Approximation Algorithms for the Maximum Connected Submodular Functions

Qinqin Gong, Zixuan Wang, Yang Lv, and Ruiqi Yang(✉)

Institute of Operations Research and Information Engineering, Beijing University of Technology, Beijing 100124, People's Republic of China
{gongqinqin,wzx1009,lvyang}@emails.bjut.edu.cn, yangruiqi@bjut.edu.cn

Abstract. Motivated by the challenge of maximizing connected coverage with limited UAVs in communication networks, we address the problem within a graph network framework $G = (V, E)$, where V represents potential UAV deployment positions and E denotes communication links between nodes. A utility function $f : 2^V \to \mathbb{R}_+$ is defined to characterize coverage efficiency. Under the constraint of limited field-of-view (FoV) UAVs, the objective is to identify a subset $S \subseteq V$ with $|S| \leq K$ that maximizes $f(S)$ while ensuring the induced subgraph $G[S]$ remains connected. We formulate this as the Maximum Connected Submodular function with Cardinality constraint (MCSC) problem and propose a $\frac{1-e^{-1}}{2\sqrt{K-1}+5}$-approximation algorithm, leveraging a novel tree decomposition technique. Additionally, we present a bicriteria $\left(\frac{(1-e^{-1})\alpha}{2\sqrt{K}+3\alpha}, \alpha^2\right)$-approximation algorithm for the problem, where $\alpha > 1$ is a constant. For a special case of the MCSC problem, where the submodular utility exhibits partial additivity when subsets are sufficiently far apart, we define the Maximum Connected h-Hop Submodular function with a Cardinality constraint (MCHSC) problem. We provide an approximation algorithm with a ratio of $(1-2\varepsilon)\left(\frac{1-e^{-1}}{5(h+1)+1} - \delta\right)$ when $K > 25h(h+1)-5$, where ε, δ are small positive constants and h captures the partial additivity property.

Keywords: Submodular optimization · Connectivity · Cardinality · Approximation algorithms

1 Introduction

In recent years, the deployment of Unmanned Aerial Vehicle (UAV) technology has facilitated its gradual integration into various fields. UAVs can function as aerial base stations, delivering wireless coverage to targeted areas or users. They play a pivotal role in disaster management by supporting emergency communication [6,15,17,23,27]. However, in many real-world scenarios, the number of available UAVs is limited. Consequently, the challenge of effectively deploying

F. V. Fomin and M. Xiao (Eds.): COCOON 2025, LNCS 15983, pp. 18–31, 2026.
https://doi.org/10.1007/978-981-95-0215-8_2

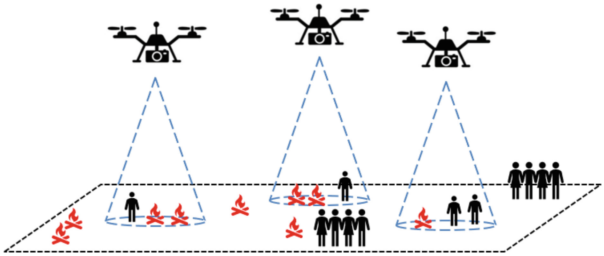

Fig. 1. A fire-affected area with trapped individuals is marked by black dashed lines. Three UAVs monitor the disaster, with their coverage zones shown by blue circular dashed lines. (Color figure online)

a communication network with a constrained number of UAVs has garnered significant attention from researchers.

Research on UAV network deployment can be broadly categorized into two main areas: (i) minimizing the number of UAVs required for deployment, and (ii) maximizing connected coverage with a limited number of UAVs. In the first category, Zhao et al. [29] proposed a centralized algorithm for optimal UAV placement. Sawalmeh et al. [21] introduced an iterative clustering and 3D placement algorithm to maximize coverage with minimal UAVs and transmit power. Zhang et al. [28] minimized UAVs for delay-bounded data collection, offering approximation algorithms with guarantees. Sabzehali et al. [22] optimized UAV counts for backhaul connectivity and full ground coverage, proposing a low-complexity algorithm. In the second category, Zhao et al. [29] developed a distributed motion control algorithm for bi-connected UAV networks. Gupta et al. [9] maximized throughput with a single UAV. Danilchenko et al. [4] addressed NP-hard connectivity constraints with fixed UAVs, proposing approximation algorithms with performance guarantees.

Our work falls into the second category of UAV deployment research, focusing on optimizing network deployment with a limited number of UAVs. The goal is to maximize coverage area and the number of rescued individuals while ensuring network connectivity (see Fig. 1). Connectivity is crucial for collaboration, conflict prevention, real-time adjustments, and data sharing, as highlighted in [14,26]. We model this problem using an undirected graph, where UAVs are nodes, and edges represent communication links. The coverage area and served users are modeled via a submodular prize function on the node set, while connectivity is reflected in the graph's structure. Since the problem is NP-hard, we focus on designing approximation algorithms.

1.1 Related Work

Research on UAV deployment problem primarily falls into two categories: one focuses on deploying networks with the minimum number of UAVs [21,22,28,29], while the other aims to optimize the deployment of a limited number of UAVs to maximize connected coverage.

Zhao et al. [29] proposed a distributed motion control algorithm for optimal UAV deployment. Gupta et al. [9] addressed 3D deployment for a single UAV using alternating optimization. Danilchenko et al. [4] developed approximation algorithms with performance guarantees for various scenarios. When UAV connectivity is ignored, the problem reduces to cardinality-constrained monotone submodular maximization. A $(1 - e^{-1})$-approximation greedy algorithm was proposed by [18], and [14] provided a $(1 - e^{-1})$-approximation when a root node must be included. With connectivity, Kuo et al. [14] proved the MCSC problem is NP-hard and proposed an $\Omega\left(\frac{1}{\sqrt{K}}\right)$-approximation algorithm for real-valued functions, specifically $\frac{1-e^{-1}}{2\sqrt{K}+11}$. They modeled the UAV network as an undirected graph. D'Angelo et al. [3] studied directed information dissemination, proposing a bicriteria algorithm with ratio $\left(\Omega\left(\frac{\varepsilon^3}{\sqrt{B}}\right), 1+\varepsilon\right)$, $\varepsilon \in (0, 1]$ Khuller et al. [12,13] studied the Budgeted Connected Dominating Set (BCDS) problem, aiming to find a connected subset S of at most K nodes to maximize dominated nodes. They proposed a $\frac{1-e^{-1}}{12}$-approximation algorithm. Xu et al. [25] developed an algorithm for maximizing throughput in UAV networks with an approximation ratio of $\frac{1-e^{-1}}{\lfloor\sqrt{K}\rfloor}$. Additionally, [26] introduced the h-hop independent submodular function, proposing a $\frac{1-e^{-1}}{2h+3}$-approximation algorithm for $\sqrt{K} \geq 2h + 3$, where h is a positive integer.

For the Maximum Connected Submodular set function with Budget constraint (MCSB) problem, ignoring connectivity reduces it to the classic knapsack problem, while considering connectivity leads to the maximum connected coverage problem with a budget, widely studied [1,11,16,19]. The MCSC algorithm builds on [14], which focuses on high-marginal-profit nodes within a smaller range using a $(1 - e^{-1})$-approximate algorithm for cardinality constraints. In contrast, [25] considers high-marginal-profit nodes with additional constraints, employing a greedy algorithm for knapsack constraints. While [25] improves the approximation ratio, [14] offers a faster running time. The MCHSC algorithm builds on [26], which maximizes users served by a connected network of K UAVs, mapping objectives to non-negative integers. This approach may limit coverage precision, as coverage sizes are not always integers. To address this, we propose an approximation algorithm for the real-valued MCHSC problem.

1.2 Organization

In Sect. 2, we define submodular functions and related concepts. In Sect. 3, we present the MCSC problem and its approximation algorithm. In Sect. 4, we introduce the MCHSC problem and its approximation algorithm. In Sect. 5, we conduct numerical experiments to validate the effectiveness of the algorithms. Finally, we conclude our paper in Sect. 6.

2 Preliminaries

We are given an undirected connected graph $G = (V, E)$, where V is the set of nodes and E is the set of edges. A path in the graph G is a sequence of distinct vertices $(v_1, v_2, \cdots, v_k)$ with a sequence of undirected edges (v_i, v_{i+1}) for $i \in \{1, 2, \cdots, k-1\}$. For any two nodes u and v in V, a path between u and v with the minimum cost of edges is called the shortest path between u and v. The cost of this shortest path, denoted by $d(u, v)$, represents the distance between u and v in the graph G. In our text, each edge has a cost of one, meaning that the shortest path between u and v is the path with minimum number of edges. For any two non-empty subsets A and B of V, let $d(A, B) = \min\{d(u, v) \mid u \in A, v \in B\}$. For a subset $S \subseteq V$, let $G[S] = (S, E[S])$, where $E[S] = \{(u, v) \in E \mid u, v \in S\}$. Given a tree T in G, let $w(T)$ denote the number of edges in T.

Definition 1. *Given a finite set V, a real-valued function $f : 2^V \rightarrow \mathbb{R}_+$ is defined as a normalized submodular function if it satisfies the following:*

(1) Normalization: $f(\emptyset) = 0$;
(2) Submodularity: $f(A) + f(B) \geq f(A \cup B) + f(A \cap B),\ \forall A, B \subseteq V$.

The submodularity property can also be equivalently expressed as: $f(A \cup \{v\}) - f(A) \geq f(B \cup \{v\}) - f(B)$ for any subsets $A \subseteq B \subseteq V$ and $v \notin B$. Additionally, f is called a monotone submodular function if it satisfies $f(A) \leq f(B)$ for any subsets $A \subseteq B \subseteq V$. Next we restate the h-hop submodular function on a graph $G = (V, E)$ as follows.

Definition 2. *[26] Given a graph $G = (V, E)$, a fnction $f : 2^V \rightarrow \mathbb{R}_+$ is an h-hop submodular function if it satisfies:*

(1) f is a monotone submodular function, and
(2) $f(A \cup B) = f(A) + f(B)$ if $d(A, B) \geq h, \forall A, B \subseteq V$.

3 Approximating MCSC Problem

We consider the MCSC problem and give an improved approximation algorithm for the problem. Given an undirected and connected graph $G = (V, E)$ and a monotone submodular function $f : 2^V \rightarrow \mathbb{R}_+$, a positive integer K, the MCSC problem aims to find a set of nodes S such that $G[S]$ is a connected subgraph of G, maximizing the value of $f(S)$ while ensuring the cardinality of S does not exceed K. Let S_K^* represent the optimal solution of this instance and $OPT = f(S_K^*)$. Additionally, let T_K^* denote the spanning tree of $G[S_K^*]$.

3.1 Algorithm and Performance Guarantees

We propose an approximation algorithm for the MCSC problem in Algorithm 1 inspired by [14]. We introduce a new parameter m, which plays an important role in analyzing our algorithm. The main results are summarized in Theorem 1 and Theorem 2. Before analyzing these two theorems, we firstly establish the following three lemmas. The proofs appear in the full version of this paper.

Algorithm 1. Algorithm for the MCSC problem

Input: Graph $G = (V, E)$, a monotone submodular function $f : 2^V \to \mathbb{R}^{\geq 0}$, positive integer K, m
Output: S.

```
S = ∅;
for r ∈ V do
    V_r = {v | d(r, v) ≤ max{m − 1, ⌊√K⌋}};
    S_r is the solution of the cardinality constrained rooted monotone submodular
    maximization problem with root r must be included [14];
    if f(S_r) > f(S) then
        r* = r;
        S = S_{r*};
    end if
end for
for v ∈ S_{r*} do
    Find the shortest path from v to r*, and add the new nodes in the path to S;
end for
return S.
```

Lemma 1. *Assume $m \leq \lfloor\sqrt{K-1}\rfloor + 1$, Algorithm 1 returns a feasible solution for the MCSC problem.*

Lemma 2. *For any tree T, there always exist $n_t \leq \frac{2w(T)}{m-1} + 1$ subtrees $T^i = (V^i, E^i)$ of T, where $|V^i| \leq m$, $\forall 1 \leq i \leq n_t$, $m \geq 2$, such that $\bigcup_{i=1}^{n_t} V^i = V(T)$ and $\bigcup_{i=1}^{n_t} E^i = E(T)$.*

Lemma 3. *Assume $m = \lfloor\sqrt{K}\rfloor$, Algorithm 1 achieves $\frac{1-e^{-1}}{2\sqrt{K}+11}$-approximation which reduces to the result stated by [14].*

Theorem 1. *Assume that $m = \lfloor\sqrt{K-1}\rfloor + 1$, Algorithm 1 achieves $\frac{1-e^{-1}}{2\sqrt{K-1}+5}$-approximation for the MCSC problem.*

Proof. We first assume that $K \geq 5$. Then, we find that $m = \lfloor\sqrt{K-1}\rfloor + 1 \geq 3$. In this case, we decompose the tree T_K^* into n_t subtrees, ensuring that the number of nodes in each subtree does not exceed m. Therefore, we derive

$$\begin{aligned} n_t &\leq 2\,\frac{K-1}{m-1} + 1 = 2\,\frac{K-1}{\lfloor\sqrt{K-1}\rfloor} + 1 \leq 2\,\frac{K-1}{\sqrt{K-1}-1} + 1 \\ &= 2\left(\sqrt{K-1} + 1 + \frac{1}{\sqrt{K-1}-1}\right) + 1 \leq 2\sqrt{K-1} + 5. \end{aligned}$$

For the case of $1 \leq K \leq 4$, the tree T_K^* contains K nodes, which can be covered by $n_t \leq K \leq 4 < 2\sqrt{K-1} + 5$ subtrees. Let $V' = \arg\max_{i \in \{1,\cdots,n_t\}} f(V^i)$. Since $|V'| \leq m$, and the distance between any two nodes is at most $m-1$. Therefore, $f(S_m^*) \geq f(V')$. By the submodularity of f, we have $OPT = f(V(T_K^*)) = f\left(\bigcup_{i=1}^{n_t} V^i\right) \leq \sum_{i=1}^{n_t} f(V^i) \leq n_t f(V')$. Select any node $r \in S_m^*$, where $S_m^* \subseteq V_r$

in Algorithm 1 and $|S^*_m| = m$. Then, we have $f(S_r) \geq (1-e^{-1})OPT' \geq (1-e^{-1})f(S^*_m)$, where OPT' denotes the optimal value of the maximum rooted submodular function f with constraints $|S| \leq m$, and $r \in S$ such that $f(S)$ is maximized. Therefore, we have $f(S) \geq f(S_{r^*}) \geq f(S_r) \geq (1-e^{-1})f(S^*_m) \geq (1-e^{-1})f(V') \geq (1-e^{-1})\frac{OPT}{n_t} \geq \frac{1-e^{-1}}{2\sqrt{K-1}+5}OPT$. □

Assume parameter m are chosen by $m = \alpha\sqrt{K} > \lfloor\sqrt{K-1}\rfloor + 1$, there exists a bicriteria approximation as follows:

Theorem 2. *For the MCSC problem, Algorithm 1 obtains a bicriteria ratio of* $\left(\frac{(1-e^{-1})\alpha}{2\sqrt{K}+3\alpha}, \alpha^2\right)$ *when* $m = \alpha\sqrt{K} > \lfloor\sqrt{K-1}\rfloor + 1$.

Proof. Let $m = \alpha\sqrt{K}$, where $m > \lfloor\sqrt{K-1}\rfloor + 1$. It follows that $m - 1 \geq \lfloor\sqrt{K-1}\rfloor + 1 \geq \lfloor\sqrt{K}\rfloor$. Consequently, $V_r = \{v \mid d(v,r) \leq m-1\}$ in line 4 of Algorithm 1. Additionally, we have $\alpha = \frac{m}{\sqrt{K}} \geq \frac{\lfloor\sqrt{K-1}\rfloor+2}{\sqrt{K}} \geq \frac{\sqrt{K-1}+1}{\sqrt{K}} \geq 1$. If $\alpha = 1$, then $\sqrt{K}$ must be an integer. However, this would imply $\sqrt{K} = \lfloor\sqrt{K}\rfloor \leq \lfloor\sqrt{K-1}\rfloor + 1$, which is impossible. Therefore, $\alpha > 1$ and $m = \alpha\sqrt{K} \geq 2$ since m is an integer. We obtain

$$\begin{aligned} n_t &= 2\,\frac{K-1}{m-1} + 1 = 2\,\frac{K-1}{\alpha\sqrt{K}-1} + 1 \\ &= 2\left(\frac{\sqrt{K}}{\alpha} + \frac{1}{\alpha^2}\right) + 2\frac{\frac{1}{\alpha^2}-1}{\alpha\sqrt{K}-1} + 1 \\ &\leq 2\frac{\sqrt{K}}{\alpha} + 3. \end{aligned}$$

Similar to the proof of Theorem 1, we derive $f(S) \geq f(S_{r^*}) \geq f(S_r) \geq (1-e^{-1})f(S^*_{\alpha\sqrt{K}}) \geq (1-e^{-1})\frac{OPT}{n_t} \geq \frac{(1-e^{-1})\alpha}{2\sqrt{K}+3\alpha}OPT$. The number of nodes in the solution S satisfies $|S| \leq m + (m-1)(m-2) = \alpha\sqrt{K} + (\alpha\sqrt{K}-1)(\alpha\sqrt{K}-2) \leq \alpha^2 K - 2\alpha\sqrt{K} + 2 \leq \alpha^2 K$. Therefore, Algorithm 1 derives a bicriteria ratio $\left(\frac{(1-e^{-1})\alpha}{2\sqrt{K}+3\alpha}, \alpha^2\right)$ when $m = \alpha\sqrt{K} > \lfloor\sqrt{K-1}\rfloor + 1$. □

4 Approximating the MCHSC Problem

In this section, we consider the MCHSC problem and propose an approximation algorithm for this problem.

Algorithm 2. Generalized Prize Assignment Algorithm

Input: An undirected, connected graph $G = (V, E)$, an h-hop submodular function $f : 2^V \to \mathbb{R}_+$ on G, a starting node $v \in V$.
Output: A prize mapping function, i.e., $p : V \to \mathbb{R}_+$.

```
p(v) = f({v});
A = {v};
U = V \ A;
while U ≠ ∅ do
    u = arg max{f(A ∪ {v}) − f(A) | v ∈ U};
    p(u) = f(A ∪ {u}) − f(A);
    A = A ∪ {u};
    U = U \ {u};
end while
return p : V → ℝ^{≥0}.
```

4.1 Node Number Minimization Quota Problem

Inspired by the classic quota problem [1], we firstly state a Node number Minimization Quota (NMQ) problem as follows. Given an undirected and connected graph $G = (V, E)$, an additive function $p : V \to \mathbb{R}_+$, and a positive number $Q > 0$, the goal is to find a tree in G with the minimum number of nodes such that the total prize satisfies $p(V(T)) \geq Q$. For convenience, we use the shortcut $p(T) = p(V(T)) = \sum_{v \in V(T)} p(v)$.

4.2 Warm-Up Algorithms

In this section, we introduce two algorithms that are utilized in the algorithm for the MCHSC problem. Given an instance of the MCHSC problem with $G = (V, E)$, a function f, and a positive integer K, let $n = |V|$ and $m = |E|$.

• *Generalized Prize Assignment Algorithm:* We can use Algorithm 2 inspired by [26] to assign prizes to the nodes in V. We extend the initial prize assignment algorithm, no longer requiring the prize allocation function to be an integer function.

• *Bicriteria Approximation for the NMQ problem:* An algorithm is a bicriteria (α, β)-approximation for the NMQ problem if for any instance of the problem, it returns a solution T such that $|T| \leq \alpha |T^*|$ and $p(T) \geq \beta Q$, where T^* is the optimal tree.

Our algorithm mainly follows by the idea proposed by Bateni et al. in [1]. The pseudo-codes are summarized in Algorithm 3 and the approximation ratio is $(5, 1 - 2\varepsilon)$ for any $\varepsilon > 0$.

Lemma 4. *[1] Given any α-approximation algorithm for the k-ST problem, there exists an $(\alpha, 1 - 2\varepsilon)$-approximation algorithm of the NMQ problem.*

Corollary 1. *Algorithm 3 is a $(5, 1-2\varepsilon)$-approximation algorithm for the NMQ problem.*

Lemma 5. *The tree* $T = T^* \bigcup (\cup_{i=1}^{K} P_i)$ *with fewer than* $(K-1)h+1$ *nodes can be decomposed into* $t \leq \alpha(h+1)+1$ *subtrees* $T^j = (V^j, E^j)$*. Each subtree satisfies* $|V^j| \leq \lfloor \frac{K}{\alpha} \rfloor$ *for all* $j \in \{1, 2, \cdots, t\}$ *when* $K > \alpha^2 h^2 + \alpha^2 h - \alpha$ *and* $\alpha > 1$*, where* $\alpha(h+1)$ *is a positive integer.*

Algorithm 3. Approximation Algorithm for the NMQ problem [1,2]

Input: An undirected, connected graph $G = (V, E)$, an additive function $p : V \to \mathbb{R}_+$, a positive number $Q > 0$, and $\varepsilon > 0$.
Output: A tree T in G, such that $|V(T)|$ is minimized and $p(T) \geq Q$.

$V(T) \leftarrow \emptyset, E(T) = \emptyset, R = \emptyset, E_0 = \emptyset, n = |V|$;
for $e \in E$ **do**
  $c(e) = 1$;
end for
for $v \in V$ **do**
  $n_v = \left\lceil \dfrac{np(v)}{\varepsilon Q} \right\rceil$;
  Let $R_v = \{r_1, r_2, \cdots, r_{n_v}\}$;
  for $r \in R_v$ **do**
    Connect the node v and node r, and let $c((v, r)) = 0$;
    $E_v = E_v \cup (v, r)$;
  end for
  $R = R \cup R_v$;
  $E_0 = E_0 \cup E_v$;
end for
Let $G' = (V', E')$, $V' = V \cup R, E' = E \cup E_0$;
Let $k = \left\lfloor \dfrac{n}{\varepsilon} \right\rfloor$, V is the set of steiner nodes, R is the set of terminals, $c : E' \to \{0, 1\}$;
Obtain the tree T' by using 5-approximation algorithm of k-ST problem[2] to the instance $I' =< G' = (V', E'), R, c, k >$;
$V(T) = V(T') \cap V$, $E(T) = E(T') \cap E$;
Return $T = (V(T), E(T))$.

4.3 Algorithms and Performance Guarantees

We propose our algorithm for the MCHSC problem in Algorithm 4. The main results are summarized as follows.

Theorem 3. *When* $K > 25h(h+1) - 5$ *and fix* $\varepsilon > 0, \delta > 0$*, Algorithm 4 achieves* $(1-2\varepsilon)\left(\frac{1-e^{-1}}{5(h+1)+1} - \delta\right)$*-approximation for the MCHSC problem.*

Proof. Assume that K and h satisfy $K > 25h(h+1) - 5$. We focus on the iteration in which z is the starting node firstly. According to Lemma 5, there exists a tree T' satisfying $|V(T')| \leq \lfloor \frac{K}{5} \rfloor$ and $p(T') \geq \frac{1-e^{-1}}{5(h+1)+1} OPT$.

In the case of $f(\{z\}) \leq q \leq \frac{1-e^{-1}}{5(h+1)+1}OPT$, the optimal value of the NMQ problem satisfies $OPT_Q \leq \left\lfloor \frac{K}{5} \right\rfloor$. After applying Algorithm 3, we obtain a tree T_q such that $|V(T_q)| \leq 5OPT_Q \leq K$. It implies that $|V(T_q)| > K$ if and only if $q > \frac{1-e^{-1}}{5(h+1)+1}OPT$. In Algorithm 4, the upper bound *right* of search interval $[left, right]$ is a value of q such that $|V(T_q)| > K$, and $right > \frac{1-e^{-1}}{5(h+1)+1}OPT$ always holds. At the end of any iteration in Algorithm 4, the lower bound *left* and the upper bound *right* satisfy $left + \delta \geq right$. Therefore, $left \geq right - \delta > \frac{1-e^{-1}}{5(h+1)+1}OPT - \delta > \left(\frac{1-e^{-1}}{5(h+1)+1} - \delta\right)OPT$, where we assume $OPT \geq 1$. As a result, the iteration with the starting node z will return a tree T_q satisfying $p(T_q) \geq (1-2\varepsilon)left > (1-2\varepsilon)\left(\frac{1-e^{-1}}{5(h+1)+1} - \delta\right)OPT$.

In any iteration i, assume the tree searched in this iteration by the Algorithm 4 is T_i. At the end of our analysis, we will establish the relationship between $f(T_i)$ and $p(T_i)$. For any node $v \in V$, let D_v denote the set of nodes that have been assigned a prize before v in this iteration. Let $D'_v = D_v \cap V(T_i)$. Therefore, we have

$$\begin{aligned} p(T_i) = \sum_{v \in V(T_i)} p(v) &= \sum_{v \in V(T_i)} f(D_v \cup \{v\}) - f(D_v) \\ &\leq \sum_{v \in V(T_i)} f(D'_v \cup \{v\}) - f(D'_v) = f(T_i). \end{aligned}$$

At the end of Algorithm 4, the solution satisfies $f(S) = \max_i\{f(T_i)\} \geq (1-2\varepsilon)\left(\frac{1-e^{-1}}{5(h+1)+1} - \delta\right)OPT$. $\square$

A feasible solution of the MCHSC problem: Let T^* denote the spanning tree of $G[S^*]$, where $S^* = \{s_1, s_2, \cdots, s_K\}$. Define $z = \arg\max\{f(\{v\}) \mid v \in S^*\}$, $S_{h-1} = \{v \mid d(v, S^*) \leq h-1\}$, and $S_h = \{v \mid d(v, S^*) > h\}$. It can be observed that if a node is in S^*, then it must also be in S_{h-1} since $d(v, S^*) = 0 \leq h-1$. Algorithm 2 assigns prizes to nodes in V with the starting node z, and label the nodes as $\{v_1, v_2, \cdots, v_n\}$ according to the order of prize assignment, where $\{v_1\} = \{z\}$. Search for the first K nodes in the set S_{h-1}, and denote their subscripts as $l_1, \cdots, l_K$. Let $Z_K = \{v_{l_1}, v_{l_2}, \cdots, v_{l_K}\}$, and represent these nodes as $\{z_1, z_2, \cdots, z_K\}$. This ensures $p(Z_K) \geq (1-e^{-1})OPT$. Each node $z_i \in Z_K$ has a path P_i to any node in T^*. Let $T = T^* \bigcup (\cup_{i=1}^K P_i)$. Therefore, the number of the edges in T is bounded by

$$\begin{aligned} |E(T)| \leq |E(T^*)| + \sum_{i=1}^{K} |P_i| &= |E(T^*)| + \sum_{i=2}^{K} |P_i| \\ &\leq (K-1) + (K-1)(h-1) = (K-1)h. \end{aligned}$$

The first inequality accounts for possible intersections between the edges in $E(T^*)$ and P_i, the second equality is due to $E(P_1) = 0$ since $\{z_1\} = \{z\} \subseteq S^*$, the third inequality reflects that P_i is the shortest path between some node in S_{h-1} and any node in T^*, ensuring $|P_i| \leq h-1$. Simultaneously, we have

Algorithm 4. Approximation Algorithm for the MCHSC problem

Input: Graph $G = (V, E)$, an h-hop submodular function $f : 2^V \to \mathbb{R}_{\geq 0}$, a positive integer K, small positive numbers ε, δ.
Output: A feasible subset $S \subseteq V$.

$S \leftarrow \emptyset$;
for $v \in V$ **do**
  if $f(\{v\}) > 0$ **then**
    Apply Algorithm 2 to get prizes for each node with the starting node v;
  else
    continue; /* consider the next node being the starting node.*/
  end if
  Let $left \leftarrow f(\{v\})$ and $right \leftarrow f(V)$
  while $left + \delta < right$ **do**
    $q \leftarrow \dfrac{left + right}{2}$;
    Apply Algorithm 3 to find a tree T_q in G, where $p(v) \leftarrow p(v), Q \leftarrow q$;
    if $|V(T_q)| \leq K$ **then**
      $left \leftarrow q$;
    else
      $right \leftarrow q$;
    end if
  end while
  $q \leftarrow left$;
  Apply Algorithm 3 to find a tree T_q in G, where $p(v) \leftarrow p(v), Q \leftarrow q$;
  if $f(V(T_q)) > f(S)$ **then**
    Let $S \leftarrow V(T_q)$;
  end if
end for
return S

$|V(T)| = |E(T)| + 1 \leq (K-1)h + 1$. Since the nodes in Z_K are included in $V(T)$, we have $p(T) \geq p(Z_K) \geq (1 - e^{-1})OPT$. The tree T may not necessarily be a feasible solution, but we can decompose it to obtain a feasible solution of the MCHSC problem.

According to Lemma 5, let $T' = (V', E')$ with $V' = \arg\max\{p(V^j) \mid j \in \{1, 2, \cdots, t\}\}$ where the starting node is z. The additivity of p implies $p(T) = \sum_{v \in V(T)} p(v) \leq \sum_{j=1}^{t} p(T^j) \leq tp(V')$. Additionally, $|V(T')| \leq \left\lfloor \frac{K}{\alpha} \right\rfloor$ when $K > \alpha^2 h^2 + \alpha^2 h - \alpha$. Let $\alpha = 5$. When $K > 25h(h+1) - 5$, there exists a tree T' satisfing $p(T') \geq \frac{1-e^{-1}}{5(h+1)+1} OPT$ and $|V(T')| \leq \left\lfloor \frac{K}{5} \right\rfloor$.

In Algorithm 4, each node with $f(\{v\}) > 0$ is used as the starting node once. Since $z = \arg\max\{f(\{v\}) \mid v \in S^*\}$, it follows that $f(S^*) \leq Kf(\{z\})$. If $f(\{z\}) = 0$, then $OPT = f(S^*) = 0$, and this case can be ignored. Therefore, the node z will be used as the starting node once. As a result, there exists a tree T' such that $p(T') \geq \frac{1-e^{-1}}{5(h+1)+1} OPT$ and $|V(T')| \leq \left\lfloor \frac{K}{5} \right\rfloor$.

5 Numerical Experiments

Consider the MCSC and MCHSC problems with K UAVs in a $4000 \times 4000\,\mathrm{m}^2$ area serving 200 users. The users have real-valued weights and follow a fat-tailed, nonuniform distribution. Each UAV hovers at $H = 300$ m, with inter-UAV communication range $R = 800$ m and user-UAV communication range r'. The coverage range of each UAV is $r = \sqrt{r'^2 - H^2}$. We divide the $4000 \times 4000\,\mathrm{m}^2$ area into 64 small squares of 500 m side length, as shown in Fig. 2. UAVs can be positioned at 49 candidate points at intersections of dashed lines, represented by coordinates (x_i, y_i) for $i \in \{1, 2, \cdots, 49\}$, ignoring altitude. Users are represented as points (x, y) in the same plane. A user is served if $(x - x_i)^2 + (y - y_i) \leq r^2$. The submodular function f is then defined. We solve this problem using Algorithm 1 and a simplified version of Algorithm 4 (running steps 2–23 from a single node, following HopAlg [26]).

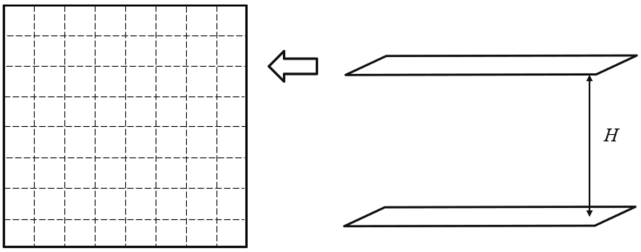

Fig. 2. A square with the size $4000 \times 4000\,\mathrm{m}^2$ and the divided method

In Algorithm 1, we set $m = \lfloor \sqrt{K - 1} \rfloor + 1$, ensuring a feasible solution with an approximation ratio of $\frac{1 - e^{-1}}{2\sqrt{K-1}+5}$. In Algorithm 4, real values are rounded down to integers, as the algorithm operates on integer fields. Additionally, we set $\varepsilon = 0.1$, $\delta = 0.02$. Initially, we set $K = 8$ UAVs and vary the coverage range from 150 to 350. We compare the coverage weight of Algorithm 1 and Algorithm 4 with GreedyAlg, RandAlg, HopAlg [26], and MCSAlg [14], as shown in Fig. 3. The corresponding running times are illustrated in Fig. 4. Algorithm 1 consistently outperforms MCSAlg, achieving up to 85.1% higher coverage weight at $r = 200$ m, with similar running times. Thus, Algorithm 1 is superior to MCSAlg. Algorithm 4 outperforms HopAlg in some ranges, with up to 30.3% higher weight at $r = 300$ m. Although its running times are higher, Algorithm 4 directly handles real-valued scenarios. Next, we set the UAV coverage range to 150 m and vary the number of UAVs K from 10 to 18 in steps of 2. We compare the coverage weight of Algorithm 1 and Algorithm 4 with GreedyAlg, RandAlg, HopAlg [26], and MCSAlg, as shown in Fig. 5. Algorithm 1 consistently outperforms MCSAlg, achieving up to 15.8% higher coverage weight with similar running times, demonstrating its superiority. Algorithm 4 outperforms HopAlg in some cases, with up to 8% higher weight at $K = 10$. Although its running times are longer, Algorithm 4 directly handles real-valued scenarios, unlike HopAlg, which requires adjustments. For Algorithm 4, with $h = 2$ as derived in [26], it provides a $\left(\frac{1-e^{-1}}{20} - 0.016\right)$-approximation solution when K and h meet the specified conditions.

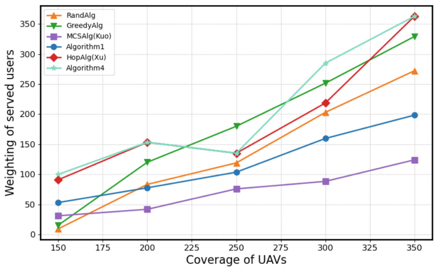

Fig. 3. The performance of different algorithms by varying r from 150 m to 350 m, R = 800 m, 200 users and K = 8.

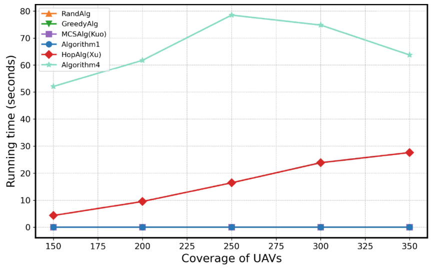

Fig. 4. The performance of different algorithms by varying the coverage range r of a UAV from 150 m to 350 m while fixing R = 800 m, when there are 200 users. K = 8.

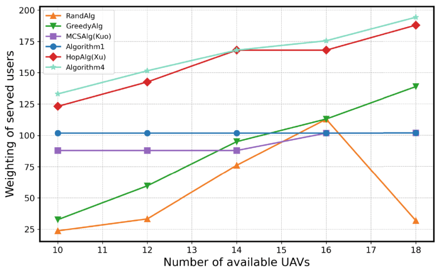

Fig. 5. The performance of different algorithms by varying the number of available UAVs K from 10 to 18 while fixing $R = 800$ m, when there are 200 users and the coverage range of UAVs is 150 m.

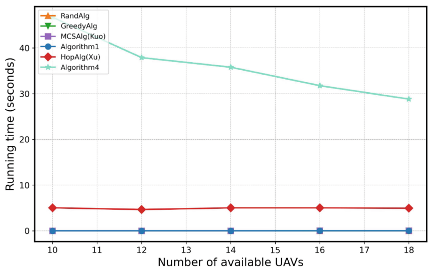

Fig. 6. The performance of different algorithms by varying the number of available UAVs K from 10 to 18 while fixing $R = 800$ m, when there are 200 users and the coverage range of UAVs is 150 m.

6 Conclusion

We addressed the maximum connected coverage problem with limited UAVs by formulating as the MCSC and the MCHSC problems, respectively. For the MCSC problem, we proposed an improved approximation algorithm by introducing a new tree-decomposition technique. For the special setting of MCHSC problem, we provided an improved algorithm with partial additive parameter. Future work could explore extending these algorithms to dynamic environments, where user positions or network conditions change over time. Additionally, incorporating constraints like energy consumption or UAV mobility patterns could enhance the proposed solutions. This would further improve their real-world applicability.

Acknowledgments. The first three authors are supported by Beijing Natural Science Foundation (Nos. Z220004, IS24001). The fourth author is supported by Natural Science Foundation of China (Nos. 12101587, 12201619).

References

1. Bateni, M.H., Hajiaghayi, M.T., Liaghat, V.: Improved approximation algorithms for (budgeted) node-weighted Steiner problems. SIAM J. Comput. **47**(4), 1275–1293 (2018)
2. Chudak, F.A., Roughgarden, T., Williamson, D.P.: Approximate k-MSTs and k-Steiner trees via the primal-dual method and Lagrangean relaxation. Math. Program. **100**, 411–421 (2004)
3. D'Angelo, G., Delfaraz, E., Gilbert, H.: Budgeted out-tree maximization with submodular prizes. arXiv preprint arXiv:2204.12162 (2022)
4. Danilchenko, K., Nutov, Z., Segal, M.: Covering users with QoS by a connected swarm of drones: graph theoretical approach and experiments. IEEE/ACM Trans. Netw. (2022)
5. Demaine, E.D., Hajiaghayi, M.T., Klein, P.N.: Node-weighted steiner tree and group steiner tree in planar graphs. In: International Colloquium on Automata, Languages, and Programming, pp. 328–340 (2009)
6. Erdelj, M., Natalizio, E.: UAV-assisted disaster management: applications and open issues. In: International Conference on Computing, Networking and Communications, pp. 1–5 (2016)
7. Garg, N.: Saving an epsilon: a 2-approximation for the k-MST problem in graphs. In: Proceedings of ACM Symposium on Theory of Computing, pp. 396-402 (2005)
8. Goemans, M.X., Williamson, D.P.: A general approximation technique for constrained forest problems. SIAM J. Comput. **24**(2), 296–317 (1995)
9. Gupta, N., Agarwal, S., Mishra, D.: UAV deployment for throughput maximization in a UAV-assisted cellular communications. In: Proceedings of International Symposium on Personal, Indoor and Mobile Radio Communications, pp. 1055-1060 (2021)
10. Huang, L., Li, J., Shi, Q.: Approximation algorithms for the connected sensor cover problem. Theoret. Comput. Sci. **809**, 563–574 (2020)
11. Johnson, D.S., Minkoff, M., Phillips, S.: The prize collecting steiner tree problem: theory and practice. In: Proceedings of ACM - SIAM Symposium on Discrete Algorithms, vol. 1(0.6), p. 4 (2000)
12. Khuller, S., Purohit, M., Sarpatwar, K.K.: Analyzing the optimal neighborhood: algorithms for budgeted and partial connected dominating set problems. In: Proceedings of ACM-SIAM Symposium on Discrete Algorithms, pp. 1702-1713 (2014)
13. Khuller, S., Purohit, M., Sarpatwar, K.K.: Analyzing the optimal neighborhood: Algorithms for partial and budgeted connected dominating set problems. SIAM J. Discret. Math. **34**(1), 251–270 (2020)
14. Kuo, T.W., Lin, K., Tsai, M.J.: Maximizing submodular set function with connectivity constraint: theory and application to networks. IEEE/ACM Trans. Netw. **23**(2), 533–546 (2014)

15. Liu, M., Yang, J., Gui, G.: DSF-NOMA: UAV-assisted emergency communication technology in a heterogeneous Internet of Things. IEEE Internet Things J. **6**(3), 5508–5519 (2019)
16. Moss, A., Rabani, Y.: Approximation algorithms for constrained node weighted steiner tree problems. SIAM J. Comput. **37**(2), 460–481 (2007)
17. Nguyen, L.D., Nguyen, K.K., Kortun, A., Duong, T.Q.: Real-time deployment and resource allocation for distributed UAV systems in disaster relief. In: Proceedings of International Workshop on Signal Processing Advances in Wireless Communications, pp. 1-5 (2019)
18. Nemhauser, G.L., Wolsey, L.A., Fisher, M.L.: An analysis of approximations for maximizing submodular set functions-I. Math. Program. **14**, 265–294 (1978)
19. Rabani, Y., Scalosub, G.: Bicriteria approximation tradeoff for the node-cost budget problem. ACM Trans. Algorithms **5**(2), 1–14 (2009)
20. Sviridenko, M.: A note on maximizing a submodular set function subject to a knapsack constraint. Oper. Res. Lett. **32**(1), 41–43 (2004)
21. Sawalmeh, A., Othman, N.S., Liu, G., Khreishah, A., Alenezi, A., Alanazi, A.: Power-efficient wireless coverage using minimum number of UAVs. Sensors **22**(1), 223 (2021)
22. Sabzehali, J., Shah, V.K., Fan, Q., Choudhury, B., Liu, L., Reed, J.H.: Optimizing number, placement, and backhaul connectivity of multi-UAV networks. IEEE Internet Things J. **9**(21), 21548–21560 (2022)
23. Wang, B., Sun, Y., Sun, Z., Nguyen, L.D., Duong, T.Q.: UAV-assisted emergency communications in social IoT: a dynamic hypergraph coloring approach. IEEE Internet Things J. **7**(8), 7663–7677 (2020)
24. Williamson, D.P., Shmoys, D.B.: The design of approximation algorithms. Cambridge university press (2011)
25. Xu, W., et al.: Throughput maximization of UAV networks. IEEE/ACM Trans. Netw. **30**(2), 881–895 (2021)
26. Xu, W., et al.: An approximation algorithm for the h-Hop independently submodular maximization problem and its applications. IEEE/ACM Trans. Netw. **31**(3), 1216–1229 (2022)
27. Zhao, N., et al.: UAV-assisted emergency networks in disasters. IEEE Wirel. Commun. **26**(1), 45–51 (2019)
28. Zhang, J., et al.: Minimizing the number of deployed UAVs for delay-bounded data collection of IoT devices. In: IEEE INFOCOM 2021-IEEE Conference on Computer Communications, pp. 1-10 (2021)
29. Zhao, H., Wang, H., Wu, W., Wei, J.: Deployment algorithms for UAV airborne networks toward on-demand coverage. IEEE J. Sel. Areas Commun. **36**(9), 2015–2031 (2018)