Maximizing Bisubmodular and k-Submodular Functions

Justin Ward*
Department of Computer Science, University of Warwick, UK
J.D.Ward@dcs.warwick.ac.uk

Stanislav Živný†
Department of Computer Science, University of Oxford, UK
standa@cs.ox.ac.uk

Abstract

Submodular functions play a key role in combinatorial optimization and in the study of valued constraint satisfaction problems. Recently, there has been interest in the class of bisubmodular functions, which assign values to disjoint pairs of sets. Like submodular functions, bisubmodular functions can be minimized exactly in polynomial time and exhibit the property of diminishing returns common to many problems in operations research. Recently, the class of k-submodular functions has been proposed. These functions generalize the notion of submodularity to k-tuples of sets, with submodular and bisubmodular functions corresponding to $k = 1$ and 2, respectively.

In this paper, we consider the problem of maximizing bisubmodular and, more generally, k-submodular functions in the value oracle model. We provide the first approximation guarantees for maximizing a general bisubmodular or k-submodular function. We give an analysis of the naive random algorithm as well as a randomized greedy algorithm inspired by the recent randomized greedy algorithm of Buchbinder et al. [FOCS'12] for unconstrained submodular maximization. We show that this algorithm approximates any k-submodular function to a factor of $1/(1 + \sqrt{k/2})$.

In the case of bisubmodular functions, our randomized greedy algorithm gives an approximation guarantee of $1/2$. We show that, as in the case of submodular functions, this result is the best possible in both the value query model, and under the assumption that $NP \neq RP$. Our analysis provides further intuition for the algorithm of Buchbinder et al. [FOCS'12] in the submodular case. Additionally, we show that the naive random algorithm gives a $1/4$-approximation for bisubmodular functions, corresponding again to known performance guarantees for submodular functions. Thus, bisubmodular functions exhibit approximability identical to submodular functions in all of the algorithmic contexts we consider.

*J.W. was supported by EPSRC grants EP/J021814/1 and EP/D063191/1.
†S.Ž. was supported by a Royal Society University Research Fellowship.

1 Introduction

Given a finite nonempty set U, a set function $f : 2^U \to \mathbb{R}_+$ defined on subsets of U is called *submodular* if for all $S, T \subseteq U$,

$$f(S) + f(T) \ \geq \ f(S \cap T) + f(S \cup T).$$

Submodular functions are a key concept in operations research and combinatorial optimization [29, 28, 38, 34, 10, 24, 19]. Examples of submodular functions include cut capacity functions, matroid rank functions, and entropy functions. Submodular functions are often considered to be a discrete analogue of convex functions [26].

Both minimizing and maximizing submodular functions, possibly under some additional conditions, have been considered extensively in the literature. Submodular function maximization is easily shown to be NP-hard [34] since it generalizes many standard NP-hard problems such as the maximum cut problem [12, 9]. In contrast, the problem of minimizing a submodular function can be solved efficiently with only polynomially many evaluations of the function [19] either by using the ellipsoid algorithm [13, 14], or by using one of several combinatorial algorithms that have been obtained in the last decade [33, 20, 17, 18, 30, 22].

Following a question by Lovász [26], a generalization of submodularity to biset functions has been introduced. Given a finite nonempty set U, a function $f : 3^U \to \mathbb{R}_+$ defined on pairs of disjoint subsets of U is called *bisubmodular* if for all pairs (S_1, S_2) and (T_1, T_2) of disjoint subsets of U,

$$f(S_1, S_2) + f(T_1, T_2) \geq f((S_1, S_2) \sqcap (T_1, T_2)) + f((S_1, S_2) \sqcup (T_1, T_2)),$$

where we define

$$(S_1, S_2) \sqcap (T_1, T_2) = (S_1 \cap T_1, S_2 \cap T_2),$$

and

$$(S_1, S_2) \sqcup (T_1, T_2) = ((S_1 \cup T_1) \setminus (S_2 \cup T_2), (S_2 \cup T_2) \setminus (S_1 \cup T_1)).$$

Examples of bisubmodular functions include rank functions of delta-matroids [4, 6]. Bisubmodularity also arises in bicooperative games [3] as well as variants of sensor placement problems and coupled feature selection problems [35]. The minimization problem for bisubmodular functions using the ellipsoid method was solved in [32]. Moreover, combinatorial [11] and strongly combinatorial [27] algorithms are known for minimizing bisubmodular functions.

In this paper, we study the natural generalization of submodular and bisubmodular functions: given a natural number $k \geq 1$ and a finite nonempty set U, a function $f : (k+1)^U \to \mathbb{R}_+$ defined on k-tuples of pairwise disjoint subsets of U is called *k-submodular* if for all k-tuples $S = (S_1, \ldots, S_k)$ and $T = (T_1, \ldots, T_k)$ of pairwise disjoint subsets of U,

$$f(S) + f(T) \ \geq \ f(S \sqcap T) + f(S \sqcup T),$$

where we define

$$S \sqcap T = (S_1 \cap T_1, \ldots, S_k \cap T_k),$$

and

$$S \sqcup T = ((S_1 \cup T_1) \setminus \bigcup_{i \in \{2,\ldots,k\}} (S_i \cup T_i), \ldots, (S_k \cup T_k) \setminus \bigcup_{i \in \{1,\ldots,k-1\}} (S_i \cup T_i)).$$

Using this notation, 1-submodular functions are submodular functions and 2-submodular functions are bisubmodular functions. (We note that Ando has used the term k-submodular to study different class of functions [1].)

Related work The name of k-submodular functions was first introduced in [15] but the concept has been known since at least [7]. k-submodularity is a special case of strong tree submodularity [23] with the tree being a star on $k+1$ vertices.

To the best of our knowledge, it is not known whether the ellipsoid method can be employed for minimizing k-submodular functions for $k \geq 3$ (some partial results can be found in [15]), let alone whether there is a (fully) combinatorial algorithm for minimizing k-submodular functions for $k \geq 3$. However, it has recently been shown that explicitly given k-submodular functions can be minimized in polynomial time [36].

Some results on maximizing special cases of bisubmodular functions have appeared in Singh, Guillory, and Bilmes [35], who showed that simple bisubmodular function can be represented as a matroid constraint and a single submodular function, thus enabling the use of existing algorithms in some special cases. Unfortunately they show that for some bisubmodular functions, this approach requires that the submodular function take negative values and so this approach does not work in general. (We note that our definition of bisubmodularity corresponds to directed bisubmodularity in [35].)

A different generalization of bisubmodularity, called skew bisubmodularity, has proved important in classifying finite-valued CSPs on domains with 3 elements [16]; this result was then generalized by a complexity classification of finite-valued CSPs on domains of arbitrary size [37]. Explicitly given skew bisubmodular functions can be minimized efficiently by results of Thapper and Živný [36]. The general question of whether all bisubmodular, and, more generally, k-submodular functions can be approximately maximized was left open.

Contributions Following the question by Lovász [26] of whether there are generalizations of submodularity that preserve some nice properties such as efficient optimization algorithms and a more recent, similar question by Vondrák,[1] we consider the class of k-submodular functions.

Specifically, we consider the problem of *maximizing* bisubmodular and, more generally, k-submodular functions in the *value oracle model*. We provide the

[1] In the SODA 2013 invited talk "Submodular Functions and Their Applications", Vondrák asked about generalizations of submodularity defined via polymorphisms (submodularity corresponds to min and max polymorphisms).

first approximation guarantees for maximizing a general bisubmodular or k-submodular function. In Section 3 we show that the naive random algorithm that simply returns a random partition of the ground set U is $1/4$-approximation for maximizing any bisubmodular function and a $1/k$-approximation for maximizing a k-submodular function with $k \geq 3$.

In Section 4, we develop a randomized greedy algorithm for k-submodular maximization inspired by the algorithm of Buchbinder et al. [5] for unconstrained submodular maximization. We show that this algorithm approximates any k-submodular function to a factor of $1/(1 + \sqrt{k/2})$.

Finally, in Section 5 we relate our results on bisubmodular functions and existing results on submodular functions via a known embedding of submodular functions into bisubmodular functions. Using this embedding we can translate inapproximability results for submodular function into analogous results for bisubmodular functions. Moreover, we show that the algorithm of Buchbinder et al. [5] may be viewed as a special case of our algorithm applied to this embedding.

Our results on bisubmodular functions and the simple $1/k$-approximation algorithm for maximizing k-submodular functions have been obtained independently by Iwata, Tanigawa, and Yoshida [21].

2 Preliminaries

We denote by $\mathbb{R}_+$ the set of all non-negative real numbers. Let U be a ground set containing n elements and $k \geq 1$ be a fixed integer. We consider functions that assign a value in $\mathbb{R}_+$ to each partial assignment of the values $\{1, \ldots, k\}$ to the elements of U. We can represent each such (partial) assignments as vectors $\mathbf{x}$ in $\{0, \ldots, k\}^U$, where we have $x_e = 0$ if element e in U is not assigned any value in $\{1, \ldots, k\}$, and otherwise have x_e equal to the value assigned to e. It will be useful to consider the partial assignment obtained from another (possibly partial) assignment $\mathbf{x}$ by "forgetting" the values assigned to all elements except for some specified set $S \subseteq U$. We represent this as the vector $\mathbf{x}\big|_S$ whose coordinates are given by $\left(\mathbf{x}\big|_S\right)_e = x_e$, for all $e \in S$ and $\left(\mathbf{x}\big|_S\right)_e = 0$ for all $e \in U \setminus S$. Note that $\mathbf{x}\big|_S$ is similar to the projection of $\mathbf{x}$ onto S, but we here require that all coordinates $e \notin S$ be set to 0, while the standard notion of projection would remove these coordinates from the resulting vector. In particular, this means that $\mathbf{x}\big|_S$ and $\mathbf{x}$ both have n coordinates.

In order to relate our results to existing work on submodular functions, we shall also use terminology from set functions. In this setting, we consider functions that assign a value to each tuple of disjoint sets $S = (S_1, \ldots, S_k)$, where $S_i \subseteq U$ and $S_i \cap S_j = \emptyset$ for all $i \neq j$. It is straightforward to check that the two notions are equivalent by having $e \in S_i$ if and only if $x_e = i$. Then, we have $x_e = 0$ if and only if e does not appear in any of the sets $S_1, \ldots, S_k$. With some abuse of notation, we shall write $e \notin S$ for an element e in U and $S = (S_1, \ldots, S_k)$ if $e \notin \cup_{1 \leq i \leq k} S_i$.

The solution space over which we optimize our functions is thus the set of

partitions of some subset $U' \subseteq U$ into k disjoint sets, where in our vector notation U' is equivalent to the set of coordinates in $\mathbf{x}$ that are non-zero. We shall refer to those solutions that partition the entire ground set U (or, alternatively, whose assignment vectors have no zero coordinates) as *partitions*, and call partitions of some $U' \subseteq U$ *partial solutions*[2] over U, to emphasize that they may not necessarily assign every element in U to a set.[3]

In this paper, we study the following particular class of functions mapping partial solutions over U to values in $\mathbb{R}_+$. Consider the operations $\min_0$ and $\max_0$ given by

$$\min_0(s,t) \stackrel{\text{def}}{=} \begin{cases} 0, & s \neq 0, t \neq 0, s \neq t \\ \min(s,t), & \text{otherwise} \end{cases}$$

and

$$\max_0(s,t) \stackrel{\text{def}}{=} \begin{cases} 0, & s \neq 0, t \neq 0, s \neq t \\ \max(s,t), & \text{otherwise,} \end{cases}$$

where $\min(s,t)$ (respectively, $\max(s,t)$) returns the smaller (respectively, the larger) of s and t with respect to the usual order on the integers.

For vectors $\mathbf{s}$ and $\mathbf{t}$ in $\{0,\ldots,k\}^U$ we let $\min_0(\mathbf{s},\mathbf{t})$ (respectively, $\max_0(\mathbf{s},\mathbf{t})$) denote the vector obtained from applying $\min_0$ (respectively, $\max_0$) to $\mathbf{s}$ and $\mathbf{t}$ coordinate-wise. Using these operations we can define the general class of k-submodular functions.

Definition 2.1. *Given a natural number $k \geq 1$ and a finite nonempty set U, a function $f : \{0,\ldots,k\}^U \to \mathbb{R}_+$ is called* k-submodular *if for all $\mathbf{s}$ and $\mathbf{t}$ in $\{0,\ldots,k\}^U$,*

$$f(\mathbf{s}) + f(\mathbf{t}) \;\geq\; f(\min_0(\mathbf{s},\mathbf{t})) + f(\max_0(\mathbf{s},\mathbf{t})). \tag{1}$$

Note that if $\mathbf{s}$ and $\mathbf{t}$ are both *partitions*, then we have $\min_0(\mathbf{s},\mathbf{t}) = \max_0(\mathbf{s},\mathbf{t}) = \mathrm{id}_0(\mathbf{s},\mathbf{t})$ where the operation id_0 on each coordinate of $\mathbf{s}$ and $\mathbf{t}$ is given by $\mathrm{id}_0(s,t) = s = t$ if $s = t$, and $\mathrm{id}_0(s,t) = 0$ otherwise. Thus, if f is a k-submodular function, we have

$$f(\mathbf{s}) + f(\mathbf{t}) \;\geq\; 2f(\mathrm{id}_0(\mathbf{s},\mathbf{t})) \tag{2}$$

for any partitions $\mathbf{s}$ and $\mathbf{t}$.

Example 2.2. The well-known Max-Cut problem demonstrates that maximizing (1-)submodular functions is NP-hard even if the objective function is given explicitly [12]. We show that the same hardness result holds for any $k \geq 1$.

[2]It is not a priori clear why a partial solution could not be an optimal solution but we shall see, in Corollary 2.4, that when maximizing k-submodular functions, where $k \geq 2$, we can focus only on partitions.

[3]Note that every partition is thus also a partial solution, but not vice versa.

Consider the following function $f_=$ defined on $\{0,\dots,k\}$ by $f_=(x,y) = 0$ if $x = y$ and $f_=(x,y) = 1$ if $x \neq y$. It is easy to check that $f_=$ is k-submodular. Moreover, given a graph (V,E) with $V = \{v_1,\dots,v_n\}$, maximizing the function $f(v_1,\dots,v_n) = \sum_{\{v_i,v_j\}\in E} f_=(v_i,v_j)$ amounts to solving the Max-k-Cut problem, which is NP-hard [31].

While quite concise, Definition 2.1 gives little intuition in the traditional setting of set functions. We now attempt to provide some such intuition. Consider two partial solutions $S = (S_1,\dots,S_k)$ and $T = (T_1,\dots,T_k)$ and let $\mathbf{s}$ and $\mathbf{t}$ be the vectors in $\{0,\dots,k\}^U$ representing S and T, respectively. Consider some element $i \in U$. We have $\min_0(s_i,t_i) = j \neq 0$ precisely when $s_i = t_i = j \neq 0$. Thus, the vector $\min_0(\mathbf{s},\mathbf{t})$ in Definition 2.1 corresponds exactly to the coordinate-wise intersection $(S_1 \cap T_1,\dots,S_k \cap T_k)$ of S and T. Moreover, $\max_0(s_i,t_i) = j \neq 0$ precisely when either $s_i = t_i \neq 0$ or when one of s_i, t_i is $j \neq 0$ and the other is 0. Thus, the vector $\max_0(\mathbf{s},\mathbf{t})$ corresponds exactly to the coordinate-wise union of S and T after we have removed any element i occurring in two different sets from both of them. That is, if we set $X_{-i} = \bigcup_{j\neq i}(S_j \cup T_j)$, then $\max_0(\mathbf{s},\mathbf{t})$ corresponds to $((S_1 \cup T_1) \setminus X_{-1},\dots,(S_k \cup T_k) \setminus X_{-k})$. Note that the removal of X_{-i} from the ith union ensures that no element occurs in two different sets in the resulting partial solution.

The following equivalences, first observed by Cohen et al. [7], allow us to relate k-submodular functions to existing families of set functions. When $k = 2$, Definition 2.1 requires that

$$\begin{aligned} &f(S_1,S_2) + f(T_1,T_2) \\ &\quad \geq f(S_1 \cap T_1, S_2 \cap T_2) + f((S_1 \cup T_1) \setminus (S_2 \cup T_2), (S_2 \cup T_2) \setminus (S_1 \cup T_2)), \end{aligned}$$

which agrees exactly with the definition of bisubmodular functions given in [10]. When $k = 1$, there is only a single set in each partial solution, and so $X_{-1} = \emptyset$. Thus, Definition 2.1 requires that

$$f(S_1) + f(T_1) \;\geq\; f(S_1 \cap T_1) + f(S_1 \cup T_1),$$

which agrees exactly with the standard definition of submodular functions [29].

Let $\mathbf{x}$ be a *partition* of the ground set U. Given a k-submodular function f, we call set function $h : 2^U \to \mathbb{R}_+$ defined for any $S \subseteq U$ by

$$h(S) \;\stackrel{\text{def}}{=}\; f(\mathbf{x}\big|_S)$$

the function *induced by* $\mathbf{x}$ and f. In the language of set functions, the function h is obtained by first assigning each element e in U to a single set X_i (where $i = x_e$). Then, $h(S)$ is simply the value of $f(S \cap X_1,\dots,S \cap X_k)$.

For a function $f : \{0,\dots,k\}^U \to \mathbb{R}_+$, a partial solution $S = (S_1,\dots,S_k)$, an element $e \in U$, and a value $i \in \{1,\dots,k\}$, we define the marginal value $f_{i,e}(S)$ by

$$f_{i,e}(S) \;\stackrel{\text{def}}{=}\; f(S_1,\dots,S_{i-1},S_i+e,S_{i+1},\dots,S_k) - f(S_1,\dots,S_k),$$

for any partial solution $S = (S_1, \ldots, S_k)$ such that $e \notin S$, where $S_i + e$ is a shorthand for $S_i \cup \{e\}$.

The following theorem shows that both the induced functions and marginal values of k-submodular functions obey certain useful properties.

Theorem 2.3. *Let $f : \{0, \ldots, k\}^U \to \mathbb{R}_+$ be a k-submodular function, where $k \geq 2$. Then,*

1. *For any partition $\mathbf{x}$, the function h induced by $\mathbf{x}$ and f is submodular.*
2. *For any element e and any partial solution $S = (S_1, \ldots, S_k)$ such that $e \notin S$,*
$$\sum_{i=1}^{k} f_{i,e}(S) \;\geq\; 0.$$

Proof. To prove the first property, let f, $\mathbf{x} = (X_1, \ldots, X_k)$, and h be as stated. For any $S, T \subseteq U$,

$$\begin{aligned}
&h(S) + h(T) \\
&\overset{(1)}{=} f(S \cap X_1, \ldots, S \cap X_k) + f(T \cap X_1, \ldots, T \cap X_k) \\
&\overset{(2)}{\geq} f((S \cap X_1) \cap (T \cap X_1), \ldots, (S \cap X_k) \cap (T \cap X_k)) \\
&\quad + f((S \cap X_1) \cup (T \cap X_1), \ldots, (S \cap X_k) \cup (T \cap X_k)) \\
&\overset{(3)}{=} f((S \cap T) \cap X_1, \ldots, (S \cap T) \cap X_k) \\
&\quad + f((S \cup T) \cap X_1, \ldots, (S \cup T) \cap X_k) \\
&\overset{(4)}{=} h(S \cap T) + h(S \cup T),
\end{aligned}$$

where (1) and (4) follow from the definition of h, (2) follows from the definition of k-submodularity and the fact that $(S \cap X_i) \cap (T \cap X_j) = \emptyset$ for all $i \neq j$ since $\mathbf{x}$ is a partition, and (3) follows from basic properties of union and intersection and the fact that X_i's are disjoint. Thus h is submodular.

In order to prove the second property, we prove the following:

$$\sum_{i=1}^{k} f(S_1, \ldots S_{i-1}, S_i + e, S_{i+1} \ldots, S_k) \geq \; k \cdot f(S_1, \ldots, S_k). \tag{3}$$

Let $S = (S_1, \ldots, S_k)$ be a partial solution and $\mathbf{s}$ the corresponding vector in $\{0, \ldots, k\}^U$. For any fixed $1 \leq i \neq j \leq k$, consider partial solutions $(S_1, \ldots, S_{i-1}, S_i + e, S_{i+1}, \ldots, S_k)$ and $(S_1, \ldots, S_{j-1}, S_j + e, S_{j+1}, \ldots, S_k)$ and let $\mathbf{s}_1$ and $\mathbf{s}_2$ be the corresponding vectors in $\{0, \ldots, k\}^U$. Since $S_p \cap S_r = \emptyset$ for all $1 \leq p \neq r \leq k$, we get $\min_0(\mathbf{s}_1, \mathbf{s}_2) = \max_0(\mathbf{s}_1, \mathbf{s}_2) = \mathbf{s}$. Thus, by using the k-submodularity inequality (1) for all pairs of $1 \leq i \neq j \leq k$, we get

$$\sum_{i=1}^{k} (k-1) \cdot f(S_1, \ldots, S_{i-1}, S_i + e, S_2, S_{i+1}, \ldots, S_k) \geq \binom{k}{2} \cdot 2 \cdot f(S_1, \ldots, S_k),$$

which, after dividing both sides by $k-1$, simplifies to inequality (3). □

In the case of $k = 2$, Ando, Fujishige, and Naito [2] have shown that the 2 properties given in Theorem 2.3 in fact give an exact characterization of the class of bisubmodular functions.

Corollary 2.4 (of Theorem 2.3)**.** *Any partial solution $S \in \{0, \ldots, k\}^U$ can be extended to a partition of U in $\{1, \ldots, k\}^U$ without any loss in the value of f.*

Proof. If S is not a partition of U then there is some e in U so that $e \notin S$ and there is at least one index i so that $f(S_1, \ldots, S_{i-1}, S_i + e, S_{i+1}, \ldots, S_k) - f(S_1, \ldots, S_k) \geq 0$ as otherwise, by summing over all $1 \leq i \leq k$, we would get a contradiction with Theorem 2.3 (2). Thus we can add e to S_i and inductively get a maximizer of f that is a partition of U. □

It is easy to show that k-submodular functions have the property of diminishing returns. (For 1-submodular functions this is an equivalent definition of submodularity [29].[4])

Proposition 2.5. *Let $f : (k+1)^U \to \mathbb{R}_+$ be a k-submodular function and let $S = (S_1, \ldots, S_k)$ and $T = (T_1, \ldots, T_k)$ be two partial solutions such that $S_i \subseteq T_i$ for all $1 \leq i \leq k$. Then for every $1 \leq i \leq k$ and $e \notin T$, $f_{i,e}(S) \geq f_{i,e}(T)$.*

Proof. Without loss of generality, we assume that $i = 1$. From the definition of k-submodularity we get $f(S_1 + e, S_2, \ldots, S_k) + f(T_1, T_2, \ldots, T_k) \geq f(S_1, \ldots, S_k) + f(T_1 + e, T_2, \ldots, T_k)$. □

Finally, we restate the following result from Lee, Sviridenko, and Vondrák [25], which shall be useful in our analysis:

Lemma 2.6 ([25, Lemma 1.1])**.** *Let f be a non-negative submodular function on U. Let $S, C \subseteq U$ and let $\{T_\ell\}_{\ell=1}^t$ be a collection of subsets of $C \setminus S$ such that each element of $C \setminus S$ appears in exactly p of these subsets. Then*

$$\sum_{\ell=1}^{t} [f(S \cup T_\ell) - f(S)] \;\geq\; p[f(S \cup C) - f(S)].$$

In fact, the following weaker statement will be sufficient for our purposes:

Corollary 2.7 (of Lemma 2.6)**.** *Let f be a non-negative submodular function on U. Let $S, C \subseteq U$ and let $\{T_\ell\}_{\ell=1}^t$ be a collection of subsets of $C \setminus S$ such that each element of $C \setminus S$ appears in exactly p of these subsets. Then*

$$\sum_{\ell=1}^{t} f(S \cup T_\ell) \;\geq\; pf(S \cup C).$$

[4] That is, $f : 2^U \to \mathbb{R}_+$ is 1-submodular if and only if $f_{i,e}(S) \geq f_{i,e}(T)$ for every $S \subseteq T$ and $e \notin T$.

Proof. Add $\sum_{\ell=1}^{t} f(S)$ to each side of the inequality in Lemma 2.6. This gives

$$\sum_{\ell=1}^{t} f(S \cup T_\ell) \geq p \cdot f(S \cup C) - p \cdot f(S) + \sum_{\ell=1}^{t} f(S)$$
$$= p \cdot f(S \cup C) + (t-p) \cdot f(S) \geq p \cdot f(S \cup C),$$

since $p \leq t$. □

3 The Naive Random Algorithm

In this section, we consider the expected performance of the naive randomized algorithm for maximizing a k-submodular function $f : \{0, \ldots, k\}^U \to \mathbb{R}_+$. Corollary 2.4 shows that any partial solution $S \in \{0, \ldots, k\}^U$ can be extended to a full partition of U in $\{1, \ldots, k\}^U$ without any loss in the value of f. Thus, we consider a random algorithm that simply selects a partition of the ground set from $\{1, \ldots, k\}^U$ uniformly at random. The proof of the following result can be found in Appendix A (case k=1 is known [9]).

Theorem 3.1. *The naive random algorithm gives a $1/4$-approximation for k-submodular functions with $k \leq 2$ and a $1/k$-approximation for k-submodular functions with $k \geq 3$.*

Example 3.2. As a tight example for $k = 2$, we consider the function $f_{[u=1,v=2]}$ on the ground set $\{u, v\}$, given by:

$$f_{[u=1,v=2]}(\mathbf{x}) = \begin{cases} 1, & x_u = 1, x_v = 2 \\ \frac{1}{2}, & x_u = 1, x_v = 0 \text{ or } x_u = 0, x_v = 2 \\ 0, & \text{otherwise} \end{cases}.$$

It is easily verified that this function is indeed bisubmodular. Moreover, the probability that a random partition will set $x_u = 1$, and $x_v = 2$ is $\frac{1}{4}$, and the function has value 0 for all other partitions. Thus, $\mathbb{E}[f_{[u=1,v=2]}(\mathbf{x})] = \frac{1}{4}$, whereas the maximum value is 1. We can generalize this to a ground sets $U = U' \cup V'$ of arbitrary size by setting $f(\mathbf{x}) = \sum_{u \in U'} \sum_{v \in V'} f_{[u=1,v=2]}(\mathbf{x})$.

Example 3.3. As a tight example for $k \geq 3$, we consider the k-submodular function $f_{[e=1]}$ on the singleton ground set $\{e\}$ given by $f_{[e=1]}(\mathbf{x}) = 1$ if $x_e = 1$ and $f_{[e=1]}(\mathbf{x}) = 0$ otherwise. It is easy to verify that this function is indeed k-submodular. Moreover, a random partition sets $s_x = 1$ with probability only $\frac{1}{k}$, and so $\mathbb{E}[f_{[e=1]}(\mathbf{x})] = \frac{1}{k}$. Note that the example is easily generalized to ground sets of arbitrary size by defining $f(\mathbf{x}) = \sum_{e \in U} f_{[e=1]}(\mathbf{x})$.

4 A Randomized Greedy Algorithm

Next, we consider the performance of a simple greedy algorithm inspired by the algorithm of Buchbinder et al. [5] for unconstrained submodular maximization.

Our algorithm begins with the initial solution $(\emptyset, \ldots, \emptyset)$ and considers elements of the ground set U in some arbitrary order, permanently adding each element to one of the sets S_i in S, based on the increase that this gives in f. Specifically, the algorithm randomly adds an element e to the set S_i with probability proportional to the resulting marginal increase $f_{i,e}(S)$ in f with respect to the current solution S. If $f_{i,e}(S) < 0$, we add e to S_i with probability 0. Note that Theorem 2.3 shows that we cannot have $f_{i,e}(S) < 0$ for all i, but it may be the case that $f_{i,e}(S) = 0$ for all i. In this case, we add e to the set S_1.

Randomized Greedy

for $i = 1$ **to** k **do**
  $S_i \leftarrow \emptyset$
end for
for each $e \in U$ **do**
  for $i = 1$ **to** k **do**
    $x_i \leftarrow \max(0, f_{i,e}(S))$
  end for
  $\beta = \sum_{i=1}^{k} x_i$
  if $\beta \neq 0$ **then**
    Let $i \in \{1, \ldots, k\}$ be chosen randomly,
      with $\Pr[i = j] = \frac{x_j}{\beta}$ for all $j \in \{1, \ldots, k\}$.
    $S_i \leftarrow S_i + e$
  else
    $S_1 \leftarrow S_1 + e$
  end if
end for

Theorem 4.1. *Let S be the solution produced by the randomized greedy algorithm on some instance $f : \{0, \ldots, k\}^U \to \mathbb{R}_+$ of k-submodular maximization with $k \geq 2$, and let OPT be the optimal solution for this instance. Then,*

$$\left(1 + \sqrt{\frac{k}{2}}\right) \mathbb{E}[f(S)] \;\geq\; f(OPT).$$

Proof. Our analysis considers 2 sequences of n solutions. First let, $S^{(i)} = (S_1^{(i)}, \ldots, S_k^{(i)})$ be the algorithm's solution after i elements have been considered, and let $U^{(i)} = \bigcup_{j=1}^{k} S_j^{(i)}$ be the set of elements that have been considered by the algorithm at this point. Then, we define the solution $O^{(i)} = (O_1^{(i)}, \ldots, O_k^{(i)})$, where $O_j^{(i)} = (OPT_j \setminus U^{(i)}) \cup S_j^{(i)}$. Intuitively, $O^{(i)}$ is the solution that agrees with $S^{(i)}$ on the placement of the elements considered by the greedy algorithm in its first i phases and agrees with OPT on the placement of all other elements. Note that in particular we have $O^{(0)} = OPT$ and $O^{(n)} = S$. In Lemma 4.2, we bound the expected decrease $\mathbb{E}[f(O^{(i)}) - f(O^{(i+1)})]$ relative to the increase

$\mathbb{E}[f(S^{(i+1)}) - f(S^{(i)})]$. Specifically, we show that

$$\mathbb{E}[f(O^{(i)}) - f(O^{(i+1)})] \;\leq\; \sqrt{\frac{k}{2}}\,\mathbb{E}[f(S^{(i+1)}) - f(S^{(i)})] \tag{4}$$

for all i. Summing the resulting inequalities for $i = 0$ to n, we then obtain

$$\sum_{i=0}^{n} \mathbb{E}[f(O^{(i)}) - f(O^{(i+1)})] \leq \sqrt{\frac{k}{2}} \sum_{i=0}^{n} \mathbb{E}[f(S^{(i+1)}) - f(S^{(i)})],$$

which simplifies to

$$\mathbb{E}[f(O^{(0)})] - \mathbb{E}[f(O^{(n)})] \leq \sqrt{\frac{k}{2}} \left(\mathbb{E}[f(S^{(n)})] - \mathbb{E}[f(S^{(0)})]\right) \leq \sqrt{\frac{k}{2}}\,\mathbb{E}[f(S^{(n)})].$$

The theorem then follows from the definitions $O^{(0)} = OPT$, and $S^{(n)} = O^{(n)} = S$. □

We now show that inequality (4) must hold.

Lemma 4.2. *For any* $0 \leq i \leq n$,

$$\mathbb{E}[f(O^{(i)}) - f(O^{(i+1)})] \leq \sqrt{\frac{k}{2}}\,\mathbb{E}[f(S^{(i+1)}) - f(S^{(i)})].$$

Proof. Let e be the element of U considered by the randomized greedy algorithm in the $(i+1)$th phase, and let $U^{(i)}$ and $O^{(i)}$ be defined as in the proof of Theorem 4.1. We condition on an arbitrary, fixed value for both $U^{(i)}$, $O^{(i)}$, and consider the expectation over choices the algorithm makes for e. Because our result will hold for an arbitrary $U^{(i)}$ or $O^{(i)}$ it then extends to the expectation over the first i choices made by the algorithm.

We define the solution $A = (A_1, \ldots, A_k)$, where $A_j = O_j^{(i)} - e$, and let $a_j = f_{j,e}(A)$ for $1 \leq j \leq k$. As in the definition of the greedy algorithm in Algorithm 4, we let $x_j = \max(0, f_{j,e}(S^{(i)}))$ for each $1 \leq j \leq k$. Then, we note that for every $1 \leq j \leq k$,

$$A_j \;=\; O_j^{(i)} - e \;=\; ((OPT_j \setminus U^{(i)}) \cup S_j^{(i)}) - e \;\supseteq\; S_j^{(i)} - e \;=\; S_j^{(i)},$$

where $S_j^{(i)} - e = S_j^{(i)}$ as e is considered in the $(i+1)$th phase. Thus, from Proposition 2.5 we have $a_j = f_{j,e}(A) \leq f_{j,e}(S^{(i)}) \leq x_j$ for all $1 \leq j \leq k$ and also, from Theorem 2.3, $\sum_{j=1}^{k} a_j \geq 0$.

Finally, we have $x_j \geq 0$ for each $1 \leq j \leq k$, by definition.

Now, let suppose that $e \in OPT_o$ for some $1 \leq o \leq k$, and that the greedy algorithm places $e \in S_j^{(i)}$ for some $1 \leq j \leq k$. Then, $O^{(i)}$ and $O^{(i+1)}$ are identical except that $O^{(i)}$ places e in the oth set, while $O^{(i+1)}$ places e in the

jth set. Thus, we have $f(O^{(i)}) = f(A) + f_{o,e}(A)$ and $f(O^{(i+1)}) = f(A) + f_{j,e}(A)$, and so

$$f(O^{(i)}) - f(O^{(i+1)}) = f_{o,e}(A) - f_{j,e}(A) = a_o - a_j,$$

and

$$f(S^{(i+1)}) - f(S^{(i)}) = f_{j,e}(S^{(i)}) = x_j.$$

For any given j, the probability that the greedy algorithm makes such a choice is precisely x_j/β, and so

$$\mathbb{E}[f(S^{(i+1)}) - f(S^{(i)})] = \frac{1}{\beta}\sum_j x_j^2,$$

and

$$\begin{aligned}\mathbb{E}[f(O^{(i)}) - f(O^{(i+1)})] &= \frac{1}{\beta}\sum_j x_j(a_o - a_j) \\ &= \frac{1}{\beta}\sum_{j\neq o} x_j(a_o - a_j).\end{aligned}$$

In order to prove the lemma it is thus sufficient to show that

$$\sum_{j\neq o} x_j(a_o - a_j) \leq \sqrt{\frac{k}{2}}\sum_j x_j^2. \tag{5}$$

For any value of $x_1, \ldots, x_k$, the left hand side of (5) is upper bounded by the optimal value of the linear program

$$\begin{aligned}\text{maximize} \quad & \sum_{j\neq o} x_j(a_o - a_j) \\ \text{subject to} \quad & a_j \leq x_j \qquad 1 \leq j \leq k \\ & \sum_j a_j \geq 0\end{aligned}$$

This is a bounded, feasible linear program in k variables a_j with $k+1$ linearly independent constraints. Let a^* be an optimal solution to this program. Then, basic linear programming theory allows us to suppose without loss of generality that a^* is in fact a basic feasible solution and hence has k tight constraints. We first note that by increasing a_o we cannot violate the final constraint and can only increase the objective, and so we may assume that $a_o = x_o$. Of the remaining k constraints, $k-1$ must be tight, of which $k-2$ must be of the first type. Hence, for all j except at most 1 value $\ell \neq o$, we in fact have $a_j = x_j$. This accounts for $k-1$ total tight constraints. The final tight constraint must imply either $a_\ell = x_\ell$ or $\sum_j a_j = 0$. Because $a_j = x_j$ for all $j \neq \ell$, the latter is equivalent to $a_\ell = -\sum_{j\neq\ell} x_j$. Moreover, because $x_j \geq 0$ for all j, setting $a_\ell = -\sum_{j\neq\ell} x_j$ always gives an objective value at least as large as setting

$a_\ell = x_\ell$. Thus, we can characterize the optimal solution to this linear program by $a_j^* = x_j$ for all $j \neq \ell$, and $a_\ell^* = -\sum_{j \neq \ell} x_j$, where ℓ is some value distinct from o.

Returning to (5), we have

$$\begin{aligned}\sum_{j \neq o} x_j(a_o - a_j) &\leq \sum_{j \neq o} x_j(a_o^* - a_j^*) \\ &= \sum_{j \neq o,\ell} x_j(x_o - x_j) + x_\ell \left(x_o + \sum_{j \neq \ell} x_j \right) \\ &= 2x_\ell x_o + \sum_{j \neq o,\ell} [x_\ell x_j + x_o x_j - x_j^2],\end{aligned}$$

for any $x_1, \ldots, x_k \geq 0$. In order to prove (5) it then suffices to show that

$$0 \leq \alpha \sum_j x_j^2 - 2x_\ell x_o - \sum_{j \neq o,\ell} [x_\ell x_j + x_o x_j - x_j^2], \tag{6}$$

where $\alpha = \sqrt{\frac{k}{2}}$. This follows directly from the fact that the right hand side of (6) can be written as the following sum of squares:

$$\begin{aligned}(x_\ell - x_o)^2 &+ \sum_{j \neq o,\ell} \left(\sqrt{\frac{\alpha - 1}{k - 2}} x_\ell - \sqrt{\frac{\alpha + 1}{2}} x_j \right)^2 \\ &+ \sum_{j \neq o,\ell} \left(\sqrt{\frac{\alpha - 1}{k - 2}} x_o - \sqrt{\frac{\alpha + 1}{2}} x_j \right)^2 .\end{aligned}$$

A verification of this can be found in Appendix C. □

A simpler analysis in Appendix B shows that a deterministic greedy algorithm gives a $(1 + k)$-approximation.

5 Conclusion

In the preceding sections we have considered the problem of maximizing k-submodular functions by both a random partition and a simple, randomized greedy algorithm. In the case of maximizing a bisubmodular function, we obtained the same approximation ratios as those already known in the submodular case: $1/4$ for the naive random solution [9] and $1/2$ via a randomized greedy approach [5]. We can make this correspondence more explicit by considering the following embedding of a submodular function into a bisubmodular function. Given a submodular function $g : 2^U \to \mathbb{R}_+$, we consider the function $f : 3^U \to \mathbb{R}_+$ defined by

$$f(S, T) \stackrel{\text{def}}{=} g(S) + g(U \setminus T) - g(U). \tag{7}$$

This embedding has been studied by Fujishige and Iwata, who show that the function f is bisubmodular and has the following property: if (S, T) is a minimizer (maximizer) of f then both S and $U \setminus T$ are minimizers (maximizers) of g [11]. Thus, exact 2-submodular function minimization (maximization) is a generalization of 1-submodular function minimization (maximization). We can in fact show a stronger result: that this embedding preserves approximability.

Suppose that some algorithm gives a α-approximation for bisubmodular maximization. Then, consider an arbitrary submodular function g and let f be the embedding of g defined as in (7). Let $OPT = (O_1, O_2)$ be a maximizer f, and suppose that the algorithm returns a solution $S = (S_1, S_2)$. Then, by Corollary 2.4 we can greedily extend S to a partition $S' = (S'_1, S'_2)$ of U. Similarly, we can assume without loss of generality that OPT is a partition of U. Then, we have $f(U \setminus S'_2) = f(S'_1)$ and $f(U \setminus O_2) = f(O_2)$, and thus

$$\begin{aligned}
g(S'_1) &= \frac{1}{2}\left(g(S'_1) + g(U \setminus S'_1)\right) \\
&= \frac{1}{2}\left(f(S'_1, S'_2) + g(U)\right) \\
&\geq \frac{1}{2}\left(\alpha f(O_1, O_2) + g(U)\right) \\
&= \frac{1}{2}\left(\alpha g(O_1) + \alpha g(U \setminus O_2) + (1-\alpha) g(U)\right) \\
&\geq \frac{1}{2}\left(\alpha g(O_1) + \alpha g(U \setminus O_2)\right) \\
&= \alpha g(O_1).
\end{aligned}$$

Since O_1 is a maximizer of g, the resulting algorithm is an α-approximation for maximizing g. Hence, the $1/2 + \epsilon$ inapproximability results of [9, 8] hold for bisubmodular maximization as well, in both the value oracle setting and under the assumption that $NP \neq RP$.

The embedding (7) also allows us to provide new intuition for the performance of the randomized greedy algorithm for submodular maximization considered by Buchbinder et al. [5]. This algorithm maintains 2 solutions, S_1 and S_2 which are initially $\emptyset$ and U. At each step, it considers an element e, and either adds e to S_1 or removes e from S_2, with probability proportional to the resulting increase in the submodular function in either case.

In comparison, we consider the case in which we embed a submodular function g into a bisubmodular function f using (7) and then run the greedy algorithm of Section 4 on f. Suppose at some step we have a current solution $T = (T_1, T_2)$ and we consider element e, and define $S_1 = T_1$ and $S_2 = U \setminus T_2$. The algorithm will add e to either T_1 or T_2 with probability proportional to the resulting increase in f. In the first case, this increase is precisely $g(T_1 + e) - g(T_1) = g(S_1 + e) - g(S_1)$, and adding e to T_1 corresponds to adding e to S_1. In the second case this increase is precisely $g(U \setminus T_2) - g(U \setminus (T_2 + e)) = g(S_2) - g(S_2 - e)$ and adding e to T_1 corresponds to removing e from S_1. Thus, the operation of the algorithm of Buchbinder et al. [5] may be viewed as a natural, straightforward

randomized greedy algorithm viewed through the lens of the embedding (7). Our analysis of bisubmodular functions can then be viewed as a generalization of their proof in the submodular case.

We do not know whether our analysis in Section 4 is tight for $k \geq 3$. More generally, we ask whether the symmetry gap technique from [39, 8] can be generalized to obtain hardness results for k-submodular maximization for $k \geq 3$.

Acknowledgments

We are grateful to Maxim Sviridenko for many insightful conversations.

References

[1] Kazutoshi Ando. K-submodular functions and convexity of their Lovász extension. *Discrete Applied Mathematics*, 122(1-3):1–12, 2002.

[2] Kazutoshi Ando, Satoru Fujishige, and Takeshi Naitoh. A characterization of bisubmodular functions. *Discrete Mathematics*, 148(1-3):299–303, 1996.

[3] Jesús M. Bilbao, Julio R. Fernández, Nieves Jiménez, and Jorge J. López. Survey of bicooperative games. In Altannar Chinchuluun, Panos M. Pardalos, Athanasios Migdalas, and Leonidas Pitsoulis, editors, *Pareto Optimality, Game Theory and Equilibria*. Springer, 2008.

[4] André Bouchet. Greedy algorithm and symmetric matroids. *Mathematical Programming*, 38(2):147–159, 1987.

[5] Niv Buchbinder, Moran Feldman, Joseph Naor, and Roy Schwartz. A tight linear time (1/2)-approximation for unconstrained submodular maximization. In *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS'12)*, pages 649–658. IEEE, 2012.

[6] Ramaswamy Chandrasekaran and Santosh N. Kabadi. Pseudomatroids. *Discrete Mathematics*, 71(3):205–217, 1988.

[7] David A. Cohen, Martin C. Cooper, Peter G. Jeavons, and Andrei A. Krokhin. The Complexity of Soft Constraint Satisfaction. *Artificial Intelligence*, 170(11):983–1016, 2006.

[8] Shahar Dobzinski and Jan Vondrák. From query complexity to computational complexity. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC'12)*, pages 1107–1116. ACM, 2012.

[9] Uriel Feige, Vahab S. Mirrokni, and Jan Vondrák. Maximizing Non-monotone Submodular Functions. *SIAM Journal on Computing*, 40(4):1133–1153, 2011.

[10] Satoru Fujishige. *Submodular Functions and Optimization*, volume 58 of *Annals of Discrete Mathematics*. North-Holland, Amsterdam, 2nd edition, 2005.

[11] Satoru Fujishige and Satoru Iwata. Bisubmodular Function Minimization. *SIAM Journal on Discrete Mathematics*, 19(4):1065–1073, 2005.

[12] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman, 1979.

[13] M. Grötschel, L. Lovasz, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–198, 1981.

[14] M. Grötschel, L. Lovasz, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, 1988.

[15] Anna Huber and Vladimir Kolmogorov. Towards Minimizing k-Submodular Functions. In *Proceedings of the 2nd International Symposium on Combinatorial Optimization (ISCO'12)*, volume 7422 of *Lecture Notes in Computer Science*, pages 451–462. Springer, 2012.

[16] Anna Huber, Andrei Krokhin, and Robert Powell. Skew Bisubmodularity and Valued CSPs. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'13)*, pages 1296–1305. SIAM, 2013.

[17] S. Iwata. A fully combinatorial algorithm for submodular function minimization. *Journal of Combinatorial Theory, Series B*, 84(2):203–212, 2002.

[18] S. Iwata. A faster scaling algorithm for minimizing submodular functions. *SIAM Journal on Computing*, 32(4):833–840, 2003.

[19] Satoru Iwata. Submodular Function Minimization. *Mathematical Programming*, 112(1):45–64, 2008.

[20] Satoru Iwata, Lisa Fleischer, and Satoru Fujishige. A combinatorial strongly polynomial algorithm for minimizing submodular functions. *Journal of the ACM*, 48(4):761–777, 2001.

[21] Satoru Iwata, Shin ichi Tanigawa, and Yuichi Yoshida. Bisubmodular function maximization and extensions. Technical Report METR 2013-16, The University of Tokyo, 2013.

[22] Satoru Iwata and James B. Orlin. A Simple Combinatorial Algorithm for Submodular Function Minimization. In *Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'09)*, pages 1230–1237, 2009.

[23] Vladimir Kolmogorov. Submodularity on a tree: Unifying $L^\natural$-convex and bisubmodular functions. In *Proceedings of the 36th International Symposium on Mathematical Foundations of Computer Science (MFCS'11)*, volume 6907 of *Lecture Notes in Computer Science*, pages 400–411. Springer, 2011.

[24] Bernhard Korte and Jens Vygen. *Combinatorial Optimization*, volume 21 of *Algorithms and Combinatorics*. Springer, 4th edition, 2007.

[25] Jon Lee, Maxim Sviridenko, and Jan Vondrák. Submodular Maximization over Multiple Matroids via Generalized Exchange Properties. *Mathematics of Operations Research*, 35(4):795–806, 2010.

[26] László Lovász. Submodular Functions and Convexity. In A. Bachem, M. Grötschel, and B. Korte, editors, *Mathematical Programming – The State of the Art*, pages 235–257, Berlin, 1983. Springer.

[27] S. Thomas McCormick and Satoru Fujishige. Strongly polynomial and fully combinatorial algorithms for bisubmodular function minimization. *Mathematical Programming*, 122(1):87–120, 2010.

[28] H. Narayanan. *Submodular Functions and Electrical Networks*. North-Holland, Amsterdam, 1997.

[29] George L. Nemhauser and Laurence A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, 1988.

[30] James B. Orlin. A faster strongly polynomial time algorithm for submodular function minimization. *Mathematical Programming*, 118(2):237–251, 2009.

[31] Christos H. Papadimitriou and Mihalis Yannakakis. Optimization, Approximation, and Complexity Classes. *Journal of Computer and System Sciences*, 43(3):425–440, 1991.

[32] Liqun Qi. Directed submodularity, ditroids and directed submodular flows. *Mathematical Programming*, 42(1-3):579–599, 1988.

[33] Alexander Schrijver. A Combinatorial Algorithm Minimizing Submodular Functions in Strongly Polynomial Time. *Journal of Combinatorial Theory, Series B*, 80(2):346–355, 2000.

[34] Alexander Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*, volume 24 of *Algorithms and Combinatorics*. Springer, 2003.

[35] Ajit P. Singh, Andrew Guillory, and Jeff Bilmes. On bisubmodular maximization. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS'12)*, volume 22 of *JLMR Workshop and Conference Proceedings*, pages 1055–1063, 2012.

[36] Johan Thapper and Stanislav Živný. The power of linear programming for valued CSPs. In *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS'12)*, pages 669–678. IEEE, 2012.

[37] Johan Thapper and Stanislav Živný. The complexity of finite-valued CSPs. In *Proceedings of the 45th ACM Symposium on the Theory of Computing (STOC'13)*, pages 695–704. ACM, 2013.

[38] Donald Topkis. *Supermodularity and Complementarity.* Princeton University Press, 1998.

[39] Jan Vondrák. Symmetry and Approximability of Submodular Maximization Problems. In *Proceedings of the 50th IEEE Symposium on Foundations of Computer Science (FOCS '09)*, pages 651–670. IEEE Computer Society, 2009.

A The Naive Random Algorithm

We present the analysis for the case in which $k \geq 3$ first, as it is simpler and will aid in motivating some of the constructions used for the case $k = 2$.

A.1 Analysis for $k \geq 3$

Let f be a k-submodular function over a ground set U of size n. It will be convenient to treat solutions to this problem as vectors in $\{0, \ldots, k\}^U$, as discussed in Section 2. Let $\mathbf{o}$ be a vector on which f takes its maximum value. Then, by Corollary 2.4, we may assume without loss of generality that $\mathbf{o}$ is a partition and so $\mathbf{o} \in \{1, \ldots, k\}^U$. Finally, let $h : 2^U \to \mathbb{R}_+$ be the submodular function induced by $\mathbf{o}$ and f.

For each $i \in U$ we consider a fixed permutation π_i on the set $\{1, \ldots, k\}$ with the property that $\pi_i(o_i) = o_i$ and $\pi_i(z) \neq z$ for all $z \in \{1, \ldots, k\} \setminus \{o_i\}$.[5] Then, we denote by $\pi(\mathbf{x})$ the vector whose ith coordinate is $\pi_i(x_i)$.

Let $P(A)$ be the set of partitions of U that agree with $\mathbf{o}$ on exactly those coordinates $i \in A$. The following lemma allows us to relate the sum of the values of all partitions in $P(A)$ to the value of $\mathbf{o}$.

Lemma A.1. *For each set $A \subseteq U$,*

$$\sum_{\mathbf{x} \in P(A)} f(\mathbf{x}) \geq (k-1)^{n-|A|} h(A).$$

Proof. Consider the sum

$$\sum_{\mathbf{x} \in P(A)} f(\pi(\mathbf{x})).$$

[5] Such a permutation can be obtained by taking, for example, $\pi_i(o_i) = o_i$, $\pi_i(o_i - 1) = o_i + 1$, and $\pi(z) = z + 1 \mod k$ for all other $z \in \{1, \ldots, k\}$.

Because $\pi_i(x_i) = o_i$ if and only if $x_i = o_i$ already, we have $\pi(\mathbf{x}) \in P(A)$ if and only if $\mathbf{x} \in P(A)$. Then, because each π_i is a bijection, we have

$$\sum_{\mathbf{x}\in P(A)} f(\mathbf{x}) = \sum_{\mathbf{x}\in P(A)} f(\pi(\mathbf{x})),$$

and so,

$$\begin{aligned}\sum_{\mathbf{x}\in P(A)} f(\mathbf{x}) &= \frac{1}{2}\left[\sum_{\mathbf{x}\in P(A)} f(\mathbf{x}) \;+ \sum_{\mathbf{x}\in P(A)} f(\pi(\mathbf{x}))\right] \\ &= \frac{1}{2}\sum_{\mathbf{x}\in P(A)} \left[f(\mathbf{x}) + f(\pi(\mathbf{x}))\right].\end{aligned} \tag{8}$$

Now, we note that $\mathbf{x}$ and $\pi(\mathbf{x})$ are both partitions. Thus, from (2) we have

$$f(\mathbf{x}) + f(\pi(\mathbf{x})) \geq 2\mathrm{id}_0(\mathbf{x}, \pi(\mathbf{x})).$$

Consider an arbitrary coordinate $i \in U$. If $i \in A$ we have $x_i = o_i$ and so $\pi_i(x_i) = x_i$ and hence $\mathrm{id}_0(x_i, \pi_i(x_i)) = x_i$. If $i \notin A$, then we have $x_i \neq o_i$ and so $\pi_i(x_i) \neq x_i$ and hence $\mathrm{id}_0(x_i, \pi_i(x_i)) = 0$. Thus,

$$2\mathrm{id}_0(\mathbf{x}, \pi(\mathbf{x})) = 2f(\mathbf{o}\big|_A) = 2h(A).$$

Combining this with (8) we have,

$$\sum_{\mathbf{x}\in P(A)} f(\mathbf{x}) \;=\; \frac{1}{2}\sum_{\mathbf{x}\in P(A)} \left[f(\mathbf{x}) + f(\pi(\mathbf{x}))\right] \geq \sum_{\mathbf{x}\in P(A)} h(A) \;=\; (k-1)^{n-|A|}h(A),$$

since there are precisely $k-1$ choices $j \neq o_i$ for x_i for each of the $n - |A|$ coordinates $i \notin A$. □

We can now prove our main result regarding the expected quality of a random partition.

Theorem A.2. *Let $\mathbf{x} \in \{1, \ldots, k\}^U$ be a partition of U chosen uniformly at random. Then, $\mathbb{E}[f(\mathbf{x})] \geq \frac{1}{k} \cdot f(\mathbf{o})$.*

Proof. We formulate the expectation as

$$\mathbb{E}[f(\mathbf{x})] = \frac{1}{k^n}\sum_{i=0}^{n} \sum_{A\in\binom{U}{i}} \sum_{\mathbf{x}\in P(A)} f(\mathbf{x}).$$

Using Lemma A.1 we obtain

$$\sum_{i=0}^{n} \sum_{A\in\binom{U}{i}} \sum_{\mathbf{x}\in P(A)} f(\mathbf{x}) \geq \sum_{i=0}^{n} \sum_{A\in\binom{U}{i}} (k-1)^{n-i}h(A). \tag{9}$$

Consider a fixed value $i \in \{0, \ldots, n\}$. Each element $e \in U$ appears in exactly $\binom{n-1}{i-1}$ of the $\binom{n}{i}$ sets $A \in \binom{U}{i}$. Because h is submodular, Corollary 2.7 then implies that

$$\sum_{A \in \binom{U}{i}} h(A) \geq \binom{n-1}{i-1} h(U) = \binom{n-1}{i-1} f(\mathbf{o}). \tag{10}$$

Combining (9) and (10) with our formulation of $\mathbb{E}[f(\mathbf{x})]$ we obtain:

$$\begin{aligned}
\mathbb{E}[f(\mathbf{x})] &\geq \frac{1}{k^n} \sum_{i=0}^{n} \binom{n-1}{i-1} (k-1)^{n-i} f(\mathbf{o}) \\
&= \frac{(k-1)^{n-1}}{k^n} \sum_{i=0}^{n} \binom{n-1}{i-1} (k-1)^{-(i-1)} f(\mathbf{o}) \\
&= \frac{(k-1)^{n-1}}{k^n} \sum_{i=0}^{n-1} \binom{n-1}{i} (k-1)^{-i} f(\mathbf{o}) \\
&= \frac{(k-1)^{n-1}}{k^n} \cdot \left(1 + \frac{1}{k-1}\right)^{n-1} \cdot f(\mathbf{o}) \\
&= \frac{(k-1)^{n-1}}{k^n} \cdot \frac{k^{n-1}}{(k-1)^{n-1}} \cdot f(\mathbf{o}) \\
&= \frac{1}{k} \cdot f(\mathbf{o}).
\end{aligned}$$

□

A.2 Analysis for $k = 2$

Now we consider the case in which f is a bisubmodular function. In our analysis of k-submodular functions for $k \geq 3$ we used a bijection π_i on $\{1, \ldots, k\}$ with the property that had the property that $\pi_i(o_i) = o_i$ and $\pi_i(z) \neq z$ for all $z \neq o_i$. However, when $k = 2$, no such bijection exists and we must adopt a different approach.

Suppose that f attains its maximum on a partition $\mathbf{o} \in \{1, 2\}^U$, and for a value $v \in \{1, 2\}$ let $\bar{v} = (v \mod 2) + 1$ (i.e. the other value in $\{1, 2\}$). Then, for any disjoint subsets A and B of U we define the (partial) solution $T(A, B)$ by

$$T(A, B)_i = \begin{cases} o_i, & i \in A \\ \bar{o_i}, & i \in B \\ 0, & \text{otherwise} \end{cases}.$$

It will simplify our analysis to work with with symmetrized values, which depend only on the sizes of the sets A and B chosen. We define

$$F_{i,j} = \binom{n}{i}^{-1} \binom{n-i}{j}^{-1} \sum_{A \in \binom{U}{i}} \sum_{B \in \binom{A \setminus B}{i}} [f(T(A, B))].$$

Then, $F_{i,j}$ gives the average value of f over all partial solutions on $i+j$ elements that agree with $\mathbf{o}$ on exactly i and disagree with it on exactly j elements. In

particular, we have $F_{n,0} = f(\mathbf{o})$, and $F_{i,n-i} = \binom{n}{i}^{-1} \sum_{A \in \binom{U}{i}} f(T(A, U \setminus A))$. Our next lemma relates these two values.

Lemma A.3. *For all i such that $0 \leq i \leq n$,*

$$F_{i,n-i} \geq \frac{i(i-1)}{n(n-1)} F_{n,0}. \tag{11}$$

Proof. We prove 2 separate inequalities which together imply the lemma. First, we shall show that for all $1 \leq i \leq n-1$,

$$F_{i,n-i} \geq F_{i-1,n-i-1}. \tag{12}$$

We prove that a related inequality holds for arbitrary sets of the appropriate size, and then average over all possible sets to obtain (12). Fix $1 \leq i \leq n-1$ and let A be any subset of U of size $i+1$. Set $B = U \setminus A$ and let x and y any two distinct elements in A. Consider the solutions $T(A-x, B+x)$ and $T(A-y, B+y)$. They are both partitions and agree on all elements except x and y. Thus, from (2)

$$\begin{aligned} &f(T(A-x, B+x)) + f(T(A-y, B+y)) \\ &\quad \geq 2\mathrm{id}_0(T(A-x, B+x), T(A-y, B+y)) \\ &\quad = 2f(T(A-x-y, B-x-y)), \end{aligned}$$

holds for any such choice of A, x, and y. Averaging the resulting inequalities over all possible choices for A, $B = U \setminus A$, x, and y and dividing both sides by 2 then gives (12).

Next, we show that for any $1 \leq i \leq n-1$,

$$F_{i-1,n-i-1} \geq \frac{i-1}{i+1} F_{i+1,n-i-1} \tag{13}$$

Again fix $i \geq 1$, let A be any subset of U of size $i+1$ and set $B = U \setminus A$. Let h be the submodular function induced by the partition $T(A, B)$ and f. Note then, that we can express h as $h(X) = T(A \cap X, B \cap X))$. We consider the sum:

$$\sum_{C \in \binom{A}{2}} [f(T(A \setminus C, B)) - T(\emptyset, B)] = \sum_{C \in \binom{A}{2}} [h(U \setminus C) - h(B)]$$

Each element of A appears in exactly $\binom{|A|-1}{2} = \binom{i}{2}$ of the sets $U \setminus C$ above (one for each way to choose a two element set C from the remaining $|A|-1$ elements). Applying Corollary 2.7 we then obtain

$$\sum_{C \in \binom{A}{2}} h(U \setminus C) \geq \binom{i}{2} h(U) = \binom{i}{2} T(A, B).$$

Averaging over all possible choices for A gives

$$\binom{i+1}{2} F_{i-1,n-i-1} \geq \binom{i}{2} F_{i+1,n-i-1},$$

which is equivalent to (13).

Combining (12) and (13) then gives the symmetrized inequality

$$F_{i,n-i} \geq \frac{i-1}{i+1} F_{i+1,n-i-1}. \tag{14}$$

The desired inequality (11) then follows from reverse induction on i. If $i = n$, then (11) is trivial. For the inductive step, we suppose that $1 \leq i \leq n-1$. Then, applying (14) followed by the induction hypothesis gives

$$F_{i,n-i} \geq \frac{i-1}{i+1} F_{i+1,n-i-1} \geq \frac{i-1}{i+1} \cdot \frac{(i+1)i}{n(n-1)} F_{n,0} = \frac{i(i-1)}{n(n-1)} F_{n,0}.$$

If $i = 0$, we cannot apply (14). In this case, however, (11) follows directly from non-negativity of f. □

Theorem A.4. *Let $\mathbf{x} \in \{1, \dots, k\}^U$ be a partition of U chosen uniformly at random. Then, $\mathbb{E}[f(\mathbf{x})] \geq \frac{1}{4} \cdot f(\mathbf{o})$.*

Proof. We can formulate the expectation in terms of our symmetric notation as

$$\begin{aligned}
\mathbb{E}[f(\mathbf{x})] &= 2^{-n} \sum_{i=0}^{n} \sum_{A \in \binom{U}{i}} T(A, U \setminus A) \\
&= 2^{-n} \sum_{i=0}^{n} \binom{n}{i} F_{i,n-i}.
\end{aligned}$$

Then, we have

$$\begin{aligned}
2^{-n} \sum_{i=0}^{n} \binom{n}{i} F_{i,n-i} &\geq 2^{-n} \sum_{i=2}^{n} \binom{n}{i} F_{i,n-i} \\
&\geq 2^{-n} \sum_{i=2}^{n} \binom{n}{i} \frac{i(i-1)}{n(n-1)} F_{n,0} \\
&= 2^{-n} \sum_{i=2}^{n} \binom{n-2}{i-2} F_{n,0} \\
&= 2^{-n} \sum_{i=0}^{n-2} \binom{n-2}{i} F_{n,0} \\
&= 2^{-n} \cdot 2^{n-2} F_{n,0} \\
&= \frac{1}{4} f(\mathbf{o}),
\end{aligned}$$

where the first inequality follows from non-negativity of f (and hence of F) and the second inequality follows from Lemma A.3. □

B A Deterministic Greedy Algorithm

In this section we consider a deterministic greedy algorithm that is even simpler than the randomized greedy algorithm from Section 4. The algorithm begins with the initial solution $(\emptyset, \ldots, \emptyset)$ and considers elements of the ground set U in some arbitrary order, permanently adding each element to one of the sets S_i in S, based on the increase that this gives in f. Specifically, the algorithm adds an element e to the set S_i with the biggest marginal increase $f_{i,e}(S)$ in f with respect to the current solution S. If there are more than one option we add e to S_i with the smallest i.

Deterministic Greedy

```
for i = 1 to k do
    S_i ← ∅
end for
for each e ∈ U do
    for i = 1 to k do
        x_i ← f_{i,e}(S)
    end for
    x = max(x_1, ..., x_k)
    Let i be the smallest value from {1, ..., k}
        so that x_i = x.
    S_i ← S_i + e
end for
```

The analysis of the deterministic greedy algorithm is similar to the analysis of the randomized greedy algorithm from Section 4 but simpler.

Theorem B.1. *Let S be the solution produced by the deterministic greedy algorithm on some instance $f : \{0, \ldots, k\}^U \to \mathbb{R}_+$ of k-submodular maximization, and let OPT be the optimal solution for this instance. Then,*

$$(1 + k)f(S) \;\geq\; f(OPT).$$

Proof. Our analysis considers 2 sequences of n solutions. First let, $S^{(i)} = (S_1^{(i)}, \ldots, S_k^{(i)})$ be the algorithm's solution after i elements have been considered, and let $U^{(i)} = \bigcup_{j=1}^{k} S_j^{(i)}$ be the set of elements that have been considered by the algorithm at this point. Then, we define the solution $O^{(i)} = (O_1^{(i)}, \ldots, O_k^{(i)})$, where $O_j^{(i)} = (OPT_j \setminus U^{(i)}) \cup S_j^{(i)}$. Intuitively, $O^{(i)}$ is the solution that agrees with $S^{(i)}$ on the placement of the elements considered by the greedy algorithm in its first i phases and agrees with OPT on the placement of all other elements. Note that in particular we have $O^{(0)} = OPT$ and $O^{(n)} = S$. Our analysis of the greedy algorithm will bound the loss in $f(O_i)$ incurred at the each stage, showing that it is bounded by the improvement made by the algorithm. In Lemma B.2, we show that for every $0 \leq i \leq n$, $f(O^{(i)}) - f(O^{(i+1)}) \;\leq\; k[f(S^{(i+1)}) - f(S^{(i)})]$.

Summing the inequality from Lemma B.2 from $i = 0$ to $n - 1$, we obtain

$$\sum_{i=0}^{n-1} \left[f(O^{(i)}) - f(O^{(i+1)}) \right] \leq \; k \sum_{i=0}^{n-1} \left[f(S^{(i+1)}) - f(S^{(i)}) \right].$$

Telescoping the summations on each side, we then have

$$f(O^{(0)}) - f(O^{(n)}) \; \leq \; k \left[f(S^{(n)}) - f(S^{(0)}) \right].$$

The theorem then follows immediately from $O^{(0)} = OPT$, $O^{(n)} = S^{(n)} = S$, and $S_0 \geq 0$. □

It remains to show the following result.

Lemma B.2. *For* $0 \leq i \leq n - 1$,

$$f(O^{(i)}) - f(O^{(i+1)}) \; \leq \; k \left[f(S^{(i+1)}) - f(S^{(i)}) \right].$$

Proof. Let e be the element considered in the ith phase of the algorithm.

We define the solution $A = (A_1, \ldots, A_k)$, where $A_j = (O_j^{(i)} - e)$, and let $a_j = f_{j,e}(A)$ for $1 \leq j \leq k$.

Now, let suppose that $e \in OPT_o$ for some $1 \leq o \leq k$, and that the greedy algorithm places $e \in S_j^{(i)}$ for some $1 \leq j \leq k$. Then, $O^{(i)}$ and $O^{(i+1)}$ are identical except that $O^{(i)}$ places e in the oth set, while $O^{(i+1)}$ places e in the jth set. Thus, we have $f(O^{(i)}) = f(A) + f_{o,e}(A)$ and $f(O^{(i+1)}) = f(A) + f_{j,e}(A)$, and so

$$f(O^{(i)}) - f(O^{(i+1)}) = f_{o,e}(A) - f_{j,e}(A) = a_o - a_j,$$

and

$$f(S^{(i+1)}) - f(S^{(i)}) = f_{j,e}(S^{(i)}) = x_j.$$

By Theorem 2.3, we have $\sum_{\ell=1}^{k} a_\ell \geq 0$ and thus $-a_j \leq \sum_{\ell \neq j} a_\ell$. Therefore, $a_o - a_j \leq a_o + \sum_{\ell \neq j} a_\ell \leq k x_j$ as $a_r \leq x_r$ for every $1 \leq r \leq k$ and $x_j = \max(x_1, \ldots, x_k)$. □

C Verification

We want to show that the right hand side of (6) can be written as the following sum of squares:

$$\begin{aligned} (x_\ell - x_o)^2 + \sum_{j \neq o,\ell} \left(\sqrt{\frac{\alpha - 1}{k - 2}} x_\ell - \sqrt{\frac{\alpha + 1}{2}} x_j \right)^2 \\ + \sum_{j \neq o,\ell} \left(\sqrt{\frac{\alpha - 1}{k - 2}} x_o - \sqrt{\frac{\alpha + 1}{2}} x_j \right)^2. \end{aligned} \tag{15}$$

In order to verify that this is the case, note that

$$(x_\ell - x_o)^2 = x_\ell^2 - 2x_\ell x_o + x_o^2$$

and

$$\begin{aligned}
&\left(\sqrt{\frac{\alpha-1}{k-2}}x_\ell - \sqrt{\frac{\alpha+1}{2}}x_j\right)^2 \\
&= \frac{\alpha-1}{k-2}x_\ell^2 - 2\sqrt{\frac{(\alpha-1)(\alpha+1)}{2(k-2)}}x_\ell x_j + \frac{\alpha+1}{2}x_j^2 \\
&= \frac{\alpha-1}{k-2}x_\ell^2 - 2\sqrt{\frac{\alpha^2-1}{2(k-2)}}x_\ell x_j + \frac{\alpha+1}{2}x_j^2 \\
&= \frac{\alpha-1}{k-2}x_\ell^2 - 2\sqrt{\frac{\frac{k}{2}-1}{2(k-2)}}x_\ell x_j + \frac{\alpha+1}{2}x_j^2 \\
&= \frac{\alpha-1}{k-2}x_\ell^2 - 2\sqrt{\frac{\frac{k-2}{2}}{2(k-2)}}x_\ell x_j + \frac{\alpha+1}{2}x_j^2 \\
&= \frac{\alpha-1}{k-2}x_\ell^2 - 2\sqrt{\frac{1}{4}}x_\ell x_j + \frac{\alpha+1}{2}x_j^2 \\
&= \frac{\alpha-1}{k-2}x_\ell^2 - x_\ell x_j + \frac{\alpha+1}{2}x_j^2,
\end{aligned}$$

and, similarly,

$$\begin{aligned}
&\left(\sqrt{\frac{\alpha-1}{k-2}}x_o - \sqrt{\frac{\alpha+1}{2}}x_j\right)^2 \\
&= \frac{\alpha-1}{k-2}x_o^2 - x_o x_j + \frac{\alpha+1}{2}x_j^2
\end{aligned}$$

Thus, (15) is equal to

$$
\begin{aligned}
& x_\ell^2 - 2x_\ell x_o + x_o^2 + \sum_{j \neq o,\ell} \left[\frac{\alpha - 1}{k-2} x_\ell^2 - x_\ell x_j + \frac{\alpha+1}{2} x_j^2 \right] \\
& \qquad + \sum_{j \neq o,\ell} \left[\frac{\alpha - 1}{k-2} x_o^2 - x_o x_j + \frac{\alpha+1}{2} x_j^2 \right] \\
&= x_\ell^2 - 2x_\ell x_o + x_o^2 + (\alpha - 1) x_\ell^2 + (\alpha - 1) x_o^2 \\
& \qquad - \sum_{j \neq o,\ell} \left[x_\ell x_j - \frac{\alpha+1}{2} x_j^2 \right] \\
& \qquad - \sum_{j \neq o,\ell} \left[x_o x_j - \frac{\alpha+1}{2} x_j^2 \right] \\
&= x_\ell^2 - 2x_\ell x_o + x_o^2 + (\alpha - 1) x_\ell^2 + (\alpha - 1) x_o^2 \\
& \qquad - \sum_{j \neq o,\ell} \left[x_\ell x_j + x_o x_j - \frac{\alpha+1}{x}^2{}_j \right] \\
&= \alpha x_\ell^2 + \alpha x_o^2 - 2x_\ell x_o + \alpha \sum_{j \neq o,\ell} x_j^2 \\
& \qquad - \sum_{j \neq o,\ell} \left[x_\ell x_j + x_o x_j - x_j^2 \right] \\
&= \alpha \sum_j x_j^2 - 2x_\ell x_o - \sum_{j \neq o,\ell} \left[x_\ell x_j + x_o x_j - x_j^2 \right].
\end{aligned}
$$