Guarantees for Maximization of k-Submodular Functions with a Knapsack and a Matroid Constraint

Kemin Yu, Min Li, Yang Zhou, and Qian Liu(✉)

School of Mathematics and Statistics, Shandong Normal University,
Jinan 250014, People's Republic of China
lq_qsh@163.com, {liminemily,zhouyang}@sdnu.edu.cn

Abstract. A k-submodular function is a generalization of a submodular function, whose definition domain is the collection of k disjoint subsets. In our paper, we apply a greedy and local search technique to obtain a $\frac{1}{6}(1-e^{-2})$-approximate algorithm for the problem of maximizing a k-submodular function subject to the intersection of a knapsack constraint and a matroid constraint. Furthermore, we use a special analytical method to improve the approximation ratio to $\frac{1}{3}(1-e^{-3})$, when the k-submodular function is monotone.

Keywords: k-submodularity · Knapsack constraint · Matroid constraint · Approximation algorithm

1 Introduction

Consider a ground set G composed of n elements and $k \in N_+$, we define $(k+1)^G$ as the family of k disjoint subset $(X_1, \ldots, X_k)$, where $X_i \subseteq G$, $\forall i \in [k]$ and $X_i \cap X_j = \emptyset$, $\forall i \neq j$. A function $f : (k+1)^G \to R$ is said to be k-submodular [7], if

$$f(\boldsymbol{x}) + f(\boldsymbol{y}) \ \geq \ f(\boldsymbol{x} \sqcup \boldsymbol{y}) + f(\boldsymbol{x} \sqcap \boldsymbol{y}),$$

for any $\boldsymbol{x} = (X_1, \ldots, X_k)$ and $\boldsymbol{y} = (Y_1, \ldots, Y_k)$ in $(k+1)^G$, where

$$\boldsymbol{x} \sqcup \boldsymbol{y} := (X_1 \cup Y_1 \setminus (\bigcup_{i \neq 1} X_i \cup Y_i), \ldots, X_k \cup Y_k \setminus (\bigcup_{i \neq k} X_i \cup Y_i)),$$

$$\boldsymbol{x} \sqcap \boldsymbol{y} := (X_1 \cap Y_1, \ldots, X_k \cap Y_k).$$

Obviously, it is a submodular function for $k = 1$.

As early as 1978, Nemhauser et al. [11] studied the monotone submodular maximization problem subject to cardinality constraints and obtained a greedy $(1 - 1/e)$-approximation algorithm. Many scholars extended submodular maximization to different constraints and design approximate algorithms, see [1–6, 10, 17, 20]. Among them,

Supported by National Science Foundation of China (No. 12001335) and Natural Science Foundation of Shandong Province of China (Nos. ZR2019PA004, ZR2020MA029, ZR2021MA100).

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022
Q. Ni and W. Wu (Eds.): AAIM 2022, LNCS 13513, pp. 156–167, 2022.
https://doi.org/10.1007/978-3-031-16081-3_14

knapsack constraint and matroid constraint are mainly concerned, and most of the algorithms can achieve the tight $1 - 1/e$ approximation ratio. However, under the intersection constraint of a knapsack and a matroid, we have not found that the algorithm can achieve $1-1/e$-approximation, since the loss of rounding is difficult to avoid. Recently, by combining greedy and local search techniques, Sarpatwa et al. [16] contributed an algorithm for reaching $\frac{1-e^{-2}}{2}$-approximation ratio.

In recent years, k-submodular maximization problem has been widely concerned and studied. There have been many research results. For k-submodular maximization without constraint, Ward and Zivny [19] gave a deterministic greedy algorithm, whose approximate ratio reached $1/3$, and a randomized greedy algorithm whose approximate ratio is $\frac{1}{1+a}$, where $a = \max\{1, \sqrt{\frac{k-1}{4}}\}$. Iwata et al. [8] improved the approximation ratio to $1/2$. Later, [14] contributed an algorithm with ratio $\frac{k^2+1}{2k^2+1}$. Under the monotonicity assumption, Ward and Zivny [19] gave a 1/2 approximation algorithm and Iwata et al. [8] improved the approximation ratio to $k/(2k-1)$, which is asymptotically tight. There are also many results for nonnegative monotone k-submodular maximization with constraints. In 2015, Ohsaka and Yoshida [13] designed a $1/2$-approximation algorithm for a total size constraint. Sakaue [15] presented a $1/2$-approximation algorithm with a matroid constraint. And for monotone k-submodular maximization subject to a knapsack constraint, Tang et al. [18] proposed an algorithm of $\frac{1-1/e}{2}$ approximate ratio. Liu et al. [9] design a combinatorial approximation algorithm for monotone k-submodular maximization subject to a knapsack and a matroid constraint and obtained a $\frac{1}{4}(1 - e^{-2})$ approximate ratio.

In this paper, we consider the k-submodular maximization subject to a knapsack and a matroid constraint, and do some work on the basis of the algorithm given by Liu et al. [9]. The main contributions of this paper are as follows:

- We extend the algorithm for k-submodular maximization problem with a knapsack and a matroid constraint to nonmonotone case, and achieve a $\frac{1}{6}(1 - e^{-2})$ approximate ratio, based on the pairwise monotone property.
- We improve the approximate ratio from $\frac{1}{4}(1 - e^{-2})$ in [9] to $\frac{1}{3}(1 - e^{-3})$ under the monotonicity assumption. In the theoretical analysis of the algorithm, we no longer rely on the results of the greedy algorithm for the unconstrained k-submodular maximization problem, and use the properties of k-submodular function to get the new result.

We organize our paper as follows. In Sect. 2, we first introduce the k-submdodular function and some corresponding results, then present the k-submodular maximization problem with a knapsack and a matroid constraint. We present our results for non-monotone case in Sect. 3. In Sect. 4, we show our theoretical analysis for monotone case.

2 Preliminaries

2.1 k-Submodular Function

For any two k disjoint subsets $\boldsymbol{x} = (X_1, \ldots, X_k)$ and $\boldsymbol{y} = (Y_1, \ldots, Y_k)$ in $(k + 1)^G$, we need to introduce a remove operation and a partial order, i.e.

$$\boldsymbol{x} \setminus \boldsymbol{y} := (X_1 \setminus Y_1, \ldots, X_k \setminus Y_k),$$

$$\boldsymbol{x} \preceq \boldsymbol{y}, \text{ if } X_i \subseteq Y_i, \forall i \in [k].$$

Define $\emptyset := (\emptyset, \ldots, \emptyset) \in (k{+}1)^G$ and $(v, i) \in (k{+}1)^G$ such that $X_i = \{v\}$ and $X_j = \emptyset$ for $\forall j \in [k]$ with $j \neq i$. Refer $U(\boldsymbol{x}) = \bigcup_{i=1}^{k} X_i$. For $v \notin U(\boldsymbol{x})$, we use $f_{\boldsymbol{x}}((v, i)) = f(\boldsymbol{x} \sqcup (v, i)) - f(\boldsymbol{x})$ to represent the marginal gain of f. A function f is said to be pairwise monotone if $f_{\boldsymbol{x}}((v, i)) + f_{\boldsymbol{x}}((v, j)) \geq 0$ for any $i \neq j \in [k]$ holds. In addition, we call that the function f is orthant submodular, if $f_{\boldsymbol{x}}((v, i)) \geq f_{\boldsymbol{y}}((v, i))$ holds, for any $\boldsymbol{x} \preceq \boldsymbol{y}$. According to the above definition, we have the equivalent definition and property of the k-submodular function as follows.

Definition 1 [19]**.** *A function $f : (k+1)^G \to R$ is k-submodular iff it is pairwise monotone and orthant submodular.*

Lemma 1 [18]**.** *Given a k-submodular f, we have*

$$f(\boldsymbol{y}) - f(\boldsymbol{x}) \leq \sum_{(v,i) \preceq \boldsymbol{y} \backslash \boldsymbol{x}} f_{\boldsymbol{x}}((v, i)),$$

for any $\boldsymbol{x} \preceq \boldsymbol{y}$.

Check the definition of k-submodular, we have the lemma as follows.

Lemma 2. *Given a k-submodular f, we set $g(\boldsymbol{x}) = f(\boldsymbol{x} \sqcup (v, i))$: $(k+1)^{G \backslash v} \to R$, then $g(\boldsymbol{x})$ is k-submodular.*

2.2 k-Submodular Maximization with a Knapsack and a Matroid Constraint

We define $\mathcal{L} \subseteq 2^G$ as the family of subsets of G. A pair $(G, \mathcal{L})$ is called as an independence system if $(\mathcal{M}1)$ and $(\mathcal{M}2)$ holds. And if $(\mathcal{M}3)$ also holds, the independence system $(G, \mathcal{L})$ is a matroid.

Definition 2. *Given a pair $\mathcal{M} = (G, \mathcal{L})$, where $\mathcal{L} \subseteq 2^G$. We call $\mathcal{M}$ is a matroid if the following holds:*

$(\mathcal{M}1)$*: $\emptyset \in \mathcal{L}$.*
$(\mathcal{M}2)$*: for any subset $A \in \mathcal{L}$, $B \subseteq A$ indicates $B \in \mathcal{L}$.*
$(\mathcal{M}3)$*: for any two subset $A, B \in \mathcal{L}$, $\mid A \mid > \mid B \mid$ indicates that there exists a point $v \in A \backslash B$, such that $B \cup \{v\} \in \mathcal{L}$.*

Given a subset $A \in \mathcal{L}$ and a pair of points (a, b), where $a \in A \cup \{\emptyset\}$ and $b \in G \backslash A$, we refer the pair (a, b) as a swap(a, b) if $A \backslash \{a\} \cup \{b\} \in \mathcal{L}$. It means that only some special points pair called swap can guarantee that $A \backslash \{a\} \cup \{b\} \in \mathcal{L}$ is still an independent set.

We highlight that the next lemma ensures that a swap(a, b) must exist between the optimal solution $\boldsymbol{x}^*$ and the current solution $\boldsymbol{x}^t$ in the later analysis. Consider the support set of the current solution $U(\boldsymbol{x}^t)$ as $A \in \mathcal{L}$ and $U(\boldsymbol{x}^*)$ as $B \in \mathcal{L}$. We will consider finding a special kind of swap$(y(b), b)$ of $U(\boldsymbol{x}^t)$, where $b \in U(\boldsymbol{x}^*)\backslash U(\boldsymbol{x}^t)$ and $y(b) \in U(\boldsymbol{x}^t)\backslash U(\boldsymbol{x}^*) \cup \{\emptyset\}$.

Lemma 3 [16]**.** *Assume two sets $A, B \in \mathcal{L}$, then we can construct a mapping $y : B\backslash A \rightarrow (A\backslash B) \cup \{\emptyset\}$, where every point $b \in B\backslash A$ satisfies $(A\backslash\{y(b)\}) \cup \{b\} \in \mathcal{L}$, and $a \in A\backslash B$ satisfies $|y^{-1}(a)| \leq 1$.*

Consider every point v in G, we give it a weight $w_v \geq 0$ and a total upper bound B. In the following, we assume that w_v is an integer, because we can always change all w_v and B proportionally without losing generality. The two constraints reduce the domain of candidate solutions, so we can only find some solutions $\boldsymbol{x} \in (k+1)^G$ such that the sum of weight w_v of all points v in $U(\boldsymbol{x})$ is less than B and $U(\boldsymbol{x})$ is an independent set. Define $w_{\boldsymbol{x}} = \sum_{v \in U(\boldsymbol{x})} w_v$. The problem can be written as

$$\max_{\boldsymbol{x} \in (k+1)^G} \{f(\boldsymbol{x}) \mid w_{\boldsymbol{x}} \leq B \text{ and } U(\boldsymbol{x}) \in \mathcal{L}\}. \tag{1}$$

In addition, in the later proof, we need to use the following lemma.

Lemma 4 [11]**.** *Given two fixed $P, D \in N_+$ and a sequence of numbers $\gamma_i \in R_+$, where $i \in [P]$, then we have*

$$\begin{aligned} &\frac{\sum_{i=1}^{P} \gamma_i}{\min_{t \in [P]} (\sum_{i=1}^{t-1} \gamma_i + D\gamma_t)} \\ &\geq 1 - (1 - \frac{1}{D})^P \geq 1 - e^{-P/D}. \end{aligned} \tag{2}$$

2.3 Algorithm

Before giving the algorithm to solve problem (1), we firstly introduce a greedy algorithm for unconstrained k-submodular by [19]. We know that a k-submodular function f is pairwise monotone due to Definition 1, that is, $f_{\boldsymbol{x}}((v, i)) + f_{\boldsymbol{x}}((v, j)) \geq 0$ for any $i \neq j \in [k]$. It means that for a fixed $\boldsymbol{x} \in (k+1)^G$ and $v \in G\backslash U(\boldsymbol{x})$, there are no two positions $i \neq j \in [k]$ such that $f_{\boldsymbol{x}}((v, i)) < 0$ and $f_{\boldsymbol{x}}((v, j)) < 0$ both hold. So we can always find a position $i \in [k]$ such that $f_{\boldsymbol{x}}((v, i)) \geq 0$ for any $v \in G\backslash U(\boldsymbol{x})$. Therefore, for every current solution $\boldsymbol{x}^t$ in the Algorithm 1, we add $v \in G\backslash U(\boldsymbol{x}^t)$ with a greedy position i_j until all points $v \in G$ are added to $U(\boldsymbol{x}^t)$.

Then we give an algorithm inspired by [16] and [18] for problems (1) called MK-KM abbreviated as maximizing k-submodular function with a knapsack constraint and a matroid constraint. Let's highlight some important nodes. Firstly, we select three elements with the largest marginal return from the optimal solution $\boldsymbol{x}^*$ by enumerating. Second, for every current solution $\boldsymbol{x}^t \in \mathcal{L}$ and the optimal solution $\boldsymbol{x}^* \in \mathcal{L}$, we can always find a swap$(y(b), b)$ satisfying $y(b) \in \boldsymbol{x}^t\backslash\boldsymbol{x}^*$ and $b \in \boldsymbol{x}^*\backslash\boldsymbol{x}^t$ by Lemma 3. But we always choose a swap(a, b) with the highest marginal profit density $\rho(a, b)$. In the

Algorithm 1. Greedy Algorithm (f, G)

Require: A function $f : (k+1)^G \to R_+$ and a set $G = [n]$
Ensure: A k-disjoint set $\boldsymbol{x} \in (k+1)^G$
$\boldsymbol{x} \leftarrow (\emptyset, \ldots, \emptyset)$
for $j = 1$ to n **do**
  $i_j \leftarrow \arg\max_{i \in [k]} f_{\boldsymbol{x}}((v, i))$
  $\boldsymbol{x} \leftarrow \boldsymbol{x} \sqcup (v, i_j)$
end for
return $\boldsymbol{x}$

line 9 of MK-KM, we reorder the $U(\boldsymbol{x}^t)$ after the operation of swap(a, b) and ensure $\boldsymbol{x}^0 \preceq \boldsymbol{x}^t$. Considering the order of each element in $(U(\boldsymbol{x}^{t-1} \setminus \boldsymbol{x}^0) \setminus \{a\}) \cup \{b\})$ as it is added to current solution in MK-KM, we add them to Greddy Algorithm in the same order. Last but not least, only when $\boldsymbol{x}^t$ is updated, S will be regenerated in line 5. Otherwise, MK-KM will continue to pick and remove the next swap in the loop from 6 to 13. So MK-KM will break the loop when $S = \emptyset$ in line 6.

Algorithm 2. MK-KM (G, B, M)

Require: A function $f : (k+1)^G \to R_+$, a budget $B \in R_+$ and a matroid $(G, \mathcal{L})$
Ensure: A k-disjoint set $\boldsymbol{x} \in (k+1)^G$ satisfying $w_{\boldsymbol{x}} \leq B$ and $U(\boldsymbol{x}) \in \mathcal{L}$
1: Let $\boldsymbol{x}^\alpha \in \arg \max_{|U(\boldsymbol{x})|=1, \boldsymbol{x} \preceq \boldsymbol{x}^*} f(\boldsymbol{x})$, $\boldsymbol{x}^\beta \in \arg \max_{|U(\boldsymbol{x})|=2, \boldsymbol{x}^\alpha \preceq \boldsymbol{x} \preceq \boldsymbol{x}^*} f(\boldsymbol{x})$
   $\boldsymbol{x}^\gamma \in \arg \max_{|U(\boldsymbol{x})|=3, \boldsymbol{x}^\beta \preceq \boldsymbol{x} \preceq \boldsymbol{x}^*} f(\boldsymbol{x})$ and $t = 0$
2: $\boldsymbol{x}^t \leftarrow \boldsymbol{x}^\gamma$ and $switch = false$
3: **while** $switch = false$ **do**
4:   $switch = true$
5:   Generate a collection of all swaps $S = S(U(\boldsymbol{x}^t \setminus \boldsymbol{x}^0))$
6:   **while** $switch = true$ and $S \neq \emptyset$ **do**
7:     Pick a swap (a, b) from S with a maximum value of $\rho(a, b) = \max_{j \in [k]} \frac{f((\boldsymbol{x}^t \setminus (a,i)) \sqcup (b,j)) - f(\boldsymbol{x}^t)}{w_b}$
8:     **if** $\rho(a, b) > 0$ and $w_{\boldsymbol{x}} - w_a + c_b \leq B$ **then**
9:       $\widetilde{\boldsymbol{x}}^t \leftarrow$ **Greedy Algorithm for** $f(\widetilde{\boldsymbol{x}}^t \sqcup \boldsymbol{x}^0)$ **over** $(U(\boldsymbol{x}^t \setminus \boldsymbol{x}^0) \setminus \{a\}) \cup \{b\}$
10:       $\boldsymbol{x}^{t+1} = \widetilde{\boldsymbol{x}}^t \sqcup \boldsymbol{x}^0$
11:       $w_{\boldsymbol{x}^{t+1}} = w_{\boldsymbol{x}^t} - w_a + w_b$
12:       $switch = false$
13:     **end if**
14:     $S = S \setminus \{(a, b)\}$
15:   **end while**
16: **end while**
17: **return** $\boldsymbol{x}$

We modify MK-KM and give MK-KM' algorithm for problem (1) with monotonicity. MK-KM' selects two elements with the largest marginal return from the optimal solution $\boldsymbol{x}^*$ by enumerating. This modification reduces the running time.

Algorithm 3. MK-KM' (G, B, M)

Require: A function $f:(k+1)^G \to R_+$, a budget $B \in R_+$ and a matroid $(G, \mathcal{L})$
Ensure: A k-disjoint set $\boldsymbol{x} \in (k+1)^G$ satisfying $w_{\boldsymbol{x}} \leq B$ and $U(\boldsymbol{x}) \in \mathcal{L}$
1: Let $\boldsymbol{x}^\alpha \in \arg\max_{|U(\boldsymbol{x})|=1, \boldsymbol{x} \preceq \boldsymbol{x}^*} f(\boldsymbol{x})$, $\boldsymbol{x}^\beta \in \arg\max_{|U(\boldsymbol{x})|=2, \boldsymbol{x}^\alpha \preceq \boldsymbol{x} \preceq \boldsymbol{x}^*} f(\boldsymbol{x})$, and $t = 0$
2: $\boldsymbol{x}^t \leftarrow \boldsymbol{x}^\beta$ and $switch = false$
3: **while** $switch = false$ **do**
4: $\quad switch = true$
5: $\quad$ Generate a collection of all swaps $S = S(U(\boldsymbol{x}^t \backslash \boldsymbol{x}^0))$
6: $\quad$ **while** $switch = true$ and $S \neq \emptyset$ **do**
7: $\quad\quad$ Pick a swap (a, b) from S with a maximum value of $\rho(a,b) = \max_{j \in [k]} \frac{f((\boldsymbol{x}^t \backslash (a,i)) \sqcup (b,j)) - f(\boldsymbol{x}^t)}{w_b}$
8: $\quad\quad$ **if** $\rho(a,b) > 0$ and $w_{\boldsymbol{x}} - w_a + c_b \leq B$ **then**
9: $\quad\quad\quad \widetilde{\boldsymbol{x}}^t \leftarrow$ **Greedy Algorithm for** $f(\widetilde{\boldsymbol{x}}^t \sqcup \boldsymbol{x}^0)$ **over** $(U(\boldsymbol{x}^t \setminus \boldsymbol{x}^0) \setminus \{a\}) \cup \{b\}$
10: $\quad\quad\quad \boldsymbol{x}^{t+1} = \widetilde{\boldsymbol{x}}^t \sqcup \boldsymbol{x}^0$
11: $\quad\quad\quad w_{\boldsymbol{x}^{t+1}} = w_{\boldsymbol{x}^t} - w_a + w_b$
12: $\quad\quad\quad switch = false$
13: $\quad\quad$ **end if**
14: $\quad\quad S = S \setminus \{(a,b)\}$
15: $\quad$ **end while**
16: **end while**
17: **return** $\boldsymbol{x}$

In order to pave the way for analysis of Sect. 4, we consider the process of the current solution $\boldsymbol{x}^t$ generated by $\boldsymbol{x}^0 \sqcup \tilde{\boldsymbol{x}}^t$. We carefully define $\tilde{\boldsymbol{x}}_j^t$ as the current solution of each iteration of the greedy algorithm of the 9th line, where $j \in \{1, \ldots, |U(\boldsymbol{x}^t) - 2|\}$ for every fixed t. Define $(v_j, i_j) = \tilde{\boldsymbol{x}}_j^t \backslash \tilde{\boldsymbol{x}}_{j-1}^t$ in Greedy Algorithm.

For the convenience of writing, we define $\boldsymbol{x}_j^t = \tilde{\boldsymbol{x}}_j^t \sqcup \boldsymbol{x}^0$. Then immediately $(v_j, i_j) = (\boldsymbol{x}_j^t \backslash \boldsymbol{x}_{j-1}^t) = ((\tilde{\boldsymbol{x}}_j^t \sqcup \boldsymbol{x}^0) \backslash (\tilde{\boldsymbol{x}}_{j-1}^t \sqcup \boldsymbol{x}^0))$ holds. For each fixed iteration step t, there are a string of iteration steps $j \in \{1, \ldots, |U(\boldsymbol{x}^t) - 2|\}$ for the nested greedy algorithm.

3 Analysis for Non-monotone k-submodular Maximization with a Knapsack Constraint and a Matroid Constraint

In this section, we will draw support from the nested greedy algorithm to solve problem (1). For nonnegative, non-monotone and unconstrained k-submodular, we need the following conclusions. Lemma 5 comes from Proposition 2.1 in [8]. If there exists a solution achieving the optimal value, we can construct an optimal solution containing all points of ground set. Therefore, for unconstrained k-submodular maximization, we only analyze the optimal solution which is the partition of ground set of Algorithm 1. And Lemma 6 ensures that we can obtain a 1/3-approximate greedy solution in the nested greedy Algorithm 1 by using $(U(\boldsymbol{x}^t \setminus \boldsymbol{x}^0) \setminus \{a\}) \cup \{b\}$ as ground set G, where $OPT_f(G)$ is the optimal value of unconstrained k-submodular f maximization over G.

Lemma 5 [8]. *For maximizing a non-monotone k-submodular f over a set G, there exists a partition of G achieving the optimal value.*

Lemma 6 [19]. *For maximizing a non-monotone k-submodular f over a set G, by greedy algorithm, we can get a solution $\boldsymbol{x}$ such that $U(\boldsymbol{x}) = G$ and $3f(\boldsymbol{x}) \geq OPT_f(G)$.*

Drawing support from the nested greedy algorithm, we reorder each iterative solution of MK-KM and analyze the approximate ratio in two cases.

Theorem 1. *Applying MK-KM algorithm to problem (1), we can obtain a $\frac{1}{6}(1-e^{-2})$-approximate ratio.*

Proof. Using Lemma 3 between the iterative solution $\boldsymbol{x}^t$ of MK-KM and the optimal solution $\boldsymbol{x}^*$, there exists swap $(y(b), b)$ satisfying $y(b) \in (U(\boldsymbol{x}^t)\backslash U(\boldsymbol{x}^*)) \cup \{\emptyset\}$ and $b \in U(\boldsymbol{x}^*)\backslash U(\boldsymbol{x}^t)$.

For any iteration step t, we construct a solution $\hat{\boldsymbol{x}}^t$. Considering all $(b, i) \preceq \boldsymbol{x}^*\backslash \boldsymbol{x}^t$, we add them to $\boldsymbol{x}^t$ and get $\hat{\boldsymbol{x}^t}$. Note that $\boldsymbol{x}^0 \preceq \boldsymbol{x}^t \preceq \hat{\boldsymbol{x}}^t$ and $U(\hat{\boldsymbol{x}}^t) = U(\boldsymbol{x}^*) \cup U(\boldsymbol{x}^t)$.

Due to Lemma 5, there exists an optimal solution containing all points in ground set G. And by Lemma 2, we know that $f(\boldsymbol{x} \sqcup \boldsymbol{x}^0)$ is a k-submodular over $U(\hat{\boldsymbol{x}}^t)\backslash U(\boldsymbol{x}^0)$. So we define that $OPT_{f(\boldsymbol{x}\sqcup \boldsymbol{x}^0)}(U(\hat{\boldsymbol{x}}^t)\backslash U(\boldsymbol{x}^0))$ is the optimal value of $f(\boldsymbol{x} \sqcup \boldsymbol{x}^0)$ over $U(\hat{\boldsymbol{x}}^t\backslash \boldsymbol{x}^0)$. Using Lemma 6 for each $\boldsymbol{x}^t$ in MK-KM, we always have

$$\begin{aligned}
&OPT_{f(\boldsymbol{x}\sqcup \boldsymbol{x}^0)}(U(\hat{\boldsymbol{x}}^t)\backslash U(\boldsymbol{x}^0)) \\
&\leq 3f(\hat{\boldsymbol{x}}^t) \\
&\leq 3f(\boldsymbol{x}^t) + 3\sum_{(b,i)\preceq \hat{\boldsymbol{x}}^t\backslash \boldsymbol{x}^t} [f(\boldsymbol{x}^t \sqcup (b,i)) - f(\boldsymbol{x}^t)] \\
&\leq 3f(\boldsymbol{x}^t) + 3\sum_{(b,i)\preceq \hat{\boldsymbol{x}}^t\backslash \boldsymbol{x}^t} [f((\boldsymbol{x}^t\backslash (y(b),j)) \sqcup (b,i)) - f((\boldsymbol{x}^t\backslash (y(b),j))].
\end{aligned} \tag{3}$$

The first inequality is due to Lemma 6. And the second is due to Lemma 1. By orthant submodularity, we get the third inequality. Recall that MK-KM breaks all loops when $S = \emptyset$ in line 6. It implies that we cannot find a qualified swap(a, b) to update the output solution $\boldsymbol{x}$. We only consider swaps$(y(b), b)$ in $S(U(\boldsymbol{x}\backslash \boldsymbol{x}^0))$ related to $b \in U(\boldsymbol{x}^*)\backslash U(\boldsymbol{x})$ instead of all candidate swaps(a, b). Now we use this construction method to analyze the algorithm in two cases.

Case 1: Consider a very special case that every swap$(y(b), b)$ was rejected just due to $\rho(y(b), b) \leq 0$ instead of knapsack constraint.

Applying formula (3) for the output solution $\boldsymbol{x}$ and constructed solution $\hat{\boldsymbol{x}}$, we get

$$\begin{aligned} & f(\boldsymbol{x}^*) \\ & \leq OPT_{f(\boldsymbol{x} \sqcup \boldsymbol{x}^0)}(U(\hat{\boldsymbol{x}}) \backslash U(\boldsymbol{x}^0)) \\ & \leq 3f(\boldsymbol{x}) + 3 \sum_{(b,i) \preceq \hat{\boldsymbol{x}} \backslash \boldsymbol{x}} [f((\boldsymbol{x} \backslash (y(b),j)) \sqcup (b,i)) - f((\boldsymbol{x} \backslash (y(b),j))]. \end{aligned} \tag{4}$$

Since $\rho(y(b), b) \leq 0$, we have

$$f((\boldsymbol{x} \backslash (y(b),j)) \sqcup (b,i)) \leq f(\boldsymbol{x}) \tag{5}$$

for all $(b, i) \preceq \hat{\boldsymbol{x}} \backslash \boldsymbol{x}$. We define $\{(y(b), j)\}_{b \in U(\hat{\boldsymbol{x}}^t \backslash \boldsymbol{x}^t)} \backslash \{\emptyset\} = \{(y_1, j_1), \ldots, (y_K, j_K)\}$, then we get

$$\begin{aligned} & \sum_{(b,i) \preceq \hat{\boldsymbol{x}} \backslash \boldsymbol{x}} [f(\boldsymbol{x}) - f((\boldsymbol{x} \backslash (y(b),j))] \\ & \leq \sum_{l=1}^{K} [f((\boldsymbol{x} \backslash ((y_1,j_1) \sqcup \cdots \sqcup (y_K,j_K))) \sqcup ((y_1,j_1) \sqcup \cdots \sqcup (y_l,j_l))) \\ & - f((\boldsymbol{x} \backslash ((y_1,j_1) \sqcup \cdots \sqcup (y_K,j_K))) \sqcup ((y_1,j_1) \sqcup \cdots \sqcup (y_{l-1},j_{l-1}))] \\ & = f(\boldsymbol{x}) - f(\boldsymbol{x} \backslash ((y_1,j_1) \sqcup \cdots \sqcup (y_K,j_K))) \\ & \leq f(\boldsymbol{x}). \end{aligned} \tag{6}$$

The first inequality is due to orthant submodularity. Because f is nonnegative, the second inequality holds. So we can get

$$f(\boldsymbol{x}^*) \leq 6f(\boldsymbol{x}). \tag{7}$$

Therefore, we find a $1/6$-approximate solution in Case 1.

Case 2: Consider the opposite of Case 1 that there exists at least one swap$(y(b), b)$ satisfying $w_{\boldsymbol{x}} - w_{y(b)} + w_b > B$.

Assume a special iteration step t^*. For the first time, there appears a swap $(y(b_*), b_*)$ in $S(U(\boldsymbol{x}^{t^*} \backslash \boldsymbol{x}^0))$ such that $w_{\boldsymbol{x}^{t^*}} - w_{y(b_*)} + w_{b_*} > B$, where $b_* \in U(\boldsymbol{x}^*) \backslash U(\boldsymbol{x}^{t^*})$ and $y(b_*) \in (U(\boldsymbol{x}^{t^*}) \backslash U(\boldsymbol{x}^*)) \cup \{\emptyset\}$.

Although this swap$(y(b_*), b_*)$ violates the knapsack constraint, we use it to construct a solution $(\boldsymbol{x}^{t^*} \setminus (y(b_*), j_{y(b_*)})) \sqcup (b_*, i_{b_*})$. By orthant submodularity, pairwise monotonicity and the greedy choice of $\boldsymbol{x}^\alpha$, $\boldsymbol{x}^\beta$ and $\boldsymbol{x}^\gamma$, we have

$$f((\boldsymbol{x}^{t^*} \setminus (y(b_*), j_{y(b_*)})) \sqcup (b_*, i_{b_*})) - f(\boldsymbol{x}^{t^*}) \leq \frac{2}{3} f(\boldsymbol{x}^0). \tag{8}$$

The detailed process of proof is shown in the Appendix. By Lemma 2, we know that $g(\boldsymbol{x}) = f(\boldsymbol{x}) - f(\boldsymbol{x}^0)$ is a k-submodular function. Then applying formula (3) for the current solution $\boldsymbol{x}^t$ and constructed solution $\hat{\boldsymbol{x}}^t$, we can get

$$g(\boldsymbol{x}^*) \leq 6[g(\boldsymbol{x}^t) + \frac{(B - w_{\boldsymbol{x}^0})}{2} \rho_{t+1}]. \tag{9}$$

for all $t \in \{1, \ldots, t^*\}$. The detailed process of proof is shown in the Appendix. We introduce a construction method inspired by K. K. Sarpatwar [16]. Its details are still in the Appendix. Due to the construction method, we can get

$$\frac{g((\boldsymbol{x}^{t^*} \setminus (y(b_*), j_{y(b_*)})) \sqcup (b_*, i_{b_*}))}{g(\boldsymbol{x}^*)} \geq \frac{1}{6}(1 - e^{-2}). \tag{10}$$

Then, combing (8) and (10), we have

$$\begin{aligned}
&f(\boldsymbol{x}^{t^*}) \\
&= f(\boldsymbol{x}^0) + g(\boldsymbol{x}^{t^*}) \\
&= f(\boldsymbol{x}^0) + g((\boldsymbol{x}^{t^*} \setminus (y(b_*), j_{y(b_*)})) \sqcup (b_*, i_{b_*})) \\
&- [g((\boldsymbol{x}^{t^*} \setminus (y(b_*), j_{y(b_*)})) \sqcup (b_*, i_{b_*}))) - g(\boldsymbol{x}^{t^*})] \\
&= f(\boldsymbol{x}^0) + g((\boldsymbol{x}^{t^*} \setminus (y(b_*), j_{y(b_*)})) \sqcup (b_*, i_{b_*})) \\
&- [f((\boldsymbol{x}^{t^*} \setminus (y(b_*), j_{y(b_*)})) \sqcup (b_*, i_{b_*}))) - f(\boldsymbol{x}^{t^*})] \\
&\geq f(\boldsymbol{x}^0) + \frac{1}{6}(1 - e^{-2})g(\boldsymbol{x}^*) - \frac{2}{3}f(\boldsymbol{x}^0) \\
&\geq \frac{1}{6}(1 - e^{-2})f(\boldsymbol{x}^*).
\end{aligned} \tag{11}$$

Therefore, we have a $\frac{1}{6}(1 - e^{-2})$-approximate solution $\boldsymbol{x}^{t^*}$ for MK-KM.

4 Analysis for Monotone k-Submodular Maximization with a Knapsack and a Matroid Constraint

A function f is said to be monotone, if $f(\boldsymbol{x}) \leq f(\boldsymbol{y})$ for any $\boldsymbol{x} \preceq \boldsymbol{y}$. It is easy to see that f must be pairwise monotone if f is monotone. Therefore, a monotone function $f : (k+1)^G \to R$ is k-submodular if and only it is orthant submodular. In this section, we introduce a special construction method inspired by Lan N. Nguyen [12], and obtain a better approximate ratio by MK-KM' algorithm.

For a fixed iteration t, recall that $(v_j, i_j) = \boldsymbol{x}_j^t \backslash \boldsymbol{x}_{j-1}^t$. Define $(v_j, i_*) \preceq \boldsymbol{x}^*$. We construct two sequences $\{\boldsymbol{o}_{j-1/2}\}$ and $\{\boldsymbol{o}_j\}$ such that $\boldsymbol{o}_{j-1/2} = (\boldsymbol{x}^* \sqcup \boldsymbol{x}_j^t) \sqcup \boldsymbol{x}_{j-1}^t$ and $\boldsymbol{o}_j = (\boldsymbol{x}^* \sqcup \boldsymbol{x}_j^t) \sqcup \boldsymbol{x}_j^t$, where $j \in \{1, \ldots, |U(\boldsymbol{x}^t)| - 2\}$ and $\boldsymbol{o}_{j=0} = \boldsymbol{x}^*$.

Note that $\boldsymbol{x}_{j-1}^t \preceq \boldsymbol{x}_j^t \preceq \boldsymbol{o}_j$ and $\boldsymbol{o}_{j-1/2} \preceq \boldsymbol{o}_j$. By Lemma 2, we know that $g(\boldsymbol{x}) = f(\boldsymbol{x}) - f(\boldsymbol{x}^0)$ is a monotone k-submodular function. Then for any $j \in \{1, \ldots, |U(\boldsymbol{x}^t)| - 2\}$, we have

$$g(\boldsymbol{o}_{j-1}) - g(\boldsymbol{o}_j) \leq g(\boldsymbol{o}_{j-1}) - g(\boldsymbol{o}_{j-1/2}) \leq g(\boldsymbol{x}_j^t) - g(\boldsymbol{x}_{j-1}^t). \tag{12}$$

The first inequality is due to monotonicity and $\boldsymbol{o}_{j-1/2} \preceq \boldsymbol{o}_j$. When $v_j \notin U(\boldsymbol{x}^*)$ or $v_j \in U(\boldsymbol{x}^*)$ with $i_j = i_*$, we have $g(\boldsymbol{o}_{j-1}) - g(\boldsymbol{o}_{j-1/2}) \leq 0$ by monotonicity. When $v_j \in U(\boldsymbol{x}^*)$ and $i_j \neq i_*$, we have $g(\boldsymbol{o}_{j-1}) - g(\boldsymbol{o}_{j-1/2}) \geq 0$. Using orthant submodularity, we get the following inequality.

$$g(\boldsymbol{o}_{j-1}) - g(\boldsymbol{o}_{j-1/2}) \leq g(\boldsymbol{x}_{j-1}^t \sqcup (v_j, i_*)) - g(\boldsymbol{x}_{j-1}^t) \tag{13}$$

Then by greedy choice, the inequality (12) holds.

Theorem 2. *According to MK-KM' algorithm, a* $\frac{1}{3}(1-e^{-3})$*-approximate solution of problem (1) can be obtained, if* f *is monotone.*

Proof. Similarly to Theorem 1, we analyze the algorithm in two cases. When we get the output solution $\boldsymbol{x}$, there is not any qualified swap (a, b) to update $\boldsymbol{x}$. We only consider swaps$(y(b), b)$ in $S(U(\boldsymbol{x}\backslash\boldsymbol{x}^0))$ related to $b \in U(\boldsymbol{x}^*)\backslash U(\boldsymbol{x})$ instead of all candidate swaps(a, b).

Case 1: Consider a very special case that every swap$(y(b), b)$ was rejected just due to $\rho(y(b), b) \le 0$ instead of knapsack constraint.

For the optimal solution $\boldsymbol{x}^*$ and the output solution $\boldsymbol{x}$, we construct two sequences $\{\boldsymbol{o}_{j-1/2}\}$ and $\{\boldsymbol{o}_j\}$, where $j \in \{1, \dots, |U(\boldsymbol{x})|-2\}$. Sum (12) for j from 1 to $(|U(\boldsymbol{x})|-2)$, we have

$$\begin{aligned} g(\boldsymbol{x}^*) - g(\boldsymbol{o}_{|U(\boldsymbol{x})|-2}) &= \sum_{j=1}^{|U(\boldsymbol{x})|-2} [g(\boldsymbol{o}_{j-1}) - g(\boldsymbol{o}_j)] \\ &\le \sum_{j=1}^{|U(\boldsymbol{x})|-2} [g(\boldsymbol{x}_j) - g(\boldsymbol{x}_{j-1})] \\ &= g(\boldsymbol{x}). \end{aligned} \tag{14}$$

Using Lemma 1, orthant submodularity and $\rho(y(b), b) \le 0$, we get

$$\begin{aligned} g(\boldsymbol{x}^*) &\le g(\boldsymbol{o}_{|U(\boldsymbol{x})|-2}) + g(\boldsymbol{x}) \\ &\le g(\boldsymbol{x}) + \sum_{(b,i)\preceq(\boldsymbol{o}_{|U(\boldsymbol{x})|-2}\backslash\boldsymbol{x})} [g(\boldsymbol{x} \sqcup (b,i)) - g(\boldsymbol{x})] + g(\boldsymbol{x}) \\ &\le 2g(\boldsymbol{x}) + \sum_{(b,i)\preceq(\boldsymbol{o}_{|U(\boldsymbol{x})|-2}\backslash\boldsymbol{x})} [g((\boldsymbol{x}\backslash(y(b),j)) \sqcup (b,i)) - g(\boldsymbol{x}\backslash(y(b),j))] \\ &\le 2g(\boldsymbol{x}) + \sum_{(b,i)\preceq(\boldsymbol{o}_{|U(\boldsymbol{x})|-2}\backslash\boldsymbol{x})} [g(\boldsymbol{x}) - g(\boldsymbol{x}\backslash(y(b),j))]. \end{aligned} \tag{15}$$

Let $\{(y(b), j)\}_{b\in U(\boldsymbol{o}_{|U(\boldsymbol{x})|}\backslash\boldsymbol{x})}\backslash\{\emptyset\} = \{(y_1, j_1), \dots, (y_K, j_K)\}$, then we have

$$\begin{aligned} g(\boldsymbol{x}^*) &\le 2g(\boldsymbol{x}) + \sum_{l=1}^{K}[g((y_1,j_1)\sqcup\cdots\sqcup(y_l,j_l)) - g((y_1,j_1)\sqcup\cdots\sqcup(y_{l-1},j_{l-1}))] \\ &\le 2g(\boldsymbol{x}) + \sum_{l=1}^{K} g((y_1,j_1)\sqcup\cdots\sqcup(y_K,j_K)) \\ &\le 3g(\boldsymbol{x}). \end{aligned} \tag{16}$$

Therefore,

$$f(\boldsymbol{x}^*) \le 3f(\boldsymbol{x}) - 2f(\boldsymbol{x}^0) \le 3f(\boldsymbol{x}). \tag{17}$$

We obtain $1/3$-approximate ratio in case 1.

Case 2: Consider the opposite of case 1 that there exists at least one swap$(y(b), b)$ satisfying $w_{\boldsymbol{x}} - w_{y(b)} + w_b > B$.

For the first time, there appears a swap $(y(b_*), b_*)$ in $S(U(\boldsymbol{x}^{t^*} \backslash \boldsymbol{x}^0))$ such that $w_{\boldsymbol{x}^{t^*}} - w_{y(b_*)} + w_{b_*} > B$, where $b_* \in U(\boldsymbol{x}^*) \backslash U(\boldsymbol{x}^{t^*})$ and $y(b_*) \in (U(\boldsymbol{x}^{t^*}) \backslash U(\boldsymbol{x}^*)) \cup \{\emptyset\}$. For each $t \in \{1, \ldots, t^*\}$, we construct two sequences $\{\boldsymbol{o}_{j-1/2}\}$ and $\{\boldsymbol{o}_j\}$ between $\boldsymbol{x}^t$ and $\boldsymbol{x}^*$, where $j \in \{1, \ldots, |U(\boldsymbol{x}^t)| - 2\}$. Summing (13) for j from 1 to $|U(\boldsymbol{x}^t)| - 2$ and using Lemma 1, we have

$$\begin{aligned} g(\boldsymbol{x}^*) &\leq g(\boldsymbol{o}_{|U(\boldsymbol{x}^t)|-2}) + g(\boldsymbol{x}^t) \\ &\leq g(\boldsymbol{x}^t) + \sum_{(b,i) \preceq (\boldsymbol{o}_{|U(\boldsymbol{x}^t)|-2} \backslash \boldsymbol{x}^t)} [g(\boldsymbol{x}^t \sqcup (b,i)) - g(\boldsymbol{x}^t)] + g(\boldsymbol{x}^t). \end{aligned} \tag{18}$$

Then applying (18) and the similar technique of (3) and (6), we can get

$$g(\boldsymbol{x}^*) \leq 3g(\boldsymbol{x}^t) + (B - w_{\boldsymbol{x}^0})\rho_{t+1}, \tag{19}$$

for all $t \in \{1, \ldots, t^*\}$. The detailed process of proof is shown in the Appendix. Similar to the proof of (10), using (19), we can get

$$\frac{g((\boldsymbol{x}^{t^*} \setminus (y(b_*), j_{y(b_*)})) \sqcup (b_*, i_{b_*}))}{g(\boldsymbol{x}^*)} \geq \frac{1}{3}(1 - e^{-3}). \tag{20}$$

We modify inequality (8) as follows. By orthant submodularity, monotonicity and the greedy choice of $\boldsymbol{x}^\alpha$, $\boldsymbol{x}^\beta$, we have

$$f((\boldsymbol{x}^{t^*} \setminus (y(b_*), j_{y(b_*)})) \sqcup (b_*, i_{b_*})) - f(\boldsymbol{x}^{t^*}) \leq \frac{f(\boldsymbol{x}^0)}{2}. \tag{21}$$

The detailed process of proof is shown in the Appendix. Combing (20) and (21), we have

$$\begin{aligned} &f(\boldsymbol{x}^{t^*}) \\ &= f(\boldsymbol{x}^0) + g((\boldsymbol{x}^{t^*} \setminus (y(b_*), j_{y(b_*)})) \sqcup (b_*, i_{b_*})) \\ &- [f((\boldsymbol{x}^{t^*} \setminus (y(b_*), j_{y(b_*)})) \sqcup (b_*, i_{b_*})) - f(\boldsymbol{x}^{t^*})] \\ &\geq f(\boldsymbol{x}^0) + \frac{1}{3}(1 - e^{-3})g(\boldsymbol{x}^*) - \frac{f(\boldsymbol{x}^0)}{2} \\ &\geq \frac{1}{3}(1 - e^{-3})f(\boldsymbol{x}^*). \end{aligned} \tag{22}$$

Hence, MK-KM' has an approximation ratio of at least $\frac{1}{3}(1 - e^{-3})$.

5 Discussion

To summarize this paper, inspired by [16] and [18], we propose a nested algorithm applicable to monotone and non-monotone k-submodular maximization with the intersection of a knapsack and a matroid constraint. For problem (1), we have a $\frac{1}{6}(1 - e^{-2})$-approximate ratio. Inspired by [12], we use a new construction method between optimal solution and current solution. For monotone k-submodular maximization with a knapsack and a matroid constraint, we achieve at least $\frac{1}{3}(1 - e^{-3})$ approximation ratio.

References

1. Bian, A.A., Buhmann, J.M., Krause, A., Tschiatschek, S.: Guarantees for greedy maximization of non-submodular functions with applications. In: Proceedings of the 34th International Conference on Machine Learning (ICML), Sydney, NSW, Australia, 2017, pp. 498–507 (2017)
2. Calinescu, G., Chekuri, C., P$\acute{a}$l, M., Vondr$\acute{a}$k, J.: Maximizing a monotone submodular function subject to a matroid constraint. SIAM J. Comput. **40**(6), 1740–1766 (2011)
3. Ene, A., Nguy$\tilde{\hat{e}}$n, H.L.: A nearly-linear time algorithm for submodular maximization with a knapsack constraint. In: Proceedings of the 46th International Colloquium on Automata, Languages and Programming (ICALP), Patras, Greece, 2019, pp. 53:1–53:12 (2019)
4. Feldman, M.: Maximization problems with submodular objective functions, Ph.D. dissertation, Computer Science Department, Technion, Haifa, Israel (2013)
5. Filmus, Y., Ward, J.: Monotone submodular maximization over a matroid via non-oblivious local search. SIAM J. Comput. **43**(2), 514–542 (2014)
6. Huang, C., Kakimura, N., Mauras, S., Yoshida, Y.: Approximability of monotone submodular function maximization under cardinality and matroid constraints in the streaming. SIAM J. Discrete Math. **36**, 355–382 (2022)
7. Huber, A., Kolmogorov, V.: Towards mininizing k-submodular functions. In: Proceedings of 2nd International Symposium on Combinatorial Optimization, pp. 451–462 (2012)
8. Iwata, S., Tanigawa, S.-I., Yoshida, Y.: Improved approximation algorithms for k-submodular function maximization. In: Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), Arlington, VA, USA, 2016, pp. 404–413 (2016)
9. Liu, Q., Yu, K., Li, M., Zhou, Y.: k-Submodular Maximization with a Knapsack Constraint and p Matroid Constraints (submitted)
10. Liu, Z., Guo, L., Du, D., Xu, D., Zhang, X.: Maximization problems of balancing submodular relevance and supermodular diversity. J. Global Optim. **82**(1), 179–194 (2021). https://doi.org/10.1007/s10898-021-01063-6
11. Nemhauser, G.L., Wolsey, L.A., Fisher, M.L.: An analysis of approximations for maximizing submodular set functions-I. Math. Program. **14**(1), 265–294 (1978)
12. Nguyen, L.N., Thai, M.T.: Streaming k-submodular maximization under noise subject to size constraint. In: Proceedings of the 37th International Conference on Machine Learning (ICML), 2020, pp. 7338–7347 (2020)
13. Ohsaka, N., Yoshida, Y.: Monotone k-submodular function maximization with size constraints. Adv. Neural. Inf. Process. Syst. **28**, 694–702 (2015)
14. Oshima, H.: Improved randomized algorithm for k-submodular function maximization. SIAM J. Discret. Math. **35**(1), 1–22 (2021)
15. Sakaue, S.: On maximizing a monotone k-submodular function subject to a matroid constraint. Discret. Optim. **23**, 105–113 (2017)
16. Sarpatwar, K.K., Schieber, B., Shachnai, H.: Constrained submodular maximization via greedy local search. Oper. Res. Lett. **47**(1), 1–6 (2019)
17. Sviridenko, M.: A note on maximizing a submodular set function subject to a knapsack constraint. Oper. Res. Lett. **32**(1), 41–43 (2004)
18. Tang, Z., Wang, C., Chan, H.: On maximizing a monotone k-submodular function under a knapsack constraint. Oper. Res. Lett. **50**(1), 28–31 (2022)
19. Ward, J., Živný, S.: Maximizing k-submodular functions and beyond. ACM Trans. Algorithms **12**(4), 47:1–47:26 (2016)
20. Yoshida, Y.: Maximizing a monotone submodular function with a bounded curvature under a knapsack constraint. SIAM J. Discret. Math. **33**(3), 1452–1471 (2019)