Multi-Agent Submodular Optimization

Richard Santiago
McGill University, Montreal, Canada
richard.santiagotorres@mail.mcgill.ca

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

Abstract

Recent years have seen 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 $f : 2^V \to \mathbb{R}$ is submodular. This progress has been coupled with a wealth of new applications for these models. Our focus is on a more general class of *multi-agent submodular optimization* (MASO) $\min / \max \sum_{i=1}^{k} f_i(S_i) : S_1 \uplus S_2 \uplus \cdots \uplus S_k \in \mathcal{F}$. Here we use $\uplus$ to denote disjoint union and hence this model is attractive where resources are being allocated across k agents, each with its own submodular cost function $f_i()$. This was introduced in the minimization setting by Goel et al. In this paper we explore the extent to which the approximability of the multi-agent problems are linked to their single-agent versions, referred to informally as the *multi-agent gap*.

We present different reductions that transform a multi-agent problem into a single-agent one. For minimization, we show that (MASO) has an $O(\alpha \cdot \min\{k, \log^2(n)\})$-approximation whenever (SO) admits an α-approximation over the convex formulation. In addition, we discuss the class of "bounded blocker" families where there is a provably tight $O(\log n)$ multi-agent gap between (MASO) and (SO). For maximization, we show that monotone (resp. nonmonotone) (MASO) 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; and the $1 - 1/e$ multi-agent gap for monotone objectives is tight. We also discuss several families (such as spanning trees, matroids, and p-systems) that have an (optimal) multi-agent gap of 1. These results substantially expand the family of tractable models for submodular maximization.

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

Keywords and phrases submodular optimization, multi-agent, approximation algorithms

Digital Object Identifier 10.4230/LIPIcs.APPROX-RANDOM.2018.23

Related Version The full version of this work can be found in [35], `https://arxiv.org/abs/1803.03767`.

Acknowledgements We thank Chandra Chekuri for valuable comments and suggestions.

1 Introduction

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 $f(S) \leq f(T)$ whenever $S \subseteq T$. Throughout, we usually assume that our functions are nonnegative and satisfy $f(\emptyset) = 0$. We work in the *value oracle model*, where for a given set S we can query the oracle to find its value $f(S)$.

Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2018).
Editors: Eric Blais, Klaus Jansen, José D. P. Rolim, and David Steurer; Article No. 23; pp. 23:1–23:20
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

For a family of feasible sets $\mathcal{F} \subseteq 2^V$ on a finite ground set V 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 V. There has been an impressive recent stream of activity around these problems for a variety of set families $\mathcal{F}$. We explore the connections between these (single-agent) problems and their multi-agent incarnations. In the *multi-agent (MA)* version, we have k agents each of which has an associated nonnegative submodular set function f_i, $i \in [k]$. 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 $\sum_i f_i(S_i)$. This leads to the multi-agent submodular optimization (MASO) versions:

$$\text{MASO}(\mathcal{F}) \qquad \text{Min} \ / \ \text{Max} \ \ \textstyle\sum_{i=1}^k f_i(S_i) : S_1 \uplus S_2 \uplus \cdots \uplus S_k \in \mathcal{F}. \tag{2}$$

The special case when $\mathcal{F} = \{V\}$ has been previously examined both for minimization (the minimum submodular cost allocation problem [17, 39, 7, 5]) and maximization (submodular welfare problem [29, 40]). For general families $\mathcal{F}$, however, we are only aware of the development in Goel et al. [12] for the minimization setting. A natural first question is whether any multi-agent problem could be directly reduced (or encoded) to a single-agent one over the same ground set V. Goel et al. give an explicit example where such a reduction does not exist. More emphatically, they show that when $\mathcal{F}$ consists of vertex covers in a graph, the *single-agent (SA)* version (i.e., (1)) has a 2-approximation while the MA version has an inapproximability lower bound of $\Omega(\log n)$.

Our first main objective is to explain the extent to which approximability for multi-agent problems is intrinsically connected to their single-agent versions, which we also refer to as the *primitive* associated with $\mathcal{F}$. We refer to the *multi-agent (MA) gap* as the approximation-factor loss incurred by moving to the MA setting.

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

$$\text{CMASO}(\mathcal{F}) \qquad \begin{aligned} \max / \min \quad & \textstyle\sum_{i=1}^k f_i(S_i) \\ \text{s.t.} \quad & S_1 \uplus \cdots \uplus S_k \in \mathcal{F} \\ & S_i \in \mathcal{F}_i \,,\ \forall i \in [k] \end{aligned} \tag{3}$$

where we are supplied with subfamilies $\mathcal{F}_i$. Many existing applications fit into this framework and some of these can be enriched through the added flexibility of the capacitated model. We illustrate this with concrete examples in the next section.

Prior work in both the single and multi-agent settings is summarized in Section 1.2. Our main contributions are presented in Section 1.3.

1.1 Some applications of (capacitated) multi-agent optimization

In this section we present several problems in the literature which are special cases of Problem (2) and the more general Problem (3). We also indicate how the extra generality of CMASO (i.e. (3)) gives modelling advantages. We start with the maximization setting.

▶ **Example 1** (The Submodular Welfare Problem). The most basic problem in the maximization setting arises when we take the family $\mathcal{F} = \{V\}$. This describes a well-known model (introduced in [29]) for allocating goods to agents, each of which has a monotone submodular valuation (utility) function over baskets of goods. This is formulated as (2) by considering nonnegative monotone functions f_i and $\mathcal{F} = \{V\}$. The CMASO framework allows us to incorporate additional constraints by defining the families $\mathcal{F}_i$ appropriately. For instance, one can impose cardinality constraints on the number of elements that an agent can take, or to only allow agent i to take a set S_i of elements satisfying some bounds $L_i \subseteq S_i \subseteq U_i$.

▶ **Example 2** (The Separable Assignment Problem)**.** An instance of the Separable Assignment Problem (SAP) consists of m items and n bins. Each bin j has an associated downwards closed collection of feasible sets $\mathcal{F}_j$, and a modular function $v_j(i)$ that denotes the value of placing item i in bin j. The goal is to choose disjoint feasible sets $S_j \in \mathcal{F}_j$ so as to maximize $\sum_{j=1}^{n} v_j(S_j)$. This well-studied problem ([11, 14, 4]) corresponds to a CMASO instance with modular objectives, $\mathcal{F} = 2^V$, and downwards closed families $\mathcal{F}_i$.

▶ **Example 3** (Sensor Placement)**.** The problem of placing sensors and information gathering has been popular in the submodularity literature [24, 26, 25]. 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 number of sensors that can be deployed. The goal is to place sensors at some of the locations so as to maximize the total "informativeness". Consider a multi-agent objective function $\sum_{i \in [k]} f_i(S_i)$, where $f_i(S_i)$ measures the informativeness of placing sensors S_i at location i. It is then natural to consider a diminishing return (i.e. submodularity) property for the f_i. We can formulate this problem as MASO where $\mathcal{F} := \{S \subseteq V : |S| \leq b\}$ imposes the budget constraint. We can also use CMASO for additional modelling flexibility. For instance, we may define $\mathcal{F}_i = \{S \subseteq V_i : |S| \leq b_i\}$ where V_i are the allowed sensors for location i and b_i an upper bound on the sensors located there.

We now discuss applications of MASO and CMASO in the minimization setting.

▶ **Example 4** (Minimum Submodular Cost Allocation)**.** The most basic problem in the minimization setting arises when we simply take $\mathcal{F} = \{V\}$. This problem, $\min \sum_{i=1}^{k} f_i(S_i) : S_1 \uplus \cdots \uplus S_k = V$, has been widely considered in the literature for both monotone [39] and nonmonotone functions [5, 7], and is referred to as the Minimum Submodular Cost Allocation (MSCA) problem[1] (introduced in [17, 39] and further developed in [5]). This is formulated as (2) by taking $\mathcal{F} = \{V\}$. The CMASO framework allows us to incorporate additional constraints into this problem. The most natural are to impose cardinality constraints on the number of elements that an agent can take, or to only allow agent i to take a set S_i of elements satisfying some bounds $L_i \subseteq S_i \subseteq U_i$.

▶ **Example 5** (Multi-agent Minimization)**.** Goel et al [12] consider the special cases of MASO($\mathcal{F}$) where the objectives are nonnegative monotone submodular and $\mathcal{F}$ is either the family of vertex covers, spanning trees, perfect matchings, or shortest st paths.

1.2 Related work

Single Agent Optimization. Minimizing a submodular function is a classical problem which can be solved in polytime [15, 36, 19]. Unconstrained maximization on the other hand is known to be inapproximable for general submodular functions but admits a polytime constant-factor approximation algorithm when f is nonnegative [2, 8].

For constrained maximization, the classical work [32, 33, 10] established an optimal $(1 - 1/e)$-approximation for nonnegative monotone maximization under a cardinality constraint, and a $(1/(k+1))$-approximation under k matroid constraints. The latter is almost tight since

[1] Sometimes referred to as submodular procurement auctions.

there is an $\Omega(\log(k)/k)$ inapproximability result [18]. For nonnegative monotone functions, [40, 4] give an optimal $(1-1/e)$-approximation based on the multilinear extension when $\mathcal{F}$ is a matroid; and [28] 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 ([1]) is the best factor known for a matroid constraint. In [27] a $1/(k+O(1))$-approximation is given for k matroid constraints with k fixed, and [16] gives a simple "multi-greedy" algorithm that matches the approximation of Lee et al. but is polytime for any k. Finally, Chekuri et al [42] introduce a general framework based on relaxation-and-rounding that allows for combining different types of constraints.

For constrained minimization the news are worse [12, 38, 20]. If $\mathcal{F}$ consists of spanning trees Goel et al [12] show a lower bound of $\Omega(n)$, while if $\mathcal{F}$ corresponds to the cardinality constraint $\{S : |S| \geq k\}$ Svitkina and Fleischer [38] 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 ([36]), triple family ([15]), or parity family ([13]). In the context of NP-Hard problems, there are almost no cases where good approximations exist. We have a 2-approximation ([12, 20]) for submodular vertex cover, and an $O(k)$-approximation for k-uniform hitting set.

Multi-agent Problems. In the maximization side, the main studied problem is Submodular Welfare Maximization ($\mathcal{F} = \{V\}$) for which the initial 1/2-approximation [29] was improved to $1 - 1/e$ by Vondrak [40]. This approximation is in fact optimal [22, 31]. We are not aware of any maximization work for MASO($\mathcal{F}$) for nontrivial families $\mathcal{F}$.

For multi-agent minimization, MSCA (i.e. $\mathcal{F} = \{V\}$) is the most studied application of MASO($\mathcal{F}$). For nonnegative monotone functions, MSCA is equivalent to the Submodular Facility Location problem from [39], where a tight $O(\log n)$ approximation is given. If the functions f_i are nonnegative and nonmonotone, then no multiplicative factor approximation exists [7]. However, the works of [5] and [7] respectively show that $O(\log n)$ and $O(k \log n)$ approximations are available for some special types of nonnegative nonmonotone objectives.

Goel et al [12] consider the minimization case of MASO($\mathcal{F}$) where the objectives are nonnegative and monotone, and $\mathcal{F}$ is a nontrivial collection of subsets of V (i.e. $\mathcal{F} \subset 2^V$). In particular, given a graph G they consider the families of vertex covers, spanning trees, perfect matchings, and shortest st paths. They provide a tight $O(\log n)$ approximation for the vertex cover problem, and show polynomial hardness for the other cases. To the best of our knowledge, [12] is the only work on MASO($\mathcal{F}$) for nontrivial collections $\mathcal{F}$.

1.3 Our contributions

We first discuss the minimization side. Here we mainly focus on nonnegative monotone objectives f_i, due to the strong hardness results discussed in Section 1.2. Our main result is showing that if the SA primitive for a family $\mathcal{F}$ admits approximation via a natural "blocking" convex relaxation (see Section 2.1), then we may extend this to its MA version with a modest blow-up in the approximation factor.

▶ **Theorem 6.** *Suppose there is a (polytime) $\alpha(n)$-approximation for monotone SO($\mathcal{F}$) minimization via the blocking convex relaxation. Then there is a (polytime) $O(\alpha(n) \cdot \min\{k, \log^2(n)\})$ approximation for monotone MASO($\mathcal{F}$) minimization.*

We remark that the $O(\log^2(n))$ factor loss due to having multiple agents (i.e the MA gap) is in the right ballpark, since the vertex cover problem has a factor 2-approximation for single-agent and a tight $O(\log n)$-approximation for the MA version [12].

We also discuss how Goel et al's $O(\log n)$-approximation for MA vertex cover is a special case of a more general phenomenon. Their analysis only relies on the fact that the feasible family (or at least its upwards closure) has a *bounded blocker property*. Given a family $\mathcal{F}$, the *blocker* $\mathcal{B}(\mathcal{F})$ of $\mathcal{F}$ consists of the minimal sets B such that $B \cap F \neq \emptyset$ for each $F \in \mathcal{F}$. We say that $\mathcal{B}(\mathcal{F})$ is β*-bounded* if $|B| \leq \beta$ for all $B \in \mathcal{B}(\mathcal{F})$.

Families with bounded blockers have been previously studied in the SA minimization setting, where the works [23, 21] show that β-approximations are always available. Our next result then establishes an $O(\log n)$ MA gap for bounded blocker families, thus improving the $O(\log^2(n))$ factor in Theorem 6 for general families. We remark that this $O(\log n)$ MA gap is tight due to examples like vertex covers [12], or submodular facility location (a trivial 1-approximation for SA and a tight $O(\log n)$-approximation [39] for MA).

▶ **Theorem 7.** *Let $\mathcal{F}$ be a family with a β-bounded blocker. Then there is a randomized $O(\beta \log n)$-approximation algorithm for monotone $MASO(\mathcal{F})$ minimization.*

While our work mainly focuses on monotone objectives, in the full version [35] we show that upwards closed families with a bounded blocker remain tractable under some special types of nonmonotone objectives introduced by Chekuri and Ene (see Section 2.3).

We conclude our minimization work by discussing a class of families which behaves well for MA minimization despite not having a bounded blocker. More specifically, we observe in Section 2.4 that crossing (and ring) families have an MA gap of $O(\log n)$.

▶ **Theorem 8.** *There is a tight $\ln(n)$-approximation for monotone MASO($\mathcal{F}$) minimization over crossing families $\mathcal{F}$.*

We now discuss our contributions for maximization. Our main result here establishes that if the SA primitive for a family $\mathcal{F}$ admits approximation via its multilinear relaxation (see Section 3.2), then we may extend this to its MA version with a constant factor loss.

▶ **Theorem 9.** *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 MASO($\mathcal{F}$) 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 MASO($\mathcal{F}$) maximization.*

We remark that the $(1 - 1/e)$ MA gap in the monotone case is tight due to examples like $\mathcal{F} = \{V\}$, where there is a trivial 1-approximation for the SA problem and a tight $(1 - 1/e)$-approximation for the MA version [40].

In Section 3 we describe a simple generic reduction that shows that for some families an (optimal) MA gap of 1 holds.

▶ **Theorem 10.** *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) MASO($\mathcal{F}$) maximization.*

In the setting of CMASO (i.e. (3)) our results from Section 3.1 provide additional modelling flexibility. They imply that one maintains decent approximations even while adding interesting side constraints. For instance, for a monotone maximization instance of CMASO where $\mathcal{F}$ corresponds to a p-matroid intersection and the $\mathcal{F}_i$ are all matroids, our results lead to a $(p + 1 + \epsilon)$-approximation algorithm. We believe that these, combined with other results from Section 3, substantially expand the family of tractable models for maximization.

2 Multi-agent submodular minimization

In this section we seek generic reductions for multi-agent minimization problems to their single-agent primitives. We work with a natural blocking convex relaxation that is obtained via the Lovász extension of a set function. We show that if the SA primitive admits approximation via such relaxation, then we may extend this to its MA version up to an $O(\min\{k, \log^2(n)\})$ factor loss. Moreover, as noted already, the $O(\log^2(n))$ approximation factor loss due to having multiple agents is in the right ballpark given the tight $O(\log n)$ MA gap for the vertex cover problem [12]. In Section 2.3 we discuss an extension of this vertex cover result to a larger class of families with a MA gap of $O(\log n)$.

2.1 The single-agent and multi-agent formulations

Due to monotonicity, one may often assume that we are working with a family $\mathcal{F}$ which is *upwards-closed*, aka a *blocking family* (cf. [21]). The advantage is that to certify whether $F \in \mathcal{F}$, we only need to check that $F \cap B \neq \emptyset$ for each element B of the family $\mathcal{B}(\mathcal{F})$ of minimal blockers of $\mathcal{F}$. We discuss the details in Appendix A.

For a set function $f : \{0,1\}^V \to \mathbb{R}$ with $f(\emptyset) = 0$ one defines its *Lovász extension* $f^L : \mathbb{R}^V_+ \to \mathbb{R}$ (introduced in [30]) 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 $i \in \{0, 1, ..., m\}$ let $S_i := \{j : z_j > v_i\}$. 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 $z \in \mathbb{R}^V_+$. It is also straightforward that f^L is monotone if f is. We use both of these properties of f^L in our proofs. We also have the following result due to Lovász.

▶ **Lemma 11** (Lovász [30]). *The function f^L is convex if and only if f is submodular.*

This gives rise to natural convex relaxations for the single-agent and multi-agent problems based on the blocking formulation $P^*(\mathcal{F}) := \{z \geq 0 : z(B) \geq 1 \text{ for all } B \in \mathcal{B}(\mathcal{F})\}$ (see Appendix A for details). The *single-agent Lovász extension formulation* (used in [20, 21]) is:

$$\text{(SA-LE)} \qquad \min f^L(z) : z \in P^*(\mathcal{F}), \tag{4}$$

and the *multi-agent Lovász extension formulation* (used in [5] for $\mathcal{F} = \{V\}$) is:

$$\text{(MA-LE)} \qquad \min \sum_{i \in [k]} f_i^L(z_i) : z_1 + z_2 + \cdots + z_k \in P^*(\mathcal{F}). \tag{5}$$

By standard methods (see Appendix B) one may solve these problems in polytime if one can separate over the blocking formulation $P^*(\mathcal{F})$. This is the case for many natural families such as spanning trees, perfect matchings, *st*-paths, and vertex covers.

It is shown in [5] that in the setting of monotone objectives and $\mathcal{F} = \{V\}$, a fractional solution of (MA-LE) can be rounded into an integral one at an $O(\log n)$ factor loss.

▶ **Theorem 12** ([5]). *Let $z_1 + z_2 + \cdots + z_k$ be a feasible solution for (MA-LE) in the setting where $\mathcal{F} = \{V\}$ (i.e. $\sum_{i \in [k]} z_i = \chi^V$) and f_i are nonnegative monotone submodular. Then there is a randomized rounding procedure that outputs an integral feasible solution $\bar{z}_1 + \bar{z}_2 + \cdots + \bar{z}_k$ such that $\sum_{i \in [k]} f_i^L(\bar{z}_i) \leq O(\log n) \sum_{i \in [k]} f_i^L(z_i)$ on expectation. That is, we get a partition $S_1, S_2, \ldots, S_k$ of V such that $\sum_{i \in [k]} f_i(S_i) \leq O(\log n) \sum_{i \in [k]} f_i^L(z_i)$ on expectation.*

2.2 A multi-agent gap of $O(\min\{k, \log^2(n)\})$

In this section we present the proof of Theorem 6. The high level idea is that we start with an optimal solution $z^* = z_1^* + z_2^* + \cdots + z_k^*$ to the multi-agent relaxation (MA-LE) 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 interpret the 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 $\min g(S) : S \in \mathcal{F}$ where

$$g(S) = \sum_{i \in [k]} f_i(S \cap V_i). \tag{6}$$

It is clear that g is nonnegative monotone submodular since the f_i are as well. Moreover, for any solution $S \in \mathcal{F}$ for this single-agent problem we obtain a MA solution of the same cost by setting $S_i = S \cap V_i$, since we then have $g(S) = \sum_{i \in [k]} f_i(S \cap V_i) = \sum_{i \in [k]} f_i(S_i)$.

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. We then have by definition of g that $g^L(z) = \sum_{i \in [k]} f_i^L(z|_{V_i})$. Moreover, if the V_i are pairwise disjoint, then we also have $\sum_{i \in [k]} f_i^L(z|_{V_i}) = \sum_{i \in [k]} f_i^L(z_i)$. We summarize this in the following result.

▶ **Proposition 13.** *Let $z = z_1 + z_2 + \cdots + z_k$ be a feasible solution to (MA-LE) where the vectors z_i have pairwise disjoint supports V_i. Then $g^L(z) = \sum_{i \in [k]} f_i^L(z|_{V_i}) = \sum_{i \in [k]} f_i^L(z_i)$.*

The next two results show how one can get a feasible solution $\hat{z} = \hat{z}_1 + \hat{z}_2 + \cdots + \hat{z}_k$ where the $\hat{z}_i$ have pairwise disjoint supports, by losing a factor of $O(\log^2(n))$ and k respectively. We remark that these two results combined prove Theorem 6.

▶ **Theorem 14.** *Suppose there is a (polytime) $\alpha(n)$-approximation for monotone SO($\mathcal{F}$) minimization based on rounding (SA-LE). Then there is a (polytime) $O(\alpha(n) \log^2(n))$-approximation for monotone MASO($\mathcal{F}$) minimization.*

Proof. Let $z^* = z_1^* + z_2^* + \cdots + z_k^*$ denote an optimal solution to (MA-LE) with value OPT_{frac}. In order to apply a black box single-agent rounding algorithm we must create a different multi-agent solution. This is done in several steps, the first few of which are standard. The key steps are *fracture, expand and return*, which arise later in the process.

Call an element v *small* if $z^*(v) \leq \frac{1}{2n}$. Note that $\sum_{v \ small} z^*(v) \leq \frac{1}{2}$, and so for any blocking set $B \in \mathcal{B}(\mathcal{F})$, we have that at most $\frac{1}{2}$ of $z^*(B)$ is contributed by small elements. We obtain a new feasible solution $z' = z'_1 + z'_2 + \cdots + z'_k$ by removing all small elements from the support of the z_i^* and then doubling the resulting vectors. By the monotonicity and homogeneity of the f_i^L, this at most doubles the cost of OPT_{frac}.

We now prune the solution $z' = z'_1 + z'_2 + \cdots + z'_k$ a bit more. Let Z_j be the elements v such that $z'(v) \in (2^{-(j+1)}, 2^{-j}]$ for $j = 0, 1, 2, \ldots, L$. Since $z'(v) > \frac{1}{2n}$ for any element in the support, we have that $L \leq \log(n)$. We call Z_j *bin* j and define $r_j = 2^j$. We round up each $v \in Z_j$ so that $z'(v) = 2^{-j}$ by augmenting the z'_i values by at most a factor of 2. We may do this simultaneously for all v by possibly "truncating" the values associated to some of the elements. As before, this is fine since the f_i^L are monotone. In the end, we call this a *uniform solution* $z'' = z''_1 + z''_2 + \cdots + z''_k$ in the sense that each $z''(v)$ is some power of 2. Note that its cost is at most $4 \cdot OPT_{frac}$.

Fracture. We now *fracture* the vectors z''_i by defining vectors $z''_{i,j} = z''_i|_{Z_j}$ for each $i \in [k]$ and each $j \in \{0, 1, \ldots, L\}$, where recall that the notation $z|_S$ denotes the truncation of z to elements of S. Notice that $z''_i = \sum_{j=0}^{L} z''_{i,j}$.

Expand. Now for each $j \in \{0, 1, \ldots, L\}$ we blow up the vectors $z''_{i,j}$ by a factor r_j. Since $z''(v) = \frac{1}{r_j}$ for each $v \in Z_j$, the resulting values yield a (probably fractional) cover of Z_j. We can then use the rounding procedure discussed in Theorem 12 (with ground set Z_j) to get an integral solution $z'''_{i,j}$ such that $\sum_i f_i^L(z'''_{i,j}) \leq O(\log n) \sum_i f_i^L(r_j \cdot z''_{i,j})$ on expectation.

Return. Now we go back to get a new MA-LE solution $\hat{z} = \hat{z}_1 + \hat{z}_2 + \cdots + \hat{z}_k$ by setting $\hat{z}_i = \sum_{j=0}^{L} \frac{1}{r_j} z'''_{i,j}$. Note that $\hat{z} = z''$ and so this is indeed feasible (and again uniform). Moreover, we have that the cost of this new solution satisfies

$$\begin{aligned}\sum_{i=1}^{k} f_i^L(\hat{z}_i) &\leq \sum_{i=1}^{k}\sum_{j=0}^{L} \frac{1}{r_j} f_i^L(z'''_{i,j}) = \sum_{j=0}^{L} \frac{1}{r_j} \sum_{i=1}^{k} f_i^L(z'''_{i,j}) \leq O(\log n) \sum_{i=1}^{k}\sum_{j=0}^{L} f_i^L(z''_{i,j}) \\ &\leq O(\log n)(L+1) \sum_{i=1}^{k} f_i^L(z''_i) \\ &\leq O(\log^2(n)) \sum_{i=1}^{k} f_i^L(z^*_i) \leq O(\log^2(n)) \cdot OPT_{MA},\end{aligned}$$

where in the first inequality we use the convexity and homogeneity of the f_i^L, in the second inequality we use again the homogeneity together with the upper bound for $\sum_i f_i^L(z'''_{i,j})$, and in the third inequality we use monotonicity and the fact that $z''_{i,j} \leq z''_i$ for all j.

Single-Agent Rounding. In the last step we use the function g defined in (6), with sets V_i corresponding to the support of the $\hat{z}_i$. 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 MA solution satisfying

$$\sum_{i=1}^{k} f_i(\hat{S}_i) = g(\hat{S}) \leq \alpha g^L(\hat{z}) = \alpha \sum_{i=1}^{k} f_i^L(\hat{z}_i) \leq \alpha \cdot O(\log^2(n)) \cdot OPT_{MA},$$

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

While in this paper we define our (SA-LE) and (MA-LE) formulations in terms of the blocking formulation $P^*(\mathcal{F})$, in the full version [35] we discuss how our results naturally extend to more general upwards closed relaxations $\{z \geq 0 : Az \geq r\}$ of the integral polyhedron $conv(\{\chi^S : S \in \mathcal{F}\})$. We also show how a more careful analysis of the above proof leads to a slightly improved MA gap of $O(\log(n)\log(\frac{n}{\log n}))$.

We now give an approximation in terms of the number of agents, which becomes preferable in settings where $k < \log^2(n)$.

▶ **Lemma 15.** *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 MASO($\mathcal{F}$) minimization.*

Proof. Let $z^* = z^*_1 + z^*_2 + \cdots + z^*_k$ denote an optimal solution to (MA-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 $\hat{z}_{i'}(v) = k z^*_i(v)$ 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 $\hat{z}_i \leq k z^*_i$ for each $i \in [k]$. Hence, given the monotonicity and homogeneity of the f_i^L we have

$$\sum_{i \in [k]} f_i^L(\hat{z}_i) \leq \sum_{i \in [k]} f_i^L(k z^*_i) = k \sum_{i \in [k]} f_i^L(z^*_i) = k \cdot OPT_{frac} \leq k \cdot OPT_{MA}.$$

Since the $\hat{z}_i$ have disjoint supports V_i, we can now use the function g defined in (6) and do a single-rounding argument as in Theorem 14. This completes the proof. ◀

2.3 A tight multi-agent gap of $O(\log n)$ for bounded blocker families

While we established an $O(\log^2(n))$ MA gap for general families based on the blocking convex formulations, the work of [12] shows an improved MA gap of $O(\log n)$ for vertex covers. In this section we generalize their result by describing a larger family class with such MA gap.

Their algorithm relies on the fact that the set family has the following *bounded blocker property*. We call a clutter (family of non-comparable sets) $\mathcal{F}$ *β-bounded* if $|F| \leq \beta$ for all $F \in \mathcal{F}$. Recall that the *blocker* of a clutter $\mathcal{F}$, denoted by $\mathcal{B}(\mathcal{F})$, is the set of all minimal B such that $B \cap F \neq \emptyset$ for all $F \in \mathcal{F}$. We then say that $\mathcal{F}$ has a β-bounded blocker if $|B| \leq \beta$ for each $B \in \mathcal{B}(\mathcal{F})$. The main SA minimization result for such families is the following.

▶ **Theorem 16** ([21, 23]). *Let $\mathcal{F}$ be a family with a β-bounded blocker. Then there is a β-approximation algorithm for monotone $SO(\mathcal{F})$ minimization. If $P^*(\mathcal{F})$ has a polytime separation oracle, then this is a polytime algorithm.*

Our next result establishes an $O(\log n)$ MA gap for families with a bounded blocker.

▶ **Theorem 17.** *Let $\mathcal{F}$ be an upwards closed family with a β-bounded blocker. Then there is a randomized $O(\beta \log n)$-approximation algorithm for the monotone $MASO(\mathcal{F})$ minimization problem. If $P^*(\mathcal{F})$ has a polytime separation oracle, then this is a polytime algorithm.*

Proof. Let $z^* = \sum_{i \in [k]} z_i^*$ be an optimal solution to (MA-LE) with value OPT_{frac}. Consider the new feasible solution given by $\beta z^* = \sum_{i \in [k]} \beta z_i^*$ and let $U = \{v \in V : \beta z^*(v) \geq 1\}$. Since $\mathcal{F}$ has a β-bounded blocker it follows that $U \in \mathcal{F}$. We now have that $\sum_{i \in [k]} \beta z_i^*$ is a feasible solution such that $\sum_{i \in [k]} \beta z_i^* \geq \chi^U$. Thus, we can use Theorem 12 to get an integral feasible solution $\sum_{i \in [k]} \bar{z}_i$ such that $\sum_{i \in [k]} \bar{z}_i \geq \chi^U$ and $\sum_{i \in [k]} f_i^L(\bar{z}_i) \leq O(\log |U|) \sum_{i \in [k]} f_i^L(\beta z_i^*) \leq \beta \cdot O(\log n) \cdot OPT_{frac}$ on expectation. ◀

While our work focuses on monotone objectives, in the full version [35] we show that upwards closed families with a bounded blocker remain tractable under some special types of nonmonotone objectives. These were introduced in [5] and [7], where they consider objectives of the form $f_i = g_i + h$ where the g_i are monotone submodular and h is symmetric submodular (in [5]) or just submodular (in [7]). Note that by taking $h \equiv 0$ (which is symmetric submodular) we recover the monotone case.

2.4 A tight multi-agent gap of $O(\log n)$ for ring and crossing families

It is well known ([36]) that submodular minimization can be solved exactly in polynomial time over a ring family. In this section we observe that the MA problem over this type of constraint admits a tight $\ln(n)$-approximation. More generally, we consider *crossing families*. A family $\mathcal{F}$ of subsets of V forms a ring family (aka lattice family) if for each $A, B \in \mathcal{F}$ we have $A \cap B, A \cup B \in \mathcal{F}$. A crossing family is one where we only require it for sets where $A \setminus B, B \setminus A, A \cap B, V - (A \cup B)$ are all non-empty. Thus any ring family is a crossing family.

For any crossing family $\mathcal{F}$ and any $u, v \in V$, let $\mathcal{F}_{uv} = \{A \in \mathcal{F} : u \in A, v \notin A\}$. It is easy to see that $\mathcal{F}_{uv}$ is a ring family. We may then solve the original MA problem by solving the associated MA problem for each non-empty $\mathcal{F}_{uv}$ and then selecting the best output solution.

So we assume now that we are given a ring family in such a way that we may compute its minimal set M (which is unique). This is a standard assumption when working with ring families (cf. submodular minimization algorithm described in [36]). Then, due to monotonicity and the fact that $\mathcal{F}$ is closed under intersections, it is not hard to see that the original problem reduces to the submodular facility location problem

$$\min \sum_{i=1}^{k} f_i(S_i) : S_1 \uplus \cdots \uplus S_k = M \ ,$$

which admits a tight $(\ln |M|)$-approximation ([39]). In particular, for the special case where we have the trivial ring family $\mathcal{F} = \{V\}$ we get a tight $\ln(n)$-approximation. The next result summarizes these observations.

▶ **Theorem 18.** *There is a tight* $\ln(n)$*-approximation for monotone* $MASO(\mathcal{F})$ *minimization over crossing families* $\mathcal{F}$*.*

3 Multi-agent submodular maximization

In this section we describe two different reductions. The first one reduces the capacitated multi-agent problem (3) to a single-agent problem. We show that several properties of the objective and family of feasible sets stay *invariant* (i.e. preserved) under the reduction. We use this to establish an (optimal) MA gap of 1 for several families. Examples of such families include spanning trees, matroids, and p-systems.

Our second reduction uses the multilinear extension of a set function. We establish that if the SA (monotone or nonmonotone) primitive admits approximation via its multilinear relaxation, then we may extend this to its MA version with a constant factor loss. Moreover, for the monotone case our MA gap is tight.

3.1 The lifting reduction

In this section we describe a generic reduction of CMASO (i.e. (3)) to a single-agent problem: $\max / \min f(S) : S \in \mathcal{L}$. 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 [10], and has since then been widely used [29, 40, 4, 37]. 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 multi-agent objective (such as the one in (3)) to a univariate objective $f : 2^E \to \mathbb{R}$ over the lifted space. Namely, for each set $S \subseteq E$ we define $f(S) = \sum_{i \in [k]} f_i(S_i)$. The function f is well-defined because each subset $S \subseteq E$ can be uniquely written as $S = \uplus_{i \in [k]}(\{i\} \times S_i)$.

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:

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

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

Clearly, this reduction is interesting if our new function f and family of sets $\mathcal{L}$ have properties which allows us to handle them computationally. This will depend on the original structure of the functions f_i, and the set families $\mathcal{F}$ and $\mathcal{F}_i$. In terms of the objective, it is

Table 1 Invariant properties under the lifting reduction.

	Multi-agent problem	**Single-agent (i.e. reduced) problem**	Result
1	$(V, \mathcal{F})$ a p-system	$(E, \mathcal{F}')$ a p-system	Appendix C
2	$\mathcal{F}$ = bases of a p-system	$\mathcal{F}'$ = bases of a p-system	Appendix C
3	$(V, \mathcal{F})$ a matroid	$(E, \mathcal{F}')$ a matroid	Appendix C
4	$\mathcal{F}$ = bases of a matroid	$\mathcal{F}'$ = bases of a matroid	Appendix C
5	$(V, \mathcal{F})$ a p-matroid intersection	$(E, \mathcal{F}')$ a p-matroid intersection	Full version [35]
6	$\mathcal{F}$ = forests (resp. spanning trees)	$\mathcal{F}'$ = forests (resp. spanning trees)	Full version [35]
7	$\mathcal{F}$ = matchings (resp. perfect matchings)	$\mathcal{F}'$ = matchings (resp. perfect matchings)	Full version [35]
8	$\mathcal{F}$ = st-paths	$\mathcal{F}'$ = st-paths	Full version [35]
9	$(V, \mathcal{F}_i)$ a matroid for all $i \in [k]$	$(E, \mathcal{H})$ a matroid	Full version [35]
10	$\mathcal{F}_i$ a ring family for all $i \in [k]$	$\mathcal{H}$ a ring family	Full version [35]

straightforward to check (as previously pointed out in [29]) that if the f_i are (nonnegative, respectively monotone) submodular functions, then f as defined above is also (nonnegative, respectively monotone) submodular. In the full version [35] 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. We show, for instance, that if the family $\mathcal{F}$ induces a matroid (or more generally a p-system) over the original ground set V, then so does the family $\mathcal{F}'$ over the lifted space E. We summarize some of these results in Table 1, and we remark that these now prove Theorem 10 (see [35] for full details).

3.2 The single-agent and multi-agent formulations

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 [3]) 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. Consider a vector $z \in [0,1]^V$ and 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.

This gives rise to natural single-agent and multi-agent relaxations for constrained submodular maximization. The *single-agent multilinear extension relaxation* is:

$$\text{(SA-ME)} \qquad \max f^M(z) : z \in P(\mathcal{F}), \tag{7}$$

and the *multi-agent multilinear extension relaxation* is:

$$\text{(MA-ME)} \qquad \max \sum_{i \in [k]} f_i^M(z_i) : z_1 + z_2 + \cdots + z_k \in P(\mathcal{F}), \tag{8}$$

where $P(\mathcal{F})$ denotes some fractional relaxation of the integral polytope $conv(\{\chi^S : S \in \mathcal{F}\})$. While the relaxation (SA-ME) has been used extensively [4, 27, 9, 6, 1] in the submodular maximization literature, we are not aware of any previous work using the multi-agent relaxation (MA-ME). We next discuss the solvability of (SA-ME).

▶ **Theorem 19** ([1, 40]). *Let $f : 2^V \to \mathbb{R}_+$ be nonnegative submodular and f^M its multilinear extension. Let $P \subseteq [0,1]^V$ be any downwards closed polytope that admits a polytime separation* *oracle, and let* $OPT = \max f^M(z) : z \in P$*. Then there is a polytime algorithm ([1]) 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 ([40]) 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 [41] 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 remark that we can solve (MA-ME) to the same approximation factor as (SA-ME). This follows from the fact that the MA problem has the form $\{\max g(w) : w \in W \subseteq \mathbf{R}^{nk}\}$ where $g(w) = g(z_1, z_2, \ldots, z_k) = \sum_{i\in[k]} f_i^M(z_i)$ and W is the downwards closed polytope $\{w = (z_1, ..., z_k) : \sum_i z_i \in P(\mathcal{F})\}$. Clearly we have a polytime separation oracle for W given that we have one for $P(\mathcal{F})$. Moreover, it is straightforward to check (see Lemma 34 on Appendix C) that $g(w) = f^M(w)$, where f is the function on the lifted space after applying the lifting reduction from Section 3.1. Thus, g is the multilinear extension of a nonnegative submodular function, and we can use Theorem 19.

3.3 A tight multi-agent gap of $1 - 1/e$

In this section we present the proof of Theorem 9. The high-level idea behind our reduction is the same as in the minimization setting (see Section 2.2). That is, we start with an (approximate) optimal solution $z^* = z_1^* + z_2^* + \cdots + z_k^*$ to the multi-agent (MA-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 then use for the SA rounding step the single-agent problem (as previously defined in (6) for the minimization setting) $\max g(S) : S \in \mathcal{F}$ where $g(S) = \sum_{i\in[k]} f_i(S \cap V_i)$.

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

▶ **Proposition 20.** *Let* $z = \sum_{i\in[k]} z_i$ *be a feasible solution to (MA-ME) where the vectors* z_i *have pairwise disjoint supports* V_i*. Then* $g^M(z) = \sum_{i\in[k]} f_i^M(z|_{V_i}) = \sum_{i\in[k]} f_i^M(z_i)$*.*

We now have all the ingredients to prove our main result in the maximization setting.

▶ **Theorem 21.** *If there is a (polytime)* $\alpha(n)$*-approximation for monotone SO(*$\mathcal{F}$*) maximization via rounding (SA-ME), there is a (polytime)* $(1-1/e)\cdot\alpha(n)$*-approximation for monotone MASO(*$\mathcal{F}$*) maximization. Furthermore, if* $\mathcal{F}$ *is downwards closed and there is a (polytime)* $\alpha(n)$*-approximation for nonmonotone SO(*$\mathcal{F}$*) maximization via rounding (SA-ME), there is a (polytime)* $0.385\cdot\alpha(n)$*-approximation for nonmonotone MASO(*$\mathcal{F}$*) 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 (MA-ME) obtained via Theorem 19, and let OPT_{frac} be the value of an optimal solution. We then have that $\sum_{i\in[k]} f_i^M(z_i^*) \geq (1-1/e)OPT_{frac} \geq (1-1/e)OPT_{MA}$.

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. Notice that by definition of the multilinear extension we have that the functions f_i^M are linear along directions $\mathbf{e_v}$ for any $v \in V$. It then follows that the function

$$h(t) = f_i^M(z_i^* + t\mathbf{e_v}) + f_{i'}^M(z_{i'}^* - t\mathbf{e_v}) + \sum_{j\in[k], j\neq i,i'} f_j^M(z_j^*)$$

is also linear for any $v \in V$ and $i \neq i' \in [k]$, since it is the sum of linear functions (on t). 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. We use this as follows.

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 feasible.

STEP 3. In the last step we use the function g defined in (6), with sets V_i corresponding to the support 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 MA solution satisfying

$$\sum_{i=1}^{k} f_i(\hat{S}_i) = g(\hat{S}) \geq \alpha g^M(\hat{z}) = \alpha \sum_{i=1}^{k} f_i^M(\hat{z}_i) \geq \alpha \sum_{i=1}^{k} f_i^M(z_i^*) \geq \alpha(1 - 1/e) OPT_{MA},$$

where the second equality uses Proposition 20. This completes the monotone case.

For the nonmonotone setting the proof 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 19. 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 multi-agent problem. ◀

4 Conclusion

A number of interesting questions remain. Perhaps the main one being whether the $O(\log^2(n))$ MA gap for minimization can be improved to $O(\log n)$? We have shown this is the case for bounded blocker and crossing families. Another question is whether the $\alpha \log^2(n)$ and αk approximations can be made truly black box? I.e., do not depend on the convex formulation.

On separate work ([34]) we discuss multivariate submodular objectives. We show that our reductions for maximization remain well-behaved algorithmically and this opens up more tractable models. This is the topic of planned future work.

References

1. Niv Buchbinder and Moran Feldman. Constrained submodular maximization via a non-symmetric technique. *arXiv preprint arXiv:1611.03253*, 2016.
2. 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.
3. 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.
4. 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.
5. 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.

6 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.

7 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.

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

9 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.

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

11 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.

12 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.

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

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

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

16 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.

17 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.

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

19 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.

20 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.

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

22 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.

23 Christos Koufogiannakis and Neal E Young. Greedy δ-approximation algorithm for covering with arbitrary constraints and submodular cost. *Algorithmica*, 66(1):113–152, 2013.

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

25 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.

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

27 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.

28 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.

29 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`.

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

31 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.

32 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.

33 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.

34 Richard Santiago and F Bruce Shepherd. Multivariate submodular optimization. *arXiv preprint arXiv:1612.05222*, 2016.

35 Richard Santiago and F Bruce Shepherd. Multi-agent submodular optimization. *arXiv preprint arXiv:1803.03767*, 2018.

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

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

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

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

40 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.

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

42 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.

A 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. Several of these results have already appeared in [21].

A.1 Blocking families and a natural relaxation for $P(\mathcal{F})$

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.

For a blocking family $\mathcal{F}$ one often works with the induced sub-family $\mathcal{F}^{min}$ of minimal sets. Then $\mathcal{F}^{min}$ has the property that it is a *clutter*, that is, $\mathcal{F}^{min}$ does not contain a pair of comparable sets, i.e., sets $F \subset F'$. If $\mathcal{F}$ is a clutter, then $\mathcal{F} = \mathcal{F}^{min}$ and there is an associated *blocking* clutter $\mathcal{B}(\mathcal{F})$, which consists of the minimal sets B such that $B \cap F \neq \emptyset$ for each $F \in \mathcal{F}$. We refer to $\mathcal{B}(\mathcal{F})$ as the *blocker* of $\mathcal{F}$.

One also checks that for an arbitrary upwards-closed family $\mathcal{F}$, we have the following.

▶ **Claim 22** (Lehman).
1. *$F \in \mathcal{F}$ if and only if $F \cap B \neq \emptyset$ for all $B \in \mathcal{B}(\mathcal{F}^{min})$.*
2. *$\mathcal{B}(\mathcal{B}(\mathcal{F}^{min})) = \mathcal{F}^{min}$.*

Thus the significance of blockers is that one may assert membership in an upwards-closed family $\mathcal{F}$ by checking intersections on sets from the blocker $\mathcal{B}(\mathcal{F}^{min})$. If we define $\mathcal{B}(\mathcal{F})$ to be the minimal sets which intersect every element of $\mathcal{F}$, then one checks that $\mathcal{B}(\mathcal{F}) = \mathcal{B}(\mathcal{F}^{min})$. These observations lead to a natural relaxation for minimization problems over the integral polyhedron $P(\mathcal{F}) = conv(\{\chi^F : F \in \mathcal{F}\})$. The *blocking formulation* for $\mathcal{F}$ is:

$$P^*(\mathcal{F}) = \{z \in \mathbb{R}^V_{\geq 0} : z(B) \geq 1 \quad \forall B \in \mathcal{B}(\mathcal{F}^{min}) = \mathcal{B}(\mathcal{F})\}. \tag{9}$$

Clearly we have $P(\mathcal{F}) \subseteq P^*(\mathcal{F})$.

A.2 Reducing to blocking families

Consider an arbitrary set family $\mathcal{F}$ over V. We may define its *upwards closure* by $\mathcal{F}^{\uparrow} = \{F' : F \subseteq F' \text{ for some } F \in \mathcal{F}\}$. 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.

As already noted $\mathcal{B}(\mathcal{F}) = \mathcal{B}(\mathcal{F}^{\uparrow}) = \mathcal{B}(\mathcal{F}^{min})$ and hence one approach is via the blocking formulation $P^*(\mathcal{F}) = P^*(\mathcal{F}^{\uparrow})$. This requires two ingredients. First, we need a separation algorithm for the blocking relaxation, 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 the blocking relaxation $P^*(\mathcal{F}^{\uparrow})$ or for the polytope $P(\mathcal{F})$.

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

▶ **Claim 23.** *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 use this observation to prove the following.

▶ **Lemma 24.** *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 23 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 24 to construct in polytime $F \in \mathcal{F}$ which obeys the same guarantee.

Moreover, notice that for Lemma 24 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 23), a separation oracle for either $P(\mathcal{F}^\uparrow)$ or $P(\mathcal{F})^\uparrow$ would be enough for the routine of Lemma 24 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 25.** *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 24 to turn feasible sets from $\mathcal{F}^\uparrow$ into feasible sets in $\mathcal{F}$.

▶ **Corollary 26.** *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 26 to construct $F \in \mathcal{F}$ which obeys the same guarantee.

B Convex relaxations for constrained submodular minimization

We will be working with upwards-closed set families $\mathcal{F}$, and their blocking relaxations $P^*(\mathcal{F})$. As we now work with arbitrary vectors $z \in [0,1]^n$, we must specify how our objective function $f(S)$ behaves on all points $z \in P^*(\mathcal{F})$. Formally, we call $g : [0,1]^V \to \mathbb{R}$ an *extension* of f if $g(\chi^S) = f(S)$ for each $S \subseteq V$. For a submodular objective function $f(S)$ there can be many extensions of f to $[0,1]^V$ (or to $\mathbb{R}^V$). The most popular one has been the so-called *Lovász Extension* (introduced in [30]) due to several of its desirable properties.

We present one of several equivalent definitions for the Lovász Extension. Let $0 < v_1 < v_2 < ... < v_m \leq 1$ be the distinct positive values taken in some vector $z \in [0,1]^V$. We also define $v_0 = 0$ and $v_{m+1} = 1$ (which may be equal to v_m). Define for each $i \in \{0, 1, ..., m\}$ 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

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

▶ **Lemma 27** (Lovász [30]). *The function f^L is convex if and only if f is submodular.*

One could now attack constrained submodular minimization by solving the problem

$$\text{(SA-LE)} \qquad \min f^L(z) : z \in P^*(\mathcal{F}), \tag{10}$$

and then seek rounding methods for the resulting solution. This is the approach used in [5, 20, 21]. We refer to the above as the *single-agent Lovász extension formulation*, abbreviated as (SA-LE).

B.1 Tractability of the single-agent formulation (SA-LE)

In this section we show that one may solve (SA-LE) approximately as long as a polytime separation algorithm for $P^*(\mathcal{F})$ is available. This is useful in several settings and in particular for our methods which rely on the multi-agent Lovász extension formulation (discussed in Section B.2).

Polytime Algorithms. One may apply the Ellipsoid Method to obtain a polytime algorithm which approximately minimizes a convex function over a polyhedron K as long as various technical conditions hold. For instance, one could require that there are two ellipsoids $E(a, A) \subseteq K \subseteq E(a, B)$ whose encoding descriptions are polynomially bounded in the input size for K. We should also have polytime (or oracle) access to the convex objective function defined over $\mathbf{R}^n$. In addition, one must be able to polytime solve the subgradient problem for f.[2] One may check that the subgradient problem is efficiently solvable for Lovász extensions of polynomially encodable submodular functions. We call f *polynomially encodable* if the values $f(S)$ have encoding size bounded by a polynomial in n (we always assume this for our functions). If these conditions hold, then methods from [15] imply that for any $\epsilon > 0$ we may find an approximately feasible solution for K which is approximately optimal. By approximate here we mean for instance that the objective value is within ϵ of the real optimum. This can be done in time polynomially bounded in n (size of input say) and $\log \frac{1}{\epsilon}$. Let us give a few details for our application.

[2] For a given y, find a subgradient of f at y.

Our convex problem's feasible space is $P^*(\mathcal{F})$ and it is easy to verify that our optimal solutions will lie in the $0-1$ hypercube H. So we may define the feasible space to be H and the objective function to be $g(z) = f^L(z)$ if $z \in H \cap P^*(\mathcal{F})$ and $= \infty$ otherwise. (Clearly g is convex in $\mathbf{R}^n$ since it is a pointwise maximum of two convex functions; alternatively, one may define the Lovász Extension on $\mathbf{R}^n$ which is also fine.) Note that g can be evaluated in polytime by the definition of f^L as long as f is polynomially encodable. We can now easily find an ellipsoid inside H and one containing H each of which has poly encoding size. We may thus solve the convex problem to within $\pm\epsilon$-optimality in time bounded by a polynomial in n and $\log \frac{1}{\epsilon}$.

▶ **Corollary 28.** *Consider a class of problems $\mathcal{F}, f$ for which f's are submodular and polynomially-encodable in $n = |V|$. If there is a polytime separation algorithm for the family of polyhedra $P^*(\mathcal{F})$, then the convex program (SA-LE) can be solved to accuracy of $\pm\epsilon$ in time bounded by a polynomial in n and $\log \frac{1}{\epsilon}$.*

B.2 The multi-agent formulation

The single-agent formulation (SA-LE) discussed above has a natural extension to the multi-agent setting. This was already introduced in [5] for the case $\mathcal{F} = \{V\}$.

$$\text{(MA-LE)} \qquad \min \sum_{i \in [k]} f_i^L(z_i) : z_1 + z_2 + \cdots + z_k \in P^*(\mathcal{F}). \tag{11}$$

We refer to the above as the *multi-agent Lovász extension formulation*, abbreviated as (MA-LE). We can solve (MA-LE) as long as we have polytime separation of $P^*(\mathcal{F})$. This follows the approach from the previous section (see Corollary 28) except our convex program now has k vectors of variables $z_1, z_2, \ldots, z_k$ (one for each agent) such that $z = \sum_i z_i$. This 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 $g(w) = g(z_1, z_2, \ldots, z_k) = \sum_{i \in [k]} f_i^L(z_i)$ is convex. Clearly we have a polytime separation routine for W, and hence we may apply Ellipsoid as in the single-agent case.

C Invariance under the lifting reduction

We prove some of the results from Table 1 in Section 3.1. For a subset of edges $S \subseteq E$ we define its *coverage* $cov(S)$ as the set of nodes $v \in V$ saturated by S. That is, $v \in cov(S)$ if there exists $i \in [k]$ such that $(i, v) \in S$. By definition of $\mathcal{F}'$ (see Section 3.1) it is straightforward that for each $S \subseteq E$ we have that

$$S \in \mathcal{F}' \iff cov(S) \in \mathcal{F} \text{ and } |S| = |cov(S)|. \tag{12}$$

For a set $S \subseteq E$, a set $B \subseteq S$ is called a *basis* of S if B is an inclusion-wise maximal independent subset of S. Our next result describes how bases and their cardinalities behave under the lifting reduction.

▶ **Lemma 29.** *Let S be an arbitrary subset of E. Then for any basis B (over $\mathcal{F}'$) of S there exists a basis B' (over $\mathcal{F}$) of $cov(S)$ such that $|B'| = |B|$. Moreover, for any basis B' of $cov(S)$ there exists a basis B of S such that $|B| = |B'|$.*

Proof. For the first part, let B be a basis of S and take $B' := cov(B)$. Since $B \in \mathcal{F}'$ we have by (12) that $B' \in \mathcal{F}$ and $|B'| = |B|$. Now, if B' is not a basis of $cov(S)$ then we can find an element $v \in cov(S) - B'$ such that $B' + v \in \mathcal{F}$. Moreover, since $v \in cov(S)$ there exists $i \in [k]$ such that $(i, v) \in S$. But then we have that $B + (i, v) \subseteq S$ and $B + (i, v) \in \mathcal{F}'$, a contradiction with the fact that B was a basis of S.

For the second part, let B' be a basis of $cov(S)$. For each $v \in B'$ let i_v be such that $(i_v, v) \in S$, and take $B := \uplus_{v \in B'} (i_v, v)$. It is clear by definition of B that $cov(B) = B'$ and $|B| = |B'|$. Hence $B \in \mathcal{F}'$ by (12). If B is not a basis of S there exists an edge $(i, v) \in S - B$ such that $B + (i, v) \in \mathcal{F}'$. But then by (12) we have that $cov(B + (i, v)) \in \mathcal{F}$ and $B' \subsetneq cov(B + (i, v)) \subseteq cov(S)$, a contradiction since B' was a basis of $cov(S)$. ◀

We say that $(V, \mathcal{F})$ is a *p-system* if for each $U \subseteq V$, the cardinality of the largest basis of U is at most p times the cardinality of the smallest basis of U. The following result is a direct consequence of Lemma 29.

▶ **Proposition 30.** *If $(V, \mathcal{F})$ is a p-system, then $(E, \mathcal{F}')$ is a p-system.*

▶ **Corollary 31.** *If $\mathcal{F}$ corresponds to the set of bases of a p-system $(V, \mathcal{I})$, then $\mathcal{F}'$ also corresponds to the set of bases of some p-system $(E, \mathcal{I}')$.*

Proof. Consider $(E, \mathcal{I}')$ where $\mathcal{I}' := \{S \subseteq E : cov(S) \in \mathcal{I} \text{ and } |cov(S)| = |S|\}$. Then by Proposition 30 we have that $(E, \mathcal{I}')$ is a p-system. It is now straightforward to check that $\mathcal{F}'$ corresponds precisely to the set of bases of $(E, \mathcal{I}')$. ◀

The following two results follow from Proposition 30 and Corollary 31 and the fact that matroids are precisely the class of 1-systems.

▶ **Corollary 32.** *If $(V, \mathcal{F})$ is a matroid, then $(E, \mathcal{F}')$ is a matroid.*

▶ **Corollary 33.** *Assume $\mathcal{F}$ is the set of bases of some matroid $\mathcal{M} = (V, \mathcal{I})$, then $\mathcal{F}'$ is the set of bases of some matroid $\mathcal{M}' = (E, \mathcal{I}')$.*

Let the functions f_i and f be as described in the lifting reduction in Section 3.1. The following result establishes the relationship between $f^M(z_1, \ldots, z_k)$ and $\sum_{i \in [k]} f_i^M(z_i)$.

▶ **Lemma 34.** *Let the functions f_i and f be as described in the lifting reduction in Section 3.1. Then for any vector $\bar{z} = (z_1, z_2, \ldots, z_k) \in [0,1]^E$, where $z_i \in [0,1]^V$ is the vector associated with agent i, we have that $f^M(\bar{z}) = f^M(z_1, z_2, \ldots, z_k) = \sum_{i \in [k]} f_i^M(z_i)$.*

Proof. We use the definition of the multilinear extension in terms of expectations (see Section 3.2). Recall that for a vector $z \in [0,1]^V$, R^z denotes a random set that contains element v_i independently with probability z_{v_i}. We use $\mathbb{P}_z(S)$ to denote $\mathbb{P}[R^z = S]$. We then have

$$\begin{aligned}
f^M(\bar{z}) &= \mathbb{E}[f(R^{\bar{z}})] = \sum_{S \subseteq E} f(S) \mathbb{P}_{\bar{z}}(S) \\
&= \sum_{S_1 \subseteq V} \sum_{S_2 \subseteq V} \cdots \sum_{S_k \subseteq V} [\sum_{i \in [k]} f_i(S_i)] \cdot \mathbb{P}_{(z_1, z_2, \ldots, z_k)}(S_1, S_2, \ldots, S_k) \\
&= \sum_{i \in [k]} \sum_{S_1 \subseteq V} \sum_{S_2 \subseteq V} \cdots \sum_{S_k \subseteq V} f_i(S_i) \cdot \mathbb{P}_{(z_1, z_2, \ldots, z_k)}(S_1, S_2, \ldots, S_k) \\
&= \sum_{i \in [k]} \sum_{S_i \subseteq V} f_i(S_i) \sum_{S_j \subseteq V, j \neq i} \mathbb{P}_{(z_1, z_2, \ldots, z_k)}(S_1, S_2, \ldots, S_k) \\
&= \sum_{i \in [k]} \sum_{S_i \subseteq V} f_i(S_i) \mathbb{P}_{z_i}(S_i) = \sum_{i \in [k]} \mathbb{E}[f_i(S_i^{z_i})] = \sum_{i \in [k]} f_i^M(z_i).
\end{aligned}$$

◀