High-Confidence Computing 4 (2024) 100208

Contents lists available at ScienceDirect

High-Confidence Computing

journal homepage: www.sciencedirect.com/journal/high-confidence-computing

Research article

H-hop independently submodular maximization problem with curvature

Yang Lv[a], Chenchen Wu[b], Dachuan Xu[a], Ruiqi Yang[a,*]

[a] Beijing Institute for Scientific and Engineering Computing, Beijing University of Technology, Beijing 100124, China
[b] College of Science, Tianjin University of Technology, Tianjin 300382, China

ARTICLE INFO

Article history:
Received 26 July 2023
Revised 10 November 2023
Accepted 14 December 2023
Available online 2 February 2024

Keywords:
Internet of things
UAV communication
Approximation algorithms

ABSTRACT

The Connected Sensor Problem (CSP) presents a prevalent challenge in the realms of communication and Internet of Things (IoT) applications. Its primary aim is to maximize the coverage of users while maintaining connectivity among K sensors. Addressing the challenge of managing a large user base alongside a finite number of candidate locations, this paper proposes an extension to the CSP: the h-hop independently submodular maximization problem characterized by curvature α. We have developed an approximation algorithm that achieves a ratio of $\frac{1-e^{-\alpha}}{(2h+3)\alpha}$. The efficacy of this algorithm is demonstrated on the CSP, where it shows superior performance over existing algorithms, marked by an average enhancement of 8.4%.

1. Introduction

The maximization of h-hop independently submodular functions with curvature is widely applicable in real-world scenarios. This problem is particularly relevant in two areas: addressing communication challenges during natural disasters and enhancing IoT sensor detection capabilities.

Natural disasters can result in irreversible loss of life and economic damage. When faced with large-scale catastrophes, communication system disruptions severely hinder rescue efforts and make it challenging to provide timely assistance to those in need. The immediate priority is to take effective measures to ensure the reliability and timeliness of rescue communications.

Unmanned aerial vehicles (UAVs) act as airborne base stations, enabling the swift restoration of communication in disaster-stricken areas [1–3]. Unlike traditional base stations, UAVs have the advantage of adaptively changing their altitude, navigating around obstacles, and enhancing the possibility of establishing line-of-sight (LoS) communication links with users. This flexibility makes them particularly suitable for unexpected situations like natural disasters, enabling them to deliver essential services to individuals in distress. In July 2021, Mishhe County in Zhengzhou was submerged by two rivers, leaving over 20,000 residents stranded. The Wing Loong UAV flew for four and a half hours into these isolated areas, restoring communication for eight hours [4]. In the future, as UAVs become lighter and battery capacity continues to improve, autonomous aerial rescue missions are poised to become mainstream.

The Connected Sensor Problem (CSP) is a classic and well-studied issue [5]. It involves selecting a connected subgraph from a set of sensor candidates to place sensors optimally. The goal is to maximize individual coverage while ensuring communication between the sensors. These sensors can also be conceptualized as UAVs. The more individuals they cover, the greater the chance for people to access the internet for rescue requests, thereby enhancing survival prospects. In communication and disaster relief scenarios, these sensors, mounted on UAVs, serve as airborne base stations. The objective is to maximize user coverage while maintaining sensor connectivity. The placement of UAVs is strategically determined based on population distribution and location, with a higher concentration of UAVs in densely populated central areas to ensure optimal device connectivity. Conversely, fewer UAVs are deployed in sparsely populated. The primary optimization goal in CSP is to ensure maximal user coverage.

The coverage issue inherently represents a submodular function, as adding additional UAVs results in diminishing returns in terms of new user coverage. In large-scale disaster relief, incorporating curvature considerations provides a more robust theoretical basis. Typically, each UAV has a limited coverage area, and it is unlikely that two UAVs can cover the same users if placed sufficiently far apart. This characteristic is common and is referred to in this paper as ‘h-hop Independence.’ The primary aim of this paper is to frame the problem as a submodular maximization challenge with curvature and h-hop independence and to develop a corresponding approximation algorithm.

Another significant application of maximizing h-hop independently submodular functions with curvature lies in IoT networks.

* Corresponding author.
E-mail address: yangruiqi@bjut.edu.cn (R. Yang).

https://doi.org/10.1016/j.hcc.2024.100208

In industrial environments, strategically placing K connected sensors at selected points forms a communication subnet, enabling these sensors to monitor a larger number of machines.

1.1. Main contributions

- We have expanded the approximation ratio to $\frac{1-e^{-\alpha}}{(2h+3)\alpha}$, enabling the algorithm to achieve better performance, particularly in scenarios requiring extensive coverage. This development ensures that our approach is competitive with, or even superior to, other methods in various contexts.
- Through numerical experiments, we have compared our algorithm with other heuristic approaches to evaluate its performance and effectiveness. Our findings indicate that the proposed approximation algorithm outperforms the optimal algorithm by 8.4%.

The structure of this paper is as follows: Section 2 reviews previous relevant research. Section 3 defines necessary symbols and background knowledge. Section 4 presents the h-Hop independently submodular maximization problem with curvature, including an approximation algorithm and its analysis. Section 5 details the positive outcomes from our numerical experiments. Section 6 concludes the paper.

2. Related works

Research in UAV communications addresses several key aspects. Firstly, the energy consumption of UAVs [6–8] is crucial due to their limited battery capacity, necessitating efficient power management for multiple functions such as flight, storage, computation, and communication. Secondly, reducing UAV latency [9,10] is vital, as UAV communication often depends on a network of relay UAVs. An excessive number of relays can lead to extended link distances, causing delays and potential network failures. Thirdly, the calculation of UAV trajectories [11,12] is important for establishing communication links with multiple users. Lastly, the strategic positioning of UAVs [13–15] is essential, especially in disaster relief, to optimize user data throughput and connectivity. This paper focuses on the deployment strategies of UAVs, considering their energy consumption and trajectories as fixed once deployed.

The maximization of submodular functions poses a significant challenge in computation and communication. It necessitates the utilization of various combinatorial optimization techniques. Nemhauer [16] proposed that under cardinality constraints, the submodular function can obtain an approximation ratio of $1-1/e$ using the greedy algorithm, which proves that the approximation ratio is compact. Based on this conclusion, better range and wider approximation than expected to be found. Conforti and Cornuejols [17] defined the total curvature α. They proved the approximation ratio of $\frac{1}{1+\alpha}$ with greedy algorithm subject to a single matroid constraint. In the particular scenario of a uniform matroid, it was demonstrated that the greedy algorithm achieves a $\frac{1}{\alpha}(1-e^{-\alpha})$ approximation guarantee. Vondrak [18] explored the application of the continuous greedy algorithm within the context of bounded curvature. This paper introduced curvature with respect to the optimum, when f has curvature at most α, continuous greedy algorithm has $\frac{1}{\alpha}(1-e^{-\alpha})$. Adding curvature on top of submodularity can lead to better solutions for large-scale CSP problems, resulting in superior approximations compared to the previous $1-1/e$ bound.

There are also some studies on the maximization of submodular functions under connectivity constraints. Kuo [19] examined the challenge of deploying K wireless routers in a network, aiming to maximize the submodular function of the deployed routers while adhering to the connectivity constraints imposed by the presence of K routers within the subnet. They proposed an algorithm with an approximation ratio of $\frac{1-1/e}{5(\sqrt{K}+1)}$. Xu made significant improvements to the tree segmentation method, reducing the overall approximation ratio to $\frac{1-1/e}{\lfloor\sqrt{K}\rfloor}$ [14]. After adding the h-hop property to the monotonic submodular function, Xu [15] proved that the overall approximation ratio became a constant approximation ratio with $\frac{1-1/e}{2h+3}$. In this paper, within a larger scope, the submodular curvature is typically not equal to 1. Introducing a curvature parameter can enhance the overall algorithm's approximation ratio.

In the realm of IoT, system interconnectivity is enabled through pervasive computing, but it faces challenges due to the Connected Sensor Problem. A single compromised sensor can significantly reduce overall network connectivity. To mitigate this, sensors must employ robust security protocols and applications [20]. Furthermore, privacy concerns arise if data traverses through monitored intermediate nodes. A dual masking encryption approach [21] is necessary to protect end-user privacy. In the CSP context, maintaining the privacy and integrity of sensor communications is critical to prevent network paralysis. Thus, privacy considerations are integral to the CSP problem.

3. Problem formulation

This section transforms the problem into an optimization framework, creating symbolic representations for the variables involved and explaining scenarios corresponding to the properties.

3.1. System model

We deal with a connected undirected graph, denoted as $G=(V,E)$. In practical scenarios, the set V represents potential locations for UAVs, while the edges in E indicate that two nodes are capable of communicating with each other. Therefore, E is a subset of pairs from $V\times V$. The variable n signifies the number of nodes, which can be expressed as $n=|V|$. Additionally, we use $d(u,v)$ to represent the shortest path length between nodes u and v. To calculate the distance between any two subsets A and B in V, denoted as $d(A,B)$, we take the minimum over all possible pairs of nodes, one from A and the other from B, using the formula $d(A,B)=\min_{u\in A,v\in B}d(u,v)$. In other words, $d(A,B)$ represents the closest distance between any point in subset A and any point in subset B.

The function to be optimized is denoted as $f:2^V\rightarrow\mathbb{Z}^{\geq 0}$, where f is a monotone submodular function with curvature α. This submodular function exhibits the property of monotone decreasing, which implies that the marginal gain of the function. The marginal gain is denoted as $f_A(B)$, where $f_A(B)=f(A\cup B)-f(B)$: $f(\emptyset)=0$ because if no UAV candidate points are chosen to place UAVs, then no users are covered, and the result is 0.

For any subsets $A\subseteq B\subseteq V$, we have $f(A)\leq f(B)$, which illustrates the property of monotonicity. This is because adding more UAVs can cover more users, making it an intuitively evident property when designing f.

For any subsets $A\subseteq B\subseteq V$ and $v\notin B$, the inequality $f(A\cup\{v\})-f(A)\geq f(B\cup\{v\})-f(B)$ holds, representing the property of submodularity. Submodularity is a classic property in CSP problems, representing diminishing marginal returns for placing UAVs.

For any subsets $A\subseteq V$ and $B\subseteq V$ such that $d(A,B)\geq h$, the following equation holds: $f(A)+f(B)=f(A\cup B)$, illustrating the concept of h-hop independence. When the distance between sets A and B is greater than or equal to h, meaning that sets A

and B are sufficiently far apart, two UAVs cannot simultaneously cover a single user. Therefore, $f(A)+f(B)=f(A\cup B)$ holds, and this property is stronger than submodularity when $d(A,B)\geq h$. $\alpha = 1-\min_{i\in V}\frac{f_i(V\setminus\{i\})}{f(\{i\})}$, where α represents the curvature of the function f. It measures the proportion of users at a location that can only be covered by a single UAV. In large-scale scenarios, a higher α value indicates a greater proportion of such users.

Xu [14] introduced an approximation algorithm with a ratio of $\frac{1-1/e}{\sqrt{K}}$ without relying on h-hop independence. However, when K is sufficiently large, the approximation ratio presented in the paper outperforms $\frac{1-1/e}{\sqrt{K}}$. Hence, assuming $2h+3\geq\sqrt{K}$, we put forth our proposed approximation algorithm.

3.2. Quota steiner tree (QST) problem

The Quota Steiner Tree problem belongs to the class of Prize Collecting Steiner Tree (PCST) problems [22]. In this problem, given an undirected graph $G=(V,E)$, a profit function $p: V\to\mathbb{Z}^{\geq 0}$ defined on the vertices, a cost function $c: E\to\mathbb{Z}^{\geq 0}$ defined on the edges, and an integer quota q, the goal is to find a subtree T that minimizes the sum of edge costs $\sum_{e\in E(T)} c(e)$, while ensuring that the sum of profits of the vertices in T is at least $\sum_{v\in V(T)} p(v)\geq q$.

Since the Quota Steiner Tree problem is a generalization of the k-MST (Minimum Spanning Tree) problem, Johnson [22] showed that by constructing an example of k-MST, the k-MST problem can be transformed into a Quota Steiner Tree problem while maintaining the same approximation ratio. By utilizing Grag's [23] proof of the 2-approximation algorithm for k-MST, we can establish the 2-approximation result for the Quota Steiner Tree problem.

3.3. Application scenarios of h-Hop

The property of "h-hop independence" is an exact abstraction for many problems. For example, it appears in problems such as the Budgeted Prize Collecting Steiner Tree Problem and the Budgeted Connected Dominating Set Problem. These problems represent specific instances where h is equal to 1 and 3, respectively.

In the Budgeted Connected Dominating Set Problem, the evaluation function is based on the capability of selected points to dominate others. Consequently, if the distance between two points is 3, indicating that the minimum number of hops between them is greater than or equal to 3, the sets of points they can dominate will not overlap. On the other hand, if there is an overlap, it implies that the minimum number of hops between the two points must be less than or equal to 2. Therefore, the Budgeted Connected Dominating Set Problem is an 3-hop independence problem.

An approximation ratio of $\frac{1}{13}\left(1-\frac{1}{e}\right)$ has been proposed for the Budgeted Connected Dominating Set Problem [24]. This work aims to improve upon this existing approximation ratio, leading to enhanced performance in solving the problem.

4. Approximation algorithm

In this section, we prove the important conclusion of the approximation algorithm with an approximation ratio of $\frac{1-e^{-\alpha}}{(2h+3)\cdot\alpha}$.

4.1. Basic idea

In the Quota Steiner Tree (QST) problem, the objective function and constraints are linear, while the actual objective function is submodular. To address this issue, a lower bound function p for f is established using a greedy algorithm. Then, with the calculated p as the reward function, a 2-approximation QST solution is obtained. During the algorithm analysis, the optimal tree solution is decomposed into $2h+3$ smaller subtrees, labeled as $T_1, T_2, \ldots, T_{2h+3}$. This decomposition enables us to find the theoretical lower bound corresponding to the algorithm.

4.2. Approximation algorithm

For undirected graph $G=(V,E)$, given the submodular function f, all points in V need to be sorted as $v_j, v_1, \ldots, v_{n-1}$ and find function $p: V\to\mathbb{Z}^{\geq 0}$. The sorting and assignment methods entail sorting them in descending order according to their maximum marginal gains.

Specifically, the algorithm begins by selecting an initial node v_j. The function value p_j for node v_j is defined as $f(v_j)$. In each iteration, a set of assigned points U is maintained. The algorithm then searches for the point v_i that yields the highest marginal gain when added to U. This is expressed as $v_i = \arg\max f(U\cup\{v\})-f(U)$. Once v_i is identified, it is assigned a value $p_j(v_i)=f(U\cup\{v_i\})-f(U)$, representing its contribution to the function value.

Algorithm 1: Greedy assignment algorithm on node in graph

Require: An undirected graph $G=(V,E)$, h-hop monotone increasing submodular function $f: 2^V\to\mathbb{Z}^{\geq 0}$, and a starting node v_j

Assign $p_j(v_j)=f(v_j)$, $U=v_j$
$U\leftarrow\emptyset$, $i=1$
while $i\leq n$ **do**
  Select the point with the largest marginal gain in $V\setminus U$ as v_i, i.e., $v_i=\arg\max_{v\in V\setminus U}(f(U\cup\{v\})-f(U))$
  Assign $p_j(v_i)=f(U\cup\{v_i\})-f(U)$
  $U\leftarrow U\cup\{v_i\}$
  $i\leftarrow i+1$
end while
return the assigned profit $p_j(v)$ of each node v in V

The problem imposes a constraint on the maximum number of nodes, denoted by K, while the QST problem considers the constraint on the sum of edge weights instead of the number of nodes. To establish a connection between the two, let us assume that in the QST problem, each edge weight is uniformly set to 1, and the total sum of edge weights is limited to $K-1$. Consequently, since a tree in the QST problem can have at most $K-1$ edges, the number of nodes in the resulting tree should be less than or equal to K, satisfying the given constraint. In the QST problem, the assigned profit p given to each node is equivalent to p_j computed by Algorithm 1. This assumption allows us to establish a correspondence between the submodular maximization problem and the QST problem. Specifically, we assume that the total assigned profit of nodes in a tree T, denoted as $p_j(T)$, is equal to the sum of the assigned profits of individual nodes in T, represented by $\sum_{v\in T} p_j(v)$, i.e. $p_j(T)=\sum_{v\in T} p_j(v)$. By transforming the submodular maximization problem into a QST problem with appropriate constraints, we can address the given problem using the tools and techniques of the QST problem.

The algorithm presented in Algorithm 2 aims to solve the h-Hop Independently Submodular Maximization Problem with Curvature.

4.3. Algorithm analysis

To analyze the approximation ratio of Algorithm 1, we begin by observing that the algorithm employs the Greedy Algorithm to assign a profit value, represented as p_j, to each data point. This process of calculating profits is carried out iteratively using

Algorithm 2: Approximation algorithm for H-Hop independently submodular maximization problem with curvature

Require: An undirected graph $G = (V, E)$, h-Hop monotone increasing submodular function$f : 2^V \rightarrow \mathbb{Z}^{\geq 0}$
```
S ← Ø
for each node v_j in V do
    Calculate the profit function p_j for each node with v_j as the
    initial point by invoking Algorithm 1
    left ← max_{v∈V} f(v), right ← f(V)
    while left + 1 < right do
        q = ⌊(left + right)/2⌋
        Use the 2-approximation algorithm for the Quota Steiner
        Tree (QST) problem, starting from v_j, to find a subtree T_j
        that minimizes |E(T_j)| while satisfying p_j(T) ≥ q
        if |E(T)| ≤ K − 1 then
            left ← q
        else
            right ← q
        end if
    end while
    Let q = left. Use the 2-approximation algorithm for the QST
    problem, starting from v_j, to find a subtree T that minimizes
    |E(T)| while satisfying p_j(T) ≥ q. The number of edges in T_j
    does not exceed K − 1, then the number of nodes does not
    exceed K.
    if f(T_j) > f(S) then
        S ← T_j
    end if
end for
return subtree S
```

the Greedy approach. In order to establish a lower bound for the ultimate output of the algorithm, it is essential to appropriately scale each step of the algorithm.

Let $L_0 = \arg\max f(S)$, where $S \subseteq V$, $|S| \leq K$, and $G[S]$ is a connected subgraph. L_0 represents the optimal solution to the problem. Assuming the optimal value is $OPT = f(L_0)$, based on the monotonicity property of f and the cardinality constraint, we have $|L_0| \leq K$ (due to the monotonicity of f, equality holds, but it does not affect the result). L_{h-1} represents any set of nodes in G that are at a distance not exceeding $h-1$ from L_0, i.e., $L_{h-1} = \{v | v \in V \setminus L_0, d(v, L_0) \leq h-1\}$, where $d(v, L_0)$ denotes the hop distance between node v and set L_0.

In the context of assigning profit values p_j to each point in V, we select j to be the index of the point with the maximum individual profit in L_0. Let us denote this point as j_0. The Greedy Algorithm only considers the points in $L_0 \cup L_{h-1}$ for the allocation process. If we define the order of allocating points as $\{v_{j_0}, v_1, v_2, \ldots, v_{n-1}\}$, then assuming that the first K allocated points from this list are in $L_0 \cup L_{h-1}$, we can represent them as $\{v_{j_0}, v_{j_1}, \ldots, v_{j_{K-1}}\}$.

In iteration l, where l is an integer ranging from 1 to $K-1$, we define D_l as the set $\{v_{j_0}, v_1, \ldots, v_{j_{l-1}}\}$. Then, we define D'_l as the intersection of D_l with $L_0 \cup L_{h-1}$, i.e. $D'_l = D_l \cap (L_0 \cup L_{h-1})$, and D''_l as the set difference between D_l and D'_l, i.e. $D''_l = D_l \setminus D'_l$. Specifically, we have $D'_l = \{v_{j_0}, v_{j_1}, \ldots, v_{j_{l-1}}\}$, and $|D'_l| = l$.

Lemma 1. *For any k such that $1 \leq k \leq K-1$, the following inequality holds: $f(L_0) \leq \alpha \sum_{v \in D'_k \setminus L_0} p_{j_0}(v) + \sum_{v \in D'_k \cap L_0} p_{j_0}(v) + |L_0 \setminus D_k| p_{j_0}(v_{j_{k+1}})$. Furthermore, it is also true that: $f(L_0) \leq K \cdot p_{j_0}(v_{j_0})$.*

Proof. For every k in the range $\{1, \ldots, K-1\}$, we have:

$$\begin{aligned} f(D_k) - f(D''_k) &= \sum_{v_{jm} \in D'_k} f_{v_{jm}}(D''_k \cup D'_m) \\ &\leq \sum_{v_{jm} \in D'_k} f_{v_{jm}}(D_m) \\ &= \sum_{v_{jm} \in D'_k} p_{j_0}(v_{jm}) \end{aligned} \tag{1}$$

The inequality is derived from the submodularity of f.

For every k in the range $\{1, \ldots, K-1\}$, we have:

$$\begin{aligned} &f(L_0 \cup D_k) \\ &= f(D''_k) + (f(D_k) - f(D''_k)) + (f(L_0 \cup D_k) - f(D_k)) \\ &\leq f(D''_k) + (f(D_k) - f(D''_k)) + \sum_{v \in L_0 \setminus D_k} p_{j_0}(v_{j_{k+1}}) \\ &\leq f(D''_k) + \sum_{v \in D'_k} p_{j_0}(v) + |L_0 \setminus D_k| p_{j_0}(v_{j_{k+1}}) \end{aligned} \tag{2}$$

In the (2), the first inequality arises from the fact that Algorithm 1 is a greedy algorithm. Specifically, $v_{j_{k+1}}$ represents the point with the maximum marginal gain in D_k. Therefore, $v_{j_{k+1}}$ has a higher marginal gain than any point in L_0 for set D_k. The second inequality arises from the fact that the margin gain of D'_{j_0} is greater than the margin gain between D''_k and D_k.

$$\begin{aligned} f(L_0 \cup D_k) =& f(L_0 \cup D''_k) + f_{D'_k \setminus L_0}(L_0 \cup D''_k) \\ =& f(L_0) + f(D''_k) + f_{D'_k \setminus L_0}(L_0 \cup D''_k) \\ =& f(L_0) + f(D''_k) \\ &+ \sum_{v_{j_i} \in D'_k \setminus L_0} f_{v_{j_i}}(L_0 \cup D''_k \cup D'_{i-1}) \\ \geq& f(L_0) + f(D''_k) \\ &+ (1-\alpha) \sum_{v_{j_i} \in D'_k \setminus L_0} f_{v_{j_i}}(L_0 \cup D''_i \cup D'_{i-1}) \\ =& f(L_0) + f(D''_k) + (1-\alpha) \sum_{v \in D'_k \setminus L_0} p_{j_0}(v) \end{aligned} \tag{3}$$

In the (3), The second equation is derived from the properties of the h-Hop. In this context, D''_k represents the subset of $L_0 \cup L_{h-1}$ where the distance between D''_k and L_0 is greater than or equal to h. As a result, we have $f(L_0 \cup D''_k) = f(L_0) + f(D''_k)$. The inequality is based on the definition of total curvature, which states that the marginal gain of v_{j_i} in a larger set is greater than or equal to the marginal gain of v_{j_i} in a smaller set multiplied by $1 - \alpha$.

Combining the above two inequalities, we observe that $f(D''_k)$ cancels out on both sides of the equation.

$$\begin{aligned} f(L_0) \leq& (\alpha - 1) \sum_{v \in D'_k \setminus L_0} p_{j_0}(v) + \sum_{v \in D'_k} p_{j_0}(v) + |L_0 \setminus D_k| p_{j_0}(v_{j_{k+1}}) \\ =& \alpha \sum_{v \in D'_k \setminus L_0} p_{j_0}(v) + \sum_{v \in L_0 \cap D_k} p_{j_0}(v) + |L_0 \setminus D_k| p_{j_0}(v_{j_{k+1}}) \end{aligned} \tag{4}$$

Let us discuss the second inequality in the lemma. Since $f(L_0)$ is being partitioned into a total of at most K marginal gain, each marginal gain is smaller than or equal to $f(v_{j_0}) = p_{j_0}(v_{j_0})$. This is due to the fact that $p_{j_0}(v_{j_0})$ represents the maximum marginal revenue among all the points in L_0.

$$f(L_0) \leq K \cdot p_{j_0}(v_{j_0}) \tag{5}$$

Lemma 2. *Using the inequality relationships, it can be concluded that $p(D_k) \geq \frac{1-e^{-\alpha}}{\alpha} \cdot f(L_0)$.*

Proof. Let us denote $p_{j_0}(v_{j_i})$ as p_i for simplicity. Since k is an integer between 0 and $K-1$, Lemma 1 consists of K inequalities. To facilitate the representation, we can write all these inequalities in matrix form.

$$
\begin{array}{cccccccc}
1 & & & & i_1 & & i_s & K
\end{array}
$$

$$
\begin{pmatrix}
K & & & & & & & \\
\alpha & & & & & & & \\
 & & K & & & & & \\
 & & \alpha & & K & & & \\
 & & \alpha & & 1 & K-1 & & \\
 & & \alpha & & 1 & \alpha & & \\
 & & \vdots & & \vdots & & K-s+1 & \\
 & & & & & & 1 & \\
\vdots & & \vdots & & & \vdots & \vdots & \\
\alpha & \ldots & \alpha & \ldots & 1 & \alpha & 1 & K-s
\end{pmatrix}
$$

This matrix is multiplied by $\left(p_{j_0} \quad p_{j_1} \quad \ldots \quad p_{j_{K-1}}\right)$.

Each value of the resulting vector after the matrix multiplication is greater than $f(L_0)$.

The inequality established by the matrix multiplication above is proven in Theorem 5.4 of Conforti [17], where the approximate ratio of the problem is demonstrated using linear transformation. They showed that $p(D_k) \geq \frac{1}{\alpha}\left[1-\left(\frac{K-\alpha}{K}\right)^K\right] f(L_0)$, and it is observed that the right-hand side of the inequality is always greater than $\frac{1-e^{-\alpha}}{\alpha} \cdot f(L_0)$.

To ensure the connectivity of the K nodes in the final output tree, a transformation is needed. We find a tree that connects all points in D'_K. Firstly, all points in L_0 are connected. All points in D'_K are the points in $L_0 \cup L_{h-1}$, where the distance between each node in D'_K and any node in L_0 does not exceed $h-1$. Therefore, apart from the nodes outside of j_0 (where j_0 is already in L_0), which are already connected to L_0, we need to connect no more than $h-1$ points from any point in D'_K to L_0. In total, connecting D'_K and connecting L_0 require $(K+(K-1)\cdot(h-1)) = (K-1)\cdot h+1$ number of points. We obtain the tree T, which connects all nodes in D'_K.

Lemma 3 ([15]). *If the tree structure contains a subtree with at most K nodes and no more than $K-1$ points within a distance of $h-1$ from the root node, it is possible to divide this tree into $2h+3$ subtrees, with each subtree having a node tree of no more than $\lfloor K/2 \rfloor$. These subtrees collectively cover the entire tree.*

Lemma 3 primarily leverages the h-hop property of the tree structure in T for edge segmentation. After the tree segmentation, it utilizes the upper bound on edges in T to determine the number of subtrees resulting from the segmentation.

According to this lemma, the tree T obtained earlier can be divided into $2h+3$ distinct subtrees, denoted as $T_1, T_2, \ldots, T_{2h+3}$. We select the subtree T_{max} with the largest p_{j_0} value among all the trees. Since the maximum value is greater than the average of all the trees, we have $p_{j_0}(T_{max}) \geq \frac{p_{j_0}(T)}{2h+3}$.

Theorem 4. *The tree T_A obtained by Algorithm 2 satisfies $f(T_A) \geq \frac{1-e^{-\alpha}}{(2h+3)\cdot\alpha} f(L_0)$, where L_0 represents the optimal solution.*

Proof. The QST problem involves finding a tree Tr with both node weights and edge weights, aiming to minimize $|E(Tr)|$ while ensuring $\sum_{v \in Tr} p(v) \geq q$. The algorithm described here is a 2-approximation algorithm. Since there exists a tree T_{max} with at most $\lfloor \frac{K}{2} \rfloor$ nodes, the number of edges in T_{max} does not exceed $\lfloor \frac{K}{2} \rfloor - 1$. Since QST is an approximation algorithm, we can find a tree with at most $2\left(\lfloor \frac{K}{2} \rfloor - 1\right)$ edges in polynomial time. Consequently, we can find a tree with at most $2\lfloor \frac{K}{2} \rfloor - 1$ nodes in polynomial time. This polynomial-time tree has fewer than K nodes, and its p_{j_0} value is greater than or equal to the p_{j_0} value of $T_{\max}$. Thus, we denote the tree found by the approximation algorithm as T_A.

In the j_0-th iteration of the algorithm, as the value of q increases in the QST problem, the number of nodes in the output tree also increases. Therefore, when $q = p'_{j_0}(T_j)$ is satisfied, the constraint is met, and the value of p'_{j_0} obtained from the binary search and QST output will not be smaller than $p'_{j_0}(T_j)$. It is known that the sum of the p_{j_0} values is less than f.

$$
\begin{aligned}
f(T_A) &\geq p(T_A) \\
&\geq p(T_j) \\
&\geq \frac{1}{2h+3} p(T) \\
&\geq \frac{1}{2h+3} p(D_K) \\
&\geq \frac{1-e^{-\alpha}}{(2h+3)\cdot\alpha} f(L_0)
\end{aligned}
\tag{6}
$$

The fourth inequality holds because the tree T is formed by the nodes of D_K, which can be expressed as $D_K \subseteq T$. Since the function p is a linear function with positive coefficients, we have $p(T) \leq p(D_K)$.

5. Numerical experiments on the application of h-Hop independent submodular maximization

Since the typical real-world application scenario is a CSP problem rather than the h-Hop submodular maximization problem, it is necessary to transform the problem into an h-Hop problem. In the paper cited as [15], it is mentioned that CSP is a subproblem with h equal to 4. Therefore, in this case, the approximation ratio for the CSP problem is $\frac{1-e^{-\alpha}}{11\alpha}$.

The problem we are considering here is the h-Hop independently submodular maximization problem with Curvature, which is well suited for large-scale scenarios. Therefore, we have chosen a large-scale factory for our analysis. Let us assume there is a 4000-meter-wide factory with 400 to 2000 devices inside. The goal is to deploy sensors to cover some of the micro-devices. These devices have a limited signal reception range, and on average, each device can receive signals from sensors within a 150-meter radius. We have placed some candidate sensor points on the top of the factory, and we need to select a subset of these points to place servers. We can have a maximum of K servers, and it is also required that the servers can be connected to each other for communication. The sensors have dedicated communication antennas, allowing them to communicate with the servers within a range of 700 m. In this case, there are 49 candidate points and 20 sensors available for deployment.

In this scenario, we deal with a sensor coverage device problem, which can be classified as a submodular problem. As the number of sensors increases, the coverage of devices also expands, resulting in a monotonic function. All sensors have identical coverage ranges, which geometrically interprets the problem as an h-Hop problem with h being less than or equal to 4. Hence, this scenario can be transformed into the problem described in the paper.

The numerical experiments involve comparing the approximate algorithm with other algorithms. The approximate algorithm used here does not exactly replicate the pseudocode mentioned earlier. Instead, a 5-approximation algorithm is employed

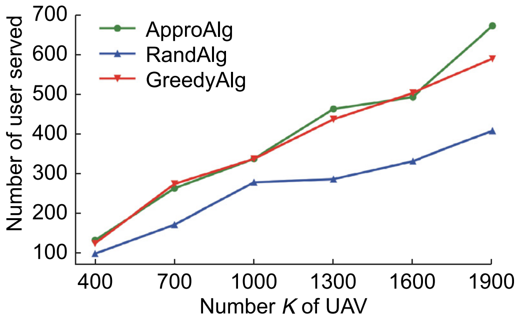

Fig. 1. The impact of different user numbers on the results.

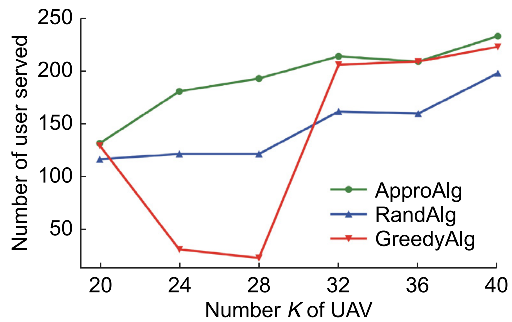

Fig. 2. The impact of different UAV numbers on the results.

to solve the QST problem. Practical applications often rely on the utilization of the 5-approximation algorithm, while the 2-approximation algorithm is a theoretical algorithm and not applicable in this context. The other two algorithms used are as follows:

(1) Random Algorithm: This algorithm randomly selects a point from the graph and iteratively adds points near it to the tree until the desired number of tree nodes, K, is reached.

(2) Greedy Algorithm: This algorithm begins with an initially empty selected set and proceeds iteratively by selecting the point with the highest marginal gain from the selected bottom set among the points close to the chosen points. The selected point is then added to the tree, and this process continues until the size of the tree reaches K.

The experimental environment for these algorithms is an Intel(R) i7-12700H (2.3 GHz) processor with 32 GB of RAM.

5.1. Algorithm performance

In our study, six different numerical experiments are conducted using curvature values of 1.00, 0.33, 0.285, 0.333, 0.2, and 0.5. These experiments demonstrate that, on average, the approximation algorithm outperforms the greedy algorithm by 8.4% and the random algorithm by 80%. As depicted in Fig. 1, the algorithm shows a linear increase in performance as the number of users grows, which is highly desirable in practical applications. Given that the optimal solution is relatively conservative, an improvement of 8.4% represents a significant advancement.

When we calculate the number of UAV from 20 to 40, with a fixed user count of 800, we find that the curvature of the submodular function is 0.33, 0.25, 0.33, 1.00, 0.25, and 0.33, respectively. Under different values of K, our algorithm outperforms or is equal to other algorithms in all scenarios. According to Fig. 2, it can be observed that as the value of K increases, the algorithm converges gradually. This is because the function is submodular, where initially adding sensors yields better results, and the marginal effect of adding more sensors diminishes, resulting in a smaller impact on the overall outcome. It is noteworthy that the greedy algorithm exhibits significant instability in this scenario. This is because the greedy algorithm might initially explore values in the vicinity of a local maximum that are very small, leading to an output that represents a poor local optimal solution. As a result, our approximation algorithm outperforms the greedy algorithm and the random algorithm by 33.2% and 196%, respectively.

6. Conclusion

In this work, we present the innovative concept of the h-hop independently submodular maximization problem with curvature. This advanced problem formulation is applicable to a wide range of practical scenarios, most notably in addressing the complexities of the Connected Sensor Problem. This approach is particularly effective in contexts involving large user scales and a limited number of candidate locations. We demonstrate that the algorithm's approximation ratio is related to α. Numerical experiments is conducted to validate the algorithm's performance on the CSP problem, revealing that our algorithm surpasses the best existing algorithm by 8.4% in terms of approximation ratio.

The CSP is a significant issue in practical applications, presenting many unresolved challenges. Firstly, there are concerns regarding privacy and the robustness of connectivity. In CSP, a connected subgraph may become disconnected if a node is compromised, potentially leading to network failure. Therefore, developing privacy algorithms to ensure network security, or designing subgraphs that remain connected even under partial attacks, is crucial. Secondly, user latency has not been thoroughly considered in UAV placement. High network latency can cause communication interruptions. While UAVs provide connectivity, they may not guarantee minimal relay count and low enough latency for effective user communication. Addressing and minimizing latency is vital for optimizing UAV placement and network connectivity, necessitating constraints on the distance between UAVs. Thirdly, since UAVs are mobile, constructing multiple subgraphs with UAVs positioned at different locations over time is feasible. Investigating these aspects will deepen our understanding of the CSP, leading to more comprehensive conclusions.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The first and third authors are supported by Beijing Natural Science Foundation Project (Z220004). The second author is supported by the Natural Science Foundation of China (12371320). The fourth author is supported by the Natural Science Foundation of China (12101587) and China Postdoctoral Science Foundation (2022M720329).

References

[1] W. Khawaja, I. Guvenc, D.W. Matolak, U.-C. Fiebig, N. Schneckenburger, A survey of air-to-ground propagation channel modeling for unmanned aerial vehicles, IEEE Commun. Surv. Tutor. 21 (3) (2019) 2361–2391.
[2] A.A. Khuwaja, Y. Chen, N. Zhao, M.-S. Alouini, P. Dobbins, A survey of channel modeling for UAV communications, IEEE Commun. Surv. Tutor. 20 (4) (2018) 2804–2821.
[3] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, M. Debbah, A tutorial on UAVs for wireless networks: Applications, challenges, and open problems, IEEE Commun. Surv. Tutor. 21 (3) (2019) 2334–2360.

[4] L.X. Fan Anqi, Chinese drones provide lifeline network services for flood-stranded residents, URL https://www.globaltimes.cn/page/202107/1229363.shtml.
[5] L. Huang, J. Li, Q. Shi, Approximation algorithms for the connected sensor cover problem, in: International Computing and Combinatorics Conference, Springer, 2015, pp. 183–196.
[6] M. Mozaffari, W. Saad, M. Bennis, M. Debbah, Mobile unmanned aerial vehicles (UAVs) for energy-efficient Internet of Things communications, IEEE Trans. Wireless Commun. 16 (11) (2017) 7574–7589.
[7] J. Xu, Y. Zeng, R. Zhang, UAV-enabled wireless power transfer: Trajectory design and energy optimization, IEEE Trans. Wirel. Commun. 17 (8) (2018) 5092–5106.
[8] Y. Zeng, R. Zhang, Energy-efficient UAV communication with trajectory optimization, IEEE Trans. Wirel. Commun. 16 (6) (2017) 3747–3760.
[9] K. Wang, C. Pan, H. Ren, W. Xu, L. Zhang, A. Nallanathan, Packet error probability and effective throughput for ultra-reliable and low-latency UAV communications, IEEE Trans. Commun. 69 (1) (2020) 73–84.
[10] Y. Yu, X. Bu, K. Yang, H. Yang, Z. Han, UAV-aided low latency mobile edge computing with mmwave backhaul, in: ICC 2019-2019 IEEE International Conference on Communications, ICC, IEEE, 2019, pp. 1–7.
[11] M. Li, N. Cheng, J. Gao, Y. Wang, L. Zhao, X. Shen, Energy-efficient UAV-assisted mobile edge computing: Resource allocation and trajectory optimization, IEEE Trans. Veh. Technol. 69 (3) (2020) 3424–3438.
[12] J. Zhou, D. Tian, Y. Yan, X. Duan, X. Shen, Joint optimization of mobility and reliability-guaranteed air-to-ground communication for UAVs, IEEE Trans. Mob. Comput. (2022).
[13] S. Li, C. Xiang, W. Xu, J. Peng, Z. Xu, J. Li, W. Liang, X. Jia, Coverage maximization of heterogeneous UAV networks, in: 2023 IEEE 43rd International Conference on Distributed Computing Systems, ICDCS, IEEE, 2023, pp. 120–130.
[14] W. Xu, Y. Sun, R. Zou, W. Liang, Q. Xia, F. Shan, T. Wang, X. Jia, Z. Li, Throughput maximization of UAV networks, IEEE/ACM Trans. Netw. 30 (2) (2021) 881–895.
[15] W. Xu, H. Xie, C. Wang, W. Liang, X. Jia, Z. Xu, P. Zhou, W. Wu, X. Chen, An approximation algorithm for the *h*-Hop independently submodular maximization problem and its applications, IEEE/ACM Trans. Netw. (2022).
[16] G.L. Nemhauser, L.A. Wolsey, M.L. Fisher, An analysis of approximations for maximizing submodular set functions—I, Math. Program. 14 (1978) 265–294.
[17] M. Conforti, G. Cornuéjols, Submodular set functions, matroids and the greedy algorithm: Tight worst-case bounds and some generalizations of the Rado-Edmonds theorem, Discrete Appl. Math. 7 (3) (1984) 251–274.
[18] J. Yu, S. Ahmed, Maximizing expected utility over a knapsack constraint, Oper. Res. Lett. 44 (2) (2016) 180–185.
[19] T.-W. Kuo, K.C.-J. Lin, M.-J. Tsai, Maximizing submodular set function with connectivity constraint: Theory and application to networks, IEEE/ACM Trans. Netw. 23 (2) (2014) 533–546.
[20] Z. Fang, H. Fu, T. Gu, Z. Qian, T. Jaeger, P. Hu, P. Mohapatra, A model checking-based security analysis framework for IoT systems, High-Confidence Comput. 1 (1) (2021) 100004.
[21] S. Zhu, R. Li, Z. Cai, D. Kim, D. Seo, W. Li, Secure verifiable aggregation for blockchain-based federated averaging, High-Confidence Comput. 2 (1) (2022) 100046.
[22] D.S. Johnson, M. Minkoff, S. Phillips, The prize collecting steiner tree problem: Theory and practice, in: SODA, vol. 1, (no. 0.6) Citeseer, 2000, p. 4.
[23] N. Garg, Saving an epsilon: A 2-approximation for the k-MST problem in graphs, in: Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing, 2005, pp. 396–402.
[24] S. Khuller, M. Purohit, K.K. Sarpatwar, Analyzing the optimal neighborhood: Algorithms for budgeted and partial connected dominating set problems, in: Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, 2014, pp. 1702–1713.