Multivariate Submodular Optimization (paper full version)

Richard Santiago
School of Computer Science, McGill University, Montreal, Canada
richard.santiagotorres@mail.mcgill.ca

F. Bruce Shepherd
Department of Computer Science, University of British Columbia, Vancouver, Canada
fbrucesh@cs.ubc.ca

Abstract

Submodular functions have found a wealth of new applications in data science and machine learning models in recent years. This has been coupled with many algorithmic advances in the area of submodular optimization: (SO) $\min / \max \; f(S) : S \in \mathcal{F}$, where $\mathcal{F}$ is a given family of feasible sets over a ground set V and $f : 2^V \rightarrow \mathbb{R}$ is submodular. Our focus is on a more general class of *multivariate submodular optimization* (MVSO) problems: $\min / \max \; f(S_1, S_2, \ldots, S_k) : S_1 \uplus S_2 \uplus \cdots \uplus S_k \in \mathcal{F}$. Here we use $\uplus$ to denote union of disjoint sets and hence this model is attractive where resources are being allocated across k agents, who share a "joint" multivariate nonnegative objective $f(S_1, S_2, \ldots, S_k)$ that captures some type of submodularity (i.e. diminishing returns) property. We provide some explicit examples and potential applications for this new framework.

For maximization, we show that practical algorithms such as accelerated greedy variants and distributed algorithms achieve good approximation guarantees for very general families (such as matroids and p-systems). For arbitrary families, we show that monotone (resp. nonmonotone) MVSO admits an $\alpha(1 - 1/e)$ (resp. $\alpha \cdot 0.385$) approximation whenever monotone (resp. nonmonotone) SO admits an α-approximation over the multilinear formulation. This substantially expands the family of tractable models for submodular maximization. For minimization, we show that if SO admits a β-approximation over *modular* functions, then MVSO admits a $\frac{\beta \cdot n}{1+(n-1)(1-c)}$-approximation where $c \in [0, 1]$ denotes the curvature of f. We show that this approximation is essentially tight even for $\mathcal{F} = \{V\}$. Finally, we give a bound in terms of k and prove that MVSO has an αk-approximation whenever SO admits an α-approximation over the convex formulation.

2012 ACM Subject Classification Theory of computation → Submodular optimization and polymatroids

Keywords and phrases submodular optimization, machine learning, multi-agent optimization, multivariate optimization, approximation algorithms

1 Introduction

Submodularity is a property of set functions with deep theoretical consequences and a wide range of applications. Optimizing submodular functions is a central subject in operations research and combinatorial optimization [47]. It appears in many important optimization frameworks including cuts in graphs, set covering problems, plant location problems, certain satisfiability problems, combinatorial auctions, and maximum entropy sampling. In machine learning it has recently been identified and utilized in domains such as viral marketing [35], information gathering [39], image segmentation [4, 37, 34], document summarization [46], and speeding up satisfiability solvers [59].

A set function $f : 2^V \to \mathbb{R}$ is *submodular* if $f(S) + f(T) \geq f(S \cup T) + f(S \cap T)$ for any $S, T \subseteq V$. We say that f is *monotone* if $f(S) \leq f(T)$ for $S \subseteq T$. Throughout, all submodular functions are nonnegative, and we usually assume $f(\emptyset) = 0$. Our functions are given by a *value oracle*, where for a given set S an algorithm can query the oracle to find its value $f(S)$.

We consider the following broad class of submodular optimization (SO) problems:

$$\text{SO}(\mathcal{F}) \qquad \text{Min} \;\; / \;\; \text{Max} \;\; f(S) : S \in \mathcal{F} \tag{1}$$

where f is a nonnegative submodular set function on a finite ground set V, and $\mathcal{F} \subseteq 2^V$ is a family of feasible sets. These problems have been well studied for a variety of set families $\mathcal{F}$. We explore the connections between these (single-agent) problems and their more general multivariate incarnations. In the *multivariate (MV)* version, we have k agents and a "joint" multivariate nonnegative objective $f(S_1, S_2, \ldots, S_k)$ that captures some type of submodularity (i.e. diminishing returns) property (see Section 1.1). As before, we are looking for sets $S \in \mathcal{F}$, however, we now have a 2-phase task: the elements of S must also be partitioned amongst the agents. Hence we have set variables S_i and seek to optimize $f(S_1, S_2, \ldots, S_k)$. This leads to the multivariate submodular optimization (MVSO) versions:

$$\text{MVSO}(\mathcal{F}) \qquad \text{Min} \;\; / \;\; \text{Max} \;\; f(S_1, S_2, \ldots, S_k) : S_1 \uplus S_2 \uplus \cdots \uplus S_k \in \mathcal{F}. \tag{2}$$

Our main objective is to study the approximability of the multivariate problems in terms of their single-agent versions. We refer to the *multivariate (MV) gap* as the approximation factor loss incurred by moving to the multivariate setting. To the best of our knowledge, neither the MVSO($\mathcal{F}$) framework for general families $\mathcal{F}$ nor the notion of MV gap have been considered before in the literature.

An important special case of MVSO occurs when the function $f(S_1, \ldots, S_k)$ can be *separated* as $f(S_1, \ldots, S_k) = \sum_{i \in [k]} f_i(S_i)$ where the f_i are all submodular; in this case we say that f is *separable*. This leads to the class of multi-agent submodular optimization (MASO) problems

$$\text{MASO}(\mathcal{F}) \qquad \text{Min} \;\; / \;\; \text{Max} \;\; \textstyle\sum_{i=1}^{k} f_i(S_i) : S_1 \uplus S_2 \uplus \cdots \uplus S_k \in \mathcal{F}, \tag{3}$$

which have been widely studied (see related work section).

1.1 Multivariate submodular optimization

We consider functions of several variables which satisfy the following type of submodularity property. A multivariate function $f : 2^{kV} \to \mathbb{R}$ is *k-multi-submodular* if for all pairs of tuples $(S_1, S_2, ..., S_k), (T_1, T_2, ..., T_k) \in 2^{kV}$ we have

$$f(S_1, ..., S_k) + f(T_1, ..., T_k) \geq f(S_1 \cup T_1, S_2 \cup T_2, .., S_k \cup T_k) + f(S_1 \cap T_1, S_2 \cap T_2, ..., S_k \cap T_k).$$

Moreover, we say that f is *normalized* if $f(\emptyset, \emptyset, \ldots, \emptyset) = 0$, and *monotone* if $f(S_1, \ldots, S_k) \leq f(T_1, \ldots, T_k)$ for all tuples $(S_1, \ldots, S_k)$ and $(T_1, \ldots, T_k)$ satisfying $S_i \subseteq T_i$ for all $i \in [k]$.

In the special case of $k = 1$ a k-multi-submodular function is just a submodular function. In Appendix A we discuss how k-multi-submodular functions can also be naturally characterized (or defined) in terms of diminishing returns. This notion of multivariate submodularity has been considered before ([16, 58]) and we discuss this in detail on Section 1.4.

Two explicit examples of (non-separable) k-multi-submodular functions (see Appendix B for proofs) are the following.

▶ **Example 1.** Consider a multilinear function $h : \mathbb{Z}_+^k \to \mathbb{R}$ given by $h(z) = \sum_{S \subseteq [k]} a_S \prod_{m \in S} z_m$. Let $f : 2^{kV} \to \mathbb{R}$ be a multivariate set function defined as $f(S_1, \ldots, S_k) = h(|S_1|, \ldots, |S_k|)$. Then f is k-multi-submodular if and only if $a_S \leq 0$ for all $S \subseteq [k]$.

▶ **Example 2.** Let $h : \mathbb{Z}_+^k \to \mathbb{R}$ be a quadratic function given by $h(z) = z^T A z$. Let $f : 2^{kV} \to \mathbb{R}$ be a multivariate set function defined as $f(S_1, \ldots, S_k) = h(|S_1|, \ldots, |S_k|)$. Then f is k-multi-submodular if and only if $A = (a_{ij})$ satisfies $a_{ij} + a_{ji} \leq 0$ for all $i, j \in [k]$.

We believe the above examples are useful for modelling "competition" between agents in many domains. In Section 1.3 we discuss one application to sensor placement problems.

1.2 Our contributions

Our first contribution is to show that the MV framework can model much more general problems than the separable multi-agent (i.e. MASO) framework. This is quantitatively captured in the following information theoretic result (see Section 3.3) where we establish a large gap between the two problems:

$$(\mathbf{MV-Min}) \quad \begin{array}{ll} \min & f(S_1, \ldots, S_k) \\ \text{s.t.} & S_1 \uplus \cdots \uplus S_k = V \end{array} \qquad (\mathbf{MA-Min}) \quad \begin{array}{ll} \min & \sum_{i=1}^k f_i(S_i) \\ \text{s.t.} & S_1 \uplus \cdots \uplus S_k = V \end{array}$$

▶ **Theorem 3.** *The MV-Min problem with a nonnegative monotone k-multi-submodular objective function cannot be approximated to a ratio $o(n/\log n)$ in the value oracle model with polynomial number of queries, whereas its separable version MA-Min has a tight $O(\log n)$-approximation polytime algorithm for nonnegative monotone submodular functions f_i.*

The above result shows that the MV model may also potentially face roadblocks in terms of tractability. Fortunately, we can show that the multivariate problem remains very well-behaved in the maximization setting. Our main result establishes that if the single-agent problem for a family $\mathcal{F}$ admits approximation via its multilinear relaxation (see Section 2.2), then we may extend this to its multivariate version with a constant factor loss.

▶ **Theorem 4.** *If there is a (polytime) $\alpha(n)$-approximation for monotone SO($\mathcal{F}$) maximization via its multilinear relaxation, then there is a (polytime) $(1-1/e) \cdot \alpha(n)$-approximation for monotone MVSO($\mathcal{F}$) maximization. Furthermore, given a downwards closed family $\mathcal{F}$, if there is a (polytime) $\alpha(n)$-approximation for nonmonotone SO($\mathcal{F}$) maximization via its multilinear relaxation, then there is a (polytime) $0.385 \cdot \alpha(n)$-approximation for nonmonotone MVSO($\mathcal{F}$) maximization.*

We note that the multilinear relaxation can be efficiently evaluated for a large class of practical and useful submodular functions [32], thus making these algorithms viable for many real-world machine learning problems.

We remark that the MV gap of $1 - 1/e$ for monotone objectives is tight, in the sense that there are families where this cannot be improved. For instance, $\mathcal{F} = \{V\}$ has a trivial

1-approximation for the single-agent problem, and a $1 - 1/e$ inapproximability factor for the separable multi-agent (i.e. MASO) version [36, 48], and hence also for the more general MVSO problem.

An immediate application of Theorem 4 is that it provides the first constant (and in fact optimal) $(1 - 1/e)$-approximation for the monotone *generalized submodular welfare* problem $\max f(S_1, S_2, \ldots, S_k) : S_1 \uplus \cdots \uplus S_k = V$. This problem generalizes the well-studied submodular welfare problem [45, 64, 38], which captures several allocation problems and has important applications in combinatorial auctions, Internet advertising, and network routing. The MV objectives can capture much more general interactions among the agents/bidders, where now a bidder's valuation does not only depend on the set S of items that she gets, but also on the items that her strategic partners and competitors get. For instance, in a bandwidth spectrum auction, this could capture a company's interest to maximize compatibility and prevent cross-border interference.

In Section 2 we describe a simple reduction that shows that for some families [1] an (optimal) MV gap of 1 holds. We also discuss how for those families, practical algorithms (such as accelerated greedy variants and distributed algorithms) can be used and lead to good approximation guarantees.

▶ **Theorem 5.** *Let $\mathcal{F}$ be a matroid, a p-matroid intersection, or a p-system. Then, if there is a (polytime) α-approximation algorithm for monotone (resp. nonmonotone) SO($\mathcal{F}$) maximization, there is a (polytime) α-approximation algorithm for monotone (resp. nonmonotone) MVSO($\mathcal{F}$) maximization.*

On the minimization side our approximation results and MV gaps are larger. This is somewhat expected due to the strong hardness results already existing for single-agent submodular minimization (see Section 1.4). However, we give essentially tight approximations in terms of the objective's curvature. The notion of curvature has been widely used for univariate functions [10, 65, 33, 2], since it allows for better approximations and it is linear time computable.

Given a tuple $(S_1, \ldots, S_k) \in 2^{kV}$ and $(i, v) \in [k] \times V$, we denote by $(S_1, \ldots, S_k) + (i, v)$ the new tuple $(S_1, \ldots, S_{i-1}, S_i + v, S_{i+1}, \ldots, S_k)$. Then, it is natural to think of the quantity

$$f_{(S_1,\ldots,S_k)}((i, v)) := f((S_1, \ldots, S_k) + (i, v)) - f(S_1, \ldots, S_k)$$

as the marginal gain of assigning element v to agent i in the tuple $(S_1, \ldots, S_k)$. We also use $f((i, v))$ to denote the quantity $f(\emptyset, \ldots, \emptyset, v, \emptyset, \ldots, \emptyset)$ where v appears in the *ith* component. Then given a normalized monotone k-multi-submodular function $f : 2^{kV} \to \mathbb{R}$ we define its *total curvature* c and its *curvature* $c(S_1, \ldots, S_k)$ *with respect to a tuple* $(S_1, \ldots, , S_k) \subseteq V^k$ as

$$c = 1 - \min_{i \in [k], v \in V} \frac{f_{(V,V,\ldots,V)-(i,v)}((i, v))}{f((i, v))} \quad , \quad c(S_1, \ldots, S_k) = 1 - \min_{i \in [k], v \in S_i} \frac{f_{(S_1,\ldots,S_k)-(i,v)}((i, v))}{f((i, v))}.$$

We prove the following curvature dependent result for k-multi-submodular objectives. We note the gap is stronger in the sense that it is relative to the single-agent *modular* problem. [2]

---

[1] A family of sets $\mathcal{F}$ is a *p-system* if for all $S \in \mathcal{F}$ and $v \in V$ there exists a set $T \subseteq S$ such that $|T| \leq p$ and $S \setminus T \cup \{v\} \in \mathcal{F}$. A *matroid* is a 1-system. Cardinality and partition constraints are examples of matroids. We refer the reader to [57, 7, 8] for a comprehensive discussion.

[2] A function $f : 2^V \to \mathbb{R}$ is modular if $f(A) + f(B) = f(A \cup B) + f(A \cap B)$ for all $A, B \subseteq V$. Modular functions can always be expressed in the form $f(S) = \sum_{v \in S} w(v)$ for some weight function $w : V \to \mathbb{R}$.

▶ **Theorem 6.** *Let f be a monotone k-multi-submodular function, and let $\mathcal{F}$ be a family that admits a (polytime) β-approximation over modular functions. Denote by $(S_1^*, \ldots, S_k^*)$ an optimal solution to monotone MVSO($\mathcal{F}$) minimization, and by $c(S_1^*, \ldots, S_k^*)$ the curvature of f with respect to $(S_1^*, \ldots, S_k^*)$. Then there is a (polytime) $\frac{\beta \sum_{i\in[k]} |S_i^*|}{1+(\sum_{i\in[k]} |S_i^*|-1)(1-c(S_1^*,\ldots,S_k^*))}$-approximation algorithm for monotone MVSO($\mathcal{F}$) minimization.*

In some situations the above result leads to approximation factors highly preferable to those obtained for general functions, given the strong polynomial hardness that most of these problems present for objectives with curvature 1. Examples of such situations include families like $\mathcal{F} = \{V\}$, spanning trees, or perfect matchings, where exact algorithms are available for modular objectives (i.e. $\beta = 1$ in those cases) and any optimal solution $(S_1^*, \ldots, S_k^*)$ satisfies $\sum_{i\in[k]} |S_i^*| = \Omega(n)$. Thus, we go from polynomial approximation factors (for objectives with curvature 1) to constant or logarithmic factors (for constant or order $1 - \frac{1}{\log n}$ curvature).

Moreover, having the curvature $c(S_1^*, \ldots, S_k^*)$ can be much more beneficial than having the total curvature c. For instance, for the problem $\min f(S_1, \ldots, S_k) : S_1 \uplus \cdots \uplus S_k = V$ with $f(S_1, \ldots, S_k) = \min\{n, \sum_{i=1}^k |S_i|\}$. Here the total curvature of f is 1 (hence leading to an n-approximation in Theorem 6), while the curvature $c(S_1^*, \ldots, S_k^*)$ with respect to any partition $(S_1^*, \ldots, S_k^*)$ is 0 (and thus leading to an exact approximation via Theorem 6).

In Section 3.3 we give evidence that Theorem 6 is essentially tight, even for $\mathcal{F} = \{V\}$ where we show the following curvature dependent information-theoretic lower bound.

▶ **Theorem 7.** *The monotone MVSO($\mathcal{F}$) minimization problem over $\mathcal{F} = \{V\}$ and objectives f with total curvature c cannot be approximated to a ratio $o(\frac{n/\log n}{1+(\frac{n}{\log n}-1)(1-c)})$ in the value oracle model with polynomial number of queries.*

Finally, we give an approximation in terms of the number of agents k, which may become preferable in settings where k is not too large.

▶ **Theorem 8.** *Suppose there is a (polytime) $\alpha(n)$-approximation for monotone SO($\mathcal{F}$) minimization based on rounding the convex relaxation. Then there is a (polytime) $k\alpha(n)$-approximation for monotone MVSO($\mathcal{F}$) minimization.*

1.3 The multivariate model and applications

Our second objective is to extend the multivariate model and show that in some cases this larger class remains tractable. Specifically, we define the *capacitated multivariate submodular optimization (CMVSO) problem* as follows:

$$\text{CMVSO}(\mathcal{F}) \qquad \begin{aligned} \max/\min \quad & f(S_1, S_2, \ldots, S_k) \\ \text{s.t.} \quad & S_1 \uplus \cdots \uplus S_k \in \mathcal{F} \\ & S_i \in \mathcal{F}_i\,,\ \forall i \in [k] \end{aligned} \tag{4}$$

where we are supplied with subfamilies $\mathcal{F}_i$.

Our results imply that one maintains good approximations even while adding interesting side constraints. For example, for a monotone maximization instance of CMVSO where $\mathcal{F}$ is a p-matroid intersection and the $\mathcal{F}_i$ are all matroids, our results from Section 2 lead to a $(\frac{1}{p+1} - \epsilon)$-approximation algorithm via the multilinear relaxation, or a $1/(p+2)$-approximation via a simple greedy algorithm. We believe that these, combined with other results from Section 2, substantially expand the family of tractable models (both in theory and practice) for maximization.

Many existing applications fit into the CMVSO framework and some of these can be enriched through the added flexibility of the capacitated model. For instance, one may include set bounds on the variables: $L_i \subseteq S_i \subseteq U_i$ for each i, or simple cardinality constraints: $|S_i| \leq b_i$ for each i. A well-studied ([17, 23, 8]) application of CMVSO in the maximization setting is the Separable Assignment Problem (SAP), which corresponds to the setting where the objective is separable and modular, the $\mathcal{F}_i$ are downward closed (i.e. hereditary) families, and $\mathcal{F} = 2^V$. The following example illustrates CMVSO's potential as a general model.

▶ **Example 9** (Sensor Placement with Multivariate Objectives)**.** The problem of placing sensors and information gathering has been popular in the submodularity literature [39, 41, 40]. We are given a set of sensors V and a set of possible locations $\{1, 2, \ldots, k\}$ where the sensors can be placed. There is also a budget constraint restricting the total number of sensors that can be deployed. The goal is to place sensors at some of the locations so as to maximize the "informativeness" gathered. This application is well suited to a k-multi-submodular objective function $f(S_1, ..., S_k)$ which measures the "informativeness" of placing sensors S_i at location i. A natural mathematical formulation for this is given by

$$\begin{array}{ll} \max & f(S_1, S_2, ..., S_k) \\ \text{s.t.} & S_1 \uplus S_2 \uplus \cdots \uplus S_k \in \mathcal{F} \\ & S_i \in \mathcal{F}_i, \end{array}$$

where $\mathcal{F} := \{S \subseteq V : |S| \leq b\}$ imposes the budget constraint and $\mathcal{F}_i$ gives additional modelling flexibility. For instance, we could impose $\mathcal{F}_i = \{S \subseteq V_i : |S| \leq b_i\}$ to constrain the types and number of sensors that can be placed at location i. Notice that in these cases both $\mathcal{F}$ and the $\mathcal{F}_i$ are matroids and hence the algorithms from Section 2.4 apply. One may form a multivariate objective by defining $f(S_1, S_2, \ldots, S_n) = \sum_i f_i(S_i) - R(S_1, S_2, \ldots, S_n)$ where the f_i's measure the benefit of placing sensors S_i at location i, and $R()$ is a redundancy function. If the f_i's are submodular and $R()$ is k-multi-supermodular, then f is k-multi-submodular. In this setting, it is natural to take the f_i's to be coverage functions (i.e. $f_i(S_i)$ measures the coverage of placing sensors S_i at location i). We next propose a family of "redundancy" functions which are k-multi-supermodular.

Supermodular penalty measures via Quadratic functions. We denote $\mathbf{S} := (S_1, S_2, \ldots, S_n)$ and define $z_{\mathbf{S}} := (|S_1|, |S_2|, \ldots, |S_n|)$. One can show (see Lemma 28 in Appendix B) that if A is a matrix satisfying $a_{ij} + a_{ji} \geq 0$, then $R(\mathbf{S}) := z_{\mathbf{S}}^T A z_{\mathbf{S}}$ is k-multi-supermodular. Then for this particular example one could for instance take redundancy coefficients a_{ij} as $\Theta(\frac{1}{d(i,j)^2})$ where $d(i, j)$ denotes the distance between locations i and j. This can be further extended so that different sensor types contribute different weights to the vector $z_{\mathbf{S}}$, e.g., define $z_{\mathbf{S}}(i) = \sum_{j \in S_i} w(j)$ for an associated sensor weight vector w.

1.4 Related work

Submodularity naturally arises in many machine learning applications such as viral marketing [35], information gathering [39], image segmentation [4, 37, 34], document summarization [46], news article recommendation [11], active learning [22], and speeding up satisfiability solvers [59].

Single Agent Optimization. The high level view of the tractability status for unconstrained (i.e., $\mathcal{F} = 2^V$) submodular optimization is that both maximization and minimization generally behave well. Minimizing a submodular set function is a classical combinatorial optimization problem which can be solved in polytime [24, 56, 30]. Unconstrained maximization, on the other hand, is known to be inapproximable for general submodular set functions but admits a polytime constant-factor approximation algorithm when f is nonnegative [6, 14].

In the constrained maximization setting, the classical work [51, 52, 16] already established an optimal $(1 - 1/e)$-approximation factor for maximizing a nonnegative monotone submodular function subject to a cardinality constraint, and a $(1/(k + 1))$-approximation for maximizing a nonnegative monotone submodular function subject to k matroid constraints. This approximation is almost tight in the sense that there is an (almost matching) factor $\Omega(\log(k)/k)$ inapproximability result [27]. For nonnegative monotone functions, [64, 8] give an optimal $(1-1/e)$-approximation based on multilinear extensions when $\mathcal{F}$ is a matroid; [42] provides a $(1 - 1/e - \epsilon)$-approximation when $\mathcal{F}$ is given by a constant number of knapsack constraints, and [44] gives a local-search algorithm that achieves a $(1/k - \epsilon)$-approximation (for any fixed $\epsilon > 0$) when $\mathcal{F}$ is a k-matroid intersection. For nonnegative nonmonotone functions, a 0.385-approximation is the best factor known [5] for maximization under a matroid constraint, in [43] a $1/(k + O(1))$-approximation is given for k matroid constraints with k fixed. A simple "multi-greedy" algorithm [25] matches the approximation of Lee et al. but is polytime for any k. Vondrak [66] gives a $\frac{1}{2}(1 - \frac{1}{\nu})$-approximation under a matroid base constraint where ν denotes the fractional base packing number. Finally, Chekuri et al [67] introduce a general framework based on relaxation-and-rounding that allows for combining different types of constraints. This leads, for instance, to $0.38/k$ and $0.19/k$ approximations for maximizing nonnegative submodular monotone and nonmonotone functions respectively under the combination of k matroids and $\ell = O(1)$ knapsacks constraints.

For constrained minimization, the news is worse [19, 60, 31]. If $\mathcal{F}$ consists of spanning trees (bases of a graphic matroid) Goel et al [19] show a lower bound of $\Omega(n)$, while in the case where $\mathcal{F}$ corresponds to the cardinality constraint $\{S : |S| \geq k\}$ Svitkina and Fleischer [60] show a lower bound of $\tilde{\Omega}(\sqrt{n})$. There are a few exceptions. The problem can be solved exactly when $\mathcal{F}$ is a ring family ([56]), triple family ([24]), or parity family ([21]). In the context of NP-Hard problems, there are almost no cases where good (say $O(1)$ or $O(\log n)$) approximations exist. We have that the submodular vertex cover admits a 2-approximation ([19, 31]), and the k-uniform hitting set has $O(k)$-approximation.

Multivariate Problems. The notion of k-multi-submodularity already appeared (under the name of multidimensional submodularity) in the classical work of Fisher et al [16], where they consider the multivariate monotone maximization problem with $\mathcal{F} = \{V\}$ as a motivating example for submodular maximization subject to a matroid constraint. They show that for this problem a simple greedy algorithm achieves a 1/2-approximation. The work of Singh et al [58] considers the special case of 2-multi-submodular functions (they call them *simple bisubmodular*). They give constant factor approximations for maximizing monotone 2-multi-submodular functions under cardinality and partition constraints, and provide applications to coupled sensor placement and coupled feature selection problems.

Other different extensions of submodular functions to multivariate settings have been studied. Some of these include bisubmodular functions [54, 1, 18, 3], k-submodular functions [28, 68, 53], or skew bisubmodular functions [29, 63, 62].

Finally, as mentioned in the introduction, an important class of (multi-agent submodular optimization) problems arises when $f(S_1, \ldots, S_k) = \sum_{i\in[k]} f_i(S_i)$. These problems have been widely studied in the case where $\mathcal{F} = \{V\}$, both for minimization ([26, 61, 13, 9]) and maximization ([16, 45, 64]), and have also been considered for more general families [19, 55].

2 Multivariate submodular maximization

We describe two different reductions. The first one reduces the capacitated multivariate problem CMVSO to a single-agent SO problem, and it is based on the simple idea of taking

k disjoint copies of the original ground set. We use this to establish an (optimal) MV gap of 1 for families such as spanning trees, matroids, and p-systems. The second reduction is based on the multilinear extension of a set function. We show that if the single-agent problem admits approximation via its multilinear relaxation (see Section 2.2), then we may extend this to its multivariate version with a constant factor loss, in the monotone and nonmonotone settings. For the monotone case the MV gap is tight.

2.1 The lifting reduction

We describe a generic reduction of CMVSO to a single-agent SO problem

$$\max / \min \bar{f}(S) : S \in \mathcal{L}.$$

The argument is based on the idea of viewing assignments of elements v to agents i in a *multi-agent bipartite graph*. This simple idea (which is equivalent to making k disjoint copies of the ground set) already appeared in the classical work of Fisher et al [16], and has since then been widely used [45, 64, 8, 58, 55]. We review briefly the reduction here for completeness and to fix notation.

Consider the complete bipartite graph $G = ([k] + V, E)$. Every subset of edges $S \subseteq E$ can be written uniquely as $S = \uplus_{i\in[k]}(\{i\} \times S_i)$ for some sets $S_i \subseteq V$. This allows us to go from a multivariate objective (such as the one in (4)) to a univariate objective $\bar{f} : 2^E \to \mathbb{R}$ over the lifted space. Namely, for each set $S \subseteq E$ we define $\bar{f}(S) = f(S_1, S_2, \ldots, S_k)$. The function $\bar{f}$ is well-defined because of the one-to-one correspondence between sets $S \subseteq E$ and tuples $(S_1, \ldots, S_k) \subseteq V^k$.

We consider two families of sets over E that capture the original constraints:

$$\mathcal{F}' := \{S \subseteq E : S_1 \uplus \cdots \uplus S_k \in \mathcal{F}\} \quad \text{and} \quad \mathcal{H} := \{S \subseteq E : S_i \in \mathcal{F}_i,\ \forall i \in [k]\}.$$

We now have:

$$\begin{array}{ccccccccc} \max/\min & f(S_1, S_2, \ldots, S_k) & = & \max/\min & \bar{f}(S) & = & \max/\min & \bar{f}(S) & \\ \text{s.t.} & S_1 \uplus \cdots \uplus S_k \in \mathcal{F} & & \text{s.t.} & S \in \mathcal{F}' \cap \mathcal{H} & & \text{s.t.} & S \in \mathcal{L} & , \\ & S_i \in \mathcal{F}_i\,,\ \forall i \in [k] & & & & & & & \end{array}$$

where in the last step we just let $\mathcal{L} := \mathcal{F}' \cap \mathcal{H}$.

Clearly, this reduction is interesting if our new function $\bar{f}$ and the family of sets $\mathcal{L}$ have properties which allow us to handle them computationally. This depends on the original structure of the function f, and the set families $\mathcal{F}$ and $\mathcal{F}_i$. The following is straightforward.

▶ **Claim 10.** *If f is a (nonnegative, respectively monotone) k-multi-submodular function, then $\bar{f}$ as defined above is also (nonnegative, respectively monotone) submodular.*

In Section 2.4 we discuss several properties of the families $\mathcal{F}$ and $\mathcal{F}_i$ that are preserved under this reduction, as well as their algorithmic consequences.

2.2 Multilinear extensions for MV problems

Given a set function $f : 2^V \to \mathbb{R}$ (or equivalently $f : \{0,1\}^n \to \mathbb{R}$), we say that $g : [0,1]^n \to \mathbb{R}$ is an *extension* of f if $g(\chi^S) = f(S)$ for each $S \subseteq V$. Clearly, there are many possible extensions that one could consider for any given set function. One that has been very useful in the submodular maximization setting due to its nice properties is the *multilinear extension.*

For a set function $f : \{0,1\}^V \to \mathbb{R}$ we define its *multilinear extension* $f^M : [0,1]^V \to \mathbb{R}$ (introduced in [7]) as

$$f^M(z) = \sum_{S \subseteq V} f(S) \prod_{v \in S} z_v \prod_{v \notin S} (1 - z_v).$$

An alternative way to define f^M is in terms of expectations. Given a vector $z \in [0,1]^V$ let R^z denote a random set that contains element v_i independently with probability z_{v_i}. Then $f^M(z) = \mathbb{E}[f(R^z)]$, where the expectation is taken over random sets generated from the probability distribution induced by z. One very useful property of the multilinear extension is the following.

▶ **Proposition 11.** *Let $f : 2^V \to \mathbb{R}$ be a submodular function and $f^M : [0,1]^n \to \mathbb{R}$ its multilinear extension. Then f^M is convex along any direction $d = \mathbf{e_{v_i}} - \mathbf{e_{v_j}}$ for $i, j \in \{1, 2, \ldots, n\}$, where $\mathbf{e_v}$ denotes the characteristic vector of $\{v\}$, i.e. the vector in $\mathbb{R}^V$ which has value 1 in the v-th component and zero elsewhere.*

This now gives rise to natural single-agent and multivariate relaxations. The *single-agent multilinear extension relaxation* is:

$$\text{(SA-ME)} \qquad \max f^M(z) : z \in P^*(\mathcal{F}), \tag{5}$$

and the *multivariate multilinear extension relaxation* is:

$$\text{(MV-ME)} \qquad \max \bar{f}^M(z_1, z_2, \ldots, z_k) : z_1 + z_2 + \cdots + z_k \in P^*(\mathcal{F}), \tag{6}$$

where $P^*(\mathcal{F})$ denotes some relaxation of the polytope $conv(\{\chi^S : S \in \mathcal{F}\})$ [3] , and $\bar{f}$ the lifted univariate function from the reduction in Section 2.1. Note that $\bar{f}$ is defined over vectors $\bar{z} = (z_1, z_2, \ldots, z_k) \in [0,1]^E$, where we think of $z_i \in \mathbb{R}^n$ as the vector associated to agent i.

The relaxation SA-ME has been used extensively [8, 43, 15, 12, 5] in the submodular maximization literature. The following result shows that when f is nonnegative submodular and the relaxation $P^*(\mathcal{F})$ is downwards closed and admits a polytime separation oracle, the relaxation SA-ME can be solved approximately in polytime.

▶ **Theorem 12** ([5, 64])**.** *Let $f : 2^V \to \mathbb{R}_+$ be a nonnegative submodular function and $f^M : [0,1]^V \to \mathbb{R}_+$ its multilinear extension. Let $P \subseteq [0,1]^V$ be any downwards closed polytope that admits a polytime separation oracle, and denote $OPT = \max f^M(z) : z \in P$. Then there is a polytime algorithm ([5]) that finds $z^* \in P$ such that $f^M(z^*) \geq 0.385 \cdot OPT$. Moreover, if f is monotone there is a polytime algorithm ([64]) that finds $z^* \in P$ such that $f^M(z^*) \geq (1 - 1/e)OPT$.*

For monotone objectives the assumption that P is downwards closed is without loss of generality. This is not the case, however, when the objective is nonmonotone. Nonetheless, this restriction is unavoidable, as Vondrák [66] showed that no algorithm can find $z^* \in P$ such that $f^M(z^*) \geq c \cdot OPT$ for any constant $c > 0$ when P admits a polytime separation oracle but it is not downwards closed.

We can solve the MV-ME relaxation to the same approximation factor that SA-ME. To see this note that the multivariate problem has the form $\{\max g(w) : w \in W \subseteq \mathbf{R}^{nk}\}$

[3] $conv(X)$ denotes the convex hull of a set X of vectors, and χ^S denotes the characteristic vector of the set S.

where W is the downwards closed polytope $\{w = (z_1, ..., z_k) : \sum_i z_i \in P^*(\mathcal{F})\}$ and $g(w) = \bar{f}^M(z_1, z_2, \dots, z_k)$. Clearly we have a polytime separation oracle for W given that we have one for $P^*(\mathcal{F})$. Moreover, g is the multilinear extension of a nonnegative submodular function (since by Claim 10 we know $\bar{f}$ is nonnegative submodular), and we can now use Theorem 12.

2.3 A tight $1 - 1/e$ MV gap

In this section we prove Theorem 4. The main idea is that we start with an (approximate) optimal solution $z^* = z_1^* + z_2^* + \cdots + z_k^*$ to the MV-ME relaxation and build a new feasible solution $\hat{z} = \hat{z}_1 + \hat{z}_2 + \cdots + \hat{z}_k$ where the $\hat{z}_i$ have supports V_i that are pairwise disjoint. We think of V_i as the set of items associated (or pre-assigned) to agent i. Once we have such a pre-assignment we consider the single-agent problem $\max g(S) : S \in \mathcal{F}$ where

$$g(S) = f(S \cap V_1, S \cap V_2, \dots, S \cap V_k). \tag{7}$$

It is clear that g is nonnegative monotone submodular since f is nonnegative monotone k-multi-submodular. Moreover, for any feasible solution $S \in \mathcal{F}$ for this single-agent problem, we obtain a multivariate solution of the same cost by setting $S_i = S \cap V_i$, since then $g(S) = f(S \cap V_1, S \cap V_2, \dots, S \cap V_k) = f(S_1, S_2, \dots, S_k)$.

For a set $S \subseteq V$ and a vector $z \in [0,1]^V$ we denote by $z|_S$ the truncation of z to elements of S. That is, we set $z|_S(v) = z(v)$ for each $v \in S$ and to zero otherwise. Then by definition of g we have that $g^M(z) = \bar{f}^M(z|_{V_1}, z|_{V_2}, \dots, z|_{V_k})$, where $\bar{f}$ is the lifted function from Section 2.1. Moreover, if the sets V_i are pairwise disjoint, then $\bar{f}^M(z|_{V_1}, z|_{V_2}, \dots, z|_{V_k}) = \bar{f}^M(z_1, z_2, \dots, z_k)$. The next result formalizes this observation.

▶ **Proposition 13.** *Let $z = \sum_{i \in [k]} z_i$ be a feasible solution to MV-ME such that the vectors z_i have pairwise disjoint supports V_i. Then $g^M(z) = \bar{f}^M(z_1, z_2, \dots, z_k)$.*

We now have all the ingredients to prove our main result for maximization. We note that a gap of $1 - 1/e$ appeared in [55] for the case of separable objectives $f(S_1, \dots, S_k) = \sum_i f_i(S_i)$. That argument uses the component-wise linearity of the multilinear extension, while our proof for non-separable objectives strongly uses the convexity property from Proposition 11.

▶ Theorem 4. If there is a (polytime) $\alpha(n)$-approximation for monotone SO($\mathcal{F}$) maximization based on rounding SA-ME, then there is a (polytime) $(1 - 1/e) \cdot \alpha(n)$-approximation for monotone MVSO($\mathcal{F}$) maximization. Furthermore, given a downwards closed family $\mathcal{F}$, if there is a (polytime) $\alpha(n)$-approximation for nonmonotone SO($\mathcal{F}$) maximization based on rounding SA-ME, then there is a (polytime) $0.385 \cdot \alpha(n)$-approximation for nonmonotone MVSO($\mathcal{F}$) maximization.

Proof. We discuss first the case of monotone objectives.

STEP 1. Let $z^* = z_1^* + z_2^* + \cdots + z_k^*$ denote an approximate solution to MV-ME obtained via Theorem 12, and let OPT_{frac} be the value of an optimal solution. We then have that $f^M(z_1^*, z_2^*, \dots, z_k^*) \geq (1 - 1/e) OPT_{frac} \geq (1 - 1/e) OPT_{MV}$.

STEP 2. For an element $v \in V$ let $\mathbf{e_v}$ denote the characteristic vector of $\{v\}$, i.e. the vector in $\mathbb{R}^V$ which has value 1 in the v-th component and zero elsewhere. Then by Proposition 11 we have that the function

$$h(t) = \bar{f}^M(z_1^*, z_2^*, \dots, z_{i-1}^*, z_i^* + t\mathbf{e_v}, z_{i+1}^*, \dots, z_{i'-1}^*, z_{i'}^* - t\mathbf{e_v}, z_{i'+1}^*, \dots, z_k^*)$$

is convex for any $v \in V$ and $i \neq i' \in [k]$. In particular, given any $v \in V$ such that there exist $i \neq i' \in [k]$ with $z_i^*(v), z_{i'}^*(v) > 0$, there is always a choice so that increasing one component and decreasing the other by the same amount does not decrease the objective value.

Let $v \in V$ be such that there exist $i \neq i' \in [k]$ with $z_i^*(v), z_{i'}^*(v) > 0$. Then, we either set $z_i^*(v) = z_i^*(v) + z_{i'}^*(v)$ and $z_{i'}^*(v) = 0$, or $z_{i'}^*(v) = z_i^*(v) + z_{i'}^*(v)$ and $z_i^*(v) = 0$, whichever does not decrease the objective value. We repeat until the vectors z_i^* have pairwise disjoint support. Let us denote these new vectors by $\hat{z}_i$ and let $\hat{z} = \sum_{i \in [k]} \hat{z}_i$. Then notice that the vector $z^* = \sum_{i \in [k]} z_i^*$ remains invariant after performing each of the above updates (i.e. $\hat{z} = z^*$), and hence the new vectors $\hat{z}_i$ remain a feasible solution.

STEP 3. In the last step we use the function g defined in (7), with sets V_i corresponding to the supports of the $\hat{z}_i$. Given our α-approximation rounding assumption for SA-ME, we can round $\hat{z}$ to find a set $\hat{S}$ such that $g(\hat{S}) \geq \alpha g^M(\hat{z})$. Then, by setting $\hat{S}_i = \hat{S} \cap V_i$ we obtain a multivariate solution satisfying

$$f(\hat{S}_1, \ldots, \hat{S}_k) = g(\hat{S}) \geq \alpha g^M(\hat{z}) = \alpha f^M(\hat{z}_1, \ldots, \hat{z}_k) \geq \alpha f^M(z_1^*, \ldots, z_k^*) \geq \alpha(1-1/e) OPT_{MV},$$

where the second equality follows from Proposition 13. This completes the monotone proof.

For the nonmonotone case the argument is very similar. Here we restrict our attention to downwards closed families, since then we can get a 0.385-approximation at STEP 1 via Theorem 12. We then apply STEP 2 and 3 in the same fashion as we did for monotone objectives. This leads to a $0.385 \cdot \alpha(n)$-approximation for the multivariate problem. ◀

2.4 Invariance under the lifting reduction

In Section 2.3 we established a MV gap of $1 - 1/e$ for monotone objectives and of 0.385 for nonmonotone objectives and downwards closed families based on the multilinear formulations. In this section we describe several families with an (optimal) MV gap of 1. Examples of such family classes include spanning trees, matroids, and p-systems.

We saw in Section 2.1 that if the original function f is k-multi-submodular then the lifted function $\bar{f}$ is submodular. We now discuss some properties of the original families $\mathcal{F}_i$ and $\mathcal{F}$ that are also preserved under the lifting reduction; these were already proved in [55]. It is shown there, for instance, that if $\mathcal{F}$ induces a matroid (or more generally a p-system) over the ground set V, then so does the family $\mathcal{F}'$ over the lifted space E. We summarize these results in Table 1, and discuss next some of the algorithmic consequences.

Table 1 Invariant properties under the lifting reduction

	Multivariate problem	**Single-agent (i.e. reduced) problem**	Result
1	f k-multi-submodular	$\bar{f}$ submodular	Section 2.1
2	f monotone	$\bar{f}$ monotone	Section 2.1
3	$(V, \mathcal{F})$ a p-system	$(E, \mathcal{F}')$ a p-system	[55]
4	$\mathcal{F}$ = bases of a p-system	$\mathcal{F}'$ = bases of a p-system	[55]
5	$(V, \mathcal{F})$ a matroid	$(E, \mathcal{F}')$ a matroid	[55]
6	$\mathcal{F}$ = bases of a matroid	$\mathcal{F}'$ = bases of a matroid	[55]
7	$(V, \mathcal{F})$ a p-matroid intersection	$(E, \mathcal{F}')$ a p-matroid intersection	[55]
8	$(V, \mathcal{F}_i)$ a matroid for all $i \in [k]$	$(E, \mathcal{H})$ a matroid	[55]
9	$\mathcal{F}_i$ a ring family for all $i \in [k]$	$\mathcal{H}$ a ring family	[55]
10	$\mathcal{F}$ = forests (resp. spanning trees)	$\mathcal{F}'$ = forests (resp. spanning trees)	[55]
11	$\mathcal{F}$ = matchings (resp. perfect matchings)	$\mathcal{F}'$ = matchings (resp. perfect matchings)	[55]
12	$\mathcal{F}$ = st-paths	$\mathcal{F}'$ = st-paths	[55]

In the setting of MVSO (i.e. (2)) this invariance allows us to leverage several results from the single-agent to the multivariate setting. These are based on the following result, which uses the fact that the size of the lifted space E is nk.

▶ **Theorem 14.** *Let $\mathcal{F}$ be a matroid, a p-matroid intersection, or a p-system. If there is a (polytime) $\alpha(n)$-approximation algorithm for monotone (resp. nonmonotone) SO($\mathcal{F}$) maximization (resp. minimization), then there is a (polytime) $\alpha(nk)$-approximation algorithm for monotone (resp. nonmonotone) MVSO($\mathcal{F}$) maximization (resp. minimization).*

For both monotone and nonmonotone maximization the approximation factors $\alpha(n)$ for the family classes described in Theorem 14 are independent of (the size of the ground set) n. Hence, we immediately get that $\alpha(nk) = \alpha(n)$ for those cases, and thus approximation factors for the corresponding multivariate and single-agent problems are the same. In our MV gap terminology this implies an MV gap of 1 for such problems. This proves Theorem 5.

In the setting of CMVSO (i.e. (4)) the results described on entries 8 and 9 of Table 1 provide additional modelling flexibility. This allows us to maintain decent approximations while combining several constraints. For instance, for a monotone maximization instance of CMVSO where $\mathcal{F}$ corresponds to a p-matroid intersection and the $\mathcal{F}_i$ are all matroids, the above invariance results lead to a $(\frac{1}{p+1} - \epsilon)$-approximation.

The results from this section also imply that algorithms that behave very well in practice (such as accelerated greedy variants [49] and distributed algorithms [50]) for the corresponding single-agent problems, can also be used for the more general multivariate setting while preserving the same approximation guarantees. We believe this makes the CMVSO framework a good candidate for potential applications in large-scale machine learning problems.

3 Multivariate submodular minimization

In this section we present different approximation factors in terms of n (i.e. the number of items) and k (i.e. the number of agents) for the monotone multivariate problem. Moreover, the approximation factors in terms of n are essentially tight.

3.1 A $\frac{\beta \cdot n}{1+(n-1)(1-c)}$-approximation

Let $f_S(v) = f(S+v) - f(S)$ denote the marginal gain of adding v to S. Given a normalized monotone submodular function $f : 2^V \to \mathbb{R}$, its *total curvature* c and its *curvature* $c(S)$ *with respect to a set* $S \subseteq V$ are defined as (in [10, 65])

$$c = \max_{j \in V} \frac{f(j) - f_{V-j}(j)}{f(j)} = 1 - \min_{j \in V} \frac{f_{V-j}(j)}{f(j)} \quad \text{and} \quad c(S) = 1 - \min_{j \in S} \frac{f_{S-j}(j)}{f(j)}.$$

We may think of this number as indicating how far the function f is from being modular (with $c = 0$ corresponding to being modular). The notion of curvature has been widely used for univariate functions [10, 65, 33, 2], since it allows for better approximations and it is linear time computable.

Given a tuple $(S_1, \ldots, S_k) \in 2^{kV}$ and $(i, v) \in [k] \times V$, we denote by $(S_1, \ldots, S_k) + (i, v)$ the new tuple $(S_1, \ldots, S_{i-1}, S_i + v, S_{i+1}, \ldots, S_k)$. It is natural to think of the quantity

$$f_{(S_1,\ldots,S_k)}((i,v)) := f((S_1, \ldots, S_k) + (i, v)) - f(S_1, \ldots, S_k)$$

as the marginal gain of assigning element v to agent i in the tuple $(S_1, \ldots, S_k)$. We also use $f((i, v))$ to denote the quantity $f(\emptyset, \ldots, \emptyset, v, \emptyset, \ldots, \emptyset)$ where v appears in the *ith* component.

Given a normalized monotone k-multi-submodular function $f : 2^{kV} \to \mathbb{R}$ we define its *total curvature* c and its *curvature* $c(S_1, \ldots, S_k)$ *with respect to a tuple* $(S_1, \ldots, S_k) \subseteq V^k$ as

$$c = 1 - \min_{i \in [k], v \in V} \frac{f_{(V,V,\ldots,V)-(i,v)}((i,v))}{f((i,v))} \quad , \quad c(S_1, \ldots, S_k) = 1 - \min_{i \in [k], v \in S_i} \frac{f_{(S_1,\ldots,S_k)-(i,v)}((i,v))}{f((i,v))}$$

There is a straightforward correspondence between the curvature of f and the curvature of its lifted version $\bar{f}$.

▶ **Claim 15.** *Let $f : 2^{kV} \to \mathbb{R}_+$ be a normalized nonnegative monotone k-multi-submodular function, and $\bar{f} : 2^E \to \mathbb{R}_+$ the corresponding lifted function. Then, f has total curvature c if and only if $\bar{f}$ has total curvature c. Also, f has curvature $c(S_1, \ldots, S_k)$ with respect to a tuple if and only if $\bar{f}$ has curvature $c(S)$ with respect to the set S in the lifted space corresponding to the tuple $(S_1, \ldots, S_k)$.*

The following curvature dependent result for univariate functions was proved in [33].

▶ **Proposition 16** ([33])**.** *Let $f : 2^V \to \mathbb{R}$ be a nonnegative monotone submodular function, and $w : V \to \mathbb{R}_+$ the modular function given by $w(v) = f(v)$. Let $c(S)$ denote the curvature of f with respect to S, and S^* denote an optimal solution to $\min f(S) : S \in \mathcal{F}$. Let $\hat{S} \in \mathcal{F}$ be a β-approximation for the problem $\min w(S) : S \in \mathcal{F}$. Then*

$$f(\hat{S}) \leq \frac{\beta|S^*|}{1 + (|S^*| - 1)(1 - c(S^*))} f(S^*).$$

We extend the above result to the setting of k-multi-submodular objectives.

▶ Theorem 6. Let f be a nonnegative monotone k-multi-submodular function, and let $\mathcal{F}$ be a family that admits a (polytime) β-approximation over modular functions. Denote by $(S_1^*, \ldots, S_k^*)$ an optimal solution to monotone MVSO($\mathcal{F}$) minimization, and by $c(S_1^*, \ldots, S_k^*)$ the curvature of f with respect to $(S_1^*, \ldots, S_k^*)$. Then there is a (polytime) $\frac{\beta \sum_{i \in [k]} |S_i^*|}{1 + (\sum_{i \in [k]} |S_i^*| - 1)(1 - c(S_1^*, \ldots, S_k^*))}$-approximation algorithm for monotone MVSO($\mathcal{F}$) minimization.

Proof. Let $\bar{f} : 2^E \to \mathbb{R}_+$ and $\mathcal{F}'$ be the lifted function and family described in the lifting reduction from Section 2.1. We then have

$$\begin{array}{llcll} \min & f(S_1, S_2, \ldots, S_k) & = & \min & \bar{f}(S) \\ \text{s.t.} & S_1 \uplus \cdots \uplus S_k \in \mathcal{F} & & \text{s.t.} & S \in \mathcal{F}' \end{array}.$$

Define a modular function $\bar{w} : E \to \mathbb{R}_+$ over the edges of the bipartite graph by $\bar{w}(i, v) = \bar{f}(i, v)$. Also, let $OPT = \min \bar{f}(S) : S \in \mathcal{F}'$ and denote by S^* such a minimizer. Then by Proposition 16 we have that any β-approximation for the modular minimization problem $\min \bar{w}(S) : S \in \mathcal{F}'$ is a $\beta|S^*|/(1 + (|S^*| - 1)(1 - c(S^*)))$-approximation for the problem $\min \bar{f}(S) : S \in \mathcal{F}'$ (and hence also for our original multivariate problem). Moreover, notice that by Claim 15 the curvature $c(S^*)$ of $\bar{f}$ with respect to S^* is the same as the curvature $c(S_1^*, \ldots, S_k^*)$ of f with respect to $(S_1^*, \ldots, S_k^*)$, where $(S_1^*, \ldots, S_k^*)$ is the tuple associated to S^*. Thus, we immediately get the desired approximation assuming that a (polytime) β-approximation is available for $\min \bar{w}(S) : S \in \mathcal{F}'$.

However, the lifted family $\mathcal{F}'$ could be more complicated than $\mathcal{F}$, and hence we would like to have an assumption depending on the original $\mathcal{F}$ (and not $\mathcal{F}'$). This can be achieved at no extra loss using the modularity of $\bar{w}$. Indeed, we can define a new modular function $w : V \to \mathbb{R}_+$ as $w(v) = \text{argmin}_{i \in [k]} \bar{w}(i, v)$ for each $v \in V$, breaking ties arbitrarily. It is then clear that $\min \bar{w}(S) : S \in \mathcal{F}' = \min w(S) : S \in \mathcal{F}$, since $\bar{w}$ always uses the cheapest copy of v (i.e. assign v to the agent i with the smallest cost for it).

We then get that any β-approximation for $\min w(S) : S \in \mathcal{F}$ is also a β-approximation for $\min \bar{w}(S) : S \in \mathcal{F}'$, and hence the desired result in terms of $\mathcal{F}$ follows. ◀

3.2 MV gap of k

Due to monotonicity, one may often assume that we are working with a family $\mathcal{F}$ which is *upwards-closed*, aka a *blocking family*. This can be done without loss of generality even if we seek polytime algorithms, since separation over a polytope with vertices $\{\chi^F : F \in \mathcal{F}\}$ implies separation over its dominant. We refer the reader to Appendix C for details.

For a normalized set function $f : \{0,1\}^V \to \mathbb{R}$ one can define its *Lovász extension* $f^L : \mathbb{R}^V_+ \to \mathbb{R}$ (introduced in [47]) as follows. Let $0 < v_1 < v_2 < ... < v_m$ be the distinct positive values taken in some vector $z \in \mathbb{R}^V_+$, and let $v_0 = 0$. For each $i \in \{0, 1, ..., m\}$ define the set $S_i := \{j : z_j > v_i\}$. In particular, S_0 is the support of z and $S_m = \emptyset$. One then defines:

$$f^L(z) = \sum_{i=0}^{m-1} (v_{i+1} - v_i) f(S_i).$$

It follows from the definition that f^L is positively homogeneous, that is $f^L(\alpha z) = \alpha f^L(z)$ for any $\alpha > 0$ and $z \in \mathbb{R}^V_+$. Moreover, it is also straightforward to see that f^L is a monotone function if f is. We have the following result due to Lovász.

▶ **Lemma 17.** *[Lovász [47]] The function f^L is convex if and only if f is submodular.*

This now gives rise to natural convex relaxations for the single-agent and multivariate problems based on some fractional relaxation $P^*(\mathcal{F})$ of the integral polyhedron $conv(\{\chi^S : S \in \mathcal{F}\})$. The *single-agent Lovász extension formulation* (used in [31, 32]) is:

$$\text{(SA-LE)} \qquad \min f^L(z) : z \in P^*(\mathcal{F}), \tag{8}$$

and the *multivariate Lovász extension formulation* is:

$$\text{(MV-LE)} \qquad \min \bar{f}^L(z_1, z_2, \ldots, z_k) : z_1 + z_2 + \cdots + z_k \in P^*(\mathcal{F}), \tag{9}$$

where $\bar{f}$ is the lifted univariate function from Section 2.1.

By standard methods (e.g. see [55]) one may solve SA-LE in polytime if one can separate over the relaxation $P^*(\mathcal{F})$. This is often the case for many natural families such as spanning trees, perfect matchings, *st*-paths, and vertex covers.

We can also solve MV-LE as long as we have polytime separation of $P^*(\mathcal{F})$. This follows from the fact that the multivariate problem has the form $\{\min g(w) : w \in W \subseteq \mathbf{R}^{nk}\}$ where W is the full-dimensional convex body $\{w = (z_1, ..., z_k) : \sum_i z_i \in P^*(\mathcal{F})\}$ and $g(w) = \bar{f}^L(z_1, z_2, \ldots, z_k)$. Clearly we have a polytime separation oracle for W given that we have one for $P^*(\mathcal{F})$. Moreover, by Lemma 17 and Claim 10, g is convex since it is the Lovász extension of a nonnegative submodular function $\bar{f}$. Hence we may apply Ellipsoid as in the single-agent case.

We now give an approximation in terms of the number of agents, which becomes preferable when k is not too large. The high-level idea behind our reduction is the same as in the maximization setting (see Section 2.3). That is, we start with an optimal solution $z^* = z_1^* + z_2^* + \cdots + z_k^*$ to the multivariate MV-LE relaxation and build a new feasible solution $\hat{z} = \hat{z}_1 + \hat{z}_2 + \cdots + \hat{z}_k$ where the $\hat{z}_i$ have supports V_i that are pairwise disjoint. We then use for the rounding step the single-agent problem (as previously defined in (7) for the maximization setting) $\min g(S) : S \in \mathcal{F}$ where $g(S) = f(S \cap V_1, S \cap V_2, \ldots, S \cap V_k)$.

Similarly to Proposition 13 which dealt with the multilinear extension, we have the following result for the Lovász extension.

▶ **Proposition 18.** *Let $z = z_1 + z_2 + \cdots + z_k$ be a feasible solution to MV-LE such that the vectors z_i have pairwise disjoint supports V_i. Then $g^L(z) = \bar{f}^L(z_1, z_2, \ldots, z_k)$.*

▶ Theorem 8. Suppose there is a (polytime) $\alpha(n)$-approximation for monotone SO($\mathcal{F}$) minimization based on rounding SA-LE. Then there is a (polytime) $k\alpha(n)$-approximation for monotone MVSO($\mathcal{F}$) minimization.

Proof. Let $z^* = z_1^* + z_2^* + \cdots + z_k^*$ denote an optimal solution to MV-LE with value OPT_{frac}. We build a new feasible solution $\hat{z} = \hat{z}_1 + \hat{z}_2 + \cdots + \hat{z}_k$ as follows. For each element $v \in V$ let $i' = \text{argmax}_{i\in[k]}\, z_i^*(v)$, breaking ties arbitrarily. Then set $\hat{z}_{i'}(v) = kz_i^*(v)$ and $\hat{z}_i(v) = 0$ for each $i \neq i'$. By construction we have $\hat{z} \geq z^*$, and hence this is indeed a feasible solution. Moreover, by construction we also have that $\hat{z}_i \leq kz_i^*$ for each $i \in [k]$. Hence, given the monotonicity and homogeneity of $\bar{f}^L$ it follows that

$$\bar{f}^L(\hat{z}_1, \hat{z}_2, \ldots, \hat{z}_k) \leq \bar{f}^L(kz_1^*, kz_2^*, \ldots, kz_k^*) = k\bar{f}^L(z_1^*, z_2^*, \ldots, z_k^*) = k{\cdot}OPT_{frac} \leq k{\cdot}OPT_{MV}.$$

Since the $\hat{z}_i$ have disjoint supports V_i, for the single-agent rounding step we can now use the function g defined in (7) with the sets V_i. Given our α-approximation rounding assumption for SA-LE, we can round $\hat{z}$ to find a set $\hat{S}$ such that $g(\hat{S}) \leq \alpha g^L(\hat{z})$. Then, by setting $\hat{S}_i = \hat{S} \cap V_i$ we obtain a multivariate solution satisfying

$$f(\hat{S}_1, \hat{S}_2, \ldots, \hat{S}_k) = g(\hat{S}) \leq \alpha g^L(\hat{z}) = \alpha\bar{f}^L(\hat{z}_1, \hat{z}_2, \ldots, \hat{z}_k) \leq \alpha k \cdot OPT_{MV},$$

where the second equality follows from Proposition 18. This completes the proof. ◀

The above theorem has interesting consequences. We now discuss one that leads to a polytime k-approximation for a much more general version of the submodular facility location problem considered by Sviktina and Tardos [61], where k denotes the number of facilities.

▶ **Corollary 19.** *There is a polytime k-approximation for the monotone MVSO($\mathcal{F}$) minimization problem over $\mathcal{F} = \{V\}$.*

Proof. Notice that the single-agent version of the above multivariate problem is the trivial $\min f(S) : S \in \{V\}$. Hence a polytime exact algorithm is available for the single-agent problem and thus a polytime k-approximation is available for the multivariate version. ◀

3.3 An $o(\frac{n}{\log n})$ lower bound hardness for $\mathcal{F} = \{V\}$

In this section we focus on the special case where $\mathcal{F} = \{V\}$. That is, we are looking for the optimal splitting of all the elements among the agents. We show that the curvature dependent approximation factors obtained in Theorem 6 are essentially tight.

We follow a technique from [20, 14, 60] and build two multivariate submodular functions that are hard to distinguish with high probability for any (even randomized) algorithm.

Assume that $k = n$, and let $\mathcal{R} := (R_1, R_2, \ldots, R_n) \subseteq V^n$ be a random partition of V. Notice that $\sum_{i=1}^n |R_i| = n$. Let $\beta = \omega(\log n)$ and such that β is integer. Consider the two nonnegative monotone n-multi-submodular functions $f_1, f_2 : 2^{nV} \to \mathbb{R}_+$ given by:

$$f_1(S_1, \ldots, S_n) = \min\{n, \sum_{i=1}^n |S_i|\} \ \ , \ \ f_2(S_1, \ldots, S_n) = \min\{f_1(S_1, \ldots, S_n), \beta + \sum_{i=1}^n |S_i \cap \bar{R}_i|\}, \tag{10}$$

where $\bar{R}_i$ denotes the complement of the set R_i, i.e. $\bar{R}_i = V - R_i$.

The work of Svitkina and Fleischer [60] show the following result for univariate functions.

▶ **Lemma 20** ([60]). *Let f_1 and f_2 be two set functions, with f_2, but not f_1, parametrized by a string of random bits r. If for any set S, chosen without knowledge of r, the probability (over r) that $f_1(S) \neq f_2(S)$ is $n^{-\omega(1)}$, then any algorithm that makes a polynomial number of oracle queries has probability at most $n^{-\omega(1)}$ of distinguishing f_1 and f_2.*

The above clearly generalizes to the setting of tuples (i.e. multivariate objectives) in a natural and straightforward way. The only difference is that our ground set in the lifted space has now size n^2 instead of n.

▶ **Lemma 21.** *Let f_1 and f_2 be two n-multivariate set functions, with f_2, but not f_1, parametrized by a string of random bits r. If for any tuple $(S_1, \ldots, S_n)$, chosen without knowledge of r, the probability (over r) that $f_1(S_1, \ldots, S_n) \neq f_2(S_1, \ldots, S_n)$ is $n^{-\omega(1)}$, then any algorithm that makes a polynomial number of oracle queries has probability at most $n^{-\omega(1)}$ of distinguishing f_1 and f_2.*

We can use Lemma 21 to show the following result for the functions defined in (10).

▶ **Lemma 22.** *Any algorithm that makes a polynomial number of oracle calls has probability $n^{-\omega(1)}$ of distinguishing the functions f_1 and f_2 above.*

Proof. By Lemma 21 it suffices to show that for any tuple $(S_1, \ldots, S_n)$ the probability (over the random choice of the partition $\mathcal{R}$) that $f_1(S_1, \ldots, S_n) \neq f_2(S_1, \ldots, S_n)$ is at most $n^{-\omega(1)}$.

Let us denote this probability by $p(S_1, \ldots, S_n)$. We first show that $p(S_1, \ldots, S_n)$ is maximized for tuples $(S_1, \ldots, S_n)$ satisfying $\sum_{i=1}^n |S_i| = n$. First suppose that $\sum_{i=1}^n |S_i| > n$. Then $p(S_1, \ldots, S_n) = \mathbb{P}[\beta + \sum_{i=1}^n |S_i \cap \bar{R}_i| < n]$. But this probability can only increase if an element is removed from some set S_i. Similarly, in the case where $\sum_{i=1}^n |S_i| < n$, we get $p(S_1, \ldots, S_n) = \mathbb{P}[\beta + \sum_{i=1}^n |S_i \cap \bar{R}_i| < \sum_{i=1}^n |S_i|] = \mathbb{P}[\sum_{i=1}^n |S_i \cap R_i| > \beta]$. But this probability can only increase if an element is added to some set S_i.

So let $(S_1, \ldots, S_n)$ be any fixed tuple satisfying $\sum_{i=1}^n |S_i| = n$, and let $m_v := \sum_{i: S_i \ni v} 1$ denote the number of sets S_i that contain a copy of v. Note that $\sum_{v \in V} m_v = \sum_{i=1}^n |S_i| = n$. Let us consider a random partition $\mathcal{R} = (R_1, R_2, \ldots, R_n)$ which is obtained by placing each element $v \in V$ independently and uniformly at random into one of the sets $R_1, R_2, \ldots, R_n$. Let X_v be a random variable for each $v \in V$, defined by $X_v = \sum_{i=1}^n |S_i \cap R_i \cap \{v\}|$. That is, $X_v = 1$ if v is assigned to an R_i such that $S_i \ni v$ (which happens with probability m_v/n), and $X_v = 0$ otherwise. Clearly, the random variables $\{X_v\}_{v \in V}$ are pairwise independent. Moreover, we have that the expected value of $\sum_{i=1}^n |S_i \cap R_i|$ is given by

$$\mu := \mathbb{E}[\sum_{i=1}^n |S_i \cap R_i|] = \mathbb{E}[\sum_{v \in V} X_v] = \sum_{v \in V} \mathbb{E}[X_v] = \sum_{v \in V} \frac{m_v}{n} = 1.$$

Then, by Chernoff bounds and using that β is an integer we obtain

$$\begin{aligned} p(S_1, \ldots, S_n) =& \mathbb{P}[\sum_{i=1}^n |S_i \cap R_i| > \beta] = \mathbb{P}[\sum_{i=1}^n |S_i \cap R_i| \geq \beta + 1] \\ =& \mathbb{P}[\sum_{v \in V} X_v \geq (1+\beta)\mu] \leq e^{-\mu\beta/3} = e^{-\beta/3} = e^{-\omega(\log n)} = n^{-\omega(1)}. \end{aligned}$$

◀

We now prove our (curvature independent) lower bound result.

▶ **Theorem 23.** *The monotone MVSO($\mathcal{F}$) minimization problem over $\mathcal{F} = \{V\}$ cannot be approximated to a ratio $o(n/\log n)$ in the value oracle model with polynomially many queries.*

Proof. Assume there is a polytime algorithm achieving an approximation factor of $\alpha = o(n/\log n)$. Choose $\beta = \omega(\log n)$ such that $\alpha\beta < n$. Consider the output of the algorithm when f_2 is given as input. The optimal solution in this case is the partition $\mathcal{R} = (R_1, \ldots, R_k)$, with $f_2(R_1, \ldots, R_k) = \beta$. So the algorithm produces a feasible solution (i.e. a partition) $(S_1^*, \ldots, S_k^*)$ satisfying $f_2(S_1^*, \ldots, S_k^*) \leq \alpha\beta < n$. However, since f_1 takes value exactly n over any partition, there is no feasible solution $(S_1, \ldots, S_n)$ such that $f_1(S_1, \ldots, S_n) < n$. This means that if the input is the function f_1 then the algorithm produces a different answer, thus distinguishing between f_1 and f_2, contradicting Lemma 22. ◀

The above result contrasts with the known $O(\log n)$ approximation ([61]) for the case where the multivariate objective is separable, that is $f(S_1, \ldots, S_k) = \sum_i f_i(S_i)$. These two facts combined now prove Theorem 3.

We use a construction from Iyer et al [33] to explicitly introduce the effect of curvature into the lower bound. Their work is for univariate functions, but it can be naturally extended to the multivariate setting. We modify the functions f_1, f_2 from (10) as follows:

$$f_i^c(S_1, \ldots, S_k) = c \cdot f_i(S_1, \ldots, S_k) + (1 - c) \sum_{i=1}^{n} |S_i| \quad \text{, for } i = 1, 2.$$

It is then straightforward to check that both f_1^c and f_2^c have total curvature c. Moreover, since $f_1(S_1, \ldots, S_k) = f_2(S_1, \ldots, S_k)$ if and only if $f_1^c(S_1, \ldots, S_k) = f_2^c(S_1, \ldots, S_k)$, by Lemma 22 it follows that any algorithm that makes polynomially many queries is not able to distinguish between f_1^c and f_2^c with high probability. In addition, the gap between the optimal solutions for these two functions is given by

$$\frac{OPT_1}{OPT_2} = \frac{cn + (1-c)n}{c\beta + (1-c)n} = \frac{n}{c\beta + (1-c)n} = \frac{n}{\beta + (n-\beta)(1-c)} = \frac{n/\beta}{1 + (n/\beta - 1)(1-c)}.$$

Then, since $\beta = \omega(\log n)$, the (curvature dependent) lower bound follows.

▶ Theorem 7. The monotone MVSO($\mathcal{F}$) minimization problem over $\mathcal{F} = \{V\}$ and objectives f with total curvature c cannot be approximated to a ratio $o(\frac{n/\log n}{1+(\frac{n}{\log n}-1)(1-c)})$ in the value oracle model with polynomial number of queries.

4 Conclusions

We introduce a new class of multivariate submodular optimization problems, and give information theoretic evidence that this class encodes much more than the separable versions arising in multi-agent objectives. We provide some explicit examples and potential applications.

For maximization, we show that practical algorithms such as accelerated greedy variants and distributed algorithms achieve good approximation guarantees under very general constraints. For arbitrary families, we show MV gaps of $1 - 1/e$ and 0.385 for the monotone and nonmonotone problems respectively, and the MV gap for monotone objectives is tight.

For minimization the news is worse. However, we give (essentially tight) approximation factors with respect to the curvature of the multivariate objective function. This may lead to significant gains in several settings.

References

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

2 Wenruo Bai and Jeffrey A Bilmes. Greed is still good: Maximizing monotone submodular+ supermodular functions. *arXiv preprint arXiv:1801.07413*, 2018.

3 André Bouchet and William H Cunningham. Delta-matroids, jump systems, and bisubmodular polyhedra. *SIAM Journal on Discrete Mathematics*, 8(1):17–32, 1995.

4 Yuri Y Boykov and M-P Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 105–112. IEEE, 2001.

5 Niv Buchbinder and Moran Feldman. Constrained submodular maximization via a nonsymmetric technique. *arXiv preprint arXiv:1611.03253*, 2016.

6 Niv Buchbinder, Moran Feldman, Joseph Seffi, and Roy Schwartz. A tight linear time (1/2)-approximation for unconstrained submodular maximization. *SIAM Journal on Computing*, 44(5):1384–1402, 2015.

7 Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a submodular set function subject to a matroid constraint. In *Integer programming and combinatorial optimization*, pages 182–196. Springer, 2007.

8 Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 40(6):1740–1766, 2011.

9 Chandra Chekuri and Alina Ene. Submodular cost allocation problem and applications. In *International Colloquium on Automata, Languages, and Programming*, pages 354–366. Springer, 2011. Extended version: arXiv preprint arXiv:1105.2040.

10 Michele Conforti and Gérard Cornuéjols. Submodular set functions, matroids and the greedy algorithm: tight worst-case bounds and some generalizations of the rado-edmonds theorem. *Discrete applied mathematics*, 7(3):251–274, 1984.

11 Khalid El-Arini, Gaurav Veda, Dafna Shahaf, and Carlos Guestrin. Turning down the noise in the blogosphere. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 289–298. ACM, 2009.

12 Alina Ene and Huy L Nguyen. Constrained submodular maximization: Beyond 1/e. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, pages 248–257. IEEE, 2016.

13 Alina Ene and Jan Vondrák. Hardness of submodular cost allocation: Lattice matching and a simplex coloring conjecture. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2014)*, 28:144–159, 2014.

14 Uriel Feige, Vahab S Mirrokni, and Jan Vondrak. Maximizing non-monotone submodular functions. *SIAM Journal on Computing*, 40(4):1133–1153, 2011.

15 Moran Feldman, Joseph Naor, and Roy Schwartz. A unified continuous greedy algorithm for submodular maximization. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 570–579. IEEE, 2011.

16 Marshall L Fisher, George L Nemhauser, and Laurence A Wolsey. *An analysis of approximations for maximizing submodular set functions-II.* Springer, 1978.

17 Lisa Fleischer, Michel X Goemans, Vahab S Mirrokni, and Maxim Sviridenko. Tight approximation algorithms for maximum general assignment problems. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 611–620. Society for Industrial and Applied Mathematics, 2006.

18 Satoru Fujishige and Satoru Iwata. Bisubmodular function minimization. *SIAM Journal on Discrete Mathematics*, 19(4):1065–1073, 2005.

19 Gagan Goel, Chinmay Karande, Pushkar Tripathi, and Lei Wang. Approximability of combinatorial problems with multi-agent submodular cost functions. In *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*, pages 755–764. IEEE, 2009.

20 Michel X Goemans, Nicholas JA Harvey, Satoru Iwata, and Vahab Mirrokni. Approximating submodular functions everywhere. In *Proceedings of the twentieth annual ACM-SIAM symposium on Discrete algorithms*, pages 535–544. Society for Industrial and Applied Mathematics, 2009.

21 Michel X. Goemans and VS Ramakrishnan. Minimizing submodular functions over families of sets. *Combinatorica*, 15(4):499–513, 1995.

22 Daniel Golovin and Andreas Krause. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research*, 42:427–486, 2011.

23 Pranava R Goundan and Andreas S Schulz. Revisiting the greedy approach to submodular set function maximization. *Optimization online*, pages 1–25, 2007.

24 Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric algorithms and combinatorial optimization*, volume 2. Springer Science & Business Media, 2012.

25 Anupam Gupta, Aaron Roth, Grant Schoenebeck, and Kunal Talwar. Constrained non-monotone submodular maximization: Offline and secretary algorithms. In *International Workshop on Internet and Network Economics*, pages 246–257. Springer, 2010.

26 Ara Hayrapetyan, Chaitanya Swamy, and Éva Tardos. Network design for information networks. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 933–942. Society for Industrial and Applied Mathematics, 2005.

27 Elad Hazan, Shmuel Safra, and Oded Schwartz. On the complexity of approximating k-set packing. *computational complexity*, 15(1):20–39, 2006.

28 Anna Huber and Vladimir Kolmogorov. Towards minimizing k-submodular functions. In *International Symposium on Combinatorial Optimization*, pages 451–462. Springer, 2012.

29 Anna Huber, Andrei Krokhin, and Robert Powell. Skew bisubmodularity and valued csps. *SIAM Journal on Computing*, 43(3):1064–1084, 2014.

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

31 Satoru Iwata and Kiyohito Nagano. Submodular function minimization under covering constraints. In *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*, pages 671–680. IEEE, 2009.

32 Rishabh Iyer, Stefanie Jegelka, and Jeff Bilmes. Monotone closure of relaxed constraints in submodular optimization: Connections between minimization and maximization: Extended version. 2014.

33 Rishabh K Iyer, Stefanie Jegelka, and Jeff A Bilmes. Curvature and optimal algorithms for learning and minimizing submodular functions. In *Advances in Neural Information Processing Systems*, pages 2742–2750, 2013.

34 Stefanie Jegelka and Jeff Bilmes. Submodularity beyond submodular energies: coupling edges in graph cuts. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1897–1904. IEEE, 2011.

35 David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM, 2003.

36 Subhash Khot, Richard J Lipton, Evangelos Markakis, and Aranyak Mehta. Inapproximability results for combinatorial auctions with submodular utility functions. In *International Workshop on Internet and Network Economics*, pages 92–101. Springer, 2005.

37 Pushmeet Kohli, M Pawan Kumar, and Philip HS Torr. P^3 & beyond: Move making algorithms for solving higher order functions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(9):1645–1656, 2009.

38 Nitish Korula, Vahab Mirrokni, and Morteza Zadimoghaddam. Online submodular welfare maximization: Greedy beats 1/2 in random order. *SIAM Journal on Computing*, 47(3):1056–1086, 2018.

39 Andreas Krause and Carlos Guestrin. Near-optimal observation selection using submodular functions. In *AAAI*, volume 7, pages 1650–1654, 2007.

40 Andreas Krause, Jure Leskovec, Carlos Guestrin, Jeanne VanBriesen, and Christos Faloutsos. Efficient sensor placement optimization for securing large water distribution networks. *Journal of Water Resources Planning and Management*, 134(6):516–526, November 2008.

41 KW Krause, MA Goodwin, and RW Smith. *Optimal software test planning through automated network analysis.* TRW Systems Group, 1973.

42 Ariel Kulik, Hadas Shachnai, and Tami Tamir. Maximizing submodular set functions subject to multiple linear constraints. In *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 545–554. Society for Industrial and Applied Mathematics, 2009.

43 Jon Lee, Vahab S Mirrokni, Viswanath Nagarajan, and Maxim Sviridenko. Non-monotone submodular maximization under matroid and knapsack constraints. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 323–332. ACM, 2009.

44 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.

45 Benny Lehmann, Daniel Lehmann, and Noam Nisan. Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behavior*, 55(2):270–296, 2006. URL: `http://EconPapers.repec.org/RePEc:eee:gamebe:v:55:y:2006:i:2:p:270-296`.

46 Hui Lin and Jeff Bilmes. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 510–520. Association for Computational Linguistics, 2011.

47 László Lovász. Submodular functions and convexity. In *Mathematical Programming The State of the Art*, pages 235–257. Springer, 1983.

48 Vahab Mirrokni, Michael Schapira, and Jan Vondrák. Tight information-theoretic lower bounds for welfare maximization in combinatorial auctions. In *Proceedings of the 9th ACM conference on Electronic commerce*, pages 70–77. ACM, 2008.

49 Baharan Mirzasoleiman, Ashwinkumar Badanidiyuru, and Amin Karbasi. Fast constrained submodular maximization: Personalized data summarization. In *ICML*, pages 1358–1367, 2016.

50 Baharan Mirzasoleiman, Amin Karbasi, Rik Sarkar, and Andreas Krause. Distributed submodular maximization. *The Journal of Machine Learning Research*, 17(1):8330–8373, 2016.

51 George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions - i. *Mathematical Programming*, 14(1):265–294, 1978.

52 George L Nemhauser and Leonard A Wolsey. Best algorithms for approximating the maximum of a submodular set function. *Mathematics of operations research*, 3(3):177–188, 1978.

53 Naoto Ohsaka and Yuichi Yoshida. Monotone k-submodular function maximization with size constraints. In *Advances in Neural Information Processing Systems*, pages 694–702, 2015.

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

55 Richard Santiago and F. Bruce Shepherd. Multi-Agent Submodular Optimization. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2018)*, 116:23:1–23:20, 2018.

56 Alexander Schrijver. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Journal of Combinatorial Theory, Series B*, 80(2):346–355, 2000.

57 Alexander Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer, 2003.

58 Ajit Singh, Andrew Guillory, and Jeff Bilmes. On bisubmodular maximization. In *Artificial Intelligence and Statistics*, pages 1055–1063, 2012.

59 Matthew Streeter and Daniel Golovin. An online algorithm for maximizing submodular functions. In *Advances in Neural Information Processing Systems*, pages 1577–1584, 2009.

60 Zoya Svitkina and Lisa Fleischer. Submodular approximation: Sampling-based algorithms and lower bounds. *SIAM Journal on Computing*, 40(6):1715–1737, 2011.

61 Zoya Svitkina and ÉVA Tardos. Facility location with hierarchical facility costs. *ACM Transactions on Algorithms (TALG)*, 6(2):37, 2010.

62 Johan Thapper and Stanislav Zivny. The power of linear programming for valued csps. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 669–678. IEEE, 2012.

63 Johan Thapper and Stanislav Živnỳ. The complexity of finite-valued csps. *Journal of the ACM (JACM)*, 63(4):37, 2016.

64 Jan Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 67–74. ACM, 2008.

65 Jan Vondrák. Submodularity and curvature: The optimal algorithm (combinatorial optimization and discrete algorithms). 2010.

66 Jan Vondrák. Symmetry and approximability of submodular maximization problems. *SIAM Journal on Computing*, 42(1):265–304, 2013.

67 Jan Vondrák, Chandra Chekuri, and Rico Zenklusen. Submodular function maximization via the multilinear relaxation and contention resolution schemes. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 783–792. ACM, 2011.

68 Justin Ward and Stanislav Živnỳ. Maximizing bisubmodular and k-submodular functions. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1468–1481. SIAM, 2014.

A Properties of k-multi-submodular functions

In this section we discuss several properties of k-multi-submodular functions. We see that some of the characterizations and results that hold for univariate submodular functions extend naturally to the multivariate setting.

We start by showing that our definition of submodularity in the multivariate setting captures the diminishing return property. Recall that we usually think of the pair $(i, v) \in [k] \times V$ as the assignment of element v to agent i. We use this to introduce some notation for adding an element to a tuple.

▶ **Definition 24.** Given a tuple $(S_1, \ldots, S_k) \in 2^{kV}$ and $(i, v) \in [k] \times V$, we denote by $(S_1, \ldots, S_k) + (i, v)$ the new tuple $(S_1, \ldots, S_{i-1}, S_i + v, S_{i+1}, \ldots, S_k)$.

Then, it is natural to think of the quantity

$$f(S_1, \ldots, S_{i-1}, S_i + v, S_{i+1}, \ldots, S_k) - f(S_1, \ldots, S_{i-1}, S_i, S_{i+1}, \ldots, S_k) \tag{11}$$

as the marginal gain of assigning element v to agent i in the tuple $(S_1, \dots, S_k)$. Notice that with the notation introduced in Definition 24 we have that (11) can be also written as

$$f((S_1, \dots, S_k) + (i, v)) - f(S_1, \dots, S_k).$$

This leads to the following diminishing returns characterizations in the multivariate setting.

▶ **Proposition 25.** *A multivariate function $f : 2^{kV} \to \mathbb{R}$ is k-multi-submodular if and only if for all tuples $(S_1, \dots, S_k) \subseteq (T_1, \dots, T_k)$ and $(i, v) \in [k] \times V$ such that $v \notin T_i$ we have*

$$f((S_1, \dots, S_k) + (i, v)) - f(S_1, \dots, S_k) \geq f((T_1, \dots, T_k) + (i, v)) - f(T_1, \dots, T_k). \quad (12)$$

Proof. We make use of the lifting reduction presented in Section 2.1. Let $\bar{f} : 2^E \to \mathbb{R}$ denote the lifted function, and let $S, T \subseteq E$ be the sets in the lifted space corresponding to the tuples $(S_1, \dots, S_k)$ and $(T_1, \dots, T_k)$ respectively. Then, since $(S_1, \dots, S_k) \subseteq (T_1, \dots, T_k)$, we know that $S \subseteq T$. Moreover, notice that

$$f((S_1, \dots, S_k) + (i, v)) - f(S_1, \dots, S_k) = \bar{f}(S + (i, v)) - \bar{f}(S)$$

and

$$f((T_1, \dots, T_k) + (i, v)) - f(T_1, \dots, T_k) = \bar{f}(T + (i, v)) - \bar{f}(T).$$

In addition, from Claim 10 in Section 2.1 we know that f is k-multi-submodular if and only if $\bar{f}$ is submodular. Then the result follows by observing the following.

$$\begin{aligned} & f \text{ is } k\text{-multi-submodular} \\ \iff & \bar{f} \text{ is submodular} \\ \iff & \bar{f}(S + (i, v)) - \bar{f}(S) \geq \bar{f}(T + (i, v)) - \bar{f}(T) \quad \text{for all } S \subseteq T \text{ and } (i, v) \notin T \\ \iff & f((S_1, \dots, S_k) + (i, v)) - f(S_1, \dots, S_k) \geq f((T_1, \dots, T_k) + (i, v)) - f(T_1, \dots, T_k) \\ & \text{for all } (S_1, \dots, S_k) \subseteq (T_1, \dots, T_k) \text{ and } v \notin T_i. \end{aligned}$$

◀

The proof of the above result also shows the following characterization of k-multi-submodular functions.

▶ **Proposition 26.** *A multivariate function $f : 2^{kV} \to \mathbb{R}$ is k-multi-submodular if and only if for all tuples $(S_1, \dots, S_k)$ and $(i, v), (j, u) \in [k] \times V$ such that $v \notin S_i$ and $u \notin S_j$ we have*

$$f((S_1, \dots, S_k)+(i, v))-f(S_1, \dots, S_k) \geq f((S_1, \dots, S_k)+(j, u)+(i, v))-f((S_1, \dots, S_k)+(j, u)). \quad (13)$$

B Examples of k-multi-submodular functions

We now provide some explicit examples of k-multi-submodular functions that lead to interesting applications.

▶ **Lemma 27.** *Consider a multilinear function $h : \mathbb{Z}_+^k \to \mathbb{R}$ given by $h(z) = \sum_{S \subseteq [k]} a_S \prod_{m \in S} z_m$. Let $f : 2^{kV} \to \mathbb{R}$ be a multivariate set function defined as $f(S_1, \dots, S_k) = h(|S_1|, \dots, |S_k|)$. Then, f is k-multi-submodular if and only if*

$$a_S \leq 0 \quad \forall S \subseteq [k]. \quad (14)$$

Proof. By Proposition 26 we know that f is k-multi-submodular if and only if condition (13) is satisfied. Let $(S_1, \ldots, S_k)$ be an arbitrary tuple and let $(i, v), (j, u) \in [k] \times V$ such that $v \notin S_i, u \notin S_j$. Denote by z^0 the integer vector with components $z^0_i = |S_i|$. That is, $z^0 = (|S_1|, |S_2|, \ldots, |S_k|) \in \mathbb{Z}^k_+$. We call z^0 the cardinality vector associated to the tuple $(S_1, \ldots, S_k)$. In a similar way, let z^1 be the cardinality vector associated to the tuple $(S_1, \ldots, S_k) + (i, v)$, z^2 the cardinality vector associated to $(S_1, \ldots, S_k) + (j, u)$, and z^3 the cardinality vector associated to $(S_1, \ldots, S_k) + (i, v) + (j, u)$. Now notice that condition (13) can be written as

$$h(z^1) - h(z^0) \geq h(z^3) - h(z^2) \tag{15}$$

for all $z^0, z^1, z^2, z^3 \in \mathbb{Z}^k_+$ such that $z^1 = z^0 + \mathbf{e_i}$, $z^2 = z^0 + \mathbf{e_j}$, and $z^3 = z^0 + \mathbf{e_i} + \mathbf{e_j}$, where $\mathbf{e_i}$ is the characteristic vector on the ith component, and similarly for $\mathbf{e_j}$.

We show that (15) is equivalent to (14). Using that $z^1_m = z^0_m$ for all $m \neq i$ and $z^1_i = z^0_i + 1$, we have

$$\begin{aligned}
h(z^1) - h(z^0) &= \sum_{S \subseteq [k]} a_S \prod_{m \in S} z^1_m - \sum_{S \subseteq [k]} a_S \prod_{m \in S} z^0_m \\
&= \sum_{S \subseteq [k]} a_S [\prod_{m \in S} z^1_m - \prod_{m \in S} z^0_m] \\
&= \sum_{S \ni i} a_S [\prod_{m \in S} z^1_m - \prod_{m \in S} z^0_m] \\
&= \sum_{S \ni i} a_S [(z^0_i + 1) \prod_{m \in S, m \neq i} z^0_m - \prod_{m \in S} z^0_m] \\
&= \sum_{S \ni i} a_S \prod_{m \in S, m \neq i} z^0_m .
\end{aligned}$$

Similarly, using that $z^3_m = z^2_m$ for all $m \neq i$ and $z^3_i = z^2_i + 1$, we have

$$\begin{aligned}
h(z^3) - h(z^2) &= \sum_{S \ni i} a_S \prod_{m \in S, m \neq i} z^2_m \\
&= \sum_{S \ni i, S \ni j} a_S \prod_{m \in S, m \neq i} z^2_m + \sum_{S \ni i, S \not\ni j} a_S \prod_{m \in S, m \neq i} z^2_m \\
&= \sum_{S \ni i, S \ni j} a_S (z^0_j + 1) \prod_{m \in S, m \neq i,j} z^0_m + \sum_{S \ni i, S \not\ni j} a_S \prod_{m \in S, m \neq i} z^0_m \\
&= \sum_{S \ni i} a_S \prod_{m \in S, m \neq i} z^0_m + \sum_{S \ni i, S \ni j} a_S \prod_{m \in S, m \neq i,j} z^0_m \\
&= h(z^1) - h(z^0) + \sum_{S \ni i, S \ni j} a_S \prod_{m \in S, m \neq i,j} z^0_m ,
\end{aligned}$$

where in the third equality we use that $z^2 = z^0 + \mathbf{e_j}$. Thus, we have

$$h(z^1) - h(z^0) \geq h(z^3) - h(z^2) \iff \sum_{S \ni i, S \ni j} a_S \prod_{m \in S, m \neq i,j} z^0_m \leq 0.$$

Since the above must hold for all $z^0, z^1, z^2, z^3 \in \mathbb{Z}^k_+$ and $i, j \in [k]$ such that $z^1 = z^0 + \mathbf{e_i}$, $z^2 = z^0 + \mathbf{e_j}$, and $z^3 = z^0 + \mathbf{e_i} + \mathbf{e_j}$, we immediately get that (15) is equivalent to (14) as we wanted to show. ◀

▶ **Lemma 28.** *Consider a quadratic function $h : \mathbb{Z}_+^k \to \mathbb{R}$ given by $h(z) = z^T Az$ for some matrix $A = (a_{ij})$. Let $f : 2^{kV} \to \mathbb{R}$ be a multivariate set function defined as $f(S_1, \ldots, S_k) = h(|S_1|, \ldots, |S_k|)$. Then, f is k-multi-submodular if and only if A satisfies*

$$a_{ij} + a_{ji} \leq 0 \quad \forall i, j \in [k]. \tag{16}$$

Proof. The proof is very similar to that of Lemma 27. By Proposition 26 we know that f is k-multi-submodular if and only if condition (13) is satisfied. Let $(S_1, \ldots, S_k)$ be an arbitrary tuple and let $(i, v), (j, u) \in [k] \times V$ such that $v \notin S_i, u \notin S_j$. Denote by z^0 the integer vector with components $z_i^0 = |S_i|$. That is, $z^0 = (|S_1|, |S_2|, \ldots, |S_k|) \in \mathbb{Z}_+^k$. We call z^0 the cardinality vector associated to the tuple $(S_1, \ldots, S_k)$. In a similar way, let z^1 be the cardinality vector associated to the tuple $(S_1, \ldots, S_k) + (i, v)$, z^2 the cardinality vector associated to $(S_1, \ldots, S_k) + (j, u)$, and z^3 the cardinality vector associated to $(S_1, \ldots, S_k) + (i, v) + (j, u)$. Now notice that condition (13) can be written as

$$h(z^1) - h(z^0) \geq h(z^3) - h(z^2) \tag{17}$$

for all $z^0, z^1, z^2, z^3 \in \mathbb{Z}_+^k$ such that $z^1 = z^0 + \mathbf{e_i}$, $z^2 = z^0 + \mathbf{e_j}$, and $z^3 = z^0 + \mathbf{e_i} + \mathbf{e_j}$, where $\mathbf{e_i}$ is the characteristic vector on the *ith* component, and similarly for $\mathbf{e_j}$.

We show that (17) is equivalent to (16). First notice that for a vector $z = (z_1, \ldots, z_k)$ the function h can be written as $h(z) = \sum_{\ell,m=1}^k a_{\ell m} z_\ell z_m$. Then, using that $z_\ell^1 = z_\ell^0$ for all $\ell \neq i$ and $z_i^1 = z_i^0 + 1$, we have

$$h(z^1) - h(z^0) = \sum_{\ell,m=1}^k a_{\ell m} z_\ell^1 z_m^1 - \sum_{\ell,m=1}^k a_{\ell m} z_\ell^0 z_m^0 = \sum_{\ell=1}^k a_{\ell i} z_\ell^0 + \sum_{m=1}^k a_{im} z_m^0 + a_{ii}.$$

Similarly, using that $z_\ell^3 = z_\ell^2$ for all $\ell \neq i$ and $z_i^3 = z_i^2 + 1$, we have

$$h(z^3) - h(z^2) = \sum_{\ell=1}^k a_{\ell i} z_\ell^2 + \sum_{m=1}^k a_{im} z_m^2 + a_{ii}.$$

Thus, using that $z^2 = z^0 + \mathbf{e_j}$ we get

$$\begin{aligned}
& h(z^1) - h(z^0) \geq h(z^3) - h(z^2) \\
\iff & \sum_{\ell=1}^k a_{\ell i} z_\ell^0 + \sum_{m=1}^k a_{im} z_m^0 + a_{ii} \;\geq\; \sum_{\ell=1}^k a_{\ell i} z_\ell^2 + \sum_{m=1}^k a_{im} z_m^2 + a_{ii} \\
\iff & \sum_{\ell=1}^k a_{\ell i}(z_\ell^0 - z_\ell^2) + \sum_{m=1}^k a_{im}(z_m^0 - z_m^2) \geq 0 \\
\iff & -a_{ji} - a_{ij} \geq 0 \\
\iff & a_{ji} + a_{ij} \leq 0.
\end{aligned}$$

◀

C Upwards-closed (aka blocking) families

In this section, we give some background for blocking families. As our work for minimization is restricted to monotone functions, we can often convert an arbitrary set family into its upwards-closure (i.e., a blocking version of it) and work with it instead. We discuss this reduction as well. The technical details discussed in this section are fairly standard and we include them for completeness.

A set family $\mathcal{F}$ over a ground set V is *upwards-closed* if $F \subseteq F'$ and $F \in \mathcal{F}$, implies that $F' \in \mathcal{F}$; these are sometimes referred to as *blocking families*. Examples of such families include vertex covers or set covers more generally, whereas spanning trees are not.

C.1 Reducing to blocking families

Now consider an arbitrary set family $\mathcal{F}$ over V. We may define its *upwards closure* by $\mathcal{F}^{\uparrow} = \{F' : F \subseteq F' \textit{ for some } F \in \mathcal{F}\}$. In this section we argue that in order to solve a monotone optimization problem over sets in $\mathcal{F}$ it is often sufficient to work over its upwards-closure.

This requires two ingredients. First, we need a separation algorithm for the relaxation $P^*(\mathcal{F})$, but indeed this is often available for many natural families such as spanning trees, perfect matchings, *st*-paths, and vertex covers. The second ingredient needed is the ability to turn an integral solution $\chi^{F'}$ from $P^*(\mathcal{F}^{\uparrow})$ or $P(\mathcal{F}^{\uparrow})$ into an integral solution $\chi^F \in P(\mathcal{F})$. We now argue that this is the case if a polytime separation algorithm is available for $P^*(\mathcal{F}^{\uparrow})$ or for the polytope $P(\mathcal{F}) := conv(\{\chi^F : F \in \mathcal{F}\})$.

For a polyhedron P, we denote its *dominant* by $P^{\uparrow} := \{z : z \geq x \textit{ for some } x \in P\}$. The following observation is straightforward.

▶ **Claim 29.** *Let H be the set of vertices of the hypercube in $\mathbb{R}^V$. Then*

$$H \cap P(\mathcal{F}^{\uparrow}) = H \cap P(\mathcal{F})^{\uparrow} = H \cap P^*(\mathcal{F}^{\uparrow}).$$

In particular we have that $\chi^S \in P(\mathcal{F})^{\uparrow} \iff \chi^S \in P^*(\mathcal{F}^{\uparrow})$.

We can now use this observation to prove the following.

▶ **Lemma 30.** *Assume we have a separation algorithm for $P^*(\mathcal{F}^{\uparrow})$. Then for any $\chi^S \in P^*(\mathcal{F}^{\uparrow})$ we can find in polytime $\chi^M \in P(\mathcal{F})$ such that $\chi^M \leq \chi^S$.*

Proof. Let $S = \{1, 2, \ldots, k\}$. We run the following routine until no more elements can be removed:

For $i \in S$
  If $\chi^{S-i} \in P^*(\mathcal{F}^{\uparrow})$ then $S = S - i$

Let χ^M be the output. We show that $\chi^M \in P(\mathcal{F})$. Since $\chi^M \in P^*(\mathcal{F}^{\uparrow})$, by Claim 29 we know that $\chi^M \in P(\mathcal{F})^{\uparrow}$. Then by definition of dominant there exists $x \in P(\mathcal{F})$ such that $x \leq \chi^M \in P(\mathcal{F})^{\uparrow}$. It follows that the vector x can be written as $x = \sum_i \lambda_i \chi^{U_i}$ for some $U_i \in \mathcal{F}$ and $\lambda_i \in (0, 1]$ with $\sum_i \lambda_i = 1$. Clearly we must have that $U_i \subseteq M$ for all i, otherwise x would have a non-zero component outside M. In addition, if for some i we have $U_i \subsetneq M$, then there must exist some $j \in M$ such that $U_i \subseteq M - j \subsetneq M$. Hence $M - j \in \mathcal{F}^{\uparrow}$, and thus $\chi^{M-j} \in P(\mathcal{F})^{\uparrow}$ and $\chi^{M-j} \in P^*(\mathcal{F}^{\uparrow})$. But then when component j was considered in the algorithm above, we would have had S such that $M \subseteq S$ and so $\chi^{S-j} \in P^*(\mathcal{F}^{\uparrow})$ (that is $\chi^{S-j} \in P(\mathcal{F})^{\uparrow}$), and so j should have been removed from S, contradiction. ◀

We point out that for many natural set families $\mathcal{F}$ we can work with the relaxation $P^*(\mathcal{F}^{\uparrow})$ assuming that it admits a separation algorithm. Then, if we have an algorithm which produces $\chi^{F'} \in P^*(\mathcal{F}^{\uparrow})$ satisfying some approximation guarantee for a monotone problem, we can use Lemma 30 to construct in polytime $F \in \mathcal{F}$ which obeys the same guarantee.

Moreover, notice that for Lemma 30 to work we do not need an actual separation oracle for $P^*(\mathcal{F}^\uparrow)$, but rather all we need is to be able to separate over $0-1$ vectors only. Hence, since the polyhedra $P^*(\mathcal{F}^\uparrow)$, $P(\mathcal{F}^\uparrow)$ and $P(\mathcal{F})^\uparrow$ have the same $0-1$ vectors (see Claim 29), a separation oracle for either $P(\mathcal{F}^\uparrow)$ or $P(\mathcal{F})^\uparrow$ would be enough for the routine of Lemma 30 to work. We now show that this is the case if we have a polytime separation oracle for $P(\mathcal{F})$. The following result shows that if we can separate efficiently over $P(\mathcal{F})$ then we can also separate efficiently over the dominant $P(\mathcal{F})^\uparrow$.

▶ **Claim 31.** *If we can separate over a polyhedron P in polytime, then we can also separate over its dominant $P^\uparrow$ in polytime.*

Proof. Given a vector y, we can decide whether $y \in P^\uparrow$ by solving

$$\begin{aligned} x + s &= y \\ x &\in P \\ s &\geq 0. \end{aligned}$$

Since can we easily separate over the first and third constraints, and a separation oracle for P is given (i.e. we can also separate over the set of constraints imposed by the second line), it follows that we can separate over the above set of constraints in polytime. ◀

Now we can apply the same mechanism from Lemma 30 to turn feasible sets from $\mathcal{F}^\uparrow$ into feasible sets in $\mathcal{F}$.

▶ **Corollary 32.** *Assume we have a separation algorithm for $P(\mathcal{F})^\uparrow$. Then for any $\chi^S \in P(\mathcal{F})^\uparrow$ we can find in polytime $\chi^M \in P(\mathcal{F})$ such that $\chi^M \leq \chi^S$.*

We conclude this section by making the remark that if we have an algorithm which produces $\chi^{F'} \in P(\mathcal{F}^\uparrow)$ satisfying some approximation guarantee for a monotone problem, we can use Corollary 32 to construct $F \in \mathcal{F}$ which obeys the same guarantee.