Streaming Algorithm for Monotone k-Submodular Maximization with Cardinality Constraints

Alina Ene [*1] **Huy L. Nguyen** [*2]

Abstract

Maximizing a monotone k-submodular function subject to cardinality constraints is a general model for several applications ranging from influence maximization with multiple products to sensor placement with multiple sensor types and online ad allocation. Due to the large problem scale in many applications and the online nature of ad allocation, a need arises for algorithms that process elements in a streaming fashion and possibly make online decisions. In this work, we develop a new streaming algorithm for maximizing a monotone k-submodular function subject to a per-coordinate cardinality constraint attaining an approximation guarantee close to the state of the art guarantee in the offline setting. Though not typical for streaming algorithms, our streaming algorithm also readily applies to the online setting with free disposal. Our algorithm is combinatorial and enjoys fast running time and small number of function evaluations. Furthermore, its guarantee improves as the cardinality constraints get larger, which is especially suited for the large scale applications. For the special case of maximizing a submodular function with large budgets, our combinatorial algorithm matches the guarantee of the state-of-the-art continuous algorithm, which requires significantly more time and function evaluations.

1. Introduction

In this paper, we study the problem of maximizing a k-submodular function subject to size constraints. In this problem, we have a finite set V of elements. The goal is to select a k-tuple $(S_1, \ldots, S_k)$ of disjoint subsets of V that maximizes an objective function f subject to the constraints that each set S_i has size at most a given budget B_i. In many applications, the objective function f is monotone and k-submodular: the function value can only increase as the sets S_i increase, and the function exhibits a natural analogue of the diminishing returns property of submodular functions. When $k = 1$, the problem coincides with the well-studied problem of maximizing a monotone submodular function subject to a cardinality constraint.

The problem is a general model for several applications (Feldman et al., 2009; Ohsaka & Yoshida, 2015). One motivating application is the influence maximization problem with multiple products or topics (Ohsaka & Yoshida, 2015). In this setting, the goal is to select a seed set S_i for each product $i \in [k]$ in order to maximize the total influence of the seed sets $(S_1, \ldots, S_k)$ subject to constraints on the sizes of each of the seed sets. In the well-studied models of influence propagation — the independent cascades and the linear threshold models — the influence function is monotone and k-submodular (Ohsaka & Yoshida, 2015).

A second motivating application arises in sensor placement with different types of sensors (Ohsaka & Yoshida, 2015). Here we have k types of sensors, each of which provides different measurements, and we have B_i sensors of type i for each $i \in [k]$. The ground set V is a set of possible locations for the sensors. The goal is to equip each location with at most one sensor in order to maximize the total information gained from the sensors.

A third motivating application arises in online advertising. In the online ad allocation problem (Feldman et al., 2009), we have k advertisers that are known in advance and n ad impressions that arrive online one at a time. Each advertiser $i \in [k]$ has a contract of B_i impressions. For each impression j and each advertiser i, there is a non-negative weight w_{ji} that captures how much value advertiser i accrues from being allocated impression j. When impression j arrives, the values $\{w_{ji} \colon i \in [k]\}$ are revealed, and the algorithm needs to allocate impression j to at most one advertiser. In the online ad allocation with free disposal, the algorithm may allocate more than B_i impressions to advertiser i, and advertiser i is charged for only the B_i

[*]Equal contribution [1]Department of Computer Science, Boston University [2]Khoury College of Computer and Information Science, Northeastern University. Correspondence to: Alina Ene <aene@bu.edu>, Huy Nguyen <hu.nguyen@northeastern.edu>.

Proceedings of the 39[th] International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

most valuable impressions allocated to it. Letting X_i denote the set of impressions allocated to advertiser i, the total revenue is $\sum_{i=1}^{k} \max\left\{\sum_{j \in S_i} w_{ji} \colon S_i \subseteq X_i, |S_i| \leq B_i\right\}$. This problem is a special case of k-submodular maximization and submodular maximization with a partition matroid constraint.

In applications such as those discussed above, the size of the data available as well as the nature of the problem make it necessary to design algorithms that process each element as it arrives, and are very efficient both in terms of time and space. In this work, we design a novel algorithm for monotone k-submodular maximization subject to size constraints that simultaneously meets all of these desiderata. Our algorithm makes a single pass over the data, and it is both a streaming algorithm as well as an online algorithm in the free disposal setting. The running time and space usage of the algorithm are optimal. The approximation guarantee of the algorithm is at least $\frac{1}{4}$ and it improves as the minimum budget $B = \min_{i \in [k]} B_i$ increases, and it approaches $\frac{1}{2(1+\ln 2)} \approx 0.2953$ as B tends to infinity. In large scale applications, like ads allocation, the budget (number of impressions) tends to be large, which is precisely the setting where our algorithm performs well. The state of the art algorithms for the problem are offline algorithms that are based on Greedy (Ohsaka & Yoshida, 2015). These algorithms achieve a slightly higher approximation guarantee of $\frac{1}{3}$ at the cost of a significantly higher space usage and higher running time. Table 1 presents a detailed comparison.

Our algorithm also readily applies to the related problem of maximizing a monotone submodular function subject to a partition matroid constraint. The submodular problem is more restrictive than the k-submodular one and thus its modeling power is more limited. However, several applications, such as the ad allocation problem discussed above, can also be modeled as submodular maximization. We show that our algorithm achieves an improved approximation guarantee for the submodular problem when the minimum budget $B = \min_{i \in [k]} B_i$ is sufficiently large. The approximation guarantee approaches 0.3178 as B tends to infinity, matching the recent result of (Feldman et al., 2021). The work (Feldman et al., 2021) gives a streaming algorithm for a general matroid constraint with an approximation guarantee of 0.3178 that holds regardless of the minimum budget. This algorithm is a continuous one that is based on the multilinear extension, which is more expensive to evaluate, leading to a high polynomial running time. Additionally, the algorithm stores multiple solutions in memory and it is not suitable for the online setting with free disposal. In contrast, our algorithm is combinatorial and more efficient both in terms of time and space. Table 2 presents a detailed comparison of our algorithms with the state of the art algorithms for submodular maximization with a partition matroid constraint.

Other related work: The works (Ohsaka & Yoshida, 2015; Nguyen & Thai, 2020) study the problem of maximizing a monotone k-submodular function subject to a common budget constraint B, i.e., the goal is to construct a solution $(S_1, \ldots, S_k)$ satisfying $|S_1 \cup \cdots \cup S_k| \leq B$. The work (Ohsaka & Yoshida, 2015) gives an offline algorithm based on Greedy that achieves a $\frac{1}{2}$ approximation, and (Nguyen & Thai, 2020) gives a streaming algorithm based on single-threshold Greedy that achieves a $\frac{1}{3} - \epsilon$ approximation. The latter algorithm makes multiple guesses for the threshold and stores multiple solutions in memory, and therefore it is not suitable for the online setting with free disposal. The recent work (Pham et al., 2021) extends the algorithm of (Nguyen & Thai, 2020) to the setting where we have a common knapsack constraint. Several works give offline algorithms for the unconstrained non-monotone k-submodular maximization problem (Ward & Živný, 2016; Iwata et al., 2016; Oshima, 2017; Soma, 2019). The work (Santiago & Shepherd, 2019) introduces a different generalization of submodularity to the multivariate setting where the goal is to select k sets that are not necessarily disjoint, and provides offline and distributed greedy algorithms for general classes of constraints.

2. Preliminaries

2.1. Notation and Definitions

We follow the notation used in prior work (Ward & Živný, 2016; Ohsaka & Yoshida, 2015). Let $[k] = \{1, 2, \ldots, k\}$. Let V be a finite ground set of elements. We consider k-tuples $\mathbf{X} = (X_1, \ldots, X_k)$ of disjoint subsets of V: $X_i \subseteq V$ for all $i \in [k]$, and $X_i \cap X_j = \emptyset$ for all $i \neq j$. We let $(k+1)^V = \{(X_1, X_2, \ldots, X_k) : X_i \subseteq V\ \forall i, X_i \cap X_j = \emptyset\ \forall i \neq j\}$ be the set of all such k-tuples. For two k-tuples $\mathbf{X} = (X_1, \ldots, X_k)$ and $\mathbf{Y} = (Y_1, \ldots, Y_k)$, we write $\mathbf{X} \preceq \mathbf{Y}$ if and only if $X_i \subseteq Y_i$ for all $i \in [k]$. We let $\operatorname{supp}(\mathbf{X}) = X_1 \cup \cdots \cup X_k$.

Let $f : (k+1)^V \to \mathbb{R}_+$ be a non-negative function. The function f is **k-submodular** if

$$f(\mathbf{X}) + f(\mathbf{Y}) \geq f(\mathbf{X} \sqcap \mathbf{Y}) + f(\mathbf{X} \sqcup \mathbf{Y})\ \forall \mathbf{X}, \mathbf{Y} \in (k+1)^V$$

where $\mathbf{X} \sqcap \mathbf{Y}$ is the k-tuple whose i-th set is $X_i \cap Y_i$ and $\mathbf{X} \sqcup \mathbf{Y}$ is the k-tuple whose i-th set is $(X_i \cup Y_i) \setminus (\cup_{j \neq i} (X_j \cup Y_j))$.

The function f is **monotone** if

$$f(\mathbf{X}) \leq f(\mathbf{Y}) \quad \forall \mathbf{X} \preceq \mathbf{Y}$$

We let $\Delta_{e,i} f(\mathbf{X})$ denote the marginal gain of adding e to the i-th set of $\mathbf{X}$:

$$\Delta_{e,i} f(\mathbf{X}) := f(X_1, \ldots, X_i \cup \{e\}, \ldots, X_k)$$

Table 1. Comparison of algorithms for monotone k-submodular maximization with size constraints. We let $B = \min_{i\in[k]} B_i$ denote the minimum budget, $r = \sum_{i=1}^{k} B_i$ denote the total budget, and $n = |V|$.

Reference	Setting	Approximation	Time	Space
(Ohsaka & Yoshida, 2015)	offline	$\frac{1}{3}$	$O(nkr)$	$O(n)$
(Ohsaka & Yoshida, 2015)	offline	$\frac{1}{3}$ with prob. $\geq 1-\delta$	$O\left(nk^2 \log\left(\frac{r}{k}\right) \log\left(\frac{r}{\delta}\right)\right)$	$O(n)$
Theorem 3.1 **(This paper)**	online, streaming	$\frac{1}{2\left(1+B\left(2^{1/B}-1\right)\right)} \geq \frac{1}{4}$ $\rightarrow \frac{1}{2(1+\ln 2)} \approx 0.2953$ as $B\rightarrow\infty$	$O(nk)$	$O(r)$

Table 2. Comparison of algorithms for monotone submodular maximization with a partition matroid constraint. We let $B = \min_{i\in[k]} B_i$ denote the minimum budget, $r = \sum_{i=1}^{k} B_i$ denote the total budget, and $n = |V|$.

Reference	Setting	Approximation	Time	Space
(Fisher et al., 1978)	offline discrete	$\frac{1}{2}$	$O(nr)$	$O(n)$
(Chakrabarti & Kale, 2015) (Chekuri et al., 2015)	online, streaming discrete	$\frac{1}{4}$	$O(n\log r)$	$O\left(r\log^{O(1)} n\right)$
Theorem 4.1 **(This paper)**	online, streaming discrete	$\max_{x\geq 0}\left(\frac{1}{x+1}\left(1-\left(1+\frac{x}{B}\right)^{-B}\right)\right) \geq \frac{1}{4}$ $\rightarrow\approx 0.3178$ as $B\rightarrow\infty$	$O(n)$	$O(r)$
(Badanidiyuru & Vondrák, 2014)	offline continuous	$1-\frac{1}{e}-\epsilon$	$O\left(\frac{nr}{\epsilon^4}\log^2\left(\frac{nr}{\epsilon}\right)\right)$	$O(n)$
(Feldman et al., 2021)	streaming continuous	$\approx 0.3178-\epsilon$	$O\left(\frac{nr\log^2 r}{\epsilon^2}\right)$	$O\left(r\log^{O(1)} n\right)$

$$- f(X_1,\ldots,X_i,\ldots,X_k)$$

Note that f being monotone is equivalent to $\Delta_{e,i} f(\mathbf{X}) \geq 0$ for all $\mathbf{X} \in (k+1)^V$, $i\in[k]$, and $e \notin \operatorname{supp}(\mathbf{X})$.

The function f is **pairwise monotone** if

$$\Delta_{e,i} f(\mathbf{X}) + \Delta_{e,j} f(\mathbf{X}) \geq 0$$

for all $\mathbf{X} \in (k+1)^V$, $e \notin \operatorname{supp}(\mathbf{X})$, and $i,j\in[k]$ such that $i \neq j$.

The function f is **orthant submodular** if

$$\Delta_{e,i} f(\mathbf{X}) \geq \Delta_{e,i} f(\mathbf{Y})$$

for all $\mathbf{X},\mathbf{Y} \in (k+1)^V$ such that $\mathbf{X} \preceq \mathbf{Y}$, $e \notin \operatorname{supp}(\mathbf{Y})$, and $i \in [k]$.

As shown in (Ward & Živnỳ, 2016), a function f is k-submodular if and only if f is orthant submodular and pairwise monotone.

2.2. Problem Definition

In the k-submodular maximization problem, we are given an objective function $f\colon (k+1)^V \rightarrow \mathbb{R}_+$ that is monotone, k-submodular, and non-negative, and positive integers $B_1,\ldots,B_k$. The goal is to find a solution $\mathbf{S} \in (k+1)^V$ that maximizes f subject to the constraint that $|S_i| \leq B_i$ for all $i\in[k]$. We let $\mathbf{S}^* \in \arg\max\{f(\mathbf{S})\colon |S_i| \leq B_i \ \forall i \in [k]\}$ denote an optimal solution to the problem. We assume without loss of generality that $f(\emptyset,\ldots,\emptyset)=0$.

We also consider the problem of maximizing a submodular set function subject to a partition matroid constraint: $\max\{f(S)\colon |S\cap P_i| \leq B_i \ \forall i\in[k]\}$, where $f\colon 2^V \rightarrow \mathbb{R}_+$ is a monotone submodular function,[1] and $P_1,\ldots,P_k$ is a partition of the ground set V. We let $S^* \in \arg\max\{f(S)\colon |S\cap P_i| \leq B_i \ \forall i\in[k]\}$ denote an optimal solution to the problem. We assume without loss of generality that $f(\emptyset)=0$.

We study the above problems in the streaming setting where the elements of the ground set V arrive one at a time, in an arbitrary (adversarial) order. We order the elements according to the stream arrival order, and we write $e \prec e'$ if and only if e arrives before e'.

The algorithms we propose in this work are also suitable for the online setting with free disposal (Feldman et al., 2009). In this setting, elements arrive online in an arbitrary order. The algorithm maintains a single feasible solution. When an element arrives, the algorithm either discards the element or it adds it to the solution, after possibly discarding another element in the current solution to make room for the new

[1] A set function f is submodular if it is 1-submodular, i.e., we have $f(X)+f(Y) \geq f(X\cap Y)+f(X\cup Y)$ for any two sets $X,Y\subseteq V$.

element.

2.3. Linear Programming Formulation

In this section, we introduce a linear programing formulation for the k-submodular maximization that will form the basis of our algorithm and its analysis. To this end, it is convenient to view each k-tuple $\mathbf{A} \in (k+1)^V$ as a partial labeling of V using labels $1, 2, \ldots, k$ as follows: element e receives label i if and only if $e \in A_i$.

We start by writing down an integer program for the problem. For each element $e \in V$ and each $i \in [k]$, we introduce a variable $x_{e,i} \in \{0,1\}$ that indicates whether element e is assigned label i (i.e., $x_{e,i} = 1$ if e is assigned label i and $x_{e,i} = 0$ otherwise). For each k-tuple $\mathbf{A} \in (k+1)^V$, we introduce a variable $y_{\mathbf{A}} \in \{0,1\}$ that indicates whether $\mathbf{A}$ is the selected k-tuple. The first set of constraints enforce that e receives label i if and only if $e \in A_i$ for the selected labeling. The second set of constraints enforce that we select exactly one k-tuple. The third set of constraints enforce that each element receives at most one label. The fourth set of constraints enforce the size constraints for the labels. By relaxing the integrality constraints $x_{e,i}, y_{\mathbf{A}} \in \{0,1\}$ to $x_{e,i}, y_{\mathbf{A}} \in [0,1]$, we obtain the LP relaxation for the problem and its dual shown in Figure 1.

Primal LP

$$\max \sum_{\mathbf{A} \in (k+1)^V} y_{\mathbf{A}} f(\mathbf{A})$$

$$\forall e \in V, i \in [k]\colon \sum_{\mathbf{A} \in (k+1)^V : \, e \in A_i} y_{\mathbf{A}} = x_{e,i}$$

$$\sum_{\mathbf{A} \in (k+1)^V} y_{\mathbf{A}} = 1$$

$$\forall e \in V\colon \sum_{i=1}^{k} x_{e,i} \le 1$$

$$\forall i \in [k]\colon \sum_{e \in V} x_{e,i} \le B_i$$

$$\forall e, i, \mathbf{A}\colon x_{e,i}, y_{\mathbf{A}} \ge 0$$

Dual LP

$$\min \beta + \sum_{e \in V} \gamma_e + \sum_{i=1}^{k} B_i \phi_i$$

$$\forall \mathbf{A} \in (k+1)^V : \sum_{i=1}^{k} \sum_{e \in A_i} \alpha_{e,i} + \beta \ge f(\mathbf{A})$$

$$\forall e \in V, i \in [k]\colon \gamma_e + \phi_i \ge \alpha_{e,i}$$

$$\forall e \in V, i \in [k]\colon \gamma_e, \phi_i \ge 0$$

Figure 1. Primal and dual LPs for k-submodular maximization.

Note that the primal LP is a valid relaxation[2] to the problem. More precisely, given any solution $\mathbf{S} = (S_1, \ldots, S_k) \in (k+1)^V$ satisfying $|S_i| \le B_i$ for all $i \in [k]$, we can construct a feasible solution to the primal LP with the same objective value by setting $y_{\mathbf{A}} = 1$ if and only if $\mathbf{A} = \mathbf{S}$ and $x_{e,i} = 1$ if and only if $e \in S_i$. Thus the optimum value of the primal LP is an upper bound on the value $f(\mathbf{S}^*)$ of an optimal solution $\mathbf{S}^*$ to the k-submodular maximization problem.

3. Monotone k-Submodular Maximization

3.1. Algorithm

In this section, we give our algorithm for maximizing a monotone k-submodular function subject to size constraints. The algorithm is shown in Algorithm 1. The algorithm follows a primal-dual approach based on the LPs shown in Figure 1, and it constructs a feasible integral solution $\mathbf{S} \in (k+1)^V$ to the primal LP and a feasible dual solution to the dual LP (the values γ_e and ϕ_i defined in the algorithm can be extended to a dual solution as shown in Lemma 3.3). In the analysis of the approximation guarantee, we will use the dual solution to upper bound the optimal value $f(\mathbf{S}^*)$

[2]One can show that the relaxation is exact, i.e., given a fractional solution x, y to the primal LP, one can round it to an integral solution $\mathbf{S} \in (k+1)^V$ of the same value. However, we will not use this fact in this paper.

Algorithm 1 Algorithm for monotone k-submodular maximization.

Parameters: C, D
$\mathbf{S} = (S_1, S_2, \ldots, S_k) \leftarrow (\emptyset, \emptyset, \ldots, \emptyset)$ // store each S_i in a deque
$\phi_i \leftarrow 0 \quad \forall i \in [k]$
for $t = 1, 2, \ldots, |V|$:
  let e be t-th element to arrive
  $i \leftarrow \arg\max_{j \in [k]} \{\Delta_{e,j} f(\mathbf{S}) - \phi_j\}$
  if $\Delta_{e,i} f(\mathbf{S}) - \phi_i \ge 0$:
    $\gamma_e \leftarrow \Delta_{e,i} f(\mathbf{S}) - \phi_i$ // defined for analysis purposes, does not need to be stored
    $\phi_i \leftarrow \left(1 + \frac{D}{B_i}\right) \phi_i + \frac{C}{B_i} \gamma_e = \phi_i + \frac{D}{B_i} \Delta_{e,i} f(\mathbf{S}) + \frac{C-D}{B_i} \gamma_e$
    if $|S_i| < B_i$:
      $S_i \leftarrow S_i \cup \{e\}$ // S_i.push_back(e)
    else:
      let e' be the earliest element of S_i // $e' = S_i$.front()
      $S_i \leftarrow (S_i \setminus \{e'\}) \cup \{e\}$ // S_i.pop_front(); S_i.push_back(e)
return $\mathbf{S}$

via weak duality. The dual values ϕ_i can be interpreted as threshold values for the marginal gains. Each threshold ϕ_i is dynamically adjusted by the algorithm based on the marginal gains and the budgets, and ϕ_i is increasing exponentially (the update rule ensures that ϕ_i increases by at least a factor of $1+\frac{D}{B_i}$ with each update). The growth rate is chosen so that, intuitively, the threshold grows by a constant factor every time the old solution is completely replaced. Furthermore, for larger budgets, the growing thresholds are able to mimic some aspect of a continuous algorithm without the computational cost.

The algorithm maintains the feasibility of the solution by removing the earliest element from the solution when the budget is exceeded. Intuitively, since the threshold increases over time, the elements meeting the earlier thresholds are also ones with smaller gains. However, this intuition does not always hold and thus an alternative practical choice is to remove the element with the lowest gain. Removing the earlier element leads to a simpler analysis, and it is the version we consider in this paper.

The following theorem states the theoretical guarantees for our algorithm. We give an outline of the analysis in this section; the omitted proofs can be found in Section A. We note that the space usage is dominated by the space needed to store the solution, and therefore the space is optimal. Additionally, the running time of the algorithm is optimal as well. To see this, consider the ad allocation problem discussed in the introduction. In this setting, the elements V are the ad impressions, and the labels $[k]$ correspond to k advertisers. Consider an instance of this problem where we have k impressions $e_{\sigma(1)}, \ldots, e_{\sigma(k)}$, where σ is an arbitrary permutation, and impression $e_{\sigma(i)}$ provides a large value v_i to the i-th advertiser and it provides value 0 to all of the other advertisers. Any algorithm that achieves a non-trivial approximation guarantee requires $O\left(k\left|V\right|\right)$ time to identify the mapping σ matching the impressions to the advertisers.

Theorem 3.1. *Consider Algorithm 1. Suppose we set $B = \min_{i\in[k]} B_i$, $D = B\left(2^{1/B}-1\right)$, and $C = 2D$. The algorithm uses $O\left(\sum_{i=1}^{k} B_i\right)$ space, $O\left(k\left|V\right|\right)$ function evaluations, and $O\left(\left|V\right|\right)$ additional time. Moreover, the algorithm returns a solution $\mathbf{S}$ satisfying*

$$\frac{f(\mathbf{S})}{f(\mathbf{S}^*)} \geq \frac{1}{2\left(1+B\left(2^{1/B}-1\right)\right)}$$

where $f(\mathbf{S}^*)$ is the value of an optimal solution. The approximation guarantee is an increasing function of B; it is $\frac{1}{4}$ when $B=1$ and it approaches $\frac{1}{2(1+\ln 2)} \approx 0.2953$ as $B\to\infty$.

The algorithm is inspired by the algorithm proposed in (Levin & Wajc, 2021) for the submodular matching problem. In the submodular matching problem, the edges of a bipartite graph arrive in a stream one edge at a time. The goal is to select a matching M of maximum value $f(M)$, where $f\colon 2^E \to \mathbb{R}_+$ is a submodular function defined on the set of edges of the graph. Our algorithm can be viewed as constructing a matching in the bipartite graph with the elements V on the left and the labels $[k]$ on the right, in the setting where the vertices on the left arrive one at a time. However, the algorithm and analysis of (Levin & Wajc, 2021) do not apply to our setting due to the inherent differences between k-submodular and submodular functions. The main similarities between our algorithm and theirs are the use of the primal-dual approach and the general form of the update rule for ϕ_i. The algorithm of (Levin & Wajc, 2021) constructs a set of elements that is not a feasible matching, and this set is used to construct a feasible solution at the end of the stream. The dual values are updated based on the marginal gains of this larger (infeasible) set. This set is larger than a feasible solution by a factor of $\Theta\left(\log\Delta\right)$, where Δ is the ratio of the largest and smallest marginal gains. In contrast, our algorithm maintains a feasible solution at all times, and our time and space usage do not have additional logarithmic factors.

3.2. Space and Time Analysis

The algorithm stores the $O\left(\sum_{i=1}^{k} B_i\right)$ elements in the solution $\mathbf{S}$ and the k dual variables $\{\phi_i\colon i\in[k]\}$. Thus the total space usage is $O\left(\sum_{i=1}^{k} B_i\right)$. In each iteration t, the algorithm evaluates the function $O(k)$ times in order to compute the marginal gains $\Delta_{e,i} f(\mathbf{S})$ for each $i\in[k]$, and it performs $O(1)$ additional operations. We maintain each set S_i in a deque, and thus removing the earliest element of S_i and adding an element to the back of S_i can be performed in $O(1)$ time. Thus the algorithm performs $O\left(k\left|V\right|\right)$ function evaluations and $O\left(\left|V\right|\right)$ additional time.

3.3. Analysis of the Approximation Guarantee

For analysis purposes, we annotate the main quantities in the algorithm using the superscript (t): $\phi_i^{(t)}, \mathbf{S}^{(t)}$ denote the respective quantities at the *end* of iteration t; $e^{(t)}$ denotes the element that arrives in iteration t; $i^{(t)}$ denotes the label selected in iteration t. We also let $X_i^{(t)} = S_i^{(1)}\cup\cdots\cup S_i^{(t)}$ denote the set of elements that were added to S_i in the first t iterations, and $\mathbf{X}^{(t)} = \left(X_1^{(t)},\ldots,X_k^{(t)}\right)$.

Our main analysis approach is to relate the marginal gains $\sum_{e^{(t)}\in\mathrm{supp}(\mathbf{X}^{(n)})} \Delta_{e^{(t)},i^{(t)}} f(\mathbf{S}^{(t-1)})$, the final solution value $f(\mathbf{S}^{(n)})$, and the optimum solution value $f(\mathbf{S}^*)$ to suitable linear combinations of the dual values $\left\{\gamma_e\colon e\in\mathrm{supp}(\mathbf{X}^{(n)})\right\}$ defined in the algorithm. We then derive our approximation guarantee by analyzing the coefficients of each γ_e in the respective linear combinations.

We start by deriving the linear combinations for $\sum_{e^{(t)} \in \mathrm{supp}(\mathbf{X}^{(n)})} \Delta_{e^{(t)}, i^{(t)}} f(\mathbf{S}^{(t-1)})$ and $f(\mathbf{S}^{(n)})$.

Lemma 3.2. *We have*

$$\sum_{e^{(t)} \in \mathrm{supp}(\mathbf{X}^{(n)})} \Delta_{e^{(t)}, i^{(t)}} f(\mathbf{S}^{(t-1)}) = \sum_{e^{(t)} \in \mathrm{supp}(\mathbf{X}^{(n)})} c_t \gamma_{e^{(t)}}$$

and

$$f(\mathbf{S}^{(n)}) \geq \sum_{e^{(t)} \in \mathrm{supp}(\mathbf{X}^{(n)})} \widetilde{c}_t \gamma_{e^{(t)}}$$

where

$$s_t = \left| \left\{ e \in X^{(n)}_{i^{(t)}} \colon e \succ e^{(t)} \right\} \right|$$

$$c_t = \frac{C}{D} \left(1 + \frac{D}{B_{i^{(t)}}} \right)^{s_t} - \frac{C}{D} + 1$$

$$\widetilde{c}_t = \begin{cases} \frac{C}{D} \left(1 + \frac{D}{B_{i^{(t)}}} \right)^{s_t} \left(1 - \left(1 + \frac{D}{B_{i^{(t)}}} \right)^{-B_{i^{(t)}}} \right) & \\ \qquad \textit{if } s_t \geq B_{i^{(t)}} & \\ c_t \qquad \textit{otherwise} & \end{cases}$$

Next, we upper bound $f(\mathbf{S}^*)$ using a linear combination of the γ_e values. To this end, we define a feasible dual solution based on the values $\left\{ \phi^{(n)}_i \colon i \in [k] \right\}$ and $\left\{ \gamma_e \colon e \in \mathrm{supp}(\mathbf{X}^{(n)}) \right\}$ defined in the algorithm.

Lemma 3.3. *For each $e \in \mathrm{supp}(\mathbf{X}^{(n)})$, we let γ_e be the dual value set by the algorithm, and we let $\gamma_e = 0$ for all $e \notin \mathrm{supp}(\mathbf{X}^{(n)})$. For each $i \in [k]$, we let ϕ_i be equal to the final value $\phi^{(n)}_i$. We set the remaining dual variables as follows:*

$$\beta = f(\mathbf{X}^{(n)})$$

$$\alpha_{e,i} = \gamma_e + \phi^{(n)}_i \qquad \forall e \in V, i \in [k]$$

The above setting is a feasible solution to the dual LP shown in Figure 1.

Proof. (*Sketch*) We have $\gamma_e \geq 0$ and $\phi^{(n)}_i \geq 0$ for all $e \in V$ and $i \in [k]$. The definition of $\alpha_{e,i}$ ensures that the dual constraint $\gamma_e + \phi^{(n)}_i \geq \alpha_{e,i}$ is satisfied with equality. Therefore it only remains to verify that the first set of constraints are satisfied, i.e., for any $\mathbf{A} \in (k+1)^V$, we need to verify that $f(\mathbf{A}) - f(\mathbf{X}^{(n)}) \leq \sum_{i=1}^k \sum_{e \in A_i} \alpha_{e,i}$. Consider a k-set $\mathbf{A} \in (k+1)^V$, and let $\mathbf{X} = \mathbf{X}^{(n)}$. We start by augmenting $\mathbf{A}$ so that it contains every element in the support of $\mathbf{X}$, and let $\widetilde{\mathbf{A}}$ denote the resulting k-set (elements are added to the same part as in $\mathbf{X}$). Since f is monotone, it follows that $f(\mathbf{A}) \leq f(\widetilde{\mathbf{A}})$ and it suffices to show that $f(\widetilde{\mathbf{A}}) - f(\mathbf{X}) \leq \sum_{i=1}^k \sum_{e \in A_i} \alpha_{e,i}$. Our definition of $\widetilde{\mathbf{A}}$ ensures that $\mathrm{supp}(\mathbf{X}) \subseteq \mathrm{supp}(\widetilde{\mathbf{A}})$. However, elements may be assigned to different parts by $\mathbf{X}$ and $\widetilde{\mathbf{A}}$, which prevents us from directly comparing $f(\widetilde{\mathbf{A}})$ to $f(\mathbf{X})$. Our approach is to modify $\widetilde{\mathbf{A}}$ so that it agrees with $\mathbf{X}$ on the placement of all the elements in $\mathbf{X}$. By exploiting the way in which the algorithm selects the part i to which to assign an element, we can upper bound the loss in function value due to these swaps to the dual values $\alpha_{e,i}$. Additionally, we relate the contribution to $f(\widetilde{\mathbf{A}})$ of the elements in $\mathrm{supp}(\widetilde{\mathbf{A}}) \setminus \mathrm{supp}(\mathbf{X})$ to the dual values $\alpha_{e,i}$. We now sketch the formal argument.

Starting with $\widetilde{\mathbf{A}}^{(0)} = \widetilde{\mathbf{A}}$, we iteratively define a sequence of k-sets $\widetilde{\mathbf{A}}^{(t)}$ such that $\mathbf{X}$ and $\widetilde{\mathbf{A}}^{(t)}$ agree on the first t elements. Consider the element $e^{(t)}$ that arrives in iteration t. Since $\mathrm{supp}(\mathbf{X}) \subseteq \mathrm{supp}(\widetilde{\mathbf{A}})$, we have the following cases:

Suppose $e^{(t)}$ is in the support of $\mathbf{X}$. Since all elements in $\mathbf{X}$ are also in $\widetilde{\mathbf{A}}$, $e^{(t)}$ is also in the support of $\widetilde{\mathbf{A}}$, but possibly in a different part. Suppose $e^{(t)} \in \widetilde{A}_j$, and recall that $e^{(t)} \in X_{i^{(t)}}$. To make $\widetilde{\mathbf{A}}^{(t)}$ and $\mathbf{X}$ agree on $e^{(t)}$, we obtain $\widetilde{\mathbf{A}}^{(t)}$ by taking $\widetilde{\mathbf{A}}^{(t-1)}$ and moving $e^{(t)}$ from part j to part $i^{(t)}$. Using the monotonicity and orthant submodularity of f, we can show that the decrease in function value $f(\widetilde{\mathbf{A}}^{(t-1)}) - f(\widetilde{\mathbf{A}}^{(t)})$ is at most the marginal gain $\Delta_{e^{(t)}, j} f(\mathbf{S}^{(t-1)})$. Using the choice of $i^{(t)}$ and the definition of the dual values, we can show that the marginal gain $\Delta_{e^{(t)}, j} f(\mathbf{S}^{(t-1)})$ is at most $\alpha_{e^{(t)}, j}$. Thus we have $f(\widetilde{\mathbf{A}}^{(t-1)}) - f(\widetilde{\mathbf{A}}^{(t)}) \leq \alpha_{e^{(t)}, j}$, as needed.

Suppose $e^{(t)}$ is in the support of $\widetilde{\mathbf{A}}$ but not $\mathbf{X}$. To make $\widetilde{\mathbf{A}}^{(t)}$ and $\mathbf{X}$ agree on $e^{(t)}$, we obtain $\widetilde{\mathbf{A}}^{(t)}$ by taking $\widetilde{\mathbf{A}}^{(t-1)}$ and removing $e^{(t)}$ from it. Suppose $e^{(t)} \in \widetilde{A}_j$; since $e^{(t)} \notin \mathbf{X}$, we have $e^{(t)} \in A_j$. Using the orthant submodularity of f, we can show that the decrease in function value $f(\widetilde{\mathbf{A}}^{(t-1)}) - f(\widetilde{\mathbf{A}}^{(t)})$ is at most the marginal gain $\Delta_{e^{(t)}, j} f(\mathbf{S}^{(t-1)})$. Since $e^{(t)} \notin \mathrm{supp}(\mathbf{X})$, $e^{(t)}$ did not meet the condition required by the algorithm. Therefore we have $\Delta_{e^{(t)}, j} f(\mathbf{S}^{(t-1)}) \leq \phi^{(t-1)}_j$ and $\gamma_{e^{(t)}} = 0$, and hence $\Delta_{e^{(t)}, j} f(\mathbf{S}^{(t-1)}) \leq \alpha_{e^{(t)}, j}$. Thus we have $f(\widetilde{\mathbf{A}}^{(t-1)}) - f(\widetilde{\mathbf{A}}^{(t)}) \leq \alpha_{e^{(t)}, j}$, as needed.

Suppose $e^{(t)}$ is not in the support of $\widetilde{\mathbf{A}}$. Thus $e^{(t)}$ is not in the suppot of $\mathbf{X}$ either, and we can set $\widetilde{\mathbf{A}}^{(t)} = \widetilde{\mathbf{A}}^{(t-1)}$. We have $f(\widetilde{\mathbf{A}}^{(t-1)}) - f(\widetilde{\mathbf{A}}^{(t)}) = 0$, as needed. ☐

By combining the above lemmas with weak duality, we obtain the following upper bound on $f(\mathbf{S}^*)$.

Lemma 3.4. *We have*

$$f(\mathbf{S}^*) \leq \sum_{e^{(t)} \in \mathrm{supp}(\mathbf{X}^{(n)})} \widehat{c}_t \gamma_{e^{(t)}}$$

where

$$\widehat{c}_t = (D+1)\, c_t + C - D + 1$$

and c_t is the coefficient from Lemma 3.2.

Proof. (*Sketch*) Lemma 3.3 and weak duality gives us the following upper bound on the optimum function value: $f(\mathbf{S}^*) \leq f(\mathbf{X}^{(n)}) + \sum_{e^{(t)} \in \mathrm{supp}(\mathbf{X}^{(n)})} \gamma_{e^{(t)}} + \sum_{i=1}^{k} B_i \phi_i^{(n)}$. Using that $\mathbf{S}^{(t)} \preceq \mathbf{X}^{(t)}$ and the orthant submodularity of f, we can show that $f(\mathbf{X}^{(n)}) \leq \sum_{e^{(t)} \in \mathrm{supp}(\mathbf{X}^{(n)})} \Delta_{e^{(t)}, i^{(t)}} f(\mathbf{S}^{(t-1)})$. Using the update rule for ϕ_i, we can show that $\sum_{i=1}^{k} B_i \phi_i^{(n)} = \sum_{e^{(t)} \in \mathrm{supp}(\mathbf{X}^{(n)})} \left(D\Delta_{e^{(t)}, i^{(t)}} f(\mathbf{S}^{(t-1)}) + (C - D)\, \gamma_{e^{(t)}} \right)$. Putting everything together and using Lemma 3.2 gives the result. $\square$

We now set the parameters C and D and derive the approximation guarantee.

Lemma 3.5. *Let $B = \min_{i \in [k]} B_i$ be the smallest budget among the k labels. Let $C = 2D$. We have:*

$$\frac{f(\mathbf{S})}{f(\mathbf{S}^*)} \geq \min_{t:\, e^{(t)} \in \mathbf{X}^{(n)}} \frac{\widetilde{c}_t}{\widehat{c}_t} \geq \frac{1}{D+1} \min \left\{ \frac{1}{2}, 1 - \left(1 + \frac{D}{B}\right)^{-B} \right\}$$

where $\widetilde{c}_t$ and $\widehat{c}_t$ are the coefficients from Lemmas 3.2 and 3.4.

Setting $D = B\left(2^{1/B} - 1\right)$ gives

$$\frac{f(\mathbf{S})}{f(\mathbf{S}^*)} \geq \frac{1}{2\left(1 + B\left(2^{1/B} - 1\right)\right)}$$

4. Monotone Submodular Maximization with a Partition Matroid Constraint

In this section, we adapt Algorithm 1 and its analysis to the problem of maximizing a monotone submodular function subject to a partition matroid constraint. The algorithm is shown in Algorithm 2. We will show that the approximation of the algorithm improves in this more restricted setting.

The following theorem states the theoretical guarantees for the algorithm. We prove the theorem in Section B. The analysis is similar to that of Algorithm 1, with the main difference being that we show a stronger upper bound on the optimal solution value $f(S^*)$ (Lemma B.2).

Theorem 4.1. *Consider Algorithm 2. Suppose we set $C = D$. The algorithm uses $O\left(\sum_{i=1}^{k} B_i\right)$ space, $O(|V|)$ function evaluations, and $O(|V|)$ additional time. Moreover, the algorithm returns a solution S satisfying*

$$\frac{f(S)}{f(S^*)} \geq \frac{1}{D+1}\left(1 - \left(1 + \frac{D}{B}\right)^{-B}\right)$$

where $f(S^*)$ is the value of an optimal solution, and $B = \min_{i \in [k]} B_i$ is the minimum budget.

Algorithm 2 Algorithm for monotone submodular maximization with a partition matroid constraint. We let $\Delta_e f(S) := f(S \cup \{e\}) - f(S)$ denote the marginal gain of e on top of S.

Parameters: C, D
$S_i \leftarrow \emptyset \quad \forall i \in [k]$ // $S_i = S \cap P_i$, store S_i in a deque
$\phi_i \leftarrow 0 \quad \forall i \in [k]$
for $t = 1, 2, \ldots, |V|$:
  let e be t-th element to arrive
  let i be the index of the part containing e, i.e., $e \in P_i$
  let $S = S_1 \cup \cdots \cup S_k$
  if $\Delta_e f(S) - \phi_i \geq 0$:
    $\gamma_e \leftarrow \Delta_e f(S) - \phi_i$ // defined for analysis purposes, does not need to be stored
    $\phi_i \leftarrow \left(1 + \frac{D}{B_i}\right)\phi_i + \frac{C}{B_i}\gamma_e = \phi_i + \frac{D}{B_i}\Delta_e f(S) + \frac{C-D}{B_i}\gamma_e$
    if $|S_i| < B_i$:
      $S_i \leftarrow S_i \cup \{e\}$ // S_i.push_back(e)
    else:
      let e' be the earliest element of S_i // $e' = S_i$.front()
      $S_i \leftarrow (S_i \setminus \{e'\}) \cup \{e\}$ // S_i.pop_front(); S_i.push_back(e)
return $S = S_1 \cup \cdots \cup S_k$

Setting $D = 1$ gives

$$\frac{f(S)}{f(S^*)} \geq \frac{1}{2}\left(1 - \left(1 + \frac{1}{B}\right)^{-B}\right)$$

Remark 4.2. The above approximation for $C = D = 1$ is least $\frac{1}{4}$ for every $B \geq 1$ and it approaches $\frac{1}{2}\left(1 - \frac{1}{e}\right) \approx 0.316$ as $B \to \infty$. For sufficiently large B, we have $1 - \left(1 + \frac{D}{B}\right)^{-B} \approx 1 - \exp(-D)$. In this case, we can set D to the value satisfying the equation $e^x = x + 2$, which is $x \approx 1.146$, and obtain an approximation of ≈ 0.3178.

5. Experimental Evaluation

In this section, we experimentally evaluate our algorithm for k-submodular maximization (Algorithm 1). Following previous work (Ohsaka & Yoshida, 2015; Nguyen & Thai, 2020), we evaluate the algorithms on instances of influence maximization with k topics and sensor placement with k measurements. We follow the experimental setup of these prior works.

Influence maximization with k topics: We consider the k-topic independent cascade model introduced in (Ohsaka & Yoshida, 2015). In this model, we have a social network represented as a directed graph $G = (V, E)$ on a set V of users. Each edge $(u, v) \in E$ is associated with k probabilities $\left\{p_{u,v}^{(i)} : i \in [k]\right\}$, where $p_{u,v}^{(i)}$ represents the strength of the influence of user u on user v for topic $i \in [k]$.

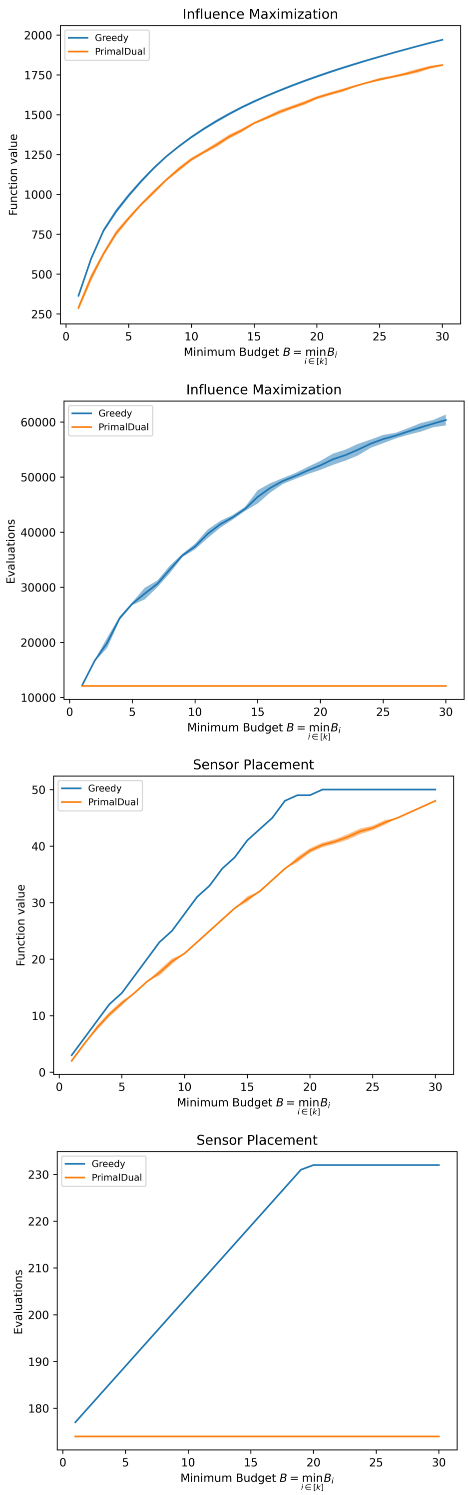


Figure 2. We report the mean and standard deviation over 5 runs. Greedy is the offline Greedy algorithm of (Ohsaka & Yoshida, 2015) implemented using lazy evaluations. PrimalDual is our Algorithm 1.

Given seed sets $(S_1, S_2, \ldots, S_k)$, where S_i is the seed set for topic i, nodes are activated independently for each topic according to the independent cascade model. More precisely, the activation process for topic i is the following. Each edge (u, v) of the graph is realized with probability $p^{(i)}_{u,v}$, independently of other edges. The set of activated nodes for topic i are the nodes reachable from the seed set S_i in this realized graph. Nodes are activated independently for each topic. The objective function is the expected number of nodes activated in at least one topic. More precisely, if we let $A_i(S_i)$ be the random variable equal to the set of nodes activated for topic i starting with seed set S_i, we have $f(S_1, \ldots, S_k) = \mathbb{E}\left[\left|\bigcup_{i \in [k]} A_i(S_i)\right|\right]$. It was shown in (Ohsaka & Yoshida, 2015) that f is monotone k-submodular.

We used the Facebook dataset from the SNAP database (Leskovec & Krevl, 2014). The graph has 4,039 nodes and 88,234 undirected edges. We replaced each undirected edge uv by two directed edges, (u, v) and (v, u). For each directed edge (u, v), we set the k probabilities $\left\{p^{(i)}_{u,v} : i \in [k]\right\}$ by randomly permuting the values $\left\{\frac{2i}{k \cdot d_v} : i \in [k]\right\}$, where d_v is the in-degree of v. We approximated f using the random sampling procedure of (Borgs et al., 2014). We set $k = 3$. We obtained similar experimental results for larger values of k, and we report the results for $k = 10$ in Section C. We set the same budget constraint B for each topic i, i.e., we set $B_i = B$ for all $i \in [k]$.

Sensor placement with k measurements: We consider the sensor placement problem introduced in (Ohsaka & Yoshida, 2015). There are k types of sensors and a set V of n sensor locations. We want to place at most one sensor in each possible location and at most B_i sensors of type i across all locations. We can represent the information collected by a sensor of type i when placed at location $e \in V$ using a random variable $X^{(i)}_e$. Let $\Omega = \left\{X^{(i)}_e : e \in V, i \in [k]\right\}$. The entropy of a subset $\mathcal{X} \subseteq \Omega$ is defined as $H(\mathcal{X}) = -\sum_{x \in \mathrm{dom}(\mathcal{X})} \Pr[x] \log \Pr[x]$. The goal is to select sensor locations $(S_1, \ldots, S_k)$, where $S_i \subseteq V$ is the set of locations of the sensors of type i, in order to maximize the entropy $H\left(\bigcup\left\{X^{(i)}_e : i \in [k], e \in S_i\right\}\right)$ of the corresponding random variables. It was shown in (Ohsaka & Yoshida, 2015) that f is monotone k-submodular.

We used the Intel Lab dataset (Bodik et al., 2004) which contains approximately 2.3 million readings from 58 sensors deployed in the Intel Berkeley research lab. As in (Ohsaka & Yoshida, 2015; Nguyen & Thai, 2020), we extracted temperature, humidity, and light values from each reading and discretized them into bins of 2 degrees Celsius each, 5 points each and 100 luxes each, respectively. A type of measurement such as temperature at a particular location has a certain amount of uncertainty, measured by the entropy of its empirical distribution in the dataset. We treat the temperature measurements at different locations as independent and $H(\mathcal{X})$ is the sum of the entropy of all sensors in $\mathcal{X}$. The different measurement types are also treated as independent. We have $k = 3$ types of measurements, and we set a budget constraint B_i for each measurement $i \in [k]$. We set the same budget constraint B for each measurement, i.e., we set $B_i = B$ for all $i \in [k]$.

Algorithms: We compare our algorithm with the state of the art Greedy algorithm proposed in (Ohsaka & Yoshida, 2015), which is an offline algorithm that makes multiple passes over the data. We implemented the Greedy algorithm using lazy evaluations. The work (Ohsaka & Yoshida, 2015) also proposed a stochastic variant of the Greedy algorithm. In our experiments, the stochastic Greedy with lazy evaluations performed worse than the deterministic Greedy with lazy evaluation both in terms of function value and number of evaluations. This is due to the fact that the Greedy algorithm allows for a more efficient implementation using lazy evaluations. For this reason, we omitted the stochastic Greedy algorithm from the plots shown in Figure 2. In all experiments, we set the parameters C and D of Algorithm 1 as follows: $D = B\left(2^{1/B} - 1\right)$, where $B = \min_{i\in[k]} B_i$ is the minimum budget, and $C = 0.5D$.

Results: The results are shown in Figure 2. We report the average and standard deviation over 5 runs of the algorithms.

Acknowledgments

AE was supported in part by NSF CAREER grant CCF-1750333, NSF grant III-1908510, and an Alfred P. Sloan Research Fellowship. HN was supported in part by NSF CAREER grant CCF-1750716 and NSF grant CCF-1909314.

References

Badanidiyuru, A. and Vondrák, J. Fast algorithms for maximizing submodular functions. In *ACM-SIAM symposium on Discrete algorithms (SODA)*, pp. 1497–1514. SIAM, 2014.

Bodik, P., Hong, W., Guestrin, C., Madden, S., Paskin, M., and Thibaux, R. Intel Lab Data. `http://db.csail.mit.edu/labdata/labdata.html`, 2004.

Borgs, C., Brautbar, M., Chayes, J., and Lucier, B. Maximizing social influence in nearly optimal time. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pp. 946–957. SIAM, 2014.

Chakrabarti, A. and Kale, S. Submodular maximization meets streaming: Matchings, matroids, and more. *Mathematical Programming*, 154(1):225–247, 2015.

Chekuri, C., Gupta, S., and Quanrud, K. Streaming algorithms for submodular function maximization. In *International Colloquium on Automata, Languages, and Programming*, pp. 318–330. Springer, 2015.

Feldman, J., Korula, N., Mirrokni, V. S., Muthukrishnan, S., and Pál, M. Online ad assignment with free disposal. In *WINE*, volume 5929 of *Lecture Notes in Computer Science*, pp. 374–385. Springer, 2009.

Feldman, M., Norouzi-Fard, A., Svensson, O., and Zenklusen, R. Streaming submodular maximization with matroid and matching constraints. *arXiv preprint arXiv:2107.07183*, 2021.

Fisher, M. L., Nemhauser, G. L., and Wolsey, L. A. An analysis of approximations for maximizing submodular set functions-ii. In *Polyhedral combinatorics*, pp. 73–87. Springer, 1978.

Iwata, S., Tanigawa, S.-i., and Yoshida, Y. Improved approximation algorithms for k-submodular function maximization. In *ACM-SIAM symposium on Discrete algorithms*, pp. 404–413. SIAM, 2016.

Leskovec, J. and Krevl, A. SNAP Datasets: Stanford large network dataset collection. `http://snap.stanford.edu/data`, 2014.

Levin, R. and Wajc, D. Streaming submodular matching meets the primal-dual method. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 1914–1933. SIAM, 2021.

Nguyen, L. and Thai, M. T. Streaming k-submodular maximization under noise subject to size constraint. In *International Conference on Machine Learning (ICML)*, pp. 7338–7347. PMLR, 2020.

Ohsaka, N. and Yoshida, Y. Monotone k-submodular function maximization with size constraints. In *Neural Information Processing Systems NeurIPS*, pp. 694–702, 2015.

Oshima, H. Derandomization for k-submodular maximization. In *International Workshop on Combinatorial Algorithms*, pp. 88–99. Springer, 2017.

Pham, C. V., Vu, Q. C., Ha, D. K. T., and Nguyen, T. T. Streaming algorithms for budgeted k-submodular maximization problem, 2021.

Santiago, R. and Shepherd, F. B. Multivariate submodular optimization. In *International Conference on Machine Learning*, pp. 5599–5609. PMLR, 2019.

Soma, T. No-regret algorithms for online k-submodular maximization. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1205–1214. PMLR, 2019.

Ward, J. and Živný, S. Maximizing k-submodular functions and beyond. *ACM Transactions on Algorithms (TALG)*, 12(4):1–26, 2016.

A. Analysis of Algorithm 1

A.1. Proof of Lemma 3.2

Proof. (Lemma 3.2) For each $e^{(t)} \in \operatorname{supp}(\mathbf{X}^{(n)})$, by rearranging the definition of $\gamma_{e^{(t)}}$, we obtain

$$\Delta_{e^{(t)},i^{(t)}} f(\mathbf{S}^{(t-1)}) = \gamma_{e^{(t)}} + \phi_{i^{(t)}}^{(t-1)} \tag{1}$$

By unrolling the update rule for ϕ_i, we obtain

$$\phi_i^{(t)} = \frac{C}{B_i} \sum_{e^{(\tau)} \in X_i^{(t)}} \left(1 + \frac{D}{B_i}\right)^{\left|\left\{e \in X_i^{(t)} \colon e \succ e^{(\tau)}\right\}\right|} \gamma_{e^{(\tau)}} \tag{2}$$

We first consider $\sum_{e^{(t)} \in \operatorname{supp}(\mathbf{X}^{(n)})} \Delta_{e^{(t)},i^{(t)}} f(\mathbf{S}^{(t-1)})$. We have

$$\begin{aligned}
&\sum_{e^{(t)} \in \operatorname{supp}(\mathbf{X}^{(n)})} \Delta_{e^{(t)},i^{(t)}} f(\mathbf{S}^{(t-1)}) \\
&\overset{(1)}{=} \sum_{e^{(t)} \in \operatorname{supp}(\mathbf{X}^{(n)})} \left(\gamma_{e^{(t)}} + \phi_{i^{(t)}}^{(t-1)}\right) \\
&\overset{(2)}{=} \sum_{e^{(t)} \in \operatorname{supp}(\mathbf{X}^{(n)})} \left(\gamma_{e^{(t)}} + \frac{C}{B_{i^{(t)}}} \sum_{e^{(\tau)} \in X_{i^{(t)}}^{(t-1)}} \left(1 + \frac{D}{B_{i^{(t)}}}\right)^{\left|\left\{e \in X_{i^{(t)}}^{(t-1)} \colon e \succ e^{(\tau)}\right\}\right|} \gamma_{e^{(\tau)}}\right) \\
&= \sum_{i=1}^{k} \sum_{e^{(t)} \in X_i^{(n)}} \left(\gamma_{e^{(t)}} + \frac{C}{B_i} \sum_{e^{(\tau)} \in X_i^{(t-1)}} \left(1 + \frac{D}{B_i}\right)^{\left|\left\{e \in X_i^{(t-1)} \colon e \succ e^{(\tau)}\right\}\right|} \gamma_{e^{(\tau)}}\right) \\
&= \sum_{i=1}^{k} \left(\sum_{e^{(t)} \in X_i^{(n)}} \gamma_{e^{(t)}} + \frac{C}{B_i} \sum_{e^{(\tau)} \in X_i^{(n)}} \gamma_{e^{(\tau)}} \sum_{e^{(t)} \in X_i^{(n)} \colon e^{(t)} \succ e^{(\tau)}} \left(1 + \frac{D}{B_i}\right)^{\left|\left\{e \in X_i^{(t-1)} \colon e \succ e^{(\tau)}\right\}\right|}\right)
\end{aligned}$$

where in the last equality we performed an exchange of summation.

Fix $e^{(\tau)} \in X_i^{(n)}$. Let $e^{(t_1)}, e^{(t_2)}, \ldots, e^{(t_m)}$ be the elements of $X_i^{(n)}$ that arrived after $e^{(\tau)}$, in the order in which they arrived. More precisely, we have $\left\{e^{(t)} \in X_i^{(n)} \colon t > \tau\right\} = \left\{e^{(t_1)}, e^{(t_2)}, \ldots, e^{(t_m)}\right\}$ and $t_1 < t_2 < \cdots < t_m$. Let $t_0 = \tau$. We have

$$\begin{aligned}
&\sum_{e^{(t)} \in X_i^{(n)} \colon e^{(t)} \succ e^{(\tau)}} \left(1 + \frac{D}{B_i}\right)^{\left|\left\{e \in X_i^{(t-1)} \colon e \succ e^{(\tau)}\right\}\right|} \\
&= \sum_{j=1}^{m} \left(1 + \frac{D}{B_i}\right)^{\left|\left\{e \in X_i^{(t_j - 1)} \colon e \succ e^{(\tau)}\right\}\right|} \\
&= \sum_{j=1}^{m} \left(1 + \frac{D}{B_i}\right)^{\left|\left\{e \in X_i^{(t_{j-1})} \colon e \succ e^{(\tau)}\right\}\right|} \\
&= \sum_{j=1}^{m} \left(1 + \frac{D}{B_i}\right)^{j-1} \\
&= \frac{B_i}{D} \left(\left(1 + \frac{D}{B_i}\right)^{m} - 1\right) \\
&= \frac{B_i}{D} \left(\left(1 + \frac{D}{B_i}\right)^{\left|\left\{e \in X_i^{(n)} \colon e \succ e^{(\tau)}\right\}\right|} - 1\right)
\end{aligned}$$

In the second equality, we used the fact that no elements were added to S_i in iteration $t \in (t_{j-1}, t_j)$, and thus $X_i^{(t_j-1)} = X_i^{(t_{j-1})}$.

Putting everything together, we obtain

$$\begin{aligned}
&\sum_{e^{(t)} \in \mathrm{supp}(\mathbf{X}^{(n)})} \Delta_{e^{(t)}, i^{(t)}} f(\mathbf{S}^{(t-1)}) \\
&= \sum_{i=1}^{k} \sum_{e^{(t)} \in X_i^{(n)}} \left(\frac{C}{D} \left(1 + \frac{D}{B_i}\right)^{\left|\left\{e \in X_i^{(n)} \colon e \succ e^{(t)}\right\}\right|} - \frac{C}{D} + 1 \right) \gamma_{e^{(t)}} \\
&= \sum_{e^{(t)} \in \mathrm{supp}(\mathbf{X}^{(n)})} \left(\frac{C}{D} \left(1 + \frac{D}{B_{i^{(t)}}}\right)^{\left|\left\{e \in X_{i^{(t)}}^{(n)} \colon e \succ e^{(t)}\right\}\right|} - \frac{C}{D} + 1 \right) \gamma_{e^{(t)}}
\end{aligned}$$

Next, we consider $f(\mathbf{S}^{(n)})$. Let

$$\widetilde{\mathbf{S}}^{(t)} := \left(S_1^{(n)} \cap \left\{e^{(1)}, \ldots, e^{(t)}\right\}, \ldots, S_k^{(n)} \cap \left\{e^{(1)}, \ldots, e^{(t)}\right\} \right)$$

Note that we have $\widetilde{\mathbf{S}}^{(t)} \preceq \mathbf{S}^{(t)}$ for all t. We have

$$\begin{aligned}
f(\mathbf{S}^{(n)}) &= f(\widetilde{\mathbf{S}}^{(n)}) - f(\widetilde{\mathbf{S}}^{(0)}) \\
&= \sum_{e^{(t)} \in \mathrm{supp}(\mathbf{S}^{(n)})} \Delta_{e^{(t)}, i^{(t)}} f(\widetilde{\mathbf{S}}^{(t-1)}) \\
&\overset{(\widetilde{\mathbf{S}}^{(t-1)} \preceq \mathbf{S}^{(t-1)})}{\geq} \sum_{e^{(t)} \in \mathrm{supp}(\mathbf{S}^{(n)})} \Delta_{e^{(t)}, i^{(t)}} f(\mathbf{S}^{(t-1)}) \\
&= \sum_{i=1}^{k} \sum_{e^{(t)} \in S_i^{(n)}} \Delta_{e^{(t)}, i} f(\mathbf{S}^{(t-1)})
\end{aligned}$$

where the inequality follows from orthant submodularity.

Using the update rules for the dual variables, we obtain

$$\begin{aligned}
f(\mathbf{S}^{(n)}) &\geq \sum_{i=1}^{k} \sum_{e^{(t)} \in S_i^{(n)}} \Delta_{e^{(t)}, i} f(\mathbf{S}^{(t-1)}) \\
&\overset{(1)}{=} \sum_{i=1}^{k} \sum_{e^{(t)} \in S_i^{(n)}} \left(\gamma_{e^{(t)}} + \phi_i^{(t-1)} \right) \\
&\overset{(2)}{=} \sum_{i=1}^{k} \sum_{e^{(t)} \in S_i^{(n)}} \left(\gamma_{e^{(t)}} + \frac{C}{B_i} \sum_{e^{(\tau)} \in X_i^{(t-1)}} \left(1 + \frac{D}{B_i}\right)^{\left|\left\{e \in X_i^{(t-1)} \colon e \succ e^{(\tau)}\right\}\right|} \gamma_{e^{(\tau)}} \right) \\
&= \sum_{i=1}^{k} \left(\sum_{e^{(t)} \in S_i^{(n)}} \gamma_{e^{(t)}} + \frac{C}{B_i} \sum_{e^{(\tau)} \in X_i^{(n)}} \gamma_{e^{(\tau)}} \sum_{e^{(t)} \in S_i^{(n)} \colon e^{(t)} \succ e^{(\tau)}} \left(1 + \frac{D}{B_i}\right)^{\left|\left\{e \in X_i^{(t-1)} \colon e \succ e^{(\tau)}\right\}\right|} \right)
\end{aligned}$$

where in the last equality we performed an exchange of summation.

Fix $e^{(\tau)} \in X_i^{(n)}$. Let $e^{(t_1)}, e^{(t_2)}, \ldots, e^{(t_m)}$ be the elements of $X_i^{(n)}$ that arrived after $e^{(\tau)}$, in the order in which they arrived. More precisely, we have $\left\{e^{(t)} \in X_i^{(n)} \colon t > \tau\right\} = \left\{e^{(t_1)}, e^{(t_2)}, \ldots, e^{(t_m)}\right\}$ and $t_1 < t_2 < \cdots < t_m$. Let $t_0 = \tau$. We consider each of the following cases in turn: $m \leq B_i$ and $m > B_i$.

Suppose that $m < B_i$. Note that $S_i^{(n)}$ is comprised of the last (at most) B_i elements of $X_i^{(n)}$. Thus we have

$$\left\{e^{(t)} \in S_i \colon e^{(t)} \succ e^{(\tau)}\right\} = \left\{e^{(t_j)} \colon 1 \le j \le m\right\}$$

Using the same calculation as above, we obtain

$$\begin{aligned}
&\sum_{e^{(t)} \in S_i^{(n)} \colon e^{(t)} \succ e^{(\tau)}} \left(1 + \frac{D}{B_i}\right)^{\left|\left\{e \in X_i^{(t-1)} \colon e \succ e^{(\tau)}\right\}\right|} \\
&= \frac{B_i}{D}\left(\left(1 + \frac{D}{B_i}\right)^{\left|\left\{e \in X_i^{(n)} \colon e \succ e^{(\tau)}\right\}\right|} - 1\right)
\end{aligned}$$

Suppose that $m \ge B_i$. We have

$$\left\{e^{(t)} \in S_i^{(n)} \colon e^{(t)} \succ e^{(\tau)}\right\} = \left\{e^{(t_j)} \colon m - B_i + 1 \le j \le m\right\}$$

and thus

$$\begin{aligned}
&\sum_{e^{(t)} \in S_i^{(n)} \colon e^{(t)} \succ e^{(\tau)}} \left(1 + \frac{D}{B_i}\right)^{\left|\left\{e \in X_i^{(t-1)} \colon e \succ e^{(\tau)}\right\}\right|} \\
&= \sum_{j=m-B_i+1}^{m} \left(1 + \frac{D}{B_i}\right)^{\left|\left\{e \in X_i^{(t_j-1)} \colon e \succ e^{(\tau)}\right\}\right|} \\
&= \sum_{j=m-B_i+1}^{m} \left(1 + \frac{D}{B_i}\right)^{\left|\left\{e \in X_i^{(t_{j-1})} \colon e \succ e^{(\tau)}\right\}\right|} \\
&= \sum_{j=m-B_i+1}^{m} \left(1 + \frac{D}{B_i}\right)^{j-1} \\
&= \frac{B_i}{D}\left(1 + \frac{D}{B_i}\right)^{m}\left(1 - \left(1 + \frac{D}{B_i}\right)^{-B_i}\right) \\
&= \frac{B_i}{D}\left(1 + \frac{D}{B_i}\right)^{\left|\left\{e \in X_i^{(n)} \colon e \succ e^{(\tau)}\right\}\right|}\left(1 - \left(1 + \frac{D}{B_i}\right)^{-B_i}\right)
\end{aligned}$$

In the second equality, we used the fact that no elements were added to S_i in iteration $t \in (t_{j-1}, t_j)$, and thus $X_i^{(t_j-1)} = X_i^{(t_{j-1})}$.

Since $S_i^{(n)}$ is comprised of the last (at most) B_i elements of $X_i^{(n)}$, we have

$$\sum_{e^{(t)} \in S_i} \gamma_{e^{(t)}} = \sum_{e^{(t)} \in X_i^{(n)} \colon \left|\left\{e \in X_i^{(n)} \colon e \succ e^{(t)}\right\}\right| < B_i} \gamma_{e^{(t)}}$$

Putting everything together, we obtain

$$\begin{aligned}
f(\mathbf{S}^{(n)}) \ge& \sum_{i=1}^{k} \sum_{e^{(\tau)} \in X_i^{(n)} \colon \left|\left\{e \in X_i^{(n)} \colon e \succ e^{(\tau)}\right\}\right| < B_i} \gamma_{e^{(\tau)}} \left(\frac{C}{D}\left(1 + \frac{D}{B_i}\right)^{\left|\left\{e \in X_i^{(n)} \colon e \succ e^{(\tau)}\right\}\right|} - \frac{C}{D} + 1\right) \\
&+ \sum_{i=1}^{k} \sum_{e^{(\tau)} \in X_i^{(n)} \colon \left|\left\{e \in X_i^{(n)} \colon e \succ e^{(\tau)}\right\}\right| \ge B_i} \gamma_{e^{(\tau)}} \frac{C}{D}\left(1 + \frac{D}{B_i}\right)^{\left|\left\{e \in X_i^{(n)} \colon e \succ e^{(\tau)}\right\}\right|}\left(1 - \left(1 + \frac{D}{B_i}\right)^{-B_i}\right)
\end{aligned}$$

$$= \sum_{e^{(\tau)} \in \mathrm{supp}(\mathbf{X}^{(n)}) : \left|\left\{e \in X^{(n)}_{i^{(\tau)}} : e \succ e^{(\tau)}\right\}\right| < B_{i^{(\tau)}}} \gamma_{e^{(\tau)}} \left(\frac{C}{D} \left(1 + \frac{D}{B_{i^{(\tau)}}}\right)^{\left|\left\{e \in X^{(n)}_{i^{(\tau)}} : e \succ e^{(\tau)}\right\}\right|} - \frac{C}{D} + 1 \right)$$

$$+ \sum_{e^{(\tau)} \in \mathrm{supp}(\mathbf{X}^{(n)}) : \left|\left\{e \in X^{(n)}_{i^{(\tau)}} : e \succ e^{(\tau)}\right\}\right| \geq B_{i^{(\tau)}}} \gamma_{e^{(\tau)}} \frac{C}{D} \left(1 + \frac{D}{B_{i^{(\tau)}}}\right)^{\left|\left\{e \in X^{(n)}_{i^{(\tau)}} : e \succ e^{(\tau)}\right\}\right|} \left(1 - \left(1 + \frac{D}{B_{i^{(\tau)}}}\right)^{-B_{i^{(\tau)}}}\right)$$

as needed. □

A.2. Proof of Lemma 3.3

Proof. (Lemma 3.3) We have $\gamma_e \geq 0$ and $\phi_i^{(n)} \geq 0$ for all $e \in V$ and $i \in [k]$. The definition of $\alpha_{e,i}$ ensures that the dual constraint $\gamma_e + \phi_i^{(n)} \geq \alpha_{e,i}$ is satisfied with equality. Therefore it only remains to verify that the first set of constraints are satisfied, i.e., for any $\mathbf{A} \in (k+1)^V$, we need to verify that

$$f(\mathbf{A}) - f(\mathbf{X}^{(n)}) \leq \sum_{i=1}^{k} \sum_{e \in A_i} \alpha_{e,i}$$

Fix a k-set $\mathbf{A} \in (k+1)^V$. Let $\mathbf{X} = \mathbf{X}^{(n)}$. We define a k-set $\widetilde{\mathbf{A}}$ by augmenting $\mathbf{A}$ with the elements in $\mathrm{supp}(\mathbf{X}) \setminus \mathrm{supp}(\mathbf{A})$. More precisely, we consider $\widetilde{\mathbf{A}} = \left(\widetilde{A}_1, \dots, \widetilde{A}_k\right)$ where $\widetilde{A}_i = A_i \cup (X_i \setminus \mathrm{supp}(\mathbf{A}))$ for all $i \in [k]$. Note that $\widetilde{\mathbf{A}} \in (k+1)^V$ and $\mathbf{A} \preceq \widetilde{\mathbf{A}}$. Since f is monotone, it follows that $f(\mathbf{A}) \leq f(\widetilde{\mathbf{A}})$. Thus it suffices to show that

$$f(\widetilde{\mathbf{A}}) - f(\mathbf{X}) \leq \sum_{i=1}^{k} \sum_{e \in A_i} \alpha_{e,i}$$

Note that we have $\mathrm{supp}(\mathbf{X}) \subseteq \mathrm{supp}(\widetilde{\mathbf{A}})$. However, an element $e \in \mathrm{supp}(\mathbf{X})$ may be assigned to different parts by $\mathbf{X}$ and $\widetilde{\mathbf{A}}$, which prevents us from directly comparing $f(\widetilde{\mathbf{A}})$ to $f(\mathbf{X})$. In the following, we modify $\widetilde{\mathbf{A}}$ so that it agrees with $\mathbf{X}$ on the placement of the elements in $\mathrm{supp}(\mathbf{X})$. By exploiting the way in which the algorithm selects the part i to which to assign an element, we will upper bound the loss in function value due to these swaps to the dual values $\alpha_{e,i}$. Additionally, we will relate the contribution to $f(\widetilde{\mathbf{A}})$ of the elements in $\mathrm{supp}(\widetilde{\mathbf{A}}) \setminus \mathrm{supp}(\mathbf{X})$ to the dual values $\alpha_{e,i}$. We now give the formal argument.

We say that two k-sets $\mathbf{Y} = (Y_1, \dots, Y_k)$ and $\mathbf{Z} = (Z_1, \dots, Z_k)$ agree on an element e if either $e \in (V \setminus \mathrm{supp}(\mathbf{Y})) \cap (V \setminus \mathrm{supp}(\mathbf{Z}))$ or $e \in Y_i \cap Z_i$ for some $i \in [k]$. We will iteratively define a sequence of k-sets $\widetilde{\mathbf{A}}^{(0)} = \widetilde{\mathbf{A}}, \widetilde{\mathbf{A}}^{(1)}, \dots, \widetilde{\mathbf{A}}^{(n-1)}, \widetilde{\mathbf{A}}^{(n)} = \mathbf{X}$ that satisfies the following two properties for every t: (i) $\widetilde{\mathbf{A}}^{(t)}$ and $\mathbf{X}$ agree on every element in $\left\{e^{(1)}, \dots, e^{(t)}\right\}$, and (ii) $\widetilde{\mathbf{A}}^{(t)}$and $\widetilde{\mathbf{A}}$ agree on every element in $\left\{e^{(t+1)}, \dots, e^{(n)}\right\}$. We set $\widetilde{\mathbf{A}}^{(0)} = \widetilde{\mathbf{A}}$. Consider $t \geq 1$, and suppose we have already defined $\widetilde{\mathbf{A}}^{(t-1)}$. Consider the element $e^{(t)}$ that arrives in iteration t. We have the following cases (recall that $\mathrm{supp}(\mathbf{X}) \subseteq \mathrm{supp}(\widetilde{\mathbf{A}})$):

- $e^{(t)} \in \mathrm{supp}(\mathbf{X})$: Recall that $e^{(t)} \in X_{i^{(t)}}$. Let j be such that $e^{(t)} \in \widetilde{A}_j$.
  If $j = i^{(t)}$, we let $\widetilde{\mathbf{A}}^{(t)} = \widetilde{\mathbf{A}}^{(t-1)}$, and we have

$$f(\widetilde{\mathbf{A}}^{(t-1)}) - f(\widetilde{\mathbf{A}}^{(t)}) = 0$$

  Therefore we may assume that $j \neq i^{(t)}$. Since $e^{(t)} \in \widetilde{A}_j \setminus X_j$, we have $e^{(t)} \in A_j$. We obtain $\widetilde{\mathbf{A}}^{(t)}$ from $\widetilde{\mathbf{A}}^{(t-1)}$ by moving $e^{(t)}$ from part j to part $i^{(t)}$, i.e., we set

$$\widetilde{A}_p^{(t)} = \begin{cases} \widetilde{A}_p^{(t-1)} & p \neq i^{(t)}, j \\ \widetilde{A}_{i^{(t)}}^{(t-1)} \cup \left\{e^{(t)}\right\} & p = i^{(t)} \\ \widetilde{A}_j^{(t-1)} \setminus \left\{e^{(t)}\right\} & p = j \end{cases}$$

For analysis purposes, we define the following intermediate k-set $\mathbf{B} = (B_1, \dots, B_k)$:

$$B_p = \begin{cases} \widetilde{A}_p^{(t-1)} & p \neq j \\ \widetilde{A}_j^{(t-1)} \setminus \left\{e^{(t)}\right\} & p = j \end{cases}$$

Note that $\mathbf{B}$ and $\widetilde{\mathbf{A}}^{(t-1)}$ only differ on $e^{(t)}$, and $e^{(t)} \notin \mathrm{supp}(\mathbf{B})$ and $e^{(t)} \in \widetilde{A}_j^{(t-1)}$. Additionally, $\mathbf{B}$ and $\widetilde{\mathbf{A}}^{(t)}$ only differ on $e^{(t)}$, and $e^{(t)} \notin \mathrm{supp}(\mathbf{B})$ and $e^{(t)} \in \widetilde{A}_{i^{(t)}}^{(t)}$. Since $\mathbf{S}^{(t-1)} \preceq \mathbf{X}^{(t-1)} \preceq \widetilde{\mathbf{A}}^{(t-1)}$ and $e^{(t)} \notin \mathrm{supp}(\mathbf{S}^{(t-1)})$, we have $\mathbf{S}^{(t-1)} \preceq \mathbf{B}$. Thus we have

$$\begin{aligned}
&f(\widetilde{\mathbf{A}}^{(t-1)}) - f(\widetilde{\mathbf{A}}^{(t)}) \\
&= \left(f(\widetilde{\mathbf{A}}^{(t-1)}) - f(\mathbf{B})\right) - \left(f(\widetilde{\mathbf{A}}^{(t)}) - f(\mathbf{B})\right) \\
&= \Delta_{e^{(t)},j} f(\mathbf{B}) - \Delta_{e,i^{(t)}} f(\mathbf{B}) \\
&\leq \Delta_{e^{(t)},j} f(\mathbf{B}) && \text{(monotonicity)} \\
&\leq \Delta_{e^{(t)},j} f(\mathbf{S}^{(t-1)}) && \text{(orthant submodularity)}
\end{aligned}$$

In the first inequality, we used that $\Delta_{e,i} f(\mathbf{B}) \geq 0$ since f is monotone. In the second inequality, we used the fact that f is orthant submodular: since $\mathbf{S}^{(t-1)} \preceq \mathbf{B}$, we have $\Delta_{e^{(t)},j} f(\mathbf{S}^{(t-1)}) \geq \Delta_{e^{(t)},j} f(\mathbf{B})$.
Using the choice of $i^{(t)}$ and the definitions of the dual values, we obtain:

$$\begin{aligned}
\Delta_{e^{(t)},j} f(\mathbf{S}^{(t-1)}) &\leq \phi_j^{(t-1)} + \Delta_{e^{(t)},i^{(t)}} f(\mathbf{S}^{(t-1)}) - \phi_{i^{(t)}}^{(t-1)} && \left(\text{definition of } i^{(t)}\right) \\
&= \phi_j^{(t-1)} + \gamma_{e^{(t)}} && (\text{definition of } \gamma_{e^{(t)}}) \\
&\leq \phi_j^{(n)} + \gamma_{e^{(t)}} \\
&= \alpha_{e^{(t)},j}
\end{aligned}$$

Therefore

$$f(\widetilde{\mathbf{A}}^{(t-1)}) - f(\widetilde{\mathbf{A}}^{(t)}) \leq \alpha_{e^{(t)},j}$$

- $e^{(t)} \in \mathrm{supp}(\widetilde{\mathbf{A}}) \setminus \mathrm{supp}(\mathbf{X})$: Let j be such that $e^{(t)} \in \widetilde{A}_j$. Note that we have $e^{(t)} \in \widetilde{A}_j^{(t-1)}$. We obtain $\widetilde{\mathbf{A}}^{(t)}$ from $\widetilde{\mathbf{A}}^{(t-1)}$ by discarding $e^{(t)}$, i.e.,

$$\widetilde{A}_p^{(t)} = \begin{cases} \widetilde{A}_p^{(t-1)} & p \neq j \\ \widetilde{A}_j^{(t-1)} \setminus \left\{e^{(t)}\right\} & p = j \end{cases}$$

Since $\mathbf{S}^{(t)} = \mathbf{S}^{(t-1)} \preceq \mathbf{X}^{(t)} = \mathbf{X}^{(t-1)} \preceq \widetilde{\mathbf{A}}^{(t-1)}$ and $e^{(t)} \notin \mathrm{supp}(\mathbf{S}^{(t-1)})$, we have $\mathbf{S}^{(t-1)} \preceq \widetilde{\mathbf{A}}^{(t)}$. Thus

$$\begin{aligned}
f(\widetilde{\mathbf{A}}^{(t-1)}) - f(\widetilde{\mathbf{A}}^{(t)}) &= \Delta_{e^{(t)},j} f(\widetilde{\mathbf{A}}^{(t)}) \\
&\leq \Delta_{e^{(t)},j} f(\mathbf{S}^{(t-1)}) && \text{(orthant submodularity)}
\end{aligned}$$

Since $e^{(t)} \notin \mathrm{supp}(\mathbf{X})$, we have

$$\Delta_{e^{(t)},j} f(\mathbf{S}^{(t-1)}) \leq \phi_j^{(t-1)} \leq \phi_j^{(n)} + \gamma_{e^{(t)}} = \alpha_{e^{(t)},j}$$

Therefore

$$f(\widetilde{\mathbf{A}}^{(t-1)}) - f(\widetilde{\mathbf{A}}^{(t)}) \leq \alpha_{e^{(t)},j}$$

Since $e^{(t)} \in \widetilde{A}_j \setminus X_j$, we have $e^{(t)} \in A_j$.

- $e^{(t)} \notin \mathrm{supp}(\widetilde{\mathbf{A}})$: We have $e^{(t)} \notin \mathrm{supp}(\mathbf{X})$ as well, and we can set $\widetilde{\mathbf{A}}^{(t)} = \widetilde{\mathbf{A}}^{(t-1)}$. We have

$$f(\widetilde{\mathbf{A}}^{(t-1)}) - f(\widetilde{\mathbf{A}}^{(t)}) = 0$$

Thus we obtain

$$\begin{aligned}
f(\widetilde{\mathbf{A}}) - f(\widetilde{\mathbf{X}}) &= f(\widetilde{\mathbf{A}}^{(0)}) - f(\widetilde{\mathbf{A}}^{(n)}) \\
&= \sum_{t=1}^{n} \left(f(\widetilde{\mathbf{A}}^{(t-1)}) - f(\widetilde{\mathbf{A}}^{(t)}) \right) \\
&\leq \sum_{j=1}^{k} \sum_{e \in A_j} \alpha_{e,j}
\end{aligned}$$

as needed. □

A.3. Proof of Lemma 3.4

Proof. (Lemma 3.4) Consider the dual solution from Lemma 3.3. By weak duality, the objective value of the dual solution is an upper bound on the objective value of any primal solution, and in particular $f(\mathbf{S}^*)$. Thus we obtain

$$f(\mathbf{S}^*) \leq f(\mathbf{X}^{(n)}) + \sum_{e^{(t)} \in \mathrm{supp}(\mathbf{X}^{(n)})} \gamma_{e^{(t)}} + \sum_{i=1}^{k} B_i \phi_i^{(n)}$$

Using that $\mathbf{S}^{(t)} \preceq \mathbf{X}^{(t)}$ for all t and that f is orthant submodular, we can upper bound $f(\mathbf{X}^{(n)})$ as follows:

$$\begin{aligned}
f(\mathbf{X}^{(n)}) &= f(\mathbf{X}^{(n)}) - f(\mathbf{X}^{(0)}) \\
&= \sum_{e^{(t)} \in \mathrm{supp}(\mathbf{X}^{(n)})} \Delta_{e^{(t)}, i^{(t)}} f(\mathbf{X}^{(t-1)}) \\
&\overset{(\mathbf{S}^{(t-1)} \preceq \mathbf{X}^{(t-1)})}{\leq} \sum_{e^{(t)} \in \mathrm{supp}(\mathbf{X}^{(n)})} \Delta_{e^{(t)}, i^{(t)}} f(\mathbf{S}^{(t-1)})
\end{aligned}$$

Additionally, we have

$$\begin{aligned}
&\sum_{i=1}^{k} B_i \phi_i^{(n)} \\
&= \sum_{i=1}^{k} B_i \sum_{t=1}^{n} \left(\phi_i^{(t)} - \phi_i^{(t-1)} \right) \\
&= \sum_{t=1}^{n} \sum_{i=1}^{k} B_i \left(\phi_i^{(t)} - \phi_i^{(t-1)} \right) \\
&= \sum_{e^{(t)} \in \mathrm{supp}(\mathbf{X}^{(n)})} B_{i^{(t)}} \left(\phi_{i^{(t)}}^{(t)} - \phi_{i^{(t)}}^{(t-1)} \right) && \left(\phi_i^{(t)} = \phi_i^{(t-1)} \text{if } i \neq i^{(t)} \text{ or } e^{(t)} \notin \mathrm{supp}(\mathbf{X}^{(n)}) \right) \\
&= \sum_{e^{(t)} \in \mathrm{supp}(\mathbf{X}^{(n)})} \left(D \Delta_{e^{(t)}, i^{(t)}} f(\mathbf{S}^{(t-1)}) + (C - D)\, \gamma_{e^{(t)}} \right) && (\text{update rule for } \phi_i)
\end{aligned}$$

Plugging into the first inequality and using Lemma 3.2, we obtain

$$\begin{aligned}
f(\mathbf{S}^*) &\leq (D+1) \sum_{e^{(t)} \in \mathrm{supp}(\mathbf{X}^{(n)})} \Delta_{e^{(t)}, i^{(t)}} f(\mathbf{S}^{(t-1)}) + (C - D + 1) \sum_{e^{(t)} \in \mathrm{supp}(\mathbf{X}^{(n)})} \gamma_{e^{(t)}} \\
&= \sum_{e^{(t)} \in \mathrm{supp}(\mathbf{X}^{(n)})} ((D+1)\, c_t + C - D + 1)\, \gamma_{e^{(t)}}
\end{aligned}$$

as needed. □

A.4. Proof of Lemma 3.5

Proof. (Lemma 3.5) We will use the following standard inequality:

$$\frac{\sum_{i=1}^n a_i}{\sum_{i=1}^n b_i} \geq \min_{i \in [n]} \frac{a_i}{b_i}$$

By applying the above inequality, we obtain

$$\frac{f(\mathbf{S})}{f(\mathbf{S}^*)} \geq \min_{t:\, e^{(t)} \in \operatorname{supp}(\mathbf{X}^{(n)})} \frac{\widetilde{c}_t}{\widehat{c}_t}$$

We now lower bound the ratio of the coefficients. We will set $C = 2D$ and we will find the best choice for D. Consider $e^{(t)} \in \operatorname{supp}(\mathbf{X}^{(n)})$, and let $j = \left|\left\{e \in X_{i^{(t)}}^{(n)} \colon e \succ e^{(t)}\right\}\right|$. If $j \geq B_{i^{(t)}}$, we have

$$\begin{aligned}
\frac{\widetilde{c}_t}{\widehat{c}_t} &= \frac{\frac{C}{D}\left(1+\frac{D}{B_{i^{(t)}}}\right)^j \left(1-\left(1+\frac{D}{B_{i^{(t)}}}\right)^{-B_{i^{(t)}}}\right)}{(D+1)\left(\frac{C}{D}\left(1+\frac{D}{B_{i^{(t)}}}\right)^j - \frac{C}{D}+1\right)+C-D+1} \\
&= \frac{\frac{C}{D}\left(1+\frac{D}{B_{i^{(t)}}}\right)^j \left(1-\left(1+\frac{D}{B_{i^{(t)}}}\right)^{-B_{i^{(t)}}}\right)}{(D+1)\frac{C}{D}\left(1+\frac{D}{B_{i^{(t)}}}\right)^j + 2 - \frac{C}{D}} \\
&= \frac{2\left(1+\frac{D}{B_{i^{(t)}}}\right)^j \left(1-\left(1+\frac{D}{B_{i^{(t)}}}\right)^{-B_{i^{(t)}}}\right)}{2\,(D+1)\left(1+\frac{D}{B_{i^{(t)}}}\right)^j} && \left(\frac{C}{D}=2\right) \\
&= \left(1-\left(1+\frac{D}{B_{i^{(t)}}}\right)^{-B_{i^{(t)}}}\right)\frac{1}{D+1}
\end{aligned}$$

If $j < B_{i^{(t)}}$, we have

$$\begin{aligned}
\frac{\widetilde{c}_t}{\widehat{c}_t} &= \frac{\frac{C}{D}\left(1+\frac{D}{B_{i^{(t)}}}\right)^j - \frac{C}{D}+1}{(D+1)\left(\frac{C}{D}\left(1+\frac{D}{B_{i^{(t)}}}\right)^j - \frac{C}{D}+1\right)+C-D+1} \\
&= \frac{\frac{C}{D}\left(1+\frac{D}{B_{i^{(t)}}}\right)^j - \frac{C}{D}+1}{(D+1)\frac{C}{D}\left(1+\frac{D}{B_{i^{(t)}}}\right)^j + 2 - \frac{C}{D}} \\
&= \frac{2\left(1+\frac{D}{B_{i^{(t)}}}\right)^j - 1}{2\,(D+1)\left(1+\frac{D}{B_{i^{(t)}}}\right)^j} && \left(\frac{C}{D}=2\right) \\
&= \frac{2-\left(1+\frac{D}{B_{i^{(t)}}}\right)^{-j}}{2\,(D+1)} \\
&\geq \frac{1}{2\,(D+1)}
\end{aligned}$$

Therefore

$$\frac{\widetilde{c}_t}{\widehat{c}_t} \geq \frac{1}{D+1}\min\left\{\frac{1}{2}, 1-\left(1+\frac{D}{B_{i^{(t)}}}\right)^{-B_{i^{(t)}}}\right\}$$

and thus

$$\min_{t:\, e^{(t)}\in\mathbf{X}^{(n)}} \frac{\widetilde{c}_t}{\widehat{c}_t} \geq \frac{1}{D+1} \min\left\{\frac{1}{2}, \min_{i\in[k]}\left(1-\left(1+\frac{D}{B_i}\right)^{-B_i}\right)\right\}$$

Letting $B = \min_{i\in[k]} B_i$ and noting that $1-\left(1+\frac{D}{x}\right)^x$ is increasing with x, we obtain

$$\min_{t:\, e^{(t)}\in\mathbf{X}^{(n)}} \frac{\widetilde{c}_t}{\widehat{c}_t} \geq \frac{1}{D+1} \min\left\{\frac{1}{2}, 1-\left(1+\frac{D}{B}\right)^{-B}\right\}$$

Putting everything together, we obtain

$$\frac{f(\mathbf{S})}{f(\mathbf{S}^*)} \geq \frac{1}{D+1} \min\left\{\frac{1}{2}, 1-\left(1+\frac{D}{B}\right)^{-B}\right\}$$

We choose D to make the two terms equal:

$$\frac{1}{2} = 1-\left(1+\frac{D}{B}\right)^{-B} \Rightarrow D = B\left(2^{1/B}-1\right)$$

and obtain

$$\frac{f(\mathbf{S})}{f(\mathbf{S}^*)} \geq \frac{1}{2\left(1+B\left(2^{1/B}-1\right)\right)}$$

as needed. □

B. Analysis of Algorithm 2

B.1. Space and Time Analysis

Analogously to Algorithm 1, Algorithm 2 stores the $O\left(\sum_{i=1}^k B_i\right)$ elements in the solution $S = S_1 \cup \cdots \cup S_k$ and the k dual variables $\{\phi_i \colon i \in [k]\}$. Thus the total space usage is $O\left(\sum_{i=1}^k B_i\right)$. In each iteration t, the algorithm evaluates the function $O(1)$ times in order to compute the marginal gain $\Delta_e f(S)$, and it performs $O(1)$ additional operations. We maintain each set S_i in a deque, and thus removing the earliest element of S_i and adding an element to the back of S_i can be performed in $O(1)$ time. Thus the algorithm performs $O\left(|V|\right)$ function evaluations and $O\left(|V|\right)$ additional time.

B.2. Analysis of the Approximation Guarantee

For analysis purposes, we annotate the main quantities in the algorithm using the superscript (t): $\phi_i^{(t)}, S^{(t)}$ denote the respective quantities at the *end* of iteration t; $e^{(t)}$ denotes the element that arrives in iteration t; $i^{(t)}$denotes the index of the part containing $e^{(t)}$ (i.e., $e^{(t)} \in P_{i^{(t)}}$). We also let $X_i^{(t)} = S_i^{(1)} \cup \cdots \cup S_i^{(t)}$ denote the set of elements that were added to S_i in the first t iterations, and $X^{(t)} = X_1^{(t)} \cup \cdots \cup X_k^{(t)}$.

We proceed similarly to the analysis from Section 3.3, with the main difference being that we show a stronger upper bound on the optimal solution value $f(S^*)$ in Lemma B.2. As before, we relate $\sum_{e^{(t)}\in X^{(n)}} \Delta_{e^{(t)}} f(S^{(t-1)})$, $f(S^{(n)})$, and $f(S^*)$ to suitable linear combinations of the dual values γ_e constructed by the algorithm. We then derive our approximation guarantee by analyzing the coefficients of each γ_e in the respective linear combinations.

We start by deriving the linear combinations for $\sum_{e^{(t)}\in X^{(n)}} \Delta_{e^{(t)}} f(S^{(t-1)})$ and $f(S^{(n)})$. The proof of the following lemma is analogous to Lemma 3.2.

Lemma B.1. *We have*

$$\sum_{e^{(t)}\in X^{(n)}} \Delta_{e^{(t)}} f(S^{(t-1)}) = \sum_{e^{(t)}\in X^{(n)}} c_t \gamma_{e^{(t)}}$$

and

$$f(S^{(n)}) \geq \sum_{e^{(t)}\in X^{(n)}} \widetilde{c}_t \gamma_{e^{(t)}}$$

where

$$c_t = \frac{C}{D}\left(1+\frac{D}{B_{i(t)}}\right)^{\left|\left\{e\in X_{i(t)}^{(n)}\colon e\succ e^{(t)}\right\}\right|} - \frac{C}{D} + 1$$
$$\widetilde{c}_t = \begin{cases} \frac{C}{D}\left(1+\frac{D}{B_{i(t)}}\right)^{\left|\left\{e\in X_{i(t)}^{(n)}\colon e\succ e^{(t)}\right\}\right|}\left(1-\left(1+\frac{D}{B_{i(t)}}\right)^{-B_{i(t)}}\right) & \textit{if } \left|\left\{e\in X_{i(t)}^{(n)}\colon e\succ e^{(t)}\right\}\right| \geq B_{i(t)} \\ c_t & \textit{otherwise} \end{cases}$$

Proof. For each $e^{(t)} \in X^{(n)}$, by rearranging the definition of $\gamma_{e^{(t)}}$, we obtain

$$\Delta_{e^{(t)}} f(S^{(t-1)}) = \gamma_{e^{(t)}} + \phi_{i(t)}^{(t-1)} \tag{3}$$

By unrolling the update rule for ϕ_i, we obtain

$$\phi_i^{(t)} = \frac{C}{B_i}\sum_{e^{(\tau)}\in X_i^{(t)}}\left(1+\frac{D}{B_i}\right)^{\left|\left\{e\in X_i^{(t)}\colon e\succ e^{(\tau)}\right\}\right|}\gamma_{e^{(\tau)}} \tag{4}$$

We first consider $\sum_{e^{(t)}\in X^{(n)}}\Delta_{e^{(t)}} f(S^{(t-1)})$. We have

$$\begin{aligned}
&\sum_{e^{(t)}\in X^{(n)}}\Delta_{e^{(t)}} f(S^{(t-1)})\\
&\overset{(3)}{=}\sum_{e^{(t)}\in X^{(n)}}\left(\gamma_{e^{(t)}}+\phi_{i(t)}^{(t-1)}\right)\\
&\overset{(4)}{=}\sum_{e^{(t)}\in X^{(n)}}\left(\gamma_{e^{(t)}}+\frac{C}{B_{i(t)}}\sum_{e^{(\tau)}\in X_{i(t)}^{(t-1)}}\left(1+\frac{D}{B_{i(t)}}\right)^{\left|\left\{e\in X_{i(t)}^{(t-1)}\colon e\succ e^{(\tau)}\right\}\right|}\gamma_{e^{(\tau)}}\right)\\
&=\sum_{i=1}^{k}\sum_{e^{(t)}\in X_i^{(n)}}\left(\gamma_{e^{(t)}}+\frac{C}{B_i}\sum_{e^{(\tau)}\in X_i^{(t-1)}}\left(1+\frac{D}{B_i}\right)^{\left|\left\{e\in X_i^{(t-1)}\colon e\succ e^{(\tau)}\right\}\right|}\gamma_{e^{(\tau)}}\right)\\
&=\sum_{i=1}^{k}\left(\sum_{e^{(t)}\in X_i^{(n)}}\gamma_{e^{(t)}}+\frac{C}{B_i}\sum_{e^{(\tau)}\in X_i^{(n)}}\gamma_{e^{(\tau)}}\sum_{e^{(t)}\in X_i^{(n)}\colon e^{(t)}\succ e^{(\tau)}}\left(1+\frac{D}{B_i}\right)^{\left|\left\{e\in X_i^{(t-1)}\colon e\succ e^{(\tau)}\right\}\right|}\right)
\end{aligned}$$

where in the last equality we performed an exchange of summation.

Fix $e^{(\tau)} \in X_i^{(n)}$. Let $e^{(t_1)}, e^{(t_2)}, \ldots, e^{(t_m)}$ be the elements of $X_i^{(n)}$ that arrived after $e^{(\tau)}$. We have

$$\begin{aligned}
&\sum_{e^{(t)}\in X_i^{(n)}\colon e^{(t)}\succ e^{(\tau)}}\left(1+\frac{D}{B_i}\right)^{\left|\left\{e\in X_i^{(t-1)}\colon e\succ e^{(\tau)}\right\}\right|}\\
&=\sum_{j=1}^{m}\left(1+\frac{D}{B_i}\right)^{\left|\left\{e\in X_i^{(t_j-1)}\colon e\succ e^{(\tau)}\right\}\right|}\\
&=\sum_{j=1}^{m}\left(1+\frac{D}{B_i}\right)^{\left|\left\{e\in X_i^{(t_{j-1})}\colon e\succ e^{(\tau)}\right\}\right|}\\
&=\sum_{j=1}^{m}\left(1+\frac{D}{B_i}\right)^{j-1}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{B_i}{D}\left(\left(1+\frac{D}{B_i}\right)^m - 1\right) \\
&= \frac{B_i}{D}\left(\left(1+\frac{D}{B_i}\right)^{\left|\left\{e\in X_i^{(n)}\,:\, e\succ e^{(\tau)}\right\}\right|} - 1\right)
\end{aligned}
$$

In the second equality, we used the fact that no elements were added to S_i in iteration $t \in (t_{j-1}, t_j)$, and thus $X_i^{(t_j-1)} = X_i^{(t_{j-1})}$.

Putting everything together, we obtain

$$
\begin{aligned}
&\sum_{e^{(t)}\in X^{(n)}} \Delta_{e^{(t)}} f(S^{(t-1)}) \\
&= \sum_{i=1}^{k} \sum_{e^{(t)}\in X_i^{(n)}} \left(\frac{C}{D}\left(1+\frac{D}{B_i}\right)^{\left|\left\{e\in X_i^{(n)}\,:\, e\succ e^{(t)}\right\}\right|} - \frac{C}{D} + 1\right)\gamma_{e^{(t)}} \\
&= \sum_{e^{(t)}\in X^{(n)}} \left(\frac{C}{D}\left(1+\frac{D}{B_{i(t)}}\right)^{\left|\left\{e\in X_{i(t)}^{(n)}\,:\, e\succ e^{(t)}\right\}\right|} - \frac{C}{D} + 1\right)\gamma_{e^{(t)}}
\end{aligned}
$$

Next, we consider $f(S^{(n)})$. Let

$$
\widetilde{S}^{(t)} := \left(S_1^{(n)} \cap \left\{e^{(1)}, \dots, e^{(t)}\right\}, \dots, S_k^{(n)} \cap \left\{e^{(1)}, \dots, e^{(t)}\right\}\right)
$$

Note that we have $\widetilde{S}^{(t)} \subseteq S^{(t)}$ for all t. We have

$$
\begin{aligned}
f(S^{(n)}) &= f(\widetilde{S}^{(n)}) - f(\widetilde{S}^{(0)}) \\
&= \sum_{e^{(t)}\in S^{(n)}} \Delta_{e^{(t)}} f(\widetilde{S}^{(t-1)}) \\
&\overset{\left(\widetilde{S}^{(t-1)}\preceq S^{(t-1)}\right)}{\geq} \sum_{e^{(t)}\in S^{(n)}} \Delta_{e^{(t)}} f(S^{(t-1)}) \\
&= \sum_{i=1}^{k} \sum_{e^{(t)}\in S_i^{(n)}} \Delta_{e^{(t)}} f(S^{(t-1)})
\end{aligned}
$$

where the inequality follows from submodularity.

Using the update rules for the dual variables, we obtain

$$
\begin{aligned}
f(S^{(n)}) &\geq \sum_{i=1}^{k} \sum_{e^{(t)}\in S_i^{(n)}} \Delta_{e^{(t)}} f(S^{(t-1)}) \\
&\overset{(3)}{=} \sum_{i=1}^{k} \sum_{e^{(t)}\in S_i^{(n)}} \left(\gamma_{e^{(t)}} + \phi_i^{(t-1)}\right) \\
&\overset{(4)}{=} \sum_{i=1}^{k} \sum_{e^{(t)}\in S_i^{(n)}} \left(\gamma_{e^{(t)}} + \frac{C}{B_i} \sum_{e^{(\tau)}\in X_i^{(t-1)}} \left(1+\frac{D}{B_i}\right)^{\left|\left\{e\in X_i^{(t-1)}\,:\, e\succ e^{(\tau)}\right\}\right|} \gamma_{e^{(\tau)}}\right) \\
&= \sum_{i=1}^{k} \left(\sum_{e^{(t)}\in S_i^{(n)}} \gamma_{e^{(t)}} + \frac{C}{B_i} \sum_{e^{(\tau)}\in X_i^{(n)}} \gamma_{e^{(\tau)}} \sum_{e^{(t)}\in S_i^{(n)}\,:\, e^{(t)}\succ e^{(\tau)}} \left(1+\frac{D}{B_i}\right)^{\left|\left\{e\in X_i^{(t-1)}\,:\, e\succ e^{(\tau)}\right\}\right|}\right)
\end{aligned}
$$

where in the last equality we performed an exchange of summation.

Fix $e^{(\tau)} \in X_i^{(n)}$. Let $e^{(t_1)}, e^{(t_2)}, \ldots, e^{(t_m)}$ be the elements of $X_i^{(n)}$ that arrived after $e^{(\tau)}$, in the order in which they arrived. More precisely, we have $\left\{e^{(t)} \in X_i^{(n)} \colon t > \tau\right\} = \left\{e^{(t_1)}, e^{(t_2)}, \ldots, e^{(t_m)}\right\}$ and $t_1 < t_2 < \cdots < t_m$. Let $t_0 = \tau$. We consider each of the following cases in turn: $m \leq B_i$ and $m > B_i$.

Suppose that $m < B_i$. Note that $S_i^{(n)}$ is comprised of the last (at most) B_i elements of $X_i^{(n)}$. Thus we have

$$\left\{e^{(t)} \in S_i \colon e^{(t)} \succ e^{(\tau)}\right\} = \left\{e^{(t_j)} \colon 1 \leq j \leq m\right\}$$

Using the same calculation as above, we obtain

$$\begin{aligned}
&\sum_{e^{(t)} \in S_i^{(n)} \colon e^{(t)} \succ e^{(\tau)}} \left(1 + \frac{D}{B_i}\right)^{\left|\left\{e \in X_i^{(t-1)} \colon e \succ e^{(\tau)}\right\}\right|} \\
&= \frac{B_i}{D}\left(\left(1 + \frac{D}{B_i}\right)^{\left|\left\{e \in X_i^{(n)} \colon e \succ e^{(\tau)}\right\}\right|} - 1\right)
\end{aligned}$$

Suppose that $m \geq B_i$. We have

$$\left\{e^{(t)} \in S_i^{(n)} \colon e^{(t)} \succ e^{(\tau)}\right\} = \left\{e^{(t_j)} \colon m - B_i + 1 \leq j \leq m\right\}$$

and thus

$$\begin{aligned}
&\sum_{e^{(t)} \in S_i^{(n)} \colon e^{(t)} \succ e^{(\tau)}} \left(1 + \frac{D}{B_i}\right)^{\left|\left\{e \in X_i^{(t-1)} \colon e \succ e^{(\tau)}\right\}\right|} \\
&= \sum_{j=m-B_i+1}^{m} \left(1 + \frac{D}{B_i}\right)^{\left|\left\{e \in X_i^{(t_j - 1)} \colon e \succ e^{(\tau)}\right\}\right|} \\
&= \sum_{j=m-B_i+1}^{m} \left(1 + \frac{D}{B_i}\right)^{\left|\left\{e \in X_i^{(t_{j-1})} \colon e \succ e^{(\tau)}\right\}\right|} \\
&= \sum_{j=m-B_i+1}^{m} \left(1 + \frac{D}{B_i}\right)^{j-1} \\
&= \frac{B_i}{D}\left(1 + \frac{D}{B_i}\right)^{m}\left(1 - \left(1 + \frac{D}{B_i}\right)^{-B_i}\right) \\
&= \frac{B_i}{D}\left(1 + \frac{D}{B_i}\right)^{\left|\left\{e \in X_i^{(n)} \colon e \succ e^{(\tau)}\right\}\right|}\left(1 - \left(1 + \frac{D}{B_i}\right)^{-B_i}\right)
\end{aligned}$$

In the second equality, we used the fact that no elements were added to S_i in iteration $t \in (t_{j-1}, t_j)$, and thus $X_i^{(t_j - 1)} = X_i^{(t_{j-1})}$.

Since $S_i^{(n)}$ is comprised of the last (at most) B_i elements of $X_i^{(n)}$, we have

$$\sum_{e^{(t)} \in S_i} \gamma_{e^{(t)}} = \sum_{e^{(t)} \in X_i^{(n)} \colon \left|\left\{e \in X_i^{(n)} \colon e \succ e^{(t)}\right\}\right| < B_i} \gamma_{e^{(t)}}$$

Putting everything together, we obtain

$$f(S^{(n)}) \geq \sum_{i=1}^{k} \sum_{e^{(\tau)} \in X_i^{(n)} \colon \left|\left\{e \in X_i^{(n)} \colon e \succ e^{(\tau)}\right\}\right| < B_i} \gamma_{e^{(\tau)}} \left(\frac{C}{D}\left(1 + \frac{D}{B_i}\right)^{\left|\left\{e \in X_i^{(n)} \colon e \succ e^{(\tau)}\right\}\right|} - \frac{C}{D} + 1\right)$$

$$+\sum_{i=1}^{k} \sum_{e^{(\tau)} \in X_i^{(n)} : \left|\left\{e \in X_i^{(n)} : e \succ e^{(\tau)}\right\}\right| \geq B_i} \gamma_{e^{(\tau)}} \frac{C}{D} \left(1 + \frac{D}{B_i}\right)^{\left|\left\{e \in X_i^{(n)} : e \succ e^{(\tau)}\right\}\right|} \left(1 - \left(1 + \frac{D}{B_i}\right)^{-B_i}\right)$$

$$= \sum_{e^{(\tau)} \in X^{(n)} : \left|\left\{e \in X_{i^{(\tau)}}^{(n)} : e \succ e^{(\tau)}\right\}\right| < B_{i^{(\tau)}}} \gamma_{e^{(\tau)}} \left(\frac{C}{D} \left(1 + \frac{D}{B_{i^{(\tau)}}}\right)^{\left|\left\{e \in X_{i^{(\tau)}}^{(n)} : e \succ e^{(\tau)}\right\}\right|} - \frac{C}{D} + 1\right)$$

$$+ \sum_{e^{(\tau)} \in X^{(n)} : \left|\left\{e \in X_{i^{(\tau)}}^{(n)} : e \succ e^{(\tau)}\right\}\right| \geq B_{i^{(\tau)}}} \gamma_{e^{(\tau)}} \frac{C}{D} \left(1 + \frac{D}{B_{i^{(\tau)}}}\right)^{\left|\left\{e \in X_{i^{(\tau)}}^{(n)} : e \succ e^{(\tau)}\right\}\right|} \left(1 - \left(1 + \frac{D}{B_{i^{(\tau)}}}\right)^{-B_{i^{(\tau)}}}\right)$$

as needed. □

Next, we upper bound $f(S^*)$ using a linear combination of the γ_e values. Compared to the k-submodular setting, we are able to obtain a stronger upper bound on $f(S^*)$.

Lemma B.2. *We have*

$$f(S^*) \leq \sum_{e^{(t)} \in X^{(n)}} \widehat{c}_t \gamma_{e^{(t)}}$$

where

$$\widehat{c}_t = (D+1)\, c_t + C - D$$

and c_t is the coefficient from Lemma B.1.

Proof. By monotonicity, we have

$$f(S^*) \leq f(S^* \cup X^{(n)})$$

Therefore it suffices to upper bound $f(S^* \cup X^{(n)})$. Our first step is to apply submodularity and upper bound $f(S^* \cup X^{(n)})$ in terms of the marginal gains $\Delta_{e^{(t)}} f(S^{(t-1)})$. Let $e^{(t_1)}, \ldots, e^{(t_a)}$ be the elements of $X^{(n)}$, where $t_1 < \cdots < t_a$. Let $e^{(t_{a+1})}, \ldots, e^{(t_b)}$ be the elements of $S^* \setminus X^{(n)}$. We have

$$\begin{aligned} f(X^{(n)} \cup S^*) &= f(X^{(n)} \cup S^*) - f(\emptyset) \\ &= \sum_{i=1}^{a+b} \Delta_{e^{(t_i)}} f\left(\left\{e^{(t_1)}, \ldots, e^{(t_{i-1})}\right\}\right) \\ &\leq \sum_{i=1}^{a+b} \Delta_{e^{(t_i)}} f(S^{(t_{i-1})}) \\ &= \sum_{e^{(t)} \in X^{(n)}} \Delta_{e^{(t)}} f(S^{(t-1)}) + \sum_{e^{(t)} \in S^* \setminus X^{(n)}} \Delta_{e^{(t)}} f(S^{(t-1)}) \end{aligned}$$

The inequality follows by submodularity: we have $S^{(t_{i-1})} \subseteq X^{(t_{i-1})} \subseteq \left\{e^{(t_1)}, \ldots, e^{(t_{i-1})}\right\}$.

For each element $e^{(t)} \notin X^{(n)}$, we have $\Delta_{e^{(t)}} f(S^{(t-1)}) \leq \phi_{i^{(t)}}^{(t-1)}$. Moreover, the dual values $\phi_i^{(t)}$ are non-negative. Therefore

$$\sum_{e^{(t)} \in S^* \setminus X^{(n)}} \Delta_{e^{(t)}} f(S^{(t-1)}) \leq \sum_{e^{(t)} \in S^* \setminus X^{(n)}} \phi_{i^{(t)}}^{(t-1)} \leq \sum_{e^{(t)} \in S^*} \phi_{i^{(t)}}^{(t-1)}$$

Using that $\phi_i^{(t)}$ is non-decreasing with t, the feasibility of S^*, and the update rule for ϕ_i, we obtain

$$\sum_{e^{(t)} \in S^*} \phi_{i^{(t)}}^{(t-1)} = \sum_{i=1}^{k} \sum_{e^{(t)} \in S^* \cap P_i} \phi_i^{(t-1)}$$

$$
\begin{aligned}
&\leq \sum_{i=1}^{k} \phi_i^{(n)} \underbrace{|S^* \cap P_i|}_{\leq B_i} \\
&\leq \sum_{i=1}^{k} \phi_i^{(n)} B_i \\
&= \sum_{i=1}^{k} B_i \left(\phi_i^{(n)} - \phi_i^{(0)}\right) \\
&= \sum_{i=1}^{k} \sum_{e^{(t)} \in X_i^{(n)}} B_i \left(\phi_i^{(t)} - \phi_i^{(t-1)}\right) \\
&= \sum_{i=1}^{k} \sum_{e^{(t)} \in X_i^{(n)}} \left(D \Delta_{e^{(t)}} f(S^{(t-1)}) + (C - D)\, \gamma_{e^{(t)}}\right) \\
&= \sum_{e^{(t)} \in X^{(n)}} \left(D \Delta_{e^{(t)}} f(S^{(t-1)}) + (C - D)\, \gamma_{e^{(t)}}\right)
\end{aligned}
$$

Putting everything together and using Lemma 3.2, we obtain

$$
\begin{aligned}
f(X^{(n)} \cup S^*) &\leq \sum_{e^{(t)} \in X^{(n)}} (D+1)\, \Delta_{e^{(t)}} f(S^{(t-1)}) + (C - D) \sum_{e^{(t)} \in X^{(n)}} \gamma_{e^{(t)}} \\
&= \sum_{e^{(t)} \in X^{(n)}} ((D+1)\, c_t + C - D)\, \gamma_{e^{(t)}}
\end{aligned}
$$

as needed. □

We now derive the approximation guarantee. The proof of the following lemma is analogous to Lemma 3.5.

Lemma B.3. *Let $B = \min_{i \in [k]} B_i$ be the minimum budget among all parts. If we set $C = D$, we obtain*

$$
\frac{f(S)}{f(S^*)} \geq \min_{t:\ e^{(t)} \in X^{(n)}} \frac{\widetilde{c}_t}{\widehat{c}_t} \geq \left(1 - \left(1 + \frac{D}{B}\right)^{-B}\right) \frac{1}{D+1}
$$

where $\widetilde{c}_t$ and $\widehat{c}_t$ are the coefficients from Lemmas B.1 and B.2.

Proof. We will use the following standard inequality:

$$
\frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} b_i} \geq \min_{i \in [n]} \frac{a_i}{b_i}
$$

By applying the above inequality, we obtain

$$
\frac{f(S)}{f(S^*)} \geq \min_{t:\ e^{(t)} \in X^{(n)}} \frac{\widetilde{c}_t}{\widehat{c}_t}
$$

We now lower bound the ratio of the coefficients. Recall that $C = D$.

Consider $e^{(t)} \in X^{(n)}$, and let $j = \left|\left\{e \in X_{i(t)}^{(n)} \colon e \succ e^{(t)}\right\}\right|$. If $j \geq B_{i^{(t)}}$, we have

$$
\frac{\widetilde{c}_t}{\widehat{c}_t} = \frac{\frac{C}{D}\left(1 + \frac{D}{B_{i^{(t)}}}\right)^{j} \left(1 - \left(1 + \frac{D}{B_{i^{(t)}}}\right)^{-B_{i^{(t)}}}\right)}{(D+1)\left(\frac{C}{D}\left(1 + \frac{D}{B_{i^{(t)}}}\right)^{j} - \frac{C}{D} + 1\right) + C - D}
$$

$$= \frac{\frac{C}{D}\left(1+\frac{D}{B_{i(t)}}\right)^{j}\left(1-\left(1+\frac{D}{B_{i(t)}}\right)^{-B_{i(t)}}\right)}{(D+1)\frac{C}{D}\left(1+\frac{D}{B_{i(t)}}\right)^{j}-\frac{C}{D}+1}$$

$$= \left(1-\left(1+\frac{D}{B_{i(t)}}\right)^{-B_{i(t)}}\right)\frac{1}{D+1} \qquad \left(\frac{C}{D}=1\right)$$

If $j < B_{i(t)}$, we have

$$\frac{\widetilde{c}_t}{\widehat{c}_t} = \frac{\frac{C}{D}\left(1+\frac{D}{B_{i(t)}}\right)^{j}-\frac{C}{D}+1}{(D+1)\frac{C}{D}\left(1+\frac{D}{B_{i(t)}}\right)^{j}-\frac{C}{D}+1}$$

$$= \frac{1}{D+1} \qquad \left(\frac{C}{D}=1\right)$$

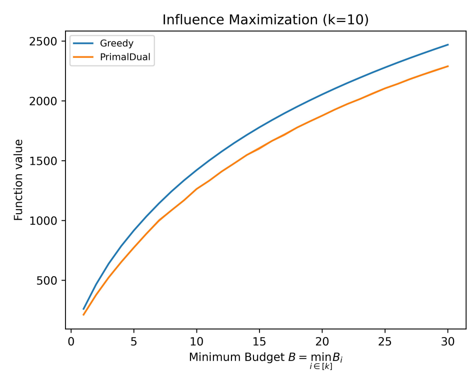
Therefore

$$\frac{f(S)}{f(S^*)} \geq \left(1-\left(1+\frac{D}{B}\right)^{-B}\right)\frac{1}{D+1}$$

where $B = \min_{i\in[k]} B_i$ is the minimum budget. $\square$

C. Additional Experimental Results

We performed the influence maximization experiment described in Section 5 with varying values of k and obtained similar results. Figure 3 reports the experimental results for influence maximization with $k = 10$ topics.



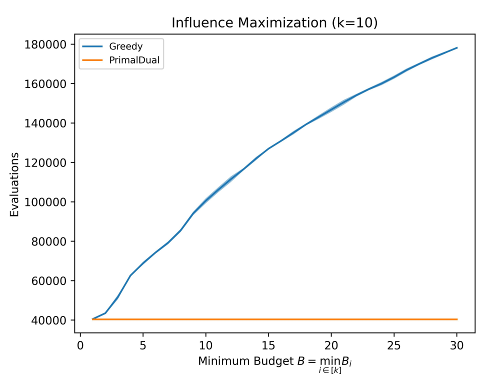


Figure 3. Experimental results for influence maximization with $k = 10$ topics. We report the mean and standard deviation over 5 runs. Greedy is the offline Greedy algorithm of (Ohsaka & Yoshida, 2015) implemented using lazy evaluations. PrimalDual is our Algorithm 1.