less than $\frac{1}{2h+3}$ of the profit sum of nodes in T, i.e., $\sum_{v \in T'} p_i(v) \geq \frac{\sum_{v \in T} p_i(v)}{2h+3} \geq \frac{1-1/e}{2h+3} OPT$.

Tree decomposition procedure

We show the tree decomposition procedure when K is odd. Then, $\lfloor \frac{K}{2} \rfloor = \frac{K-1}{2}$. On the other hand, $\lfloor \frac{K}{2} \rfloor = \frac{K}{2}$ when K is even. The procedure with the case that K is even is omitted, due to its similarity with the case that K is odd.

Recall that tree T is the union of a spanning tree T^* in $G[L_0]$ and $K-1$ paths $P_1, P_2, \ldots, P_{K-1}$, where the number of edges in P_k is no more than $h-1$ with $1 \leq k \leq K-1$, see Fig. 2. Without loss of generality, we further assume that $P_1, P_2, \ldots, P_{K-1}$ are edge-disjoint. Otherwise, the paths with edge-sharing can be converted to edge-disjoint paths, by duplicating the shared edges.

Let node $v_i \in L_0$ be the root of tree T. Denote by T_v the subtree of T rooted at node v for any node $v \in T$, and denote by $w(T_v)$ the number of edges in T_v. We decompose tree T by a Depth-First Search (DFS) starting from node v_i, until the number of edges in the residual tree is no more than $\frac{K-1}{2} - 1 = \frac{K-3}{2}$. The detailed tree decomposition procedure is given as follows.

Assume that v is the node being visited by the DFS. If the number of edges in tree T_v is no more than $\frac{K-3}{2} - 1$, i.e., $w(T_v) \leq \frac{K-3}{2} - 1$, nothing is done and the tree decomposition procedure continues; otherwise ($w(T_v) \geq \frac{K-3}{2}$ as $w(T_v)$ is an integer), a tree will be decomposed from T as follows. We later show that node v must be contained in tree T^*, where T^* is a spanning tree in $G[L_0]$ by the optimal solution L_0.

Assume that node T_v has n_v children $v'_1, v'_2, \ldots, v'_{n_v}$. Denote by tree T'_l the union of edge (v, v'_l) and subtree $T_{v'_l}$ rooted at a child v'_l, i.e., $T'_l = (v, v'_l) \cup T_{v'_l}$, where $1 \leq l \leq n_v$.

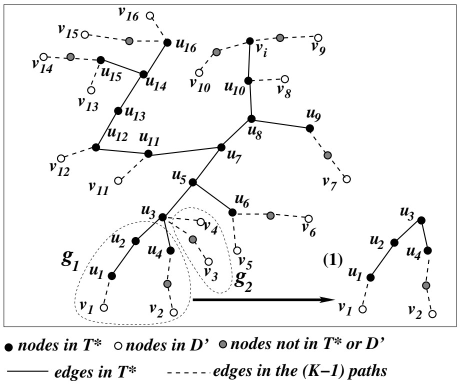
Following the work in [30], the n_v subtrees $T'_1, T'_2, \ldots, T'_{n_v}$ can be partitioned into, say $n' (\geq 2)$, groups $g_1, g_2, \ldots, g_{n'}$ such that the number of edges of subtrees in each group is no more than $\frac{K-3}{2}$ (i.e., $\sum_{T'_l \in g_j} w(T'_l) \leq \frac{K-3}{2}$ for each j with $1 \leq j \leq n'$), while the number of edges in the subtrees of *any two groups* is larger than $\frac{K-3}{2}$ (i.e., $\sum_{T'_l \in g_j \cup g_{j'}} w(T'_l) > \frac{K-3}{2}$ for each pair of j and j' with $1 \leq j, j' \leq n'$ and $j \neq j'$). For example, Fig. 3(a) shows that tree T_{u_3} rooted at u_3 consists of four subtrees, and these four subtrees are partitioned into $n' = 2$ groups, where $K = 17$ and $\frac{K-3}{2} = 7$. Also, it can be seen that the numbers of edges in the subtrees of groups g_1 and g_2 are 6 and 3, respectively. Then, $w(g_1) = 6 \leq \frac{K-3}{2} = 7$ and $w(g_2) = 3 \leq \frac{K-3}{2} = 7$, while $w(g_1) + w(g_2) = 6 + 3 = 9 > \frac{K-3}{2} = 7$.

For each group g_j with $1 \leq j \leq n'$, denote by n^*_j the number of edges in $g_j \cap T^*$, where T^* is a spanning tree in graph $G[L_0]$. For example, consider two groups g_1 and g_2 in Fig. 3(a). It can be seen that $n^*_1 = 3$ and $n^*_2 = 0$.

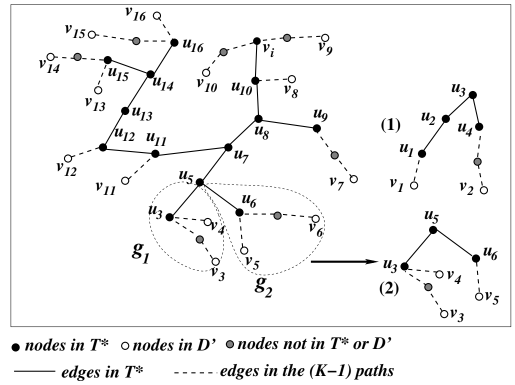
A tree T''_j is decomposed from T by distinguishing into two cases. **Case (i):** the number of edges in the subtrees of a group g_j is no less than $\frac{K-3}{2} - n^*_j - (h-2)$, i.e.,

$$w(g_j) = \sum_{T'_l \in g_j} w(T'_l) \geq \frac{K-3}{2} - n^*_j - (h-2). \quad (1)$$

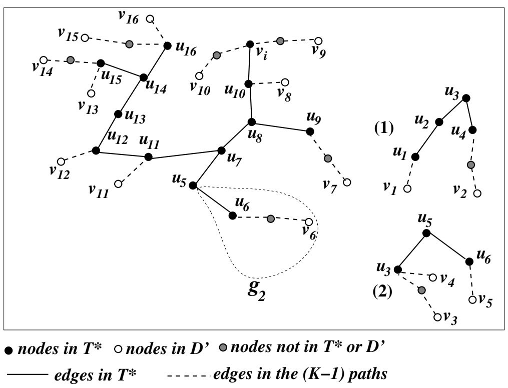
A tree T''_j in **Case (i)** is constructed, which is the union of the subtrees in group g_j, as each subtree in g_j contains the



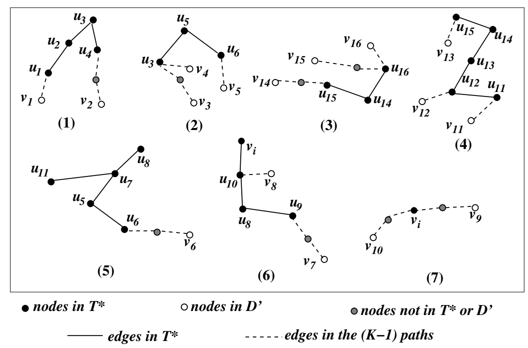
(a) **Case (i)**: the number of edges in the subtrees of group g_1 i.e., 6, is no less than $\frac{K-3}{2} - n_1^* - (h-2) = 3$, where $K = 17$, $n_1^* = |E(g_1 \cap T^*)| = 3$, and $h = 3$



(b) **Case (ii):** the number of edges in the subtrees of any group g_j, i.e., 4, is no more than $\frac{K-3}{2} - n_j^* - (h-2) = 5$, where $K = 17$, $n_j^* = |E(g_j \cap T^*)| = 1$ with $1 \leq j \leq 2$, and $h = 3$



(c) The residual tree of T after the tree decomposition in **Case (ii)**, and the edge (u_5, u_6) in $g_2 \cap T^*$ is not removed



(d) The obtained 7 subtrees after the tree decomposition, where $7 \leq 2h+3 = 9$, and the number of edges in each subtree is no greater than $\frac{K-3}{2} = 7$

Fig. 3. The execution illustrations of the tree decomposition procedure.

root v of T_v. Finally, the edges in T_j'' are removed from T. For example, Fig. 3(a) shows that the number of edges in the subtrees of group g_1 is $w(g_1) = 3 + 3 = 6 > \frac{K-3}{2} - n_1^* - (h-2) = \frac{17-3}{2} - 3 - (3-2) = 3$. Tree T_j'' thus is the union of the subtrees in group g_1, see Fig. 3(a).

Case (ii): the number of edges in all subtrees of each group g_j is no more than $\frac{K-3}{2} - n_j^* - (h-2)$, i.e., $w(g_j) < \frac{K-3}{2} - n_j^* - (h-2)$ for each j with $1 \leq j \leq n'$. For example, consider node u_5 in Fig. 3(b), where the number of edges in the subtrees of group g_1 (or g_2) is 4, while the value of $\frac{K-3}{2} - n_j^* - (h-2)$ is $\frac{17-3}{2} - 1 - (3-2) = 5$. The n' groups $g_1, g_2, \ldots, g_{n'}$ are sorted in non-increasing order of the numbers of their edges in T^*. Without loss of generality, assume that $n_1^* \geq n_2^* \geq \cdots \geq n_{n'}^*$, where $n_j^* = |E(g_j \cap T^*)|$ with $1 \leq j \leq n'$. Notice that node v is contained in each subtree $g_j \cap T^*$ with $1 \leq j \leq n'$, as v is contained in each subtree of g_j and v is in T^*.

A tree T_j'' in **Case (ii)** is constructed from T as follows. First, let T_j'' be the union of the subtrees in group g_1. Then, we duplicate the edges in $g_2 \cap T^*$ and add the edges to T_j''. Notice that T_j'' is connected after adding the edges in $g_2 \cap T^*$, since node v is contained in both $g_1 \cap T^*$ and $g_2 \cap T^*$. Finally, recall that for each node v_k in $D' \setminus \{v_i\}$, there is a path P_k between v_k and a node u_k in T^* such that the number of edges in P_k is no more than $h-1$. We continue adding a path P_k of a node v_k in $(D' \setminus \{v_i\}) \cap g_2$ to T_j'' as long as the number of edges in T_j'' is no more than $\frac{K-3}{2}$. For example, Fig. 3(b) illustrates such a tree, where the edge in $g_2 \cap T^*$ is (u_5, u_6), and path P_5 consisting of only edge (v_5, u_6) is added to T_j''.

Having constructed tree T_j'', the edges in T_j'' except the edges in $g_2 \cap T^*$ are removed from T, see Fig. 3(c) for the residual tree of T after the tree decomposition in **Case (ii)**.

Denote by $\mathcal{T}$ the set of the decomposed subtrees from T by the tree decomposition procedure. For example, Fig. 3(d) shows that seven subtrees are obtained through the tree decomposition of tree T. It can be seen that the number of edges of each tree in $\mathcal{T}$ is no more than $\frac{K-3}{2}$. Then, the number of nodes of each tree is no greater than $\frac{K-3}{2} + 1 = \frac{K-1}{2} \leq \frac{K}{2}$.

Bound the number of decomposed subtrees in $\mathcal{T}$

Lemma 3: Assume that $\sqrt{K} \geq 2h+3$. Then, the tree T in G with no more than $(K-1)h+1$ nodes can be decomposed into no more than $2h+3$ subtrees, such that the number of nodes in each subtree is no more than $\frac{K}{2}$. Then, there is a subtree T' among the $2h+3$ subtrees with no more than $\frac{K}{2}$ nodes such that the profit sum of nodes in T' is no less than $\frac{1-1/e}{2h+3} \cdot OPT$, i.e., $|V(T')| \leq \frac{K}{2}$ and $\sum_{v \in T'} p_i(v) \geq \frac{1-1/e}{2h+3} \cdot OPT$.

Proof: It can be seen that the number of nodes of each subtree in $\mathcal{T}$ is no greater than $\frac{K}{2}$. We show that the number of subtrees in $\mathcal{T}$ is no more than $2h+3$. Recall that, before splitting any subtree off from tree T, T consists of a spanning tree T^* in $G[L_0]$ and $K-1$ paths $P_1, P_2, \ldots, P_{K-1}$, see Fig. 2(a). It can be seen that the numbers of edges in T^* and T are $K-1$ and $(K-1)h$, respectively, by Lemma 2.

Let $\mathcal{T} = \{T_1'', T_2'', \ldots, T_x'', T_{x+1}''\}$ be the set of decomposed subtrees of T by the tree decomposition procedure, where the number of subtrees in $\mathcal{T}$ is $x+1$ and x is a nonnegative integer.

Following the tree composition procedure, some edges of T^* will be removed when decomposing each subtree T_j'' from T, where $1 \le j \le x+1$. The set of the removed edges can be represented as $(E(T_j'') \cap E(T^*)) \setminus E(T^*)$. Let $n_j^* = |((E(T_j'') \cap E(T^*)) \setminus E(T^*)|$. It can be seen that $\sum_{j=1}^{x+1} n_j^* \le K-1$, as any edge in T^* will not be contained in other subtrees once it has been removed from T. Especially, we have

$$\sum_{j=1}^{x} n_j^* \le \sum_{j=1}^{x+1} n_j^* \le K-1, \tag{2}$$

where T_{x+1}'' is the final decomposed subtree.

We show that the number of removed edges from T after decomposing each subtree T_j'' is no less than $\frac{K-3}{2} - n_j^* - (h-2)$, i.e.,

$$w(T_j'' \setminus T) \ge \frac{K-3}{2} - n_j^* - (h-2). \tag{3}$$

Assume that a node v in T is being visited by the DFS in the tree decomposition procedure when T_j'' is decomposed. Then, T_j'' is a subtree of T_v. Subtree T_j'' may be obtained in either Case (i) or Case (ii) of the tree decomposition procedure, see Fig. 3. For Case (i) (see Fig. 3(a)), we have

$$w(T_j'' \setminus T) = w(T_j'') \ge \frac{K-3}{2} - n_j^* - (h-2), \text{ by Ineq. (1).} \tag{4}$$

On the other hand, assume that T_j'' is obtained by Case (ii) in the tree decomposition procedure. It can be seen that the number of edges in T_j'' is at least $\frac{K-3}{2} - (h-2)$, i.e.,

$$w(T_j'') \ge \frac{K-3}{2} - (h-2). \tag{5}$$

Otherwise $(w(T_j'') < \frac{K-3}{2} - (h-2))$, we have $w(T_j'') \le \frac{K-3}{2} - (h-2) - 1 = \frac{K-3}{2} - (h-1)$. We then can add another path $P_{k'}$ of a node $v_{k'}$ in $(D' \setminus \{v_i\}) \cap g_2$, such that the number of edges in T_j'' is at most $\frac{K-3}{2} - (h-1) + |E(P_{k'})| \le \frac{K-3}{2}$, as the number of edges in $P_{k'}$ is no more than $h-1$. This however contradicts the construction of tree T_j''.

It can be seen that the set of edges removed from T after decomposing subtree T_j'' by Case (ii) is $E(T_j'') \setminus E(g_2 \cap T^*)$, since the edges in $E(g_2 \cap T^*)$ are not removed from T in the tree decomposition, see Fig. 3(b) and Fig. 3(c). We then have

$$\begin{aligned}
& w(T_j'' \setminus T) \\
= \;& w(T_j'') - |E(g_2 \cap T^*)| \\
\ge \;& \frac{K-3}{2} - (h-2) - |E(g_2 \cap T^*)|, \text{ by Ineq. (5)} \\
= \;& \frac{K-3}{2} - n_2^* - (h-2), \text{ as } n_2^* = |E(g_2 \cap T^*)| \\
\ge \;& \frac{K-3}{2} - n_1^* - (h-2), \text{ as } n_1^* \ge n_2^*.
\end{aligned} \tag{6}$$

Combining Ineq. (4) and Ineq. (6), Ineq. (3) holds.

Since there are $(K-1)h$ edges in tree T initially, the number of edges removed from T after decomposing the first x subtrees is no greater than $(K-1)h$. We thus have

$$\begin{aligned}
(K-1)h \;\ge\;& \sum_{j=1}^{x} w(T_j'' \setminus T) \\
=\;& \sum_{j=1}^{x} \left(\frac{K-3}{2} - n_j^* - (h-2)\right), \text{ by Ineq. (3)} \\
=\;& \left(\frac{K-1}{2} - (h-1)\right) \cdot x - \sum_{j=1}^{x} n_j^* \\
\ge\;& \left(\frac{K-1}{2} - (h-1)\right) \cdot x - (K-1), \text{ by Ineq. (2).}
\end{aligned} \tag{7}$$

Then,

$$\begin{aligned}
2(h+1) \;\ge\;& \left(1 - \frac{2(h-1)}{K-1}\right) \cdot x \\
\ge\;& \left(1 - \frac{2(h-1)}{(2h+3)^2 - 1}\right) \cdot x, \\
& \text{by the assumption that } \sqrt{K} \ge 2h+3.
\end{aligned} \tag{8}$$

By re-arranging Ineq. (8), we have

$$x \le 2h + 3 - \frac{5h+7}{2h^2 + 5h + 5}. \tag{9}$$

Since x is an integer, we have

$$x \le 2h + 2. \tag{10}$$

Then, the number of subtrees in $\mathcal{T}$ is $x+1 \le 2h+3$. For example, Fig. 3(d) shows that seven subtrees are obtained after the tree decomposition of tree T, $|\mathcal{T}| = 7 \le 2h+3 = 9$, and the number of edges of each subtree is no more than $\frac{K-3}{2} = 7$. The lemma then follows. ■

C. Analysis of the approximation ratio

Lemma 4: Given node $v_i \in L_0$ with the maximum profit, assign profits to nodes in G with profit function $p_i : V \mapsto \mathbb{Z}^{\ge 0}$. Then, the 2-approximation algorithm for the QST problem in [12], [16] can find a tree in G with no more than K nodes such that the profit sum of nodes in the tree is no less than a quota q if $q \le \lceil \frac{1-1/e}{2h+3} \cdot OPT \rceil$. Equivalently, if the algorithm in [12], [16] delivers a tree with more than K nodes, then the quota q is larger than $\lceil \frac{1-1/e}{2h+3} \cdot OPT \rceil$.

Proof: Following Lemma 3, there is a tree T' in G with no more than $\frac{K}{2}$ nodes such that the profit sum of nodes in T' is no less than $\lceil \frac{1-1/e}{2h+3} \cdot OPT \rceil$, as the profit sum is an integer. Therefore, tree T' is a feasible solution to the QST problem when the quota $q \le \lceil \frac{1-1/e}{2h+3} \cdot OPT \rceil$. Then, the optimal solution to the QST problem with a quota q contains no more than $\frac{K}{2}$ nodes. We thus conclude that the tree delivered by the 2-approximation algorithm for the QST problem [12], [16] contains no more than $2 \cdot \frac{K}{2} = K$ nodes. ■

We finally analyze the approximation ratio of the proposed approximation algorithm by the following theorem.

Theorem 1: Given an undirected, connected graph $G = (V, E)$, an h-hop independently submodular function $f : 2^V \mapsto \mathbb{Z}^{\ge 0}$, and a positive integer K with $K \le |V|$, Then, there is an approximation algorithm, `Algorithm` 2, for the h-hop independently submodular maximization problem, which delivers a $\frac{1-1/e}{2h+3}$-approximate solution, where h is a given positive integer with $h \geq 2$, and e is the base of the natural logarithm.

Proof: Consider node $v_i \in L_0$ in the optimal solution with the maximum profit, and a profit function $p_i : V \mapsto \mathbb{Z}^{\geq 0}$. It can be seen that $ub = lb+1$ when `Algorithm` 2 terminates, where ub and lb are the upper and lower bounds on the value of $\lceil \frac{1-1/e}{2h+3} \cdot OPT \rceil$. Also, the algorithm in [12], [16] for the QST problem delivers a tree with no more than K nodes when the quota $q = lb$, while it delivers a tree more than K nodes when the quota $q = ub$. Then, $ub > \lceil \frac{1-1/e}{2h+3} \cdot OPT \rceil$ by Lemma 4. We thus have $ub \geq \lceil \frac{1-1/e}{2h+3} \cdot OPT \rceil + 1$, due to that the value of ub is an integer. Therefore, $lb \geq \lceil \frac{1-1/e}{2h+3} \cdot OPT \rceil \geq \frac{1-1/e}{2h+3} \cdot OPT$. That is, the tree delivered by the algorithm for the QST problem with quota $q = lb (\geq \frac{1-1/e}{2h+3} \cdot OPT)$ in [12], [16] contains no more than K nodes. Therefore, the approximation ratio of `Algorithm` 2 is $\frac{1-1/e}{2h+3}$. ■

V. Performance Evaluation

A. Experimental environment settings

We consider an application of the problem for deploying a connected UAV network to serve ground users in a disaster area. Consider a disaster area of 3×3 km^2 square [42], in which 500 to 3,000 users are located, where the human density follows the fat-tailed distribution, i.e., many people are located at a small portion of places while a few people are located at other places [27]. The number of deployed UAVs K varies from 10 to 50. Then, the approximation ratio of the proposed algorithm is $\frac{1-1/e}{11}$, where e is the base of the natural logarithm. We assume that each UAV hovers at altitude $H_{uav} = 300$ m [1]. The communication range R between any two UAVs is 600 m, while the communication range r' between a user and a UAV is 500 m [42].

To evaluate the performance of the proposed algorithm `ApproAlg` for the maximum connected coverage problem, we adopt the following three benchmarks. (i) Algorithm `MotionCtrl` [42] finds a distributed motion control solution for deploying K UAVs to cover as many as users while maintaining the connectivity of the UAVs. (ii) Algorithm `MCS` [32] delivers a $\frac{1-1/e}{\sqrt{K}}$-approximate solution to the problem of deploying K UAVs in a disaster area, such that a submodular function of the deployed UAVs is maximized, subject to the connectivity constraint that the subnetwork induced by the K UAVs is connected. (iii) Algorithm `GreedyLabel` [17] first assigns profits for deploying a UAV at different hovering locations in a greedy way, followed by identifying a connected subgraph with no more than K nodes such that the profit sum of nodes in the subgraph is maximized.

B. Algorithm Performance

We first study the algorithm performance by varying the number m of users from 500 to 3,000, when there are $K(=30)$ UAVs. Fig. 4 shows that the number of users served by algorithm `ApproAlg` is about from 8.5% to 12.5% higher than those by algorithms `MotionCtrl`, `MCS`, and `GreedyLabel`. For example, the numbers of users served by the four algorithms `ApproAlg`, `MotionCtrl`, `MCS`, and `GreedyLabel` are 2,600, 1,670, 2,395, and 1,800, respectively when there are 3,000 users in the disaster area. Fig. 4 demonstrates that more users are served by each of the four algorithms, with the increase on the number m of users.

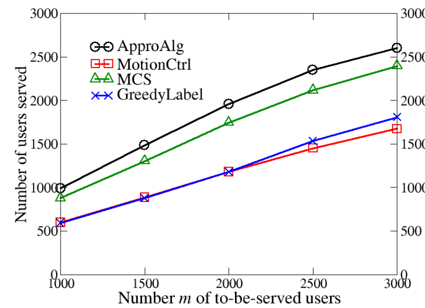


Fig. 4. The performance of different algorithms by varying the number m of to-be-served users from 500 to 3,000, when there are $K = 30$ UAVs.

We then investigate the performance of different algorithms by increasing the number K of UAVs from 10 to 50, when there are $m = 3,000$ users. Fig. 5 plots that the number of users served by each algorithm increases with more UAVs. In addition, the deployed UAVs by algorithm `ApproAlg` serve 97% ($\approx \frac{2,915}{3,000}$) of users when there are $K = 40$ UAVs, while the deployed UAVs by the other three algorithms serve no more than 88% ($\approx \frac{2,633}{3,000}$) of users.

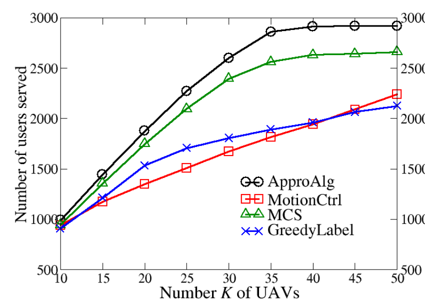


Fig. 5. The performance of different algorithms by increasing the number K of UAVs from 10 to 50, when there are $m = 3,000$ to-be-served users.

We finally study the performance of different algorithms by varying the communication range R between two UAVs from 500 m to $1,000$ m while fixing the communication range r' of a user at 500 m, when $m = 3,000$, $K = 30$. Fig. 6 illustrates that the number of users served by each of the four algorithms `ApproAlg`, `MotionCtrl`, `MCS`, and `GreedyLabel` increases with the growth of the communication range R between two UAVs. The rationale behind the phenomenon is that less numbers of relaying UAVs are needed when the communication range R is larger, and more UAVs thus can be used to serve the users. Fig. 6 also plots the difference between the numbers of users served by algorithms `ApproAlg`, `MotionCtrl`, `MCS`, and `GreedyLabel`. For example, the number of users served by algorithm `ApproAlg` is about 20% larger than the one by algorithm `MCS` when the communication range R between two UAVs is 500 m, while the number by algorithm `ApproAlg` is only about 2.2% larger than the one by algorithm `MCS` when $R = 1,000$ m.

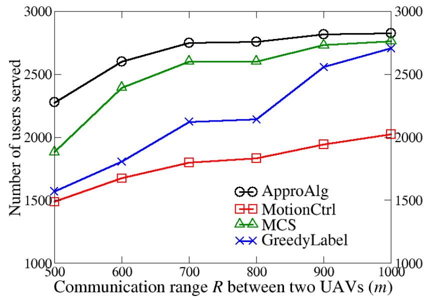


Fig. 6. The performance of different algorithms by varying the communication range R between two UAVs from 500 m to 1,000 m while fixing $r' = 500$ m, when $m = 3,000$ users, $K = 30$.

VI. Related Work

The use of UAVs as aerial base stations (BS) recently has gained lots of attentions in public communications. For example, Zhao *et al.* [42] presented a motion control algorithm for deploying a given number K of UAVs to cover as many as users while maintaining the connectivity among UAVs. Liu *et al.* [22] considered the similar problem and proposed a deep reinforcement learning (DRL) based algorithm. Yang *et al.* [37] investigated the problem of scheduling the movement of multiple UAVs to fairly provide communication services to mobile ground users for a given period, by using the DRL method, too. Shi *et al.* [26] studied the problem of planning the flying trajectories of multiple UAVs for a period such that the average UAV-to-user pathloss in the network is minimized, assuming that a user can be served by only a single UAV during the period. They decoupled the problem into multiple subproblems, and solved the subproblems separately.

There are several studies on the special cases of the h-hop independently submodular maximization problem, subject to the connectivity constraint that the induced subgraph of G by a subset of nodes in V is connected. For example, Khuller *et al.* [17], [18] devised a $\frac{1-1/e}{12}$-approximation algorithm for the *budgeted connected dominating set (BCDS) problem*, which is to find a set S of K nodes in a graph G such that the number of nodes dominated by the nodes in S is maximized, subject to the constraint that the induced subgraph $G[S]$ is connected. Notice that the objective function of the BCDS problem is 3-hop independently submodular with $h = 3$. Huang *et al.* [15] proposed a $\frac{1-1/e}{8(\lceil 2\sqrt{2}\alpha\rceil+1)^2}$-approximation algorithm for the maximum connected coverage problem with $h = 4$. where $\alpha = \frac{r}{R}$, r and R are the sensing range and communication range of a sensor respectively, and $0 < r \leq R$. It can be seen that $\frac{1-1/e}{128} \leq \frac{1-1/e}{8(\lceil 2\sqrt{2}\alpha\rceil+1)^2} \leq \frac{1-1/e}{32}$, as $0 < \alpha \leq 1$. Yu *et al.* [38], [39] recently improved the approximation ratio to $\frac{1-1/e}{8(\lceil \frac{4}{\sqrt{3}}\alpha\rceil+1)^2}$, where $\frac{1-1/e}{128} \leq \frac{1-1/e}{8(\lceil \frac{4}{\sqrt{3}}\alpha\rceil+1)^2} \leq \frac{1-1/e}{32}$. It can be seen that both approximation ratios in [15] and [38], [39] are no greater than $\frac{1-1/e}{32}$.

There are other investigations on maximizing the values of other submodular functions, not h-hop independently submodular functions, subject to connectivity constraints. For example, Kuo *et al.* [20] considered the problem of deploying K wireless routers in a wireless network such that a submodular function of the deployed K routers is maximized, subject to the constraint that the subnetwork induced by the K routers is connected, for which they proposed a $\frac{1-1/e}{5(\sqrt{K}+1)}$-approximation algorithm, where e is the base of the natural logarithm.

There are flourishing studies on maximizing the value of a submodular function *without* connectivity constraints. For monotone submodular functions, Nemhauser *et al.* [25] considered a problem of choosing K elements from a set such that a submodular function of the chosen K elements is maximized. They devised a $(1-1/e)$-approximation algorithm for the problem and showed that the result is tight. They also extended their result to the submodular function maximization problem under the constraint of the intersection of M matroids, and proposed a $\frac{1}{M+1}$-approximation algorithm [11], and this approximation ratio later is further improved to $\frac{1}{M+\epsilon}$ by Lee *et al.* [21] when $M \geq 2$, where ϵ is a given constant with $0 < \epsilon \leq 1$. Calinescu *et al.* [7] and Filmus *et al.* [10] proposed a randomized $(1-1/e)$-approximation algorithm for maximizing a submodular problem under a matroid constraint, respectively, while Buchbinder devised a deterministic $(1/2+\epsilon)$-approximation algorithm [4], [5]. Sviridenko [28] proposed a $(1-1/e)$-approximation algorithm for maximizing a submodular function subject to a linear constraint, while Kulik *et al.* [19] extended to the solution to multiple linear constraints by giving a $(1-1/e-\epsilon)$-approximation algorithm.

VII. Conclusions

In this paper, we studied the novel h-hop independently submodular maximization problem, which generalizes many optimization problems arisen in different domains, such as the MCCP problem of deploying a connected UAV network to serve as many users as possible. We then devised a $\frac{1-1/e}{2h+3}$-approximation algorithm for the problem, where e is the base of the natural logarithm. The proposed algorithm has many potential applications, and one direct corollary from this result is a $\frac{1-1/e}{11}$-approximate solution to the MCCP problem when $h = 4$, which significantly improves its currently best $\frac{1-1/e}{32}$-approximate solution [39].

Acknowledgement

The work by Wenzheng Xu was supported by the National Natural Science Foundation of China (NSFC) with grant number 61602330. The work by Dezhong Peng was supported by the NSFC with Grant numbers 61971296 and U19A2078. The work by Weifa Liang was supported by a grant from City University of Hong Kong with Project No. CityU 9380137/CS. The work by Zichuan Xu is funded by the NSFC with grant numbers 62172068 and 61802048. The work by Pan Zhou was funded by the NSFC with grant number 61972448. The work by Weigang Wu was supported by the Guangdong Natural Science Foundation of China with grant number 2018B030312002.

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] I. Bor-Yaliniz and H. Yanikomeroglu, "The new frontier in RAN heterogeneity: multi-tier drone-cells," *IEEE Commun. Mag.*, vol. 54, no. 11, pp. 48–55, Nov. 2016.

[4] N. Buchbinder and M. Feldman, "Deterministic algorithms for submodular maximization problems," *ACM Trans. Algorithms (TALG)*, vol. 14, no.3, pp. 32:1–32:20, 2018.

[5] N. Buchbinder, M. Feldman, and M. Garg, "Deterministic $(1/2 + \epsilon)$-approximation for submodular maximization over a matroid," in *Proc. 30th Annu. ACM-SIAM Symp. Discrete Algorithms (SODA)*, 2019, pp. 241–254.

[6] N. Buchbinder, M. Feldman, J. Seffi, and R. Schwartz, "A tight linear time (1/2)-approximation for unconstrained submodular maximization," *SIAM J. Comput.*, vol. 44, no. 5, pp. 1384–1402, 2015.

[7] G. Calinescu, C. Chekuri, M. Pál, and J. Vondrák, "Maximizing a submodular set function subject to a matroid constraint (extended abstract)," in *Proc. Int. Conf. Integer Programming Combinatorial Optimization (IPCO)*, 2007, pp. 182–196.

[8] S. Chandrasekharan, K. Gomez, A. Al-Hourani, S. Kandeepan, T. Rasheed, L. Goratti, L. Reynaud, D. Grace, I. Bucaille, T. Wirth, and S. Allsopp, "Designing and implementing future aerial communication networks," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 26–34, 2016.

[9] L. Deng, W. Xu, W. Liang, J. Peng, Y. Zhou, L. Duan, and S. K. Das, "Approximation algorithms for min-max cycle cover problems with neighborhoods," *IEEE/ACM Trans. Netw. (ToN)*, vol. 28, no. 4, pp. 1845–1858, Aug. 2020.

[10] Y. Filmus and J. Ward, "A tight combinatorial algorithm for submodular maximization subject to a matroid constraint," in *Proc. 53rd Annu. IEEE Symp. Found. Comput. Sci. (FOCS)*, 2012, pp. 659–668.

[11] M. L. Fisher, G. L. Nemhausert and L. A. Wolsey, "An analysis of the approximations for maximizing submodular set functions-II," *Math. Programming Study*, vol. 8, pp. 73–87, 1978.

[12] 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.

[13] J. Gong, T. H. Chang, C. Shen, and X. Chen, "Flight time minimization of UAV for data collection over wireless sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 1942–1954, 2018.

[14] Q. Guo, J. Peng, W. Xu, W. Liang, X. Jia, Z. Xu, Y. Yang, and M. Wang, "Minimizing the longest tour time among a fleet of UAVs for disaster area surveillance," *IEEE Trans. Mobile Comput. (TMC)*, DoI: 10.1109/TMC.2020.3038156, 2022.

[15] L. Huang, J. Li, and Q. Shi, "Approximation algorithms for the connected sensor cover problem," in *Proc. 21st Int. Conf. Comput. Combinatorics (COCOON)*, 2015, pp. 183–196.

[16] 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.

[17] 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.

[18] 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.

[19] A. Kulik, H. Shachnai, and T. Tamir, "Maximizing submodular set functions subject to multiple linear constraints," in *Proc. 20th Annu. ACM-SIAM Symp. Discrete Algorithms (SODA)*, 2009, pp. 545–554.

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

[21] J. Lee, M. Sviridenko, and J. Vondrák, "Submodular maximization over multiple matroids via generalized exchange properties," in *Proc. 12th Int. Workshop Approximation Algorithms Combinatorial Optimization & 13th Int. Workshop Randomization Approximation Tech. Comput. Sci. (APPROX-RANDOM)*, 2009, pp. 244–257.

[22] 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.*, vol. 36, no. 9, pp. 2059–2070, Sep. 2018.

[23] Y. Liu and W. Liang, "Approximate coverage in wireless sensor networks," in *Proc. 30th Annu. IEEE Conf. Local Comput. Netw. (LCN)*, 2005, pp. 68–75.

[24] 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.

[25] G. Nemhauser and L. A. Wolsey "An analysis of the approximations for maximizing submodular set functions-I," *Math. Programming*, vol. 14, pp. 265–294, 1978.

[26] W. Shi, J. Li, N. Cheng, F. Lyu, S. Zhang, H. Zhou, and X. Shen, "Multi-drone 3-D trajectory planning and scheduling in drone-assisted radio access networks," *IEEE Trans. Veh. Techn.*, vol. 68, no. 8, pp. 8145–8158, 2019.

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

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

[29] W. Xu, T. Li, W. Liang, J. X. Yu, N. Yang, and S. Gao "Identifying structural hole spanners to maximally block information propagation," *Inform. Sci. (INS)*, vol. 505, pp. 100–126, 2019.

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

[31] W. Xu, W. Liang, X Lin, and J. X. Yu, "Finding top-k influential users in social networks under the structural diversity model," *Inform. Sci. (INS)*, vol. 355, pp. 110–126, 2016.

[32] W. Xu, W. Liang, Z. Xu, J. Peng, D. Peng, T. Liu, X. Jia, and S. K. Das, "Approximation algorithms for the generalized team orienteering problem and its applications," *IEEE/ACM Trans. Netw. (ToN)*, vol. 29 no. 1, pp. 176–189, Feb. 2021.

[33] 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. (TKDE)*, vol. 29, no. 5, pp. 1017–1030, 2017.

[34] 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)*, DoI: 10.1109/TNET.2021.3125982, 2022.

[35] W. Xu, T. Xiao, J. Zhang, W. Liang, Z. Xu, X. Liu, X. Jia, and S. K. Das, "Minimizing the deployment cost of UAVs for delay-sensitive data collection in IoT networks," *IEEE/ACM Trans. Netw. (ToN)*, doi: 10.1109/TNET.2021.3123606, 2022.

[36] W. Xu, Z. Xu, J. Peng, W. Liang, T. Liu, X. Jia, and S. K. Das "Approximation algorithms for the team orienteering problem," in *Proc. 39th IEEE Int. Conf. Comput. Commun.*, 2020, pp. 1389–1398.

[37] P. Yang, X. Cao, X. Xi, W. Du, Z. Xiao, and D. Wu, "Three-dimensional continuous movement control of drone cells for energy-efficient communication coverage," *IEEE Trans. Veh. Techn.*, vol. 68, no. 7, pp. 6535–6546, 2019.

[38] N. Yu, H. Dai A. X. Liu, and B. Tian, "Placement of connected wireless chargers," in *Proc. 37th IEEE Conf. Comput. Commun. (INFOCOM)*, 2018, pp. 387–395.

[39] 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.

[40] Y. Zeng and R. Zhang "Energy-efficient UAV communication with trajectory optimization," *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 3747–3760, June 2017.

[41] J. Zhang, Z. Li, W. Xu, J. Peng, W. Liang, Z. Xu, X. Ren, and X. Jia, "Minimizing the number of deployed UAVs for delay-bounded data collection of IoT devices," in *Proc. 40th IEEE Int. Conf. Comput. Commun. (INFOCOM)*, 2021, pp. 1–10.

[42] 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.