TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 08/14 pp906–915
DOI: 10.26599/TST.2022.9010033
Volume 28, Number 5, October 2023

# Approximating $(m_B, m_P)$-Monotone BP Maximization and Extensions

Ruiqi Yang, Suixiang Gao, Lu Han, Gaidi Li*, and Zhongrui Zhao

**Abstract:** The paper proposes the optimization problem of maximizing the sum of suBmodular and suPermodular (BP) functions with partial monotonicity under a streaming fashion. In this model, elements are randomly released from the stream and the utility is encoded by the sum of partial monotone suBmodular and suPermodular functions. The goal is to determine whether a subset from the stream of size bounded by parameter $k$ subject to the summarized utility is as large as possible. In this work, a threshold-based streaming algorithm is presented for the BP maximization that attains a $((1-\kappa)/(2-\kappa)-\mathcal{O}(\varepsilon))$-approximation with $\mathcal{O}(1/\varepsilon^4 \log^3(1/\varepsilon)\log((2-\kappa)k/(1-\kappa)^2))$ memory complexity, where $\kappa$ denotes the parameter of supermodularity ratio. We further consider a more general model with fair constraints and present a greedy-based algorithm that obtains the same approximation. We finally investigate this fair model under the streaming fashion and provide a $((1-\kappa)^4/(2-2\kappa+\kappa^2)^2-\mathcal{O}(\varepsilon))$-approximation algorithm.

**Key words:** submodular maximization; streaming model; threshold technique; approximation algorithm

## 1 Introduction

Numerous optimization tasks in theoretical computer science, machine learning, and combinatorial optimization can be designated to find a subset satisfying some constraint with a submodular utility function. A set function $B\colon 2^N \to \mathbf{R}^+$ on a finite ground set $N$ is submodular if it holds $B(S) + B(T) \geqslant B(S \cup T) + B(S \cap T)$. An equivalent form of submodular is defined from the diminishing marginal returns perspective, that is, $B\colon 2^N \to \mathbf{R}^+$ is submodular if for any pair $S \subseteq T \subseteq N$ and $e \notin T$, it holds that $B(S \cup \{e\}) - B(S) \geqslant B(T \cup \{e\}) - B(T)$.

Additionally, a set function is monotone if $B(S) \leqslant B(T)$. The monotonicity property of set functions plays an important role in submodular optimization. Indeed almost every problem encoded with a submodular function has been queried in both monotone and non-monotone settings. Naturally, the submodular optimization with monotone objectives usually enjoys enhanced performance guarantees compared with the case of general objectives. That means that there exist gaps in approximation ratios between the monotonic and non-monotonic submodular optimization problems. For non-negative set functions[2], there is also a parameter of monotonicity ratio that measures how much of the function is monotonic.

In the current work, we investigate a general constrained maximization problem encoded by the sum of suBmodular and suPermodular functions, which is defined as "suBmodular-suPmodular (BP) maximization"[3]. We restate it as follows:

$$\arg\max_{S\subseteq N, S\in I} B(S) + P(S) \qquad (1)$$

where submodular term $B\colon 2^N \to \mathbf{R}^+$ and supermodular term $P\colon 2^N \to \mathbf{R}^+$ are non-negative and monotone, respectively, and $I := \{S \subseteq N, |S| \leqslant k\}$ denotes a $k$-

• Ruiqi Yang, Gaidi Li, and Zhongrui Zhao are with Beijing Institute for Scientific and Engineering Computing, Beijing University of Technology, Beijing 100124, China. E-mail: {yangruiqi, ligd}@bjut.edu.cn; zhaozhongrui@emails.bjut.edu.cn.
• Suixiang Gao is with School of Mathematical Sciences, University of Chinese Academy Sciences, Beijing 100049, China. E-mail: sxgao@ucas.ac.cn.
• Lu Han is with School of Science, Beijing University of Posts and Telecommunications, Beijing 100876, China. E-mail: hl@bupt.edu.cn.
• A preliminary version of this paper has been presented at the 22nd International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT 2021)[1].
∗ To whom correspondence should be addressed.
Manuscript received: 2022-04-29; revised: 2022-07-17; accepted: 2022-08-22

cardinality constraint. Indeed, the BP maximization is fairly difficult, and it has been shown that no constant approximation algorithm exists in polynomial time. By introducing a supermodular curvature, Bai and Bilmes[3] investigated a greedy-based parameterized algorithm for the discussed BP maximization problem.

Recently, applications with huge problem instances have led to studies on submodular optimization under streaming scenarios. In this streaming fashion, elements are not stored in advance, and it is assumed that the elements are read one by one. Threshold-based algorithms are rapidly developed to handle with the problems under streaming[4–9], intuitively selecting elements with marginal gain by at least a proper lower bound. In addition, extended local search-based streaming algorithms have also been proposed[10, 11].

In the current work, we consider the BP maximization under the above streaming model and provide a threshold-based streaming algorithm with theoretical performance guarantees. Then, we introduce the BP maximization with fairness constraints and present a greedy-based algorithm. The results can be found in our conference version[1]. Compared with our conference version, we study a general $(m_B, m_P)$-monotone BP maximization under streaming in this journal version and provide an extended threshold-based streaming algorithm with theoretical performance guarantees. Moreover, we investigate the fair $(m_B, m_P)$-monotone BP maximization under the streaming model and propose a local-search based streaming algorithm for the new model. The results are concluded as follows:

- We develop a semi-streaming algorithm that yields a subset $S$ obeying $\frac{B(S) + P(S)}{m_B \cdot B(O) + m_P \cdot P(O)} \geqslant \frac{1-\kappa}{2-\kappa}$ with $O(k)$ memory complexity (formally summarized as Theorem 1) and then implement it to a full streaming algorithm, which obtains a subset $S$ satisfying $\frac{B(S) + P(S)}{m_B \cdot B(O) + m_P \cdot P(O)} \geqslant \frac{1-\kappa}{2-\kappa} - \mathcal{O}(\varepsilon)$ with $\mathcal{O}\left(\frac{1}{\varepsilon^4}\log^3\left(\frac{1}{\varepsilon}\right)\log\frac{(2-\kappa)k}{(1-\kappa)^2}\right)$ memory complexity for the $(m_B, m_P)$-monotone BP maximization under the one-pass streaming scenario (formally summarized as Theorem 2).
- Then, we provide a greedy-based algorithm that obtains a set $S$ obeying $\frac{B(S) + P(S)}{m_B \cdot B(O) + m_P \cdot P(O)} \geqslant \frac{1-\kappa}{2-\kappa} - \mathcal{O}(\varepsilon)$ for the more general fair $(m_B, m_P)$-monotone BP maximization problem (formally summarized as Theorem 3).
- We provide a streaming algorithm that obtains a set $S$ obeying $\frac{B(S)+P(S)}{m_B \cdot B(O)+m_P \cdot P(O)} \geqslant \frac{(1-\kappa)^4}{(2-2\kappa+\kappa^2)^2} - \mathcal{O}(\varepsilon)$ for the fair $(m_B, m_P)$-monotone BP maximization problem under streaming (formally summarized as Theorem 4).

**Paper organization.** Section 2 reviews some existing related works on BP maximization and streaming algorithms for submodular maximization problems. Section 3 presents some basic concepts of the curvature and monotonicity ratios for the set functions and formally describes the $(m_B, m_P)$-monotone BP maximization problem. Section 4 discusses the $(m_B, m_P)$-monotone BP maximization problem under the streaming model and analyzes algorithms that work for semi (see Section 4.1) and full (see Section 4.2) streaming settings in respectively. Section 5 studies a more general $(m_B, m_P)$-monotone BP maximization problem with fairness constraints. In particular, Section 5.1 presents a greedy-based algorithm, while Section 5.2 provides a streaming algorithm for the discussed problem. Finally, Section 6 finally gives a conclusion for our work.

## 2 Related Work

Numerous studies on BP maximization have been conducted. In this section, we present a quick overview of the literature. The offline problem $\max_{S \in N, S \in I} B(S) + P(S)$ for $k$-cardinality and $p$-matroid constraints has been well-studied, starting with greedy algorithms based on curvatures[3]. Recent studies have obtained new bounds on the approximation ratios via distorted greedy methods[12, 13]. Submodular maximizations with various constraints are popular phenomenon with algorithms developed for submodular functions defined on integer lattices[14–16]. In addition, an extended BP maximization of the sum of a DR submodular plugging a supermodular function on a lattice has been proposed in a previous work[17].

Streaming algorithms for submodular optimization problems with efficient and low-memory requirements have been extensively studied. In the streaming scenario, the inputs of elements are sorted in an arbitrary sequence, and the algorithm is only allowed to visit the stream one by one with respect to this sorted order. The aim of the streaming setting is to find a solution with a fairly high quality while using a limited memory that is usually independent of the input size. Often, the algorithm makes one (single) pass if it is allowed to visit the stream once. Indeed, single-streaming algorithms have been proposed for the submodular problem with a variety of constraints, such as knapsack[18, 19], matroid[20, 21], and a more general independence system[22–24]. In other cases, the algorithms are allowed to visit the input multiple times, which are summarized as multi-pass streaming algorithms. Boosting multi-pass streaming algorithms are widely popular approaches, with algorithms developed for submodular maximization[25, 26], and beyond. Online model is related to our streaming model in which elements are read online and the algorithm is required to maintain a feasible solution without considering the space requirement. Some studies have investigated online submodular maximization[27–29]. Recently, BP maximization under the online model has also been studied[30].

## 3 Preliminary

We begin with this section by investigating the notations to be used throughout this paper. Let $N$ be an element ground whose size may not be accessed in advance. For any $G\colon 2^N \to \mathbf{R}^+$, we denote by $G(e|S) = G(S \cup \{e\}) - G(S)$ the marginal contribution of adding $e$ to $S$ according to $G$. We are motivated by the monotonicity ratio that measures the loss of adding elements to a set and restate it as below.

**Definition 1**[2] Given any non-negative set function $G\colon 2^N \to \mathbf{R}^+$, the monotonicity ratio of this function is defined as the scalar $m_G \in [0, 1]$, such that,

$$m_G = \min_{S \subseteq T \subseteq N} \frac{G(T)}{G(S)} \tag{2}$$

We assume $G(T)/G(S) = 1$ whenever $G(S) = 0$. Additionally, the value of $m_G$ is exactly equal to 1 if the set function $G$ is monotone. A non-negative set function is defined as $m$-monotone if its monotonicity ratio is at least $m$. We introduce an extended total curvature of the $m_B$-monotone submodular function by combining the total curvature of the non-decreasing submodular function[31] and the abovementioned monotonicity ratio.

**Definition 2** Given any non-negative $m_f$-monotone submodular function $B\colon 2^N \to \mathbf{R}^+$, the total curvature of this function is defined as

$$\kappa_B = 1 - \min_{e \in N} \frac{B(e|N \setminus \{e\})}{B(e)} \tag{3}$$

Similarly, function $P\colon 2^N \to \mathbf{R}^+$ is defined as supermodular if for any pair $S \subseteq T \subseteq N$ and element $e \notin T$, it holds the inequality $P(e|S) \leqslant P(e|T)$. A supermodular function $P$ is called as $m_P$-monotone if it is captured by the monotonicity ratio $m$. We similarly obtain an $m_P$-monotone supermodular curvature as follows.

**Definition 3** Given any non-negative $m_P$-monotone supermodular function $P\colon 2^N \to \mathbf{R}^+$, the total curvature of this function is defined as

$$\kappa^P = 1 - \min_{e \in N} \frac{P(e)}{P(e|N \setminus \{e\})} \tag{4}$$

**($m_B, m_P$)-Monotone BP maximization problem.** In this model, given an $m_B$-monotone submodular function and an $m_P$-monotone supermodular function, the goal is to select a subset of size at most $k$, such that the summarized utility $B(S) + P(S)$ is maximized. We formally state the problem as follows:

$$\arg \max_{S \subseteq V, S \in I} \{B(S) + P(S)\} \tag{5}$$

where function $B$ is non-negative $m_B$-monotone and submodular, and function $P$ is non-negative $m_P$-monotone and supermodular.

## 4 ($m_B, m_P$)-Monotone BP Maximization under Streaming

In this section, we present a threshold-based algorithm for the $(m_B, m_P)$-Monotone BP maximization problem under the streaming setting. Recall that in this model, the element ground set $N$ cannot be initially saved, and the elements are released one by one in a stream. Additionally, we are given two non-negative $m_B$-monotone submodular and $m_P$-monotone supermodular functions. The goal is to find a high-quality solution $S$ of size at most $k$ that maximizes $B(S) + P(S)$, while consuming less memory than what is necessary for storing the entire stream. Let $\kappa = \kappa^P$ for clarity in the following sections. The properties of the algorithm we provide are listed as the following theorem.

**Theorem 1** Assume we have access to the optimum threshold value $\tau = \frac{1}{k} \cdot \frac{1-\kappa}{2-\kappa}(m_B B(O) + m_P P(O))$ where $O$ denotes an optimal solution to the $(m_B, m_P)$-monotone BP maximization problem, then Algorithm 1 gets a solution set $S$ obeying

$$\frac{B(S) + P(S)}{m_B \cdot B(O) + m_P \cdot P(O)} \geqslant \frac{1-\kappa}{2-\kappa}.$$

### 4.1 Threshold-based algorithm

The formal algorithm we utilize to show Theorem 1 is listed as Algorithm 1. This algorithm starts with an empty set $S = \varnothing$. For any arriving element $e = e_t$ from the stream, we instantiate a common threshold value as

$$\tau = \frac{1}{k} \cdot \frac{1-\kappa}{2-\kappa}(m_B \cdot B(O) + m_P \cdot P(O)).$$

**Algorithm 1 Threshold-based algorithm**

**Input:** Stream $N = \{e_1, e_2, \ldots, e_n\}$, integer $k$, $m_B$-monotone submodular function $B$, and $m_P$-monotone supermodular function $P$

Set $\tau = \frac{1}{k} \cdot \frac{1-\kappa}{2-\kappa}(m_B \cdot B(O) + m_P \cdot P(O))$ and $S \leftarrow \varnothing$
**while** every element arriving $e$ from stream $N$ **do**
  **if** $|S| < k$ and $B(u|S) + P(u) \geqslant \tau$ **then**
    $S \leftarrow S + e$
  **end if**
**end while**
**return** $S$

If the size of the current solution is less than $k$ and the extended marginal contribution $B(e|S) + P(e)$ of element $e$ with respect to $S$ is larger than $\tau$, then we add the element $e$ to $S$. The algorithm returns the solution $S$ after the stream is entirely visited, or the cardinality constraint is encountered.

We start the analysis of the threshold-based algorithm with the two cases of $|S| = k$ and $|S| < k$, respectively. Our first step toward proving Theorem 1 starts with the lemma below, which gives a lower bound of $B(S) + P(S)$ by the optimum threshold.

**Lemma 1** If $|S| = k$, then we have

$$B(S) + P(S) \geqslant k\tau.$$

**Proof** In this case, without loss of generality, assume $S = \{e_1, e_2, \ldots, e_k\}$ were ordered by to their visited order and set $S^i = \{e_1, e_2, \ldots, e_i\}$ as the first $i$ added elements of $S$. Additionally, we set $S^0 = \varnothing$. Thus we have the following inequality:

$$B(S) + P(S) = \sum_{i=1}^{k}\{B(e_i|S^{i-1}) + P(e_i|S^{i-1})\} \geqslant \sum_{i=1}^{k}\{B(e_i|S^{i-1}) + P(e_i)\} \geqslant k\tau,$$

where the first inequality directly follows by the supermodularity of $P$. ■

Then, we study the second case of $|S| < k$ and also obtain a lower bound on the value of $B(S) + P(S)$ by the optimum.

**Lemma 2** If $|S| < k$, then we have

$$B(S) + P(S) \geqslant m_B \cdot B(O) + m_P \cdot P(O) - \frac{k\tau}{1-\kappa}.$$

**Proof** Consider any $e = e_t \in O \setminus S$. We have $B(e_t|S_t) + P(e_t) < \tau$ because of $|S| < k$, and $e$ is only visited, where $S_t$ denotes the state of $S$ at the time of visiting $e_t$. Then, summing up the above inequalities for all $e \in O \setminus S$, we yield

$$\begin{aligned} & m_B \cdot B(O) - B(S) + m_P \cdot P(O) \leqslant \\ & B(O|S) + m_P \cdot P(O) \leqslant \\ & P(S) + \sum_{e \in O\setminus S}\left\{B(e|S) + P(e|(S \cup O)\{e\})\right\} \leqslant \\ & P(S) + \sum_{e \in O\setminus S}\left\{B(e|S) + \frac{P(e)}{1-\kappa}\right\} \leqslant \\ & P(S) + \frac{k\tau}{1-\kappa}. \end{aligned}$$

Rearranging the above inequality implies that

$$B(S) + P(S) \geqslant m_B \cdot B(O) + m_P \cdot P(O) - \frac{k\tau}{1-\kappa}.$$

Thus, the proof is finished. ■

Combining the two cases discussed above, we derive the lower bound of $B(S) + P(S)$, as stated in Theorem 1.

**Proof of Theorem 1** Based on the previous lemmas we drive the following inequality as follows:

$$B(S) + P(S) \geqslant \min\left\{k\tau, m_B \cdot B(O) + m_P \cdot P(O) - \frac{k\tau}{1-\kappa}\right\}.$$

To obtain the best lower of the above inequality we set

$$k\tau = m_B \cdot B(O) + m_P \cdot P(O) - \frac{k\tau}{1-\kappa},$$

and obtain

$$\tau = \frac{1}{k} \cdot \frac{1-\kappa}{2-\kappa}(m_B \cdot B(O) + m_P \cdot P(O)).$$ ■

### 4.2 Full threshold-based algorithm

The previous threshold-based algorithm requires knowledge of the curvature $\kappa$, as well as monotonicity ratios $m_B$ and $m_P$. However, it is very rare that the precise values of the above parameters are known in practice. Unless $B$ and $P$ are non-decreasing monotones, then $m_B = m_P = 1$. If $g$ is modular then $\kappa = 0$, and if $P$ is fully curved, then $\kappa = 1$. Often, the parameters $m_B, m_P$, and $\kappa$ are data dependent and only a crude common lower bound $L \leqslant \min\{\kappa, m_B, m_P\}$ is known. Thus, we develop a meta algorithm that guesses the values of $m_B, m_P$, and $\kappa$, as well as $(\kappa, m_B, m_P)$-sweep, listed as Algorithm 2. We now discuss the process of handling the last challenge of estimating the value of $\tau$. Indeed, we provide a full threshold-based algorithm, listed as Algorithm 3, and then obtain our main result by combining with the $(\kappa, m_B, m_P)$-sweep. Denote $\mathcal{H}(\kappa) = \frac{1-\kappa}{2-\kappa}$ for clarity. We conclude that

**Algorithm 2** $(\boldsymbol{\kappa}, \boldsymbol{m_B}, \boldsymbol{m_P})$**-sweep**

**Input:** Utilities $B$ and $P$, Algorithm $\mathcal{A}$, lower bound $L$, $\varepsilon \in (0,1)$

```
Set S_{i,j,r} ← ∅ for any i, j, r, T ← ⌈(1/ε) ln(1/max{ε, L})⌉
for i = 0 : T do
  κ ← (1 − δ)^i
  for j = 0 : T do
    m_B ← (1 − δ)^j
    for r = 0 : T do
      m_P ← (1 − δ)^r
      S_{i,j,r} ← A(κ, m_B, m_P)
    end for
  end for
end for
return S ← arg max_{i,j,r} {B(S_{i,j,r}) + P(S_{i,j,r})}
```

**Algorithm 3 Full threshold-based algorithm**

**Input:** Stream $N = \{e_1, e_2, \ldots, e_n\}$, integer $k$, $m_B$-monotone submodular function $B$, and $m_P$-monotone supermodular function $P$

```
Set M_0 ← 0
while every element arriving e from stream N do
  M_t ← max{M_{t−1}, H(κ) · (m_B · B(e) + m_P · P(e))}
  O_t ← {(1 + ε)^i | (1 + ε)^i ∈ [M_t/((1+ε)K), M_t/((1−κ)H(κ))]}
  Delete the threshold values and set S_τ with τ < M_t/((1+ε)K)
  for τ ∈ O_t do
    if τ is a new instantiated threshold then
      S_τ ← ∅
    end if
    if |S_τ| < k and B(e|S_τ) + P(e) ⩾ τ then
      S_τ ← S_τ + e
    end if
  end for
end while
return S = arg max_τ {B(S_τ) + P(S_τ)}
```

$$\max_{u\in N}\left\{\{\mathcal{H}(\kappa)\cdot(m_B\cdot B(e)+m_P\cdot P(e))\right\}\leqslant \mathcal{H}(\kappa)\cdot(m_B\cdot B(O)+m_P\cdot P(O))\leqslant \frac{\mathcal{K}}{1-\mathcal{K}}\cdot\max_{u\in N}\left\{\{\mathcal{H}(\kappa)\cdot(m_B\cdot B(e)+m_P\cdot P(e))\right\},$$

where the first inequality held by $e$ is a candidate to be $O$ for any $e \in N$, and the second inequality is held by the submodularity of $B$ and the supermodular curvature of $P$. Consequently, given any pair $m_B$ and $m_P$, if one has acquired to this maximum singleton value $\mathcal{M} = \max_{u\in\mathcal{V}} \mathcal{H}(\kappa)\cdot(m_B B(e) + m_P P(e))$, then we could guess the threshold value $\tau$ by a geometric series with the form of $(1+\varepsilon)^i$ among the range of $\left[\frac{\mathcal{M}}{(1+\varepsilon)k}, \frac{\mathcal{M}}{1-\kappa}\right]$. It follows that there must exist a threshold $\tilde{\tau}$ satisfying

$$k\tilde{\tau} \leqslant \mathcal{H}(\kappa)\cdot(m_B\cdot B(O)+m_P\cdot P(O)) \leqslant (1+\varepsilon)k\tilde{\tau},$$

and the amount of guesses is bounded by

$$\log_{1+\varepsilon}\frac{(1+\varepsilon)\cdot k\cdot\max_{e\in N}\{\mathcal{H}(\kappa)\cdot(m_B\cdot B(e)+m_P\cdot P(e))\}}{(1-\kappa)\cdot\max_{e\in N}\mathcal{H}(\kappa)\cdot(m_B\cdot B(e)+m_P\cdot P(e))},$$

which is bounded by $\mathcal{O}\left(\frac{1}{\varepsilon}\log\frac{k}{1-\kappa}\right)$. Indeed, the value $\max_{e\in\mathcal{V}}\mathcal{H}(\kappa)\cdot(m_B B(e)+m_P P(e))$ also cannot be accessed to us in advance. Let $\mathcal{M}_t = \max_{e\in\mathcal{N}'}\mathcal{H}(\kappa)\cdot(m_B B(e)+m_P P(e))$ be the maximum singleton value at the current moment, where $N'$ denotes the element set until current time $t$. We can denote the value of $\frac{1}{(1+\varepsilon)k}\max_{e\in N'}\mathcal{H}(\kappa)\cdot(m_B\cdot B(e)+m_P\cdot P(e))$ as the lower bound for threshold $\tau$. Moreover, utilizing the single element value seen so far, a proposition is stated below for the guessing process.

**Proposition 1** Assume we have access to a threshold $\tau > \frac{1}{\mathcal{H}(\kappa)}\cdot\max_{e\in N'}\mathcal{H}(\kappa)\cdot(m_B\cdot B(e) + m_P\cdot P(e))$, then the elements of $\mathcal{N}'$ must not be added to $S$ with respect to $\tau$.

We conclude that it is sufficient to explicitly maintain a copy of Algorithm 1 for values of $\tau$ that fall within the range $\left[\frac{\mathcal{M}_t}{(1+\varepsilon)\cdot k}, \frac{\mathcal{M}_t}{(1-\kappa)\mathcal{H}(\kappa)}\right]$. The memory complexity is stated as

$$\log_{1+\varepsilon}\frac{\frac{1}{(1-\kappa)\mathcal{H}(\kappa)}\cdot\max_{e\in N'}\{\mathcal{H}(\kappa)\cdot(m_B\cdot B(e)+m_P\cdot P(e))\}}{\frac{1}{(1+\varepsilon)\cdot k}\cdot\max_{e\in N'}\{\mathcal{H}(\kappa)\cdot(m_B\cdot B(e)+m_P\cdot P(e))\}},$$

which is at most $\mathcal{O}\left(\frac{1}{\varepsilon}\log\frac{k}{(1-\kappa)\mathcal{H}(\kappa)}\right)$. We now conclude the main results using the theorem below.

**Theorem 2** Assume that some $\varepsilon \in (0,1)$, $(\kappa, m_B, m_P)$-sweep requires at most $\mathcal{O}\left(\frac{1}{\varepsilon^3}\log^3\left(\frac{1}{\varepsilon}\right)\right)$ calls to Algorithm 3 and returns a set $S$ obeying

$$\frac{B(S)+P(S)}{m_B\cdot B(O)+m_P\cdot P(O)} \geqslant \frac{1-\kappa}{2-\kappa} - \mathcal{O}(\varepsilon).$$

Moreover, the memory complexity can be bounded by $\mathcal{O}\left(\frac{1}{\varepsilon^4}\log^3\left(\frac{1}{\varepsilon}\right)\log\frac{(2-\kappa)k}{(1-\kappa)^2}\right)$.

**Proof** As the approximation guarantee mainly follows the proof of Theorem 1, we omit it here. Recall that the total amount of guesses in Algorithm 2 is at most $\mathcal{O}\left(\frac{1}{\varepsilon^3}\log^3\frac{1}{\varepsilon}\right)$. Additionally, we need a maximum of $\mathcal{O}\left(\frac{1}{\varepsilon}\log\frac{k}{(1-\kappa)\mathcal{H}(\kappa)}\right)$ memory to guess the desired threshold value for any given $\kappa$, $m_B$, and $m_P$. Thus, the total memory complexity can be upper bounded by $\mathcal{O}\left(\frac{1}{\varepsilon^4}\log^3\left(\frac{1}{\varepsilon}\right)\log\frac{(2-\kappa)k}{(1-\kappa)^2}\right)$. ■

## 5 ($m_B$, $m_P$)-Monotone BP Maximization in Fairness

In this section, we introduce a fair $(m_B, m_P)$-monotone BP maximization problem and provide an extended distorted fair greedy for this problem. Recall that in our model, we assume there is a finite element partition of $N$, i.e., $N = \uplus_{i=1}^{q} \mathcal{P}_i$. For any part $i \in \{1, 2, \ldots, q\}$, there exist upper and lower bounds denoted by $u_i$ and $\ell_i$, respectively. Let $I = \{S \subseteq V\colon |S| \leqslant k$ and for any $i$, $|S \cap \mathcal{P}_i| \in [\ell_i, u_i]\}$. The goal is to find a subset $S \subseteq V$ with $S \in I$, such that $B(S) + P(S)$ is maximized.

### 5.1 Fair ($m_B$, $m_P$)-monotone BP maximization

We conclude that there may be certain violations of the fairness constraints when the global $k$-cardinality is reached. This is because the elegant greedy justly selects elements during the iterations with maximum marginal values by enumerating over the ground set. A novelty concept of an extendable set has been introduced in a previous work[32], which we restated it as below. A set $S$ is *extendable* if it is a subset $S \subseteq S'$ of some feasible solution set $S' \in \mathcal{I}$. We now restate the useful property as follows.

**Proposition 2** A set $S$ is extendable if and only if for any $\mathcal{P}_i$, it holds that

- $|S \cap \mathcal{P}_i| \leqslant u_i$ and
- $\sum_i \max\{|S \cap \mathcal{P}_i|, \ell_i\} \leqslant k$.

The formal algorithm, listed as Algorithm 4, starts with $\mathcal{S} = \varnothing$ and then selects at each iteration element with the maximum distorted marginal value from an extendable set $U$. We denote by $U = \{e \in N\colon S \cup \{e\}$ is extendable$\}$ the extendable set and denote by $B(e|\mathcal{S}) + \alpha \cdot P(e)$ the distorted marginal value of adding $e$ to $S$.

A mapping constructed from the returned solution set to the optimum is presented as follows.

**Lemma 3** There exists a mapping between $S = \{e_1, e_2, \ldots, e_k\}$ and $O = \{o_1, o_2, \ldots, o_k\}$ satisfying the following conditions:

- for any $i$, both $e_i$ and $o_i$ are partitioned into the same group;
- or $\mathcal{P}[e_i] \neq \mathcal{P}[o_i]$, where $\mathcal{P}[e]$ denotes the part of $e$ belonging to, then $|S \cap \mathcal{P}[e_i]| > |O \cap \mathcal{P}[e_i]|$ and $|S \cap \mathcal{P}[o_i]| < |O \cap \mathcal{P}[e_i]|$.

**Proof** One can recursively construct the mapping as follows, matching $e_i$ and $o_i$ together as long as they belong to the same part. Once the remaining elements belong to different parts, they are then matched randomly. ■

**Algorithm 4 Distorted-fair-greedy**

**Input:** Ground set $N = \{e_1, e_2, \ldots, e_n\}$, integer $k$, $m_B$-monotone submodular function $B$, and $m_P$-monotone supermodular function $P$

1: Set $S \leftarrow \varnothing$
2: **while** $|S| < k$ **do**
3: $\quad U \leftarrow \{e \in N\colon S \cup \{e\}$ is extendible$\}$
4: $\quad S \leftarrow S + \arg\max_{e \in U}\{B(e|S) + \alpha \cdot P(e)\}$
5: **end while**
6: **return** $S$

Following the mapping process described above, it can be concluded that $S \setminus \{e_i\} \cup \{o_i\}$ is feasible for any $i$. Next, we obtain the main results, as shown below.

**Theorem 3** Given some $\varepsilon \in (0, 1)$ and let $\alpha = \dfrac{2}{2-\kappa}$. Then, $(\kappa, m_B, m_P)$-sweep requires a maximum of $\mathcal{O}\left(\dfrac{1}{\varepsilon}\log\left(\dfrac{1}{\varepsilon}\right)\right)$ calls to Algorithm 4 and returns a set $S$ obeying

$$\frac{B(S) + P(S)}{m_B \cdot B(O) + m_P \cdot P(O)} \geqslant \frac{1-\kappa}{2-\kappa} - O(\varepsilon).$$

**Proof** Assume we obtain the true supermodularity curvature $\kappa$ in advance. Then, we have the following inequalities:

$$\begin{aligned}
&B(S) + \alpha \cdot P(S) \geqslant \\
&\quad \sum_{i=1}^{k}\{B(u_i|S_{i-1}) + \alpha \cdot P(u_i)\} \geqslant \\
&\quad \sum_{i=1}^{k}\{B(o_i|S_{i-1}) + \alpha \cdot P(o_i)\} \geqslant \\
&\quad \sum_{i=1}^{k}\{B(o_i|S_k \cup O_{i-1}) + \alpha(1-\kappa)P(o_i|S_k \cup O_{i-1})\} \geqslant \\
&\quad \{m_B \cdot B(O) - B(S) + \alpha(1-\kappa)(m_P \cdot P(O) - P(S))\}.
\end{aligned}$$

The first inequality follows by the supermodularity, the second follows the extensibility of $S_{i-1} \cup \{o_i\}$, the third combines the definitions of submodularity and $\kappa$, and the last one obtains by the monotonicity ratios. Setting $\alpha = \dfrac{2}{2-\kappa}$, we obtain

$$\frac{B(S) + P(S)}{m_B \cdot B(O) + m_P \cdot P(O)} \geqslant \frac{1-\kappa}{2-\kappa}.$$

Moreover, we only guess a nearly approximate value of $\kappa$ in at most $\mathcal{O}(\frac{1}{\varepsilon}\log(\frac{1}{\varepsilon}))$ calls. Thus, the theorem is considered a desirable tool for this process. ■

### 5.2 Fair ($m_B$, $m_P$)-monotone BP maximization under streaming

In this section we present our algorithm for the fair $(m_B, m_P)$-monotone BP maximization in the streaming setting and then prove its approximation guarantee below. Our algorithm mainly follows a previously reported framework[32] and is formally described as Algorithm 5.

We also are motivated by the extendable subsets of the stream constructing a matroid system. Note that the remaining constraints would give rise to a partition matroid by removing the lower-bound constraints. We assume that there are efficient streaming algorithms for the $(m_B, m_P)$-monotone BP maximization under a matroid constraint, which we could use in a black-box style. We denote by $\mathcal{A}$ the streaming algorithm for the above $(m_B, m_P)$-monotone BP maximization problem. We conclude that the solution produced by Algorithm $\mathcal{A}$ should violate the lower-bound constraints. Otherwise, we simply return the solution as the output, after which we augment the solution to a feasible one by using "backup" elements obtained during the stream processing. We similarly obtain a claim for the fair $(m_B, m_P)$-monotone BP maximization.

**Lemma 4** We denote by $\mathcal{A}$ the streaming algorithm for the $(m_B, m_P)$-monotone BP maximization under a matroid constraint. Then we find a streaming algorithm for fair $(m_B, m_P)$-monotone BP maximization with the same approximation ratio and memory usage as $\mathcal{A}$.

Applying the above lemma to Algorithm 6, we obtain the following result.

**Theorem 4** Given some $\varepsilon \in (0, 1)$ and let $\alpha = 1 + \frac{1}{(1-\kappa)^2}$. Then, $(\kappa, m_B, m_P)$-Sweep requires at most $\mathcal{O}(\frac{1}{\varepsilon}\log(\frac{1}{\varepsilon}))$ calls to Algorithm 5 and returns a set $S$ obeying

**Algorithm 5 Distorted-fair-stream**

**Input:** Stream $N = \{e_1, e_2, \dots, e_n\}$, integer $k$, $m_B$-monotone submodular function $B$, and $m_P$-monotone supermodular function $P$

Set $S_{\mathcal{A}} \leftarrow \varnothing$ and $B_i = \varnothing$ for all $i \in \{1, 2, \dots, q\}$
**for** every element arriving $e$ belonging to part $\mathcal{P}_i$ **do**
  Fed $e$ to Algorithm $\mathcal{A}$
  **if** $|B_i| < \ell_i$ **then**
    $B_i \leftarrow B_i + e$
  **end if**
**end for**
$S_{\mathcal{A}} \leftarrow$ solution of Algorithm $\mathcal{A}$
$S \leftarrow S_{\mathcal{A}}$ augmented with elements in sets $B_i$
**return** $S$

**Algorithm 6 Local search under stream**

**Input:** Stream $N = \{e_1, e_2, \dots, e_n\}$, $m_B$-monotone submodular function $B$, $m_P$-monotone supermodular function $P$, independence oracle for matroid system $(N, I)$, and $\alpha = 1 + \frac{1}{(1-\kappa)^2}$

Set $S_0 \leftarrow \varnothing$, $t \leftarrow 1$
**for** every element arriving $e = e_t$ from the stream $N$ **do**
  **if** $S_{t-1} + e_t \in I$ **then**
    $S_t \leftarrow S_{t-1} + e_t$
  **else**
    Set $C_t$ as the single cycle in $S_{t-1} + e_t$
    Set $e'_t \in \arg\min_{e'_t \in C_t - e_t} H(e'_t\colon S_{t-1})$
  **end if**
  **if** $H(e_t|S_{t-1}) \geqslant \alpha H(e'_t : S_{t-1})$ **then**
    Set $S_t \leftarrow S_{t-1} + e_t - e'_t$
  **else**
    Set $S_t \leftarrow S_{t-1}$
  **end if**
**end for**
**return** $S_n$

$$\frac{B(S)+P(S)}{m_B \cdot B(O) + m_P \cdot P(O)} \geqslant \frac{(1-\kappa)^4}{(2-2\kappa+\kappa^2)^2} - O(\varepsilon).$$

We begin with a local search based streaming algorithm for $(m_B, m_P)$-monotone BP maximization under a matroid constraint, which appears as Algorithm 6. Next, we present some necessary terminologies as follows. Let $H(S) = B(S) + P(S)$ for any $S \subseteq N$. Notably, $N$ is ordered in an arbitrary order in our streaming setting. We also introduce $\nu(H, S, e) = H(e|S')$ as the incremental value of encountering $e$ with respect to $S$ where $S' = \{s \in S\colon s < e\}$ denotes the state of $S$ before $e$. It follows that $H(S) = \sum_{s\in S} \nu(H, S, e)$ and $(1-\kappa)\nu(H, T, e) \leqslant \nu(H, S, e)$ for any pair $S \subseteq T \subseteq N$. Additionally, it holds $(1-\kappa)\nu(H_T, S, e) \leqslant \nu(H, S\cup T, e)$ for any $S, T \subseteq N$. We denote by $A$ the set of all elements that have ever appeared in the solution of Algorithm 6, i.e., $A = \cup_{t=0}^{n} S_t$. For any $e = e_t \in N$, we denote by $S_t^-$ and $S_t^+$ the states of $S$ before and after $e_t$ is processed, respectively. Obviously, $S_t^- = S_t^+$ if the element $e_t$ is not accepted by the algorithm. We further denote $\delta_t = H(S_t^+) - H(S_t^-)$ as the contribution of processing element $e_t$ and then $\delta_t = 0$ for any $e_t \in N \setminus A$ and $H(S) = \sum_{e_t\in N} \delta_t$. Then, we obtain a lower bound for the above contribution by the following lemma.

**Lemma 5** For any $e_t \in A$, we have

$$\delta_t \geqslant (1-\kappa)\nu(H, S_t^-, e'_t).$$

**Proof** Given that $e = e_t \in A$ is replaced by $e'_t$ via Algorithm 6 and $\alpha = 1 + (1-\kappa)^2$, then it holds that $H(e_t|S_t^-) \geqslant \alpha\nu(H, S_t^-, e_t')$. Let $Z = S_t^- - e_t' = S_t^+ + e_t$. Thus,

$$\delta_t = H(Z + e_t) - H(Z + e_t') \geqslant$$
$$(1-\kappa)H(e_t|S_t^-) - H(e_t'|Z) \geqslant$$
$$(1-\kappa)H(e_t|S_t^-) - \frac{\nu(H, S_t^-, e_t')}{1-\kappa} \geqslant$$
$$(1-\kappa)\nu(H, S_t^-, e_t').$$

Then the lemma follows. ■

For any $d = d_t \in A \setminus S$, we denote by $e_{\tilde{t}} = e(d_t)$ the element that $d$ has been swapped for aid $d_t \in S_{\tilde{t}}^+ \setminus S_{\tilde{t}}^-$. Moreover, let $\chi(d_t) = \nu(H, S_{\tilde{t}}^-, d_t)$. Then, the summed marginal values of $\chi(\cdot)$ in $A \setminus S$ can be bounded below.

**Lemma 6**

$$\sum_{d_t \in A\setminus S} \chi(d_t) \leqslant \frac{H(S)}{1-\kappa}.$$

**Proof** We have the following inequality:

$$\sum_{d_t \in A\setminus S} \chi(d_t) = \sum_{u\in U} \nu(H, S_{\tilde{t}}^-, d_t) \leqslant$$
$$\sum_{u\in U} \nu(H, S_{\tilde{t}}^-, d_t) \leqslant \frac{\delta_{\tilde{t}}}{1-\kappa} = \frac{H(S)}{1-\kappa}.$$

Then, the lemma follows. ■

**Lemma 7** We now bound the value of $H(A)$ based on the above lemma. We have the following inequality:

$$H(A) \leqslant \left(1 + \frac{1}{(1-\kappa)^2}\right) H(S).$$

**Proof** Note that

$$H(A) - H(S) = \sum_{d\in A\setminus S} \nu(H_S, A, d) \leqslant$$
$$\sum_{d_t\in A\setminus S} \frac{\chi(d_t)}{1-\kappa} \leqslant \frac{H(S)}{(1-\kappa)^2},$$

where the last inequality is obtained by Lemma 6. ■

For any feasible $T \in I$, there exists an exchange lemma between $T$ and $A \setminus S$. We summarize this as the following lemma.

**Lemma 8** There is an injection $\pi : T \to S \cup \varnothing$ that satisfies the following conditions:

• Each $s \in S$ appears in $\pi(t)$ at most a single time choice of $t \in T$;

• For any $t \in T$, it holds

$$\nu(H, S_t^-, e_t') \leqslant \nu(H, S, \pi(e_t)),$$

where $\pi(e_t) \in S$.

Then by setting $T = O$, where $O$ denotes an optimal solution for this $(m_B, m_P)$-monotone BP maximization under a matroid constraint. We now bound $B(S)+P(S)$.

**Proof of Theorem 4** Consider any $e_t \in O \setminus A$, and assume $\pi$: $O \to A$ is the mapping produced by Lemma 8. Then we get

$$\sum_{e_t\in O\setminus A} \nu(H, S_t^-, e_t') \leqslant$$
$$\sum_{e_t\in O\setminus A} \nu(H, S, \pi(e_t)) \leqslant$$
$$\sum_{s\in S} \nu(H, S, \pi(e_t)) \leqslant H(S).$$

In addition, we have

$$m_B B(O) + m_P P(O) \leqslant H(O \cup A) \leqslant$$
$$H(A) + \frac{1}{1-\kappa} \sum_{e_t\in O\setminus A} H(e_t|A) \leqslant$$
$$H(A) + \frac{1}{(1-\kappa)^2} \sum_{e_t\in O\setminus A} H(e_t|S_t^-) \leqslant$$
$$H(A) + \frac{1+(1-\kappa)^2}{(1-\kappa)^4} \sum_{e_t\in O\setminus A} \nu(H, S_t^-, e_t') \leqslant$$
$$\frac{(2-2\kappa+\kappa^2)^2}{(1-\kappa)^4} H(S).$$

Thus the proof is completed. ■

## 6 Conclusion

In this work we first provided a threshold-based streaming algorithm with a performance guarantee for the $(m_B, m_P)$-monotone BP maximization under streaming. Then, we studied a general $(m_B, m_P)$-monotone BP maximization with fairness constraints and presented a fair greedy-based algorithm. Finally, we considered the fair $(m_B, m_P)$-monotone BP maximization under streaming and provided a local search based streaming algorithm with computable guarantees. In our future work, we will pay attention to the settings of $(m_B, m_P)$-monotone BP maximization with more complex constraints in practice.

## Acknowledgment

This work was supported by the National Natural Science Foundation of China (No. 12101587), the China Postdoctoral Science Foundation (No. 2022M720329), the National Natural Science Foundation of China (No. 12001523), the Beijing Natural Science Foundation Project (No. Z200002), and the National Natural Science Foundation of China (No. 12131003).

## References

[1] R. Yang, S. Gao, L. Han, G. Li, and Z. Zhao, Approximating BP maximization with distorted-based strategy, in *Proc. 22nd Int. Conf. on Parallel and Distributed Computing: Applications and Technologies*, Guangzhou, China, 2022,

pp. 452–459.

[2] L. Mualem and M. Feldman, Using partial monotonicity in submodular maximization, arXiv prefprint arXiv: 2202.03051, 2022.

[3] W. Bai and J. Bilmes, Greed is still good: Maximizing monotone submodular + supermodular (BP) functions, in *Proc. 35$^{th}$ Int. Conf. on Machine Learning*, Stockholm, Sweden, 2018, pp. 304–313.

[4] A. Badanidiyuru, B. Mirzasoleiman, A. Karbasi, and A. Krause, Streaming submodular maximization: Massive data summarization on the fly, in *Proc. 20$^{th}$ ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, New York, NY, USA, 2014, pp. 671–680.

[5] M. Feldman, A. Norouzi-Fard, O. Svensson, and R. Zenklusen, The one-way communication complexity of submodular maximization with applications to streaming and robustness, in *Proc. 52$^{nd}$ Ann. ACM SIGACT Symp. on Theory on Computing*, Chicago, IL, USA, 2020, pp. 1363–1374.

[6] R. Gomes and A. Krause, Budgeted nonparametric learning from data streams, in *Proc. 27$^{th}$ Int. Conf. on Machine Learning*, Haifa, Israel, 2010, pp. 391–398.

[7] E. Kazemi, M. Mitrovic, M. Zadimoghaddam, S. Lattanzi, and A. Karbasi, Submodular streaming in all its glory: Tight approximation, minimum memory and low adaptive complexity, in *Proc. 36$^{th}$ Int. Conf. on Machine Learning*, Long Beach, CA, USA, 2019, pp. 3311–3320.

[8] B. Mirzasoleiman, A. Badanidiyuru, and A. Karbasi, Fast constrained submodular maximization: Personalized data summarization, in *Proc. 33$^{rd}$ Int. Conf. on Machine Learning*, New York City, NY, USA, 2016, pp. 1358–1367.

[9] A. Norouzi-Fard, J. Tarnawski, S. Mitrović, A. Zandieh, A. Mousavifar, and O. Svensson, Beyond $1/2$-approximation for submodular maximization on massive data streams, in *Proc. 35$^{th}$ Int. Conf. on Machine Learning*, Stockholm, Sweden, 2018, pp. 3829–3838.

[10] N. Buchbinder, M. Feldman, and R. Schwartz, Online submodular maximization with preemption, in *Proc. 26$^{th}$ Ann. ACM-SIAM Symp. on Discrete Algorithms*, San Diego, CA, USA, 2015, pp. 1202–1216.

[11] C. Chekuri, S. Gupta, and K. Quanrud, Streaming algorithms for submodular function maximization, in *Proc. 42$^{nd}$ Int. Colloquium on Automata, Languages, and Programming*, Kyoto, Japan, 2015, pp. 318–330.

[12] Z. Liu, H. Chang, D. Du, and X. Zhang, Improved algorithms for non-submodular function maximization problem, *Theor. Comput. Sci.*, vol. 931, pp. 49–55, 2022.

[13] Z. Liu, L. Guo, D. Du, D. Xu, and X. Zhang, Maximization problems of balancing submodular relevance and supermodular diversity, *J. Glob. Optim.*, vol. 82, no. 1, pp. 179–194, 2022.

[14] A. Kuhnle, J. D. Smith, V. Crawford, and M. Thai, Fast maximization of non-submodular, monotonic functions on the integer lattice, in *Proc. 35$^{th}$ Int. Conf. on Machine Learning*, Stockholm, Sweden, 2018, pp. 2786–2795.

[15] T. Soma, N. Kakimura, K. Inaba, and K. Kawarabayashi, Optimal budget allocation: Theoretical guarantee and efficient algorithm, in *Proc. 31$^{st}$ Int. Conf. on Machine Learning*, Beijing, China, 2014, pp. 351–359.

[16] T. Soma and Y. Yoshida, Maximizing monotone submodular functions over the integer lattice, *Math. Program.*, vol 172, nos. 1&2, pp. 539–563, 2018.

[17] Z. Zhang, D. Du, Y. Jiang, and C. Wu, Maximizing DR-submodular + supermodular functions on the integer lattice subject to a cardinality constraint, *J, Glob. Optim.*, vol. 80, no. 3, pp. 595–616, 2021.

[18] C. C. Huang and N. Kakimura, Improved streaming algorithms for maximizing monotone submodular functions under a knapsack constraint, *Algorithmica*, vol. 83, no. 3, pp. 879–902, 2021.

[19] C. C. Huang, N. Kakimura, and Y. Yoshida, Streaming algorithms for maximizing monotone submodular functions under a knapsack constraint, *Algorithmica*, vol. 82, no. 4, pp. 1006–1032, 2020.

[20] A. Chakrabarti and S. Kale, Submodular maximization meets streaming: Matchings, matroids, and more, *Math. Program.*, vol. 154, nos. 1&2, pp. 225–247, 2015.

[21] M. Feldman, A. Norouzi-Fard, O. Svensson, and R. Zenklusen, Streaming submodular maximization with matroid and matching constraints, arXiv prefprint arXiv: 2107.07183, 2022.

[22] M. Feldman, A. Karbasi, and E. Kazemi, Do less, get more: Streaming submodular maximization with subsampling, in *Proc. 32$^{nd}$ Ann. Conf. on Neural Information Processing Systems*, Montréal, Canada, 2018, pp. 730–740.

[23] R. Haba, E. Kazemi, M. Feldman, and A. Karbasi, Streaming submodular maximization under a $k$-set system constraint, in *Proc. 37$^{th}$ Int. Conf. on Machine Learning*, Vienna, Austria, 2020, pp. 3939–3949.

[24] B. Mirzasoleiman, S. Jegelka, and A. Krause, Streaming non-monotone submodular maximization: Personalized video summarization on the fly, in *Proc. 32$^{nd}$ AAAI Conf. Artificial Intelligence and 30th Innovative Applications of Artificial Intelligence Conf. and 8th AAAI Symp. Educational Advances in Artificial Intelligence*, New Orleans, LA, USA, 2018, pp. 1379–1386.

[25] C. C. Huang, and N. Kakimura, Multi-pass streaming algorithms for monotone submodular function maximization, *Theory Comput. Syst.*, vol. 66, no. 1, pp. 354–394, 2022.

[26] A. Kuhnle, Quick streaming algorithms for the maximizing of monotone submodular functions in linear time, in *Proc. 24$^{th}$ Int. Conf. on Artificial Intelligence and Statistics*, Virtual Event, 2021, pp. 1360–1368.

[27] M. H. Bateni, M. Hajiaghayi, and M. Zadimoghaddam, Submodular secretary problem and extensions, *ACM Trans. Algorithms*, vol. 9, no. 4, p. 32, 2013.

[28] N. Buchbinder, M. Feldman, Y. Filmus, and M. Garg, Online submodular maximization: Beating $1/2$ made simple, *Math. Program.*, vol. 183, nos. 1&2, pp. 149–169, 2020.

[29] A. Gupta, A. Roth, G. Schoenebeck, and K. Talwar, Constrained non-monotone submodular maximization: Offline and secretary algorithms, in *Proc. 6$^{th}$ Int. Workshop on Internet and Network Economics*, Stanford, CA, USA, 2010, pp. 246–257.

[30] Z. Liu, L. Chen, H. Chang, D. Du, and X. Zhang, Online

algorithms for BP functions maximization, *Theor. Comput. Sci.*, vol. 858, pp. 114–121, 2021.

[31] M. Conforti and 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.*, vol. 7, no. 3, pp. 251–274, 1984.

[32] M. EI. Halabi, S. Mitrovi, A. Norouzi-Fard, J. Tardos, and J. Tarnawski, Fairness in streaming submodular maximization: Algorithms and hardness, in *Proc. 34$^{th}$ Int. Conf. Neural Information Processing Systems*, Vancouver, Canada, 2020, pp. 13609–13622.

**Ruiqi Yang** received the PhD degree from Beijing University of Technology, China in 2020. From June 2020 to June 2022, he worked as a postdoctoral researcher at the School of Mathematical Sciences, University of Chinese Academy of Sciences, along with his co-supervisor professor Suixiang Gao. He is currently working as a teacher at Beijing University of Technology. His research interests include combinatorial optimization, approximation algorithms, and submodular optimization.

**Suixiang Gao** received the PhD degree from Institute of Applied Mathematics, Chinese Academy of Sciences, China in 1998. He is currently a professor at the School of Mathematical Sciences, University of Chinese Academy of Sciences. His research focuses on combinatorial optimization, network optimization, and approximation algorithms.

**Lu Han** received the PhD degree from Beijing University of Technology, China in 2019. Currently, she is an assistant professor at Beijing University of Posts and Telecommunications. Her research interests include combinatorial optimization, approximation algorithm, and machine learning.

**Gaidi Li** received the PhD degree from the Academy of Mathematics and Systems Science, Chinese Academy of Sciences in 2004. She is currently an associate professor at the Faculty of Science, Beijing University of Technology. Her research interests include nonlinear optimization, approximation algorithms, and combinatorial optimization.

**Zhongrui Zhao** received the BEng degree from Qingdao University of Technology, China in 2020. He is currently a master student at Beijing University of Technology. His research interests include combinatorial optimization, approximation algorithm, and submodular optimization.