Submodular maximization and its generalization through an intersection cut lens

Liding Xu *   Leo Liberti *

March 4, 2023

Abstract

We study a mixed-integer set $\mathcal{S} := \{(x,t) \in \{0,1\}^n \times \mathbb{R} : f(x) \geq t\}$ arising in the submodular maximization problem, where f is a submodular function defined over $\{0,1\}^n$. We use intersection cuts to tighten a polyhedral outer approximation of $\mathcal{S}$. We construct a continuous extension F of f, which is convex and defined over the entire space $\mathbb{R}^n$. We show that the epigraph epi(F) of F is an $\mathcal{S}$-free set, and characterize maximal $\mathcal{S}$-free sets including epi(F). We propose a hybrid discrete Newton algorithm to compute an intersection cut efficiently and exactly. Our results are generalized to the hypograph or the superlevel set of a submodular-supermodular function, which is a model for discrete nonconvexity. A consequence of these results is intersection cuts for Boolean multilinear constraints. We evaluate our techniques on max cut, pseudo Boolean maximization, and Bayesian D-optimal design problems within a MIP solver.

Keywords: MINLP, submodular maximization, submodular-supermodular functions, intersection cuts, Boolean multilinear functions, D-optimal design.

1 Introduction

In this paper, we consider the submodular maximization problem:

$$\max_{t \in \mathbb{R}} t \quad \text{s.t.} \quad f(x) \geq t, \quad x \in \{0,1\}^n \cap \mathcal{X}. \tag{1}$$

where $f : \{0,1\}^n \to \mathbb{R}$ is a submodular function and $\mathcal{X} \subseteq \mathbb{R}^n$ is a set describing additional constraints. We study valid inequalities for the mixed-integer set $\text{hypo}_{\{0,1\}^n}(f) := \{(x,t) \in \{0,1\}^n \times \mathbb{R} : f(x) \geq t\}$, which is the hypograph of f over the Boolean hypercube $\{0,1\}^n$.

The maximization of arbitrary submodular functions (i.e., Eq. (1)) can be reduced to a Mixed-Integer Linear Program (MILP) with exponentially many linear inequalities [60]. No polynomial-time algorithm is yet known to separate these inequalities. The Benders-like exact approach based on a branch-and-cut algorithm proposed in [27] provides global dual bounds for primal solutions, and achieves a finite convergence rate.

Many submodular maximization problems (e.g., max cut with positive edge weights [66], D-optimal design [64], and utility maximization [2]) have natural MILP or mixed-integer nonlinear programming (MINLP) formulations, which can be solved using general-purpose global optimization solvers. The algorithm underlying these solvers is typically a branch-and-cut algorithm, which

---

*LIX CNRS, École Polytechnique, Institut Polytechnique de Paris, Palaiseau, 91128, France. E-mail: liding.xu@polytechnique.edu, liberti@lix.polytechnique.fr

uses polyhedral outer approximations to construct LP relaxations [14, 15, 70]. For submodular maximization problems with convex MINLP formulations, a state-of-art algorithm also uses polyhedral outer approximations [23].

Intersection cuts can be used to strengthen polyhedral outer approximations of a nonconvex set $\mathcal{S}$ that is considered hard to optimize over. The construction of intersection cuts [24] requires two key ingredients: a corner polyhedron relaxation of $\mathcal{S}$, and an $\mathcal{S}$-free set, which is a convex set that does not contain any interior point of $\mathcal{S}$. (Inclusion-wise) maximal $\mathcal{S}$-free sets generate strong intersection cuts not dominated by other intersection cuts.

Intersection cuts were initially devised in the continuous setting (the papers [73, 74], cited in [46, Ch. III], appeared before the classic paper [9]), where they could approximate the hypograph $\mathcal{S}$ of a convex function over a polytope. There is a unique maximal $\mathcal{S}$-free set: the epigraph of that convex function. Later, intersection cuts were used in the discrete setting [9], where $\mathcal{S}$ is a lattice. Several more families of lattice-free sets (e.g., splits, triangles, and spheres [24, 51]) were described later.

The submodular maximization problem plays an intermediate role between these settings. On the one hand, the submodular function f is defined over the Boolean hypercube $\{0,1\}^n$. Therefore, the graph of f projected on $\mathbb{R}^n$ is a subset of a lattice. On the other hand, as a discrete analogue to convex functions, f has a convex (thus continuous) extension over the hypercube $[0,1]^n$, namely the Lovász extension [52]. We can extend the Lovász extension to a convex function, which we call F, over the entire n-dimensional Euclidean space $\mathbb{R}^n$. This (continuous) function F inherits a rich combinatorial structure from f.

The difference of two submodular functions (call them f_1, f_2) is a *submodular-supermodular* (SS) function. SS functions generalize submodular functions, which are also discrete analogues of difference-of-convex (DC) functions. The epigraphs/superlevel sets of DC functions can represent various nonconvex sets, e.g., quadratic sets [57] and signomial-term sets [77]. This representation facilitates the derivation of intersection cuts [67]. In fact, SS functions may represent some discrete nonconvex functions arising in combinatorial optimization. For example, we will show that any Boolean multilinear function is an SS function.

Let $\mathcal{S} = \mathrm{hypo}_{\{0,1\}^n}(f)$. In this paper, we use convex extensions in order to construct some $\mathcal{S}$-free (namely, hypograph-free) sets with Boolean structure. The hypograph set $\mathrm{hypo}_{\{0,1\}^n}(f)$ is a special case of the constraint set $\mathcal{S} := \{(x,t) \in \{0,1\}^n \times \mathbb{R} : f_1(x) - f_2(x) \geq \ell t\}$ with $\ell \in \{0,1\}$. We also aim at extending our results to this more general set $\mathcal{S}$. Finally, we propose an efficient algorithm to compute intersection cuts derived from $\mathcal{S}$-free sets. To the best of our knowledge, intersection cuts have not been applied directly to approximate problems with submodular and/or supermodular structures.

We implement intersection cuts within the SCIP solver [14] and test them on MAX CUT, PSEUDO BOOLEAN MAXIMIZATION, and BAYESIAN D-OPTIMAL DESIGN problems. We show the strengths and weaknesses of intersection cuts under these different settings.

1.1 Contributions

Our primary contribution is the construction of hypograph-free sets. We show that a maximally hypograph-free set $\mathcal{C} \times \mathbb{R}$ can be lifted from a maximal $\{0,1\}^n$-free set. We also give an alternative construction of hypograph-free sets by exploiting the submodularity. We relate the analytical properties of F to its combinatorial properties, which inherit those of the Lovász extension. We show that the epigraph $\mathrm{epi}(\mathsf{F})$ of F is a hypograph-free set, larger than the epigraph of the Lovász extension. However, unlike in the continuous setting, $\mathrm{epi}(\mathsf{F})$ is not maximally hypograph-free. We give necessary and sufficient conditions on maximal hypograph-free sets that include $\mathrm{epi}(\mathsf{F})$.

The second contribution is the computation of intersection cuts. We reduce the intersection cut separation problem to solving univariate nonlinear equations, which we achieve by a hybrid discrete Newton algorithm like [41]. We show that facets of epi(F) can be separated in strongly polynomial time. This implies that the (sub)-gradients required by the Newton algorithm can be computed in a strongly polynomial time. The hybrid discrete Newton algorithm finds a zero point of a univariate nonlinear equation in a finite number of steps. By contrast, the conventional bisection algorithm only guarantees ϵ-approximated solutions for $\epsilon > 0$.

Lastly, we extend the previous findings to constraint sets involving an SS function. We show that any Boolean multilinear function is an SS function. This result yields intersection cuts for multilinear constraints in binary variables.

1.2 Literature review

The work of Jack Edmonds [36] plays a prominent role in the study of the combinatorial properties of submodular functions. We refer to [66] for basic concepts and definitions. The convex envelope of a submodular function f is its Lovász extension [8, 52]. Submodular functions are a subclass of discrete convex functions, and we refer to [58] for more details about discrete convex analysis.

Valid inequalities for the hypographs of general submodular functions are called the *base inequalities* [60]. For a class of special submodular functions, lifting procedures [2, 69] can strengthen the base inequalities. The base inequalities can be separated either using heuristics [2] or at integer points in a Benders-like framework [27]. The method defined in [8] combines valid inequalities for the submodular and supermodular components of an SS function. We refer to [6, 7, 17, 18, 44, 48, 61, 68, 77, 78] for more details about the exploitation of submodular/supermodular functions in mathematical programs. Supermodular polynomials in binary variables are defined and studied in [17, 61]. The submodularity of the D-OPTIMAL DESIGN problem is exploited in [63, 68].

As mentioned above, intersection cuts generate valid inequalities for sets that are hard to optimize over. Gomory introduced the corner polyhedron [42], and his celebrated mixed-integer cuts [43] are special intersection cuts derived from split disjunctions [59]. The definition of intersection cuts for arbitrary set $\mathcal{S}$ is due to [35, 40]. We refer to [3, 4, 10, 12, 25, 26, 28, 34, 35, 62] for a more in-depth analysis. The method given in [72] can generate valid inequalities that cut off points outside $\mathcal{S}$-free sets. We refer to [4, 13, 49, 50, 54, 55] for relevant recent developments in mixed-integer conic programming.

For the cases where the nonconvexity of $\mathcal{S}$ is not just due to integer variables, we refer to [37] for bilevel programs, [16] for outer-product sets, [57, 56] for quadratic constraint sets, [77] for signomial-term sets, and [38] for bilinear sets. The method given in [67] constructs intersection cuts for sets arising from factorable programs that contain DC functions [47].

Next, we discuss valid inequalities for polynomial programming, because we use polynomial programs in binary variables as a benchmark in our computational study. In [16], intersection cuts approximate a nonconvex lifted set, namely the outer product set arising from the extended formulation of a polynomial program. Lifted sets link decision variables to auxiliary variables representing (graphs of) monomials up to a given degree. We remark that in most combinatorial optimization problems, decision variables are binaries. The polynomial program of interest is then a Boolean Multilinear Program (BMP). The corresponding lifted set is the *Boolean multilinear set* [29, 39], the convex hull of which is the so-called Boolean multilinear polytope. Valid inequalities for the Boolean multilinear polytope may be stronger than those for the convex hull of the outer product set. Various Gomory-Chvátal-based inequalities [30, 31, 32, 33] are valid for the multilinear polytope. The separation and strength of these inequalities depend on the hypergraph representing the underlying sparsity pattern of the multilinear set.

We consider a constrained polynomial program, and assume that some of its constraints are neither integrality constraints nor variable bound constraints. After lifting, those constraints are linear constraints and thus define a convex set $\mathcal{S}_1$. The lifted set $\mathcal{S}_2$ is nonconvex, and $\mathcal{S}_1 \not\subseteq \mathcal{S}_2$. The polynomial program is then equivalent to linear optimization over $\mathrm{conv}(\mathcal{S}_1 \cap \mathcal{S}_2)$. However, in general, $\mathrm{conv}(\mathcal{S}_1 \cap \mathcal{S}_2) \neq \mathcal{S}_1 \cap \mathrm{conv}(\mathcal{S}_2)$, so the convexification of the lifted set may not yield an equivalent convex problem. To address this issue, one attempt is to directly consider $\mathrm{conv}(\mathcal{S}_1 \cap \mathcal{S}_2)$ and generate valid inequalities for it. Some work in this sense exists for certain interesting special cases, e.g. the intersection of multilinear sets with additional constraint sets such as cardinality constraints [20]. Another attempt is to consider constraints in projected formulations, e.g., in mixed-integer quadratically constrained quadratic programs [65]. Since the representation complexity of the projected formulation is smaller than that of the extended formulation, this approach is also amenable to computation. In [21, 57], intersection cuts for the set defined by a quadratic constraint are derived. If additionally, the nonbasic variables of the LP relaxation are integer, the monoidal technique [22] can strengthen such intersection cuts.

However, generating valid inequalities for Boolean multilinear constraints, and, more generally, constructing $\mathcal{S}$-free sets for nonlinear constraints on discrete variables, remain problems of considerable interest. In this paper, we look at these questions through a "submodularity lens".

1.3 Notation

We let $[n] := \{1, \cdots, n\}$ for any positive integer n. We denote $\mathcal{B} := \{0,1\}^n$, $\bar{\mathcal{B}} := [0,1]^n$. We assume that $[n]$ is equipped with the natural number order. $\mathbf{1}$ denotes the all-one vector, and $\mathbf{0}$ denotes the all-zero vector. For $S \subseteq [n]$, we denote by $\mathsf{supp}(S) \in \mathcal{B}$ the characteristic vector of S. For vectors a, b, we let (a, b) be their concatenation, and extend this notation naturally to the case where b is a scalar. Given a set $\mathcal{D} \subseteq \mathbb{R}^n$ and a function $g : \mathcal{D} \to \mathbb{R}$, we adopt the usual notation $\mathrm{epi}_{\mathcal{D}}(g), \mathrm{gra}_{\mathcal{D}}(g), \mathrm{hypo}_{\mathcal{D}}(g)$ to denote the epigraph, graph and hypograph of g over $\mathcal{D}$, respectively. For example, $\mathrm{gra}_{\mathcal{D}}(g) := \{(x, t) \in \mathcal{D} \times \mathbb{R} : g(x) = t\}$. When $\mathcal{D}$ is omitted in the subscript, it is assumed to be $\mathbb{R}^n$. For any set $\mathcal{S}$, we denote by $\mathrm{bd}(\mathcal{S}), \mathrm{ext}(\mathcal{S}), \mathrm{int}(\mathcal{S})$ its boundary, extreme points, interior, respectively. When $\mathcal{S}$ is not full-dimensional, $\mathrm{relint}(\mathcal{S}), \mathrm{relbd}(\mathcal{S})$ denote its relative interior and relative boundary.

1.4 Outline

The rest of the paper is organized as follows. In Sect. 2, we recall some preliminaries for intersection cuts. In Sect. 3, we study extensions of submodular functions. In Sect. 4, we study the hypograph-free sets for the submodular function. In Sect. 5, we generalize the previous results for sets involving SS functions. In Sect. 6, we consider applications for intersection cuts to Boolean multilinear constraints and BAYESIAN D-OPTIMAL DESIGN. In Sect. 7, we propose the hybrid discrete Newton algorithm for computing intersection cuts. In Sect. 8, we analyze the computational results.

2 Intersection cut preliminaries

In this section, we review the basic concept of intersection cuts. Assume that we are solving the optimization problem $\min_{z \in \mathcal{S}} cz$. Given a polyhedral outer approximation $\mathcal{P}$ of a nonconvex set $\mathcal{S}$, an LP relaxation is $\min_{z \in \mathcal{P}} cz$. Then an optimal *relaxation point* $\tilde{z}$ is a vertex of $\mathcal{P}$. An intersection cut is a particular cut that separates $\tilde{z}$ from $\mathcal{S}$.

Definition 1. *Given $\mathcal{S} \in \mathbb{R}^p$, a closed set $\mathcal{C}$ is called (convex) $\mathcal{S}$-free, if $\mathcal{C}$ is convex and $\mathrm{int}(\mathcal{C}) \cap \mathcal{S} = \varnothing$.*

The construction of intersection cuts [24] needs two components: a corner polyhedron relaxation $\mathcal{R}$ of $\mathcal{S}$ with apex $\tilde{z}$ ($\mathcal{R}$ can be extracted from $\mathcal{P}$), and an $\mathcal{S}$-free set $\mathcal{C}$ containing $\tilde{z}$ in its interior. Then an intersection cut separates $\tilde{z}$ from $\operatorname{conv}(\mathcal{R} \smallsetminus \operatorname{int}(\mathcal{C}))$ (a set which, we note, contains $\mathcal{S}$) as follows. The half-space and ray representations of the corner polyhedron $\mathcal{R}$ is as follows:

$$\mathcal{R} := \{z \in \mathbb{R}^p : A(z - \tilde{z}) \leq 0\} = \{z \in \mathbb{R}^p : \exists \eta \in \mathbb{R}^p_+ \ z = \tilde{z} + \sum_{j=1}^p \eta_j r^j\}, \tag{2}$$

where A is a p-by-p invertible matrix, r^j is the j-th column of $-A^{-1}$ and an extreme ray of $\mathcal{R}$.

Define the *step length*

$$\eta_j^* := \sup_{\eta_j \geq 0} \{\eta_j : \tilde{z} + \eta_j r^j \in \mathcal{C}\}. \tag{3}$$

The point $\tilde{z}$ is separated by an intersection cut

$$\sum_{j=1}^p \frac{1}{\eta_j^*} A_j(z - \tilde{z}) \leq -1.$$

Let $\mathcal{C}, \mathcal{C}^*$ be two $\mathcal{S}$-free sets such that $\mathcal{C} \subseteq \mathcal{C}^*$. Then the intersection cut derived from $\mathcal{C}^*$ dominates the intersection cut derived from $\mathcal{C}$. This makes maximal $\mathcal{S}$-free sets relevant in the study.

3 Extensions of submodular functions

In this section, we study continuous extensions of submodular functions. W.l.o.g., we assume in the sequel that for any submodular function f, $f(\mathbf{0}) = 0$ holds (by a translation of a constant). It is known that the Lovász extension [52] extends f from $\mathcal{B}$ to $\bar{\mathcal{B}}$. Based on this extension, we construct another extension F of f defined over the entire space $\mathbb{R}^n$, and study its analytical and combinatorial structures.

We first look at some polyhedra associated with the submodular function f [8, 66]. Its *extended polymatroid* is defined as

$$EPM_f := \{s \in \mathbb{R}^n : \forall x \in \mathcal{B}, \ sx \leq f(x)\},$$

its convex hull of the epigraph f over $\mathcal{B}$ is defined as

$$Q_f := \operatorname{conv}(\operatorname{epi}_{\mathcal{B}}(f)).$$

Recall that $\operatorname{ext}(EPM_f)$ are the vertices of EPM_f, and we can further define a polyhedron

$$EE_f := \{(x, t) \in \mathbb{R}^{n+1} : \forall s \in \operatorname{ext}(EPM_f), \ sx \leq t\}. \tag{4}$$

In fact, EE_f includes Q_f, because of the following lemma:

Lemma 1 ([8]). $Q_f = EE_f \cap (\bar{\mathcal{B}} \times \mathbb{R})$.

Therefore, $x \in \bar{\mathcal{B}}$ defines trivial facets of Q_f, and non-trivial facets of Q_f are $sx \leq t$, where s is a vertex of EPM_f.

These polyhedra in turn give rise to some functions associated with f. A convex function g is a *convex underestimating function* of f over $\mathcal{B}$, if for all $x \in \mathcal{B}$, $g(x) \leq f(x)$. The *convex envelope*

$\text{env}_{\mathcal{B}}(f)$ is thus the maximal convex underestimating function of f over $\mathcal{B}$. Since Q_f is the epigraph of $\text{env}_{\mathcal{B}}(f)$, by Lemma 1,

$$\text{env}_{\mathcal{B}}(f) : \bar{\mathcal{B}} \to \mathbb{R} : x \to \max_{s \in \text{ext}(EPM_f)} sx,$$

where its domain is $\bar{\mathcal{B}}$. We remark that $\text{env}_{\mathcal{B}}(f)$ is equivalent to the Lovász extension of f [8]. We will show that the cardinality $|\text{ext}(EPM_f)|$ is not polynomial to n. Thereby, when computing $\text{env}_{\mathcal{B}}(f)$, it is inefficient to evaluate all sx for $s \in \text{ext}(EPM_f)$. In fact, the value and the (sub)-gradients of $\text{env}_{\mathcal{B}}(f)$ on $\bar{\mathcal{B}}$ can be computed in a strongly polynomial time [8].

We define the envelope of f extended to $\mathbb{R}^n$ as

$$\mathsf{F} : \mathbb{R}^n \to \mathbb{R} : x \to \max_{s \in \text{ext}(EPM_f)} sx.$$

We note that F simply enlarges the domain of $\text{env}_{\mathcal{B}}(f)$ from $\bar{\mathcal{B}}$ to $\mathbb{R}^n$. This extension is algebraically simple, but analytical properties of $\mathsf{F}(x)$ outside $\bar{\mathcal{B}}$ will be studied in further detail. We find that EE_f is the epigraph of F, i.e., $EE_f = \text{epi}(\mathsf{F})$, so F is a convex function. Since every facet $sx \leq t$ of EE_f is in one-to-one correspondence to a linear underestimator function sx of F, we call EE_f the *extended envelope epigraph*.

A fundamental question on F is how to compute its value and (sub)-gradients efficiently, because this is crucial in constructing intersection cuts. Since the Lovász extension $\text{env}_{\mathcal{B}}(f)$ is a restriction of F on the hypercube $\bar{\mathcal{B}}$, we can compute F efficiently (in a strongly polynomial time) on $\bar{\mathcal{B}}$. In the following, we will show how to extend this method to compute F over the entire space $\mathbb{R}^n$. This extension requires us to study the properties of F and EE_f.

We first look at combinatorial structures associated with the facets of EE_f. Recall that a permutation π on $[n]$ is a bijective map from $[n]$ to itself. The map $\pi(i) \in [n]$ is the image of an element $i \in [n]$ under this permutation. We denote by Π the set of permutations on $[n]$. We define the following sets and vectors related to permutations.

Definition 2. *Given a permutation $\pi \in \Pi$ and an integer $i \in \{0, \cdots, n\}$, define $\pi([i]) := \{\pi(1), \cdots, \pi(i)\}$ $(\pi([0]) := \varnothing)$, and define $v^i(\pi) := \text{supp}(\pi([i]))$. Define the map $\sigma : \Pi \to \mathbb{R}^n$ such that it satisfies $\sigma(\pi)_{\pi(i)} = f(v^i(\pi)) - f(v^{i-1}(\pi))$ for all $\pi \in \Pi$ and for all $i \in [n]$.*

The set of vertices $\text{ext}(EPM_f)$ is the image of Π under the map σ.

Lemma 2 ([36]). *$\sigma(\Pi) = \text{ext}(EPM_f)$.*

Every permutation $\pi \in \Pi$ induces a vertex $\sigma(\pi)$ of $\text{ext}(EPM_f)$ through the map σ, so the cardinality of $\text{ext}(EPM_f)$ is $n!$ (not polynomial to n). The above lemma shows that every facet of EE_f (a non-trivial facet of Q_f) is given as $\sigma(\pi)x \leq t$, and every linear underestimator of F is given as $\sigma(\pi)x$.

Proposition 1. *Given a permutation $\pi \in \Pi$, for all $i \in [n] \cup \{0\}$, the facet-defining inequality $\sigma(\pi)x \leq t$ is supported by $(v^i(\pi), f(v^i(\pi)))$, i.e., $\sigma(\pi)v^i(\pi) = f(v^i(\pi))$.*

Proof.

$$\sigma(\pi)v^i(\pi) = \sum_{j \in [i]} \sigma(\pi)_{\pi(j)} = \sum_{j \in [i]} \left(f(v^j(\pi)) - f(v^j(\pi)) \right) = f^i(v^i(\pi)) - f(0) = f(v^i(\pi)),$$

where the first equation follows from Defn. 2, the second equation follows from Lemma 2, and the last two equations follow from the expansion of the sum. □

Conversely to Prop. 1, given a point in the graph of f, we can construct all the facets supported by it.

Corollary 1. *For a point $v \in \mathcal{B}$, let ι be the number of ones in v. If a permutation $\pi \in \Pi$ satisfies that $v = v^{\iota}(\pi)$, then $(v, f(v))$ supports the facet-defining inequality $\sigma(\pi)x \leq t$ of EE_f.*

At the moment, we find that one can easily obtain the facial structure of EE_f from that of Q_f. We ask how to separate facets of EE_f. Since EE_f is the epigraph of F, the shape of EE_f is determined by F, so it suffices to look at F.

From a convex analysis perspective, the nonsmooth polyhedral function F is the maximum of a set of linear functions, so it is convex and positive homogeneous of degree 1. This means that F is subdifferentiable [45]. Moreover, F has the following analytical properties.

Proposition 2. *For all $x', x \in \mathbb{R}^n$ and all $s \in \partial\mathsf{F}(x')$, $\mathsf{F}(x') = sx'$ and $\mathsf{F}(x) \geq sx$. Moreover, $\partial\mathsf{F}(x') = \mathrm{conv}(\mathrm{argmax}_{s \in \mathrm{ext}(EPM_f)} sx')$.*

Proof. As $\mathsf{F}(x) = \max_{s \in \mathrm{ext}(EPM_f)} sx$, F is the maximum of a set of linear functions. This implies that it is positive-homogeneous of degree-1 and convex, and it is easy to show the other results. $\square$

Given $\tilde{x} \in \mathbb{R}^n$, the evaluation of $\mathsf{F}(\tilde{x})$ is called the *extended polymatroid vertex maximization problem*, as by definition $\mathsf{F}(\tilde{x})$ equals

$$\max_{s \in \mathrm{ext}(EPM_f)} s\tilde{x}. \tag{5}$$

By Prop. 2, an optimal solution s^* is a subgradient of F at $\tilde{x}$ (i.e., $s^* \in \partial\mathsf{F}(\tilde{x})$). By Lemma 2, $\max_{s \in \mathrm{ext}(EPM_f)} s\tilde{x} = \max_{\pi \in \Pi} \sigma(\pi)\tilde{x}$, so (5) asks for a permutation π^* that maximizes $\sigma(\pi^*)\tilde{x}$. One of the main findings in this section is an algorithm to solve (5).

To tackle (5), we look at a related relaxed problem, namely the *extended polymatroid maximization problem*, that is well studied:

$$\max_{s \in EPM_f} s\tilde{x}. \tag{6}$$

If $\tilde{x} \geq 0$, a strongly polynomial time *sorting algorithm* can solve the extended polymatroid maximization [36]: Let $\pi^* \in \Pi$ be a permutation such that $\tilde{x}_{\pi^*(1)} \geq \cdots \geq \tilde{x}_{\pi^*(n)}$, then an optimal solution to (6) is $\sigma(\pi^*)$.

We note that the vertices $\mathrm{ext}(EPM_f)$ are a finite set, so (5) is always bounded; EPM_f is the Minkowski sum of $\mathrm{ext}(EPM_f)$ and a set of recession rays, so EPM_f is unbounded. This means that (6) can be unbounded.

Lemma 3. *[[8, 36]] When $\tilde{x} \geq 0$, the optimum of (6) must be a vertex, and (5) is equivalent to (6); when $\tilde{x}$ has some negative entries, (5) is unbounded and is not equivalent to (6).*

Therefore, (5) is not equivalent to (6) for $x \in \mathbb{R}^n \setminus \mathbb{R}^n_+$. However, we can show that the sorting algorithm also solves the problem (5) for any case.

Proposition 3. *The output of the sorting algorithm is optimal to the extended polymatroid vertex maximization problem (5).*

Proof. Let π^* be the permutation found by the sorting algorithm. By Lemma 2, $\sigma(\pi^*)$ is in $\mathrm{ext}(EPM_f)$ and hence a feasible solution to (5). Next, we prove the optimality of $\sigma(\pi^*)$. Let $d := \min_{i \in [n]} \tilde{x}_i$, then the translated vector $\tilde{x} - d\mathbf{1} = (\tilde{x}_i - d)_{i \in [n]} \geq \mathbf{0}$. The following inequalities hold:

$$\sigma(\pi^*)\tilde{x} \leq \max_{s \in \mathrm{ext}(EPM_f)} s\tilde{x} = \max_{s \in \mathrm{ext}(EPM_f)} s(\tilde{x} - d\mathbf{1} + d\mathbf{1})$$
$$\leq \max_{s \in \mathrm{ext}(EPM_f)} s(\tilde{x} - d\mathbf{1}) + \max_{s \in \mathrm{ext}(EPM_f)} s(d\mathbf{1}). \tag{7}$$

It is easy to show that $(\tilde{x} - d\mathbf{1})_{\pi^*(1)} \geq \cdots \geq (\tilde{x} - d\mathbf{1})_{\pi^*(n)}$. As $\tilde{x} - d\mathbf{1} \geq \mathbf{0}$, by Lemma 3 and the sorting algorithm, $\sigma(\pi^*)(\tilde{x} - d\mathbf{1}) = \max_{s \in EPM_f} s(\tilde{x} - d\mathbf{1}) = \max_{s \in \text{ext}(EPM_f)} s(\tilde{x} - d\mathbf{1})$. It follows from Prop. 1 that $\sigma(\pi)v^n(\pi) = \sigma(\pi)\mathbf{1} = f(\mathbf{1})$. As the entries of $d\mathbf{1}$ are identical, for any $\pi \in \Pi$, $\sigma(\pi)(d\mathbf{1}) = df(\mathbf{1})$. Therefore, for any $s \in \text{ext}(EPM_f)$, $s(d\mathbf{1}) = df(\mathbf{1})$, so $\sigma(\pi^*)(d\mathbf{1}) = \max_{s \in \text{ext}(EPM_f)} s(d\mathbf{1})$. Looking at the inequalities (7), they become equations, because

$$\sigma(\pi^*)\tilde{x} \leq \max_{s \in \text{ext}(EPM_f)} s\tilde{x} \leq \sigma(\pi^*)(\tilde{x} - d\mathbf{1}) + \sigma(\pi^*)(d\mathbf{1}) = \sigma(\pi^*)\tilde{x}.$$

Therefore, $\sigma(\pi^*)$ is an optimal solution to (5). $\qquad\square$

Given $\tilde{x} \in \mathbb{R}^n$, the sorting algorithm outputs a permutation on it. The sorting algorithm is translation-invariant, i.e., translating each entry of $\tilde{x}$ by the same value does not change the output permutation. A byproduct of Prop. 3 is that the translation invariance implies the ray-linearity of F.

Corollary 2. *Let $\tilde{x} \in \mathbb{R}^n$, then F is linear on $\tilde{x} + \lambda\mathbf{1}$ w.r.t. $\lambda \in \mathbb{R}$.*

We look at the boundary of EE_f. By Prop. 1 and Cor. 1, for all $x \in \mathcal{B}$, the point $(x, f(x))$ supports some facets of EE_f.

Theorem 1. $EE_f \cap \text{hypo}_{\mathcal{B}}(f) = \text{gra}_{\mathcal{B}}(f) \subseteq \text{bd}(EE_f)$.

Proof. We consider a point $v \in \mathcal{B}$ and look at the line $\ell = \{(v, t) : t \in \mathbb{R}\}$. It can be separated into the restricted epigraph $\ell_+ := \{(v, t) : f(v) \leq t\}$ and the restricted hypograph $\ell_- := \{(v, t) : f(v) \geq t\}$, as $\ell_+ \cap \ell_- = (v, f(v))$ and $\ell = \ell_+ \cup \ell_-$. First, we know that, by definition of Q_f and Lemma 1, $\ell_+ \subseteq Q_f \subseteq EE_f$. Second, by Prop. 1, the point $(v, f(v))$ supports some facets of EE_f, so the point (v, t) with $t < f(v)$ is separated by these facets from EE_f. Thereby, we know that $\ell_- \cap EE_f = \{(v, f(v))\}$. To summarize, we know that $EE_f \cap \ell = \ell_+$ and $(v, f(v)) \in \text{bd}(EE_f)$. As $\text{gra}_{\mathcal{B}}(f) = \cup_{v \in B}\{(v, f(v))\}$, we have that $\text{gra}_{\mathcal{B}}(f) \subseteq \text{bd}(EE_f)$. As the hypograph $\text{hypo}_{\mathcal{B}}(f) = \cup_{v \in \mathcal{B}}\{(v, t) : f(v) \geq t\}$ (union of restricted hypographs), we have that $EE_f \cap \text{hypo}_{\mathcal{B}}(f) = \text{gra}_{\mathcal{B}}(f)$. $\qquad\square$

As already mentioned, F is convex and $EE_f = \text{epi}(\mathsf{F})$, so F is also a continuous extension of f. As EE_f includes Q_f, F further extends $\text{env}_{\mathcal{B}}(f)$ (the Lovász extension).

We now understand the facial structure of EE_f, which will help us construct hypograph-free sets. We also know how to compute the value and subgradients of F at any point in $\mathbb{R}^n$, which is important for constructing intersection cuts.

4 Hypograph-free sets for submodular functions

In this section, we construct two types of hypograph-free sets for the submodular function f.

First, we show that one can lift a maximal $\mathcal{B}$-free set into a maximal $\text{hypo}_{\mathcal{B}}(f)$-free set.

Theorem 2. *Let $f : \mathcal{B} \to \mathbb{R}$ be an arbitrary function, and let $\mathcal{K}$ be a maximal $\mathcal{B}$-free set in $\mathbb{R}^n$. Then $\mathcal{C} := \mathcal{K} \times \mathbb{R}$ is a maximal $\text{hypo}_{\mathcal{B}}(f)$-free set.*

Proof. We note that $\text{int}(\mathcal{C}) = \text{int}(\mathcal{K}) \times \mathbb{R}$. It is easy to show that $\mathcal{C}$ is $\text{hypo}_{\mathcal{B}}(f)$-free, since $\text{int}(\mathcal{C}) \cap \text{hypo}_{\mathcal{B}}(f) = \varnothing$. Assume that there exists a $\text{hypo}_{\mathcal{B}}(f)$-free set $\mathcal{C}'$ including $\mathcal{C}$. Then the recession cone of $\mathcal{C}'$ must include that of $\mathcal{C}$, so $\mathcal{C}' = \mathcal{K}' \times \mathbb{R}$ for some closed convex set $\mathcal{K}'$ including $\mathcal{K}$. Moreover, $\mathcal{K}'$ must be a $\mathcal{B}$-free set, otherwise, there exists a point $x \in \mathcal{B} \cap \text{int}(\mathcal{K}')$ such that $(x, f(x)) \in \text{int}(\mathcal{K}') \times \mathbb{R} = \text{int}(\mathcal{C}')$. However, since $\mathcal{K}$ is maximally $\mathcal{B}$-free, this implies that $\mathcal{K} = \mathcal{K}'$. As a result, $\mathcal{C} = \mathcal{C}'$, so $\mathcal{C}$ is maximal. $\qquad\square$

This construction does not rely on any structure of f, as it just lifts a $\mathcal{B}$-free set. For any $j \in [n]$, the simple lifted split $\{x \in \mathbb{R}^n : 0 \le x_j \le 1\} \times \mathbb{R}$ is a maximal $\text{hypo}_{\mathcal{B}}(f)$-free set. We next construct $\text{hypo}_{\mathcal{B}}(f)$-free sets using the submodularity, for both theoretical and computational interests.

We show that the extended envelope epigraph is a hypograph-free set.

Proposition 4. EE_f, Q_f *are* $\text{hypo}_{\mathcal{B}}(f)$-*free sets.*

Proof. Since $\text{gra}_{\mathcal{B}}(f) \subseteq \text{bd}(EE_f)$, we conclude that $EE_f \cap \text{hypo}_{\mathcal{B}}(f) \subseteq \text{bd}(EE_f)$ and hence $\text{int}(EE_f) \cap \text{hypo}_{\mathcal{B}}(f) = \varnothing$. Additionally, EE_f is convex and hence $\text{hypo}_{\mathcal{B}}(f)$-free. As $Q_f \subseteq EE_f$, Q_f is $\text{hypo}_{\mathcal{B}}(f)$-free set. $\qquad\square$

It is known that for a convex function, its maximal hypograph-free set is its epigraph. However, for the submodular function f, we will show that its extended epigraph EE_f is not a maximal hypograph-free set. A high-level way to test the maximality of EE_f is as follows. The set Q_f is the convex hull of $\text{epi}_{\mathcal{B}}(f)$. Geometrically, Q_f is the "minimal" convex set including $\text{epi}_{\mathcal{B}}(f)$. This is a conflict as we aim to obtain an inclusion-wise "maximal" $\text{hypo}_{\mathcal{B}}(f)$-free set. We can remove some facets from Q_f and thus enlarge this polyhedron. After removing trivial facets of Q_f, the enlarged polyhedron is the extended envelope epigraph EE_f. However, this enlargement is still not enough.

We look at a concrete characterization of the "correct" enlarging of EE_f. The following fundamental theorem gives a sufficient and necessary condition on (maximal) hypograph-free sets including EE_f.

Theorem 3. *Let* $\mathcal{C}$ *be a full-dimensional closed convex set in* $\mathbb{R}^{n+1}$ *including* EE_f. *Then* $\mathcal{C}$ *is a* $\text{hypo}_{\mathcal{B}}(f)$-*free set if and only if* $\mathcal{C}$ *is* $\text{gra}_{\mathcal{B}}(f)$-*free. Moreover,* $\mathcal{C}$ *is a maximal* $\text{hypo}_{\mathcal{B}}(f)$-*free set if and only if* $\mathcal{C}$ *is a polyhedron and there is at least one point of* $\text{gra}_{\mathcal{B}}(f)$ *in the relative interior of each facet of* $\mathcal{C}$.

Proof. We note that by Thm. 1, $\text{gra}_{\mathcal{B}}(f) \subseteq \text{bd}(EE_f) \subseteq EE_f \subseteq \mathcal{C}$. Thereby, $\text{gra}_{\mathcal{B}}(f) \cap \text{int}(\mathcal{C}) = \varnothing$ (i.e., $\mathcal{C}$ is $\text{gra}_{\mathcal{B}}(f)$-free) if and only if $\text{gra}_{\mathcal{B}}(f) \subseteq \text{bd}(\mathcal{C})$.

We consider the $\mathcal{S}$-freeness first. We prove the forward direction. Assume that $\mathcal{C}$ is a $\text{hypo}_{\mathcal{B}}(f)$-free set. Suppose, to aim at a contradiction, that there exists a point $(v, f(v)) \in \text{int}(\mathcal{C}) \cap \text{gra}_{\mathcal{B}}(f)$. Then there exists a sufficiently small $\epsilon > 0$ such that $(v, f(v) - \epsilon) \in \text{int}(\mathcal{C})$, but $(v, f(v) - \epsilon) \in \text{hypo}_{\mathcal{B}}(f)$, which leads to a contradiction. We prove the reverse direction. Assume that $\mathcal{C}$ is $\text{gra}_{\mathcal{B}}(f)$-free. Suppose, to aim at a contradiction, that there exists a point $(v, f(v) - \delta) \in \text{int}(\mathcal{C})$ with $v \in \mathcal{B}$ and $\delta > 0$. As, for some $\epsilon > 0$, $(v, f(v) + \epsilon) \subseteq \text{int}(EE_f) \subseteq \text{int}(\mathcal{C})$, by convexity of $\mathcal{C}$, $(v, f(v)) \in \text{int}(\mathcal{C})$, which leads to a contradiction. This implies that $\mathcal{C}$ is $\text{hypo}_{\mathcal{B}}(f)$-free if and only if $\text{gra}_{\mathcal{B}}(f)$-free (or $\text{gra}_{\mathcal{B}}(f) \subseteq \text{bd}(\mathcal{C})$).

We consider the maximality next. Due to [11], a full-dimensional lattice-free set is maximal if it is a polyhedron and there is at least one lattice point in the relative interior of each facet. As $\text{gra}_{\mathcal{B}}(f)$ is a finite set, the proof strategy is similar although it is not a subset of any lattice. Then the result follows. $\qquad\square$

The above theorem is purely geometrical. Since submodular functions are combinatorial objects, we translate this theorem into a combinatorial language. We first define a combinatorial object in the Boolean hypercube $\mathcal{B}$.

Definition 3. *Let* $x^0, x^1, \cdots, x^n$ *be* $n + 1$ *distinct points of* $\mathcal{B}$. *They are called monotone, if* $\mathbf{0} = x^0 < x^1 < \cdots < x^n = \mathbf{1}$. *We call the corresponding ordered set* $(x^0, \cdots, x^n) \subseteq \mathcal{B}$ *a monotone chain in* $\mathcal{B}$.

Therefore, we use a monotone chain to represent a set of monotone points. Then we have the following observation.

Proposition 5. *The set of monotone chains is in one-to-one correspondence to the set Π of permutations via the map V defined as follows: for all $\pi \in \Pi$, $V(\pi) := (v^i(\pi) \mid i \in \mathcal{N} \cup \{0\})$.*

Proof. By Prop. 1, since $\varnothing = \pi([0]) \subsetneq \cdots \subsetneq \pi([n]) = [n]$, by Defn. 2, $0 = v^0(\pi) < \cdots < v^n(\pi) = 1$, so $V(\pi)$ is a monotone chain. Conversely, given a monotone chain $(x^0, \cdots, x^n)$, its inverse map π exists and satisfies that $\pi(0) = 0$; and for all $i \in [n]$, $\pi(i)$ is the index of the unique non-zero entry of $x^i - x^{i-1}$. $\qquad\square$

We find that permutations and monotone chains are indeed equivalent. We note that any $n+1$ distinct points from $\mathrm{gra}_{\mathcal{B}}(f)$ are affinely independent in $\mathbb{R}^{n+1}$ and hence support a hyperplane in $\mathbb{R}^{n+1}$. Thereby, we can infer from Prop. 1 and Prop. 5 that

Corollary 3. *If $(x^0, \cdots, x^n)$ is a monotone chain in $\mathcal{B}$, then distinct points $(x^0, f(x^0)), \cdots, (x^n, f(x^n))$ of $\mathrm{gra}_{\mathcal{B}}(f)$ define (or support) a facet of the extended envelope epigraph EE_f.*

We say that this monotone chain *induces the facet*. In fact, we find that facets of EE_f, permutations on $[n]$, and monotone chains in $\mathcal{B}$ are in one-to-one correspondence. Therefore, we can view them as the same objects. Especially, Prop. 5 relates permutations and monotone chains. We give the following characterization of permutations on $[n]$.

Definition 4. *A subset Π' of permutations of Π is called a cover, if $\bigcup_{\pi \in \Pi'} V(\pi) = \mathcal{B}$; moreover, Π' is called a minimal cover, if additionally, for all $\pi \in \Pi'$, $V(\pi) \setminus \bigcup_{\pi' \in \Pi' : \pi' \neq \pi} V(\pi')$ is not empty.*

We want to enlarge EE_f by removing its facets, this is equivalent to removing permutations from Π. Let Π' be a subset of permutations of Π, and $\mathcal{C}(\Pi') := \{(x,t) : \forall \pi \in \Pi', \ \sigma(\pi)x \leq t\}$ denotes the relaxation of the extended envelope epigraph induced by Π'. It is obvious that $EE_f = \mathcal{C}(\Pi) \subseteq \mathcal{C}(\Pi')$ for any $\Pi' \subseteq \Pi$. The following corollary translates Thm. 3 in a combinatorial language.

Corollary 4. *Let Π' be a subset of permutations of Π. $\mathcal{C}(\Pi')$ is $\mathrm{hypo}_{\mathcal{B}}(f)$-free if and only if Π' is a cover. $\mathcal{C}(\Pi')$ is maximally $\mathrm{hypo}_{\mathcal{B}}(f)$-free if and only if Π' is a minimal cover.*

Proof. First, we note that $C(\Pi')$, as a relaxation of EE_f includes $\mathrm{gra}_{\mathcal{B}}(f)$. Next, we assume that Π' is a cover. Then points of $\mathrm{gra}_{\mathcal{B}}(f)$ support facets of $C(\Pi')$. By Thm. 3, $C(\Pi')$ is $\mathrm{hypo}_{\mathcal{B}}(f)$-free if and only if it is a cover. Finally, Π' is a minimal cover, if and only if then each facet of $C(\Pi')$ has a point of $\mathrm{gra}_{\mathcal{B}}(f)$ in its interior. By Thm. 3, the later is equivalent to that $C(\Pi')$ is maximally $\mathrm{hypo}_{\mathcal{B}}(f)$-free. $\qquad\square$

We now can disprove the maximality EE_f by a counter-example. Thanks to the Cor. 4, we can use a counting argument to show that we can remove facets from EE_f. This results in a new enlarged $\mathrm{hypo}_{\mathcal{B}}(f)$-free polyhedron.

Proposition 6. *EE_f is not maximally hypograph-free.*

Proof. It suffices to find a counter-example. Consider $n = 3$, $\mathcal{B} = \{0,1\}^3$, there are 6 permutations, and 6 monotone chains (see Fig. 1). We assume that, in a non-degenerate case, the associated extended envelope epigraph EE_f has 6 facets induced by 6 chains respectively. The vertices $(0,0,0)$ and $(1,1,1)$ are visited by all the chains, while the other vertices are visited twice each. Therefore, a chain cannot "exclusively" visit a vertex, so the corresponding facet cannot contain one point
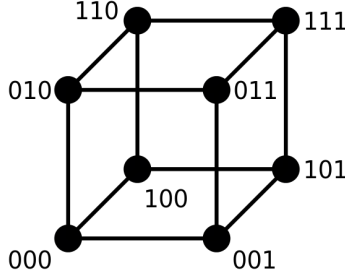
10

Figure 1: $\mathcal{B} = \{0, 1\}^3$

of $\mathrm{gra}_\mathcal{B}(f)$ in its relative interior. In fact, we can remove some facets from the extended envelope epigraph. We keep three chains:

$$((0,0,0), (0,0,1), (0,1,1), (1,1,1)),$$
$$((0,0,0), (0,1,0), (1,1,0), (1,1,1)),$$
$$((0,0,0), (1,0,0), (1,0,1), (1,1,1)).$$

These chains induce 3 facets such that at least one point of $\mathrm{gra}_\mathcal{B}(f)$ is in the relative interior of each facet and each point of $\mathcal{B}$ is in these 3 facets, so the polyhedron defined by these 3 facets is a $\mathrm{hypo}_\mathcal{B}(f)$-free set larger than EE_f.

□

We explain the hardness to enlarge EE_f. We build a bipartite graph $G := (\mathcal{B} \cup \Pi, E)$. An edge e of E connects a vertex $v \in \mathcal{B}$ to a permutation $\pi \in \Pi$ if $v \in V(\pi)$. Then, a minimal cover is a subset Π' of Π such that i) each vertex of $\mathcal{B}$ is incident to at least one permutation in Π'; ii) each permutation in Π' is incident to a vertex of $\mathcal{B}$ that no other permutation in Π' is incident to. As $|\mathcal{B}| = 2^n$ and $|\Pi| = n!$, the size of such a graph is not polynomial to n. Therefore, one may need additional structural information to enlarge EE_f efficiently.

We relax the submodular maximization problem (1) via a polyhedral outer approximation $\mathcal{P}$ of $\mathrm{hypo}_\mathcal{B}(f)$. Let X be the orthogonal projection of $\mathcal{P}$ on x-space. We remark that, within a branch-and-cut algorithm, X might be within a low-dimensional face of $\bar{\mathcal{B}}$. Let $\tilde{z} := (\tilde{x}, \tilde{t})$ be a solution to the LP relaxation $\max_{(x,t) \in \mathcal{P}} t$. We assume that $\tilde{x} \notin \mathcal{B}$, otherwise, $\tilde{x}$ is already an optimal solution to (1). The polyhedral outer approximation $\mathcal{P}$ gives rise to a piece-wise linear concave overestimating function of f over X: $\bar{f}(x) := \max_{(x,t) \in \mathcal{P}} t$, such that $\max_{(x,t) \in \mathcal{P}} t = \max_{x \in X} \bar{f}(x)$. We then have the following observation.

Proposition 7. *Assume that f is not affine over X, and let $x^* \in \mathrm{relint}(X)$. Then $\bar{f}(x^*) > \mathsf{F}(x^*)$, i.e., $(x^*, \bar{f}(x^*)) \in \mathrm{int}(EE_f)$.*

Proof. As $\bar{f}$ is concave overestimator of f over X and F is convex underestimator of f over X, $\bar{f} \geq \mathsf{F}$ over X. Suppose, to aim at a contradiction, that $\bar{f}(x^*) = \mathsf{F}(x^*)$. Define a concave function $g := \bar{f} - \mathsf{F}$, then for all $x \in X$, $g(x) \geq 0$, and $g(x^*) = 0$. By its concavity, there exists an affine overestimating function a of g, such that $g(x^*) = a(x^*) = 0$, and for all $x \in X$, $0 \leq g(x) \leq a(x)$. As $x^* \in \mathrm{relint}(X)$, the affinity of a implies that $a = g = 0$ over X, i.e., $\bar{f} = \mathsf{F}$ over X. So f is concave and convex over X and thus affine over X, which is a contradiction. □

The measure of the relative boundary $\mathrm{relbd}(X)$ is zero, so we can assume that a mild *relative interior condition* that $\tilde{x} \in \mathrm{relint}(X)$ holds with probability one. Then the relaxation point $\tilde{z}$ is in the relative interior of the extended envelope epigraph with probability one.

11

5 Hypograph-free and superlevel-free sets for SS functions

This section considers hypograph and superlevel sets for an SS function $f := f_1 - f_2$, where f_1 and f_2 are two submodular functions. This generalizes our previous results for the hypograph set of the submodular function, and thus one can generate intersection cuts for a larger family of discrete nonconvex sets.

More specifically, we consider the following nonconvex set

$$S := \{(x,t) \in \mathcal{B} \times \mathbb{R} : f(x) \geq \ell t\}, \tag{8}$$

with $\ell \in \{0,1\}$. Given a relaxation point $(\tilde{x}, \tilde{t}) \notin S$, we want to find cutting planes separating this point from S.

Let $\mathsf{F}_1 := \max_{s \in EPM_{f_1}} sx$ and $\mathsf{F}_2 := \max_{s \in EPM_{f_2}} sx$ be extended envelopes of f_1, f_2, respectively. As F_1 (resp. F_2) is a convex extension of f_1 (resp. f_2), we have that $S = \{(x,t) \in \mathcal{B} \times \mathbb{R} : \mathsf{F}_1(x) - \mathsf{F}_2(x) \geq \ell t\}$. By relaxing $\mathcal{B}$ to $\mathbb{R}^n$, a (nonconvex) continuous outer approximation of S is

$$\bar{S} := \{(x,t) \in \mathbb{R}^n \times \mathbb{R} : \mathsf{F}_1(x) - \mathsf{F}_2(x) \geq \ell t\}. \tag{9}$$

Moreover, for all $x \in \mathcal{B}$, $(x,t) \in \bar{S}$ if and only if $(x,t) \in S$.

Special cases. When $\ell = 1$, S is the hypograph of the SS function f; when $\ell = 0$, S is the 0-superlevel set of the SS function f. Setting $f_2 = 0$ and $\ell = 1$, the set S becomes the hypograph $\{(x,t) \in \mathcal{B} \times \mathbb{R} : f_1(x) \geq t\}$, which is studied in the previous section. Setting $f_1 = 0$, the relaxed set $\bar{S}$ becomes $\{(x,t) \in \mathcal{B} \times \mathbb{R} : \mathsf{F}_2(x) \leq -\ell t\}$. If $(\tilde{x}, \tilde{t}) \notin \bar{S}$, since $\mathsf{F}_2(x) \geq \gamma^* x$ and $\mathsf{F}_2(\tilde{x}) = \gamma^* \tilde{x}$ for any $\gamma^* \in \partial \mathsf{F}_2(\tilde{x})$, then the simple outer approximation cut $\gamma^* x \leq -\ell t$ is a valid inequality for $\bar{S}$ (hence for S).

In general, we should separate intersection cuts specifically for SS functions. Let $\gamma^* \in \partial \mathsf{F}_2(\tilde{x})$ be a solution to (5) associated with f_2, and we define the set

$$\mathcal{C}_{\tilde{x}} := \{(x,t) \in \mathbb{R}^n \times \mathbb{R} : \mathsf{F}_1(x) - \gamma^* x \leq \ell t\}. \tag{10}$$

The following proposition gives S-free sets.

Proposition 8. *The set $\mathcal{C}_{\tilde{x}}$ is an S-free set. Moreover, if $(\tilde{x}, \tilde{t}) \notin \bar{S}$, then $\mathcal{C}_{\tilde{x}}$ does not contain $\tilde{x}$ in its interior.*

Proof. We first prove that $\mathcal{C}_{\tilde{x}}$ is $\bar{S}$-free. By definition, $\gamma^* x \leq \mathsf{F}_2(x)$, which implies that $\mathsf{F}_1(x) - \gamma^* x \geq \mathsf{F}_1(x) - \mathsf{F}_2(x)$. Therefore, for $(x,t) \in \mathrm{int}(\mathcal{C}_{\tilde{x}})$, we have that $\ell t > \mathsf{F}_1(x) - \gamma^* x \geq \mathsf{F}_1(x) - \mathsf{F}_2(x)$, which implies that $(x,t) \notin \bar{S}$. Hence, $\mathrm{int}(\mathcal{C}_{\tilde{x}}) \cap \bar{S} = \varnothing$. Additionally, $\mathcal{C}_{\tilde{x}}$ is convex. These two facts imply that $\mathcal{C}_{\tilde{x}}$ is $\bar{S}$-free. Since $S \subseteq \bar{S}$, $\mathcal{C}_{\tilde{x}}$ is also an S-free set. Next, assume that $(\tilde{x}, \tilde{t}) \notin \bar{S}$, then $\ell \tilde{t} > \mathsf{F}_1(\tilde{x}) - \mathsf{F}_2(\tilde{x}) \leq \mathsf{F}_1(\tilde{x}) - \gamma^* \tilde{x}$, so $(\tilde{x}, \tilde{t}) \in \mathrm{int}(\mathcal{C}_{\tilde{x}})$. $\square$

In [57, 67, 77], the authors study the sub/superlevel sets of some DC functions. Their construction of S-free sets relies on a common reverse-linearization technique: reverse the set S by changing the sign of its defining inequality, and linearize one convex function.

In our case, f is an SS function, so we first need to extend the submodular and supermodular components of f. After the extension, we obtain a DC function. Then, we can apply the reverse-linearization technique to its continuous extension.

6 Two applications

In this section, we discuss applications of intersection cuts to Boolean multilinear programming and D-optimal design. We exploit submodular structures in these two problems.

6.1 Boolean multilinear constraints

We consider the construction of $\mathcal{S}$-free sets for Boolean multilinear constraints. Since $x \in \{0,1\} \Leftrightarrow x^2 = x$, a polynomial function defined on binary variables is equivalent to a multilinear function on binary variables. A Boolean multilinear function is sometimes called a pseudo Boolean function.

A similar case is the construction of $\mathcal{S}$-free sets for continuous quadratic constraints [57]. We call this construction the "continuous approach". It applies eigenvalue decomposition to factor the symmetric matrices representing quadratic terms in a quadratic constraint. After factoring, the reformulated constraint contains a DC function. This reformulation is amenable to the reverse-linearization technique, by which one obtains the so-called continuous-quadratic-free sets [57]. Multilinear terms, however, are represented by tensors. It is doubtful whether this construction can be extended so as to produce DC functions from tensors.

Here we consider an alternative discrete approach. It exploits the submodularity and the super-modularity of Boolean multilinear functions. In [17, 60], a class of Boolean multilinear functions is shown to be supermodular. We give a submodular-supermodular decomposition for general Boolean multilinear functions in the following.

Proposition 9. *Given a Boolean multilinear function $f : \mathcal{B} \to \mathbb{R}$ defined as $f(x) := \sum_{k \in [K]} a_k \prod_{j \in A_k} x_j$ with K multilinear terms, where $A_k \subseteq [n]$. Let $f = f_1 - f_2$ where $f_1(x) := \sum_{\substack{k \in [K] \\ a_k < 0}} a_k \prod_{j \in A_k} x_j$ and*

$f_2(x) := -\sum_{\substack{k \in [K] \\ a_k > 0}} a_k \prod_{j \in A_k} x_j$. *Then f_1, f_2 are submodular over $\mathcal{B}$.*

Proof. Given a Cartesian product set $D := \prod_{j \in [n]} D_j$ ($D_j \subseteq \mathbb{R}$), a function $g : D \to \mathbb{R}$ is a generalized supermodular function over D, if for every $x, y \in D$, $g(x) + g(y) \leq g(x \vee y) + g(x \wedge y)$. Each multilinear term function $\prod_{j \in A_k} x_j$ is a Cobb-Douglas function [71], which is a generalized supermodular function over $\mathbb{R}^n_+$. It is known [71] that, if restricting the domain (e.g. $\mathbb{R}^n_+$) to its subdomain (e.g. $\mathcal{B}$) still yields a Cartesian product set, then the supermodularity is preserved. Moreover, a negative combination of supermodular functions is a submodular function. Therefore, f_1, f_2 are submodular functions over $\mathcal{B}$. □

Since every Boolean multilinear function is an SS function, we can construct $\mathcal{S}$-free sets for the corresponding superlevel set or hypograph set.

Corollary 5. *Given a multilinear function $f : \mathcal{B} \to \mathbb{R} : x \to f(x) := \sum_{k \in [K]} a_i \prod_{j \in A_k} x_j$ ($A_k \subseteq [n]$) as in Prop. 9, assume that $f = f_1 - f_2$ where $f_1(x) := \sum_{\substack{k \in [K] \\ a_k < 0}} a_k \prod_{j \in A_k} x_j$ and $f_2(x) := \sum_{\substack{k \in [K] \\ a_k > 0}} -a_k \prod_{j \in A_k} x_j$. Let $\mathcal{S}$, $\overline{\mathcal{S}}$, and $\mathcal{C}_{\tilde{x}}$ be as (8), (9), (10), respectively. Then, the set $\mathcal{C}_{\tilde{x}}$ is an $\mathcal{S}$-free set. Moreover, if $\tilde{x} \notin \overline{\mathcal{S}}$, then $\mathcal{C}_{\tilde{x}}$ does not contain $\tilde{x}$ in its interior.*

Proof. By Prop. 9, we know that both f_1 and f_2 are submodular. Hence, the result follows by applying Prop. 8. □

Importing the notation in Prop. 9, a BMP problem has the following form:

$$\max \quad t \tag{11a}$$

$$\sum_{k \in \mathcal{K}_0} a_{ik} \prod_{j \in A_k} x_j \geq t \tag{11b}$$

$$\forall i \in [m] \quad \sum_{k \in \mathcal{K}_i} a_{ik} \prod_{j \in A_k} x_j \geq 0 \tag{11c}$$

$$\forall j \in [n] \qquad\qquad x_j \in \{0,1\}, \qquad\qquad (11d)$$

where m is the number of constraints, K is the number of distinct multilinear terms in the BMP, $\mathcal{K}_i \subseteq [K]$ is the index set of multilinear terms in the i-th constraint (0 for objective). Unconstrained BMP has several synonyms: PSEUDO BOOLEAN MAXIMIZATION or MULTILINEAR UNCONSTRAINED BINARY OPTIMIZATION (MUBO).

To construct $\mathcal{S}$-free sets for Boolean multilinear constraints in the BMP, we need to write them as the standard form (8). For all $i \in [m]$ or $i = 0$, let

$$f_i(x) := \sum_{k \in \mathcal{K}_i} a_{ik} \prod_{j \in A_k} x_j,$$

and write

$$f_i(x) = f_{i1}(x) - f_{i2}(x),$$

where $f_{i1} := \sum_{k \in \mathcal{K}_i : a_{ik} < 0} a_{ik} \prod_{j \in A_k} x_j$ and $f_{i2} := -\sum_{k \in \mathcal{K}_i : a_{ik} > 0} a_{ik} \prod_{j \in A_k} x_j$ are two submodular functions.

The objective and constraints of (11) can be represented as

$$f_{i1}(x) - f_{i2}(x) \geq \ell_i t$$

(for all $i \in [m]$, $\ell_i = 0$, and $\ell_0 = 1$), which, by Cor. 5, is in the standard form.

Separating intersection cuts requires LP relaxations or corner polyhedra. One can first lift multilinear terms to obtain an extended formulation:

$$\max \qquad\qquad t \qquad\qquad\qquad (12a)$$

$$\sum_{k \in \mathcal{K}_0} a_{0k} y_k \geq t \qquad\qquad (12b)$$

$$\forall i \in [m] \quad \sum_{k \in \mathcal{K}_i} a_{ik} y_k \geq 0 \qquad\qquad (12c)$$

$$k \in [K] \qquad\qquad y_k = \prod_{j \in A_k} x_j \qquad\qquad (12d)$$

$$\forall j \in [n] \qquad\qquad x_j \in \{0,1\} \qquad\qquad (12e)$$

The standard Boolean linearization technique [29] can reformulate a multilinear term $\prod_{j \in A_k} x_j$ by its underestimators and overestimators:

$$\forall j \in A_k \quad y_k \leq x_j \qquad\qquad (13a)$$

$$y_k \geq |A_k| + 1 - \sum_{j \in A_k} x_j \qquad\qquad (13b)$$

$$y_k \in [0,1] \qquad\qquad (13c)$$

$$\forall j \in A_k \quad x_j \in \{0,1\}, \qquad\qquad (13d)$$

where $|A_k|$ is the cardinality of A_k. Then, by linearizing each nonlinear constraint (12d) as linear constraints in (13), one obtains a MILP reformulation of (12).

To construct LP relaxations and corner polyhedra, one can simply drop the integrality constraints $x_j \in \{0,1\}$. The direct LP relaxation for the MILP reformulation is also an LP relaxation for the BMP (12). This gives us a corner polyhedron in the extended space (x, y, t). The $\mathcal{S}$-free

set lives in a projected space (i.e., (x, t)-space). By extracting (x, t) entries of rays of the corner polyhedron, we project the corner polyhedron into the (x, t)-space.

Given a corner polyhedron, it is straightforward to construct intersection cuts for the BMP: we separate intersection cuts constructed from the $\mathcal{S}$-free sets given by Prop. 8.

We note that Boolean quadratic constraints belong to Boolean multilinear constraints, and continuous quadratic constraints relax Boolean quadratic constraints. Both the continuous and discrete approaches can construct valid $\mathcal{S}$-free sets for Boolean quadratic constraints. We remark that maximal continuous-quadratic-free sets are no longer maximally Boolean-quadratic-free. It is easy to see that the discrete approach preserves the term-wise sparsity patterns of the SS functions and requires no factorizations. Therefore, the discrete approach is computationally amenable to ill-conditioned or sparse coefficient matrices.

6.2   D-optimal design

In statistical estimation, optimal designs are a class of experimental designs that are optimal with respect to some statistical criterion. We derive an extended convex MINLP formulation for the BAYESIAN D-OPTIMAL DESIGN problem. In this formulation, the problem is a cardinality-constrained submodular maximization problem.

Let $\mathbb{S}^m$ denote the set of m-by-m symmetric matrices, and let $\mathbb{S}^m_+$ (resp. $\mathbb{S}^m_{++}$) denote the set of m-by-m positive semi-definite (resp. positive definite) matrices. Given a set of full row-rank matrices $\{M_j \in \mathbb{R}^{m \times r_k}\}_{j \in [n]}$, an optimal design problem usually has the following form:

$$\max \quad \Phi \left(\sum_{j \in [n]} M_j M_j^\top x_j \right) \tag{14a}$$

$$\sum_{j \in [n]} x_j = k \tag{14b}$$

$$\forall j \in [n] \qquad x_j \in \{0, 1\}, \tag{14c}$$

where k is the size of the design and $\Phi : \mathbb{S}^m \to \mathbb{R}$ is the design criterion. The matrix $M(x) := \sum_{j \in [n]} M_j M_j^\top x_j$ is called the *information matrix*. For the D-optimal criterion [18, 64], Ψ is the log determinant function ldet.

People usually study BAYESIAN D-OPTIMAL DESIGN, where a statistical prior on the parameters $\{M_i\}_{i \in [n]}$ adds a regularization term ϵI into the information matrix $M(x)$. This additional term is also due to the well-posedness: when $x = 0$, $\mathrm{ldet}(\epsilon I)$ is well defined. Then, the submodular maximization version of the BAYESIAN D-OPTIMAL DESIGN problem has the following formulation:

$$\max \quad \mathrm{ldet} \left(\epsilon I + \sum_{j \in [n]} M_j M_j^\top x_j \right) \tag{15a}$$

$$\sum_{j \in [n]} x_j = k \tag{15b}$$

$$\forall j \in [n] \qquad x_j \in \{0, 1\}, \tag{15c}$$

The log determinant function is concave and has a semi-definite programming (SDP) and geometric programming representation [5]. The scalability of the mixed-integer log determinant formulation above is limited by the current state of SDP solvers. Based on the second order cone

15

representation of the determinant function $\det(M(x))$ [64], we give an extended formulation for (15):

$$\max \quad t \tag{16a}$$

$$t \leq \sum_{i \in [m]} \log(J_{ii}) \tag{16b}$$

$$\sum_{j \in [n] \cup \{0\}} M_j Z_j = J \tag{16c}$$

$$J \text{ is lower triangular} \tag{16d}$$

$$j \in [n] \cup \{0\}\, j \in [m] \quad \|Z_j e_i\|^2 \leq u_{ji} x_j \tag{16e}$$

$$i \in [m] \quad \sum_{j \in [n] \cup \{0\}} u_{ji} \leq J_{ii} \tag{16f}$$

$$\sum_{j \in [n]} x_j = k \tag{16g}$$

$$x \in \{1\} \times \mathcal{B} \tag{16h}$$

$$J \in \mathbb{R}^{m \times m} \tag{16i}$$

$$j \in [n] \quad Z_j \in \mathbb{R}^{r_j \times m} \tag{16j}$$

$$j \in [n] \cup \{0\}\, i \in [m] \quad u_{ji} \in \mathbb{R}_+^{r_j \times m}, \tag{16k}$$

where $M_0 = \epsilon^{1/2} I$ is an auxiliary matrix. One can represent this formulation by low-dimensional convex cones [5], e.g., (rotated) second-order cones, and exponential cones. Therefore, this extended formulation is amenable to computation.

Proposition 10. (16) *is equivalent to* (15)*, and the objective of* (16) *is submodular w.r.t.* x.

Proof. One can modify the original D-optimal design problem by adding a slack variable $x_0 = 1$. Applying the logarithmic transformation to results in [64], (16) is equivalent to (15). It follows from [63, 68] that (16) is submodular w.r.t. x. $\qquad\square$

A global optimization solver like `SCIP` can linearize the constraints in the extended formulation (16), and thus produces an LP relaxation in the extended space. We can obtain a corner polyhedron as the approach dealing with the BMP. Then, we can construct intersection cuts from hypograph-free sets.

7   Separation problem

In this section, we consider the separation problem to generate an intersection cut using an $\mathcal{S}$-free set. Summarizing the previous sections, the $\mathcal{S}$-free set is in the form of

$$\mathcal{C} := \{(x, t) \in \mathbb{R}^n \times \mathbb{R} : \mathsf{G}(x) \leq \ell t\},$$

where $\mathsf{G}(x) = \max_{s \in \text{ext}(EPM_g)} sx$ is the extended envelope of some submodular function g over $\mathcal{B}$ and $\ell \in \{0, 1\}$. We remark that the extended envelope epigraph EE_f in (4) is a special case with $\ell = 1$ and $g = f$; the set $\mathcal{C}_{\tilde{x}}$ in (10) is also a special case that $g(x) = f_1(x) - \gamma^* x$.

Assume that $z^* := (\tilde{x}, \tilde{t})$ is a vertex of a corner polyhedron $\mathcal{R}$, and $z^* \in \text{int}(\mathcal{C})$. Recalling the cut coefficient formula in Sect. 2, the separation problem is reduced to calculate the step length along each ray r^j:

$$\eta_j^* = \sup_{\eta_j \geq 0}\{\eta_j : z^* + \eta_j r^j \in \mathcal{C}\}. \tag{17}$$

This line search problem asks for the step length to the border of $\mathcal{C}$ along the ray r^j from the interior point z^*. We denote by r_x^j, r_t^j the projection of r^j on x- and t- spaces. Looking at the function defining $\mathcal{C}$, the intersection step length η_j^* is the zero point of the following function:

$$\zeta^j : \mathbb{R}_+ \to \mathbb{R} : \eta_j \to \zeta^j(\eta_j) := \ell(\tilde{t} + r_t^j \eta_j) - \mathsf{G}(\tilde{x} + r_x^j \eta_j).$$

This function enjoys the following properties.

Proposition 11. ζ^j *is a concave piece-wise linear function over* $[0, +\infty]$ *with* $\zeta^j(0) > 0$. *If* $\eta_j^* < \infty$ *and there exists an* $\eta_j' > 0$ *with* $\zeta^j(\eta_j') = 0$, *then* $\eta_j' = \eta_j^*$, *i.e., the solution* η_j^* *must be unique. For all* $s^* \in \text{argmax}_{s \in \text{ext}(EPM_g)} s(\tilde{x} + \eta_j r_x^j)$, $\ell r_t^j - s^* r_x^j$ *is a subgradient in* $\partial \zeta^j(\eta_j)$. *For* $\eta_j > \eta_j^*$, $\partial \zeta^j(\eta_j) \leq \partial \zeta^j(\eta_j^*)$.

Proof. Since the extended envelope G is the maximum of linear functions, it is convex and piece-wise linear, so ζ^j is concave and piece-wise linear. Since $\zeta^j(0) = \ell\tilde{t} - \mathsf{G}(\tilde{x})$, it follows from the assumption $z^* \in \text{int}(\mathcal{C})$ that $\ell\tilde{t} > \mathsf{G}(\tilde{x})$ and thus $\zeta^j(0) > 0$. Since $\mathcal{C}$ is closed and convex, $\eta_j' = \eta_j^*$ if and only if $z^* + \eta_j' r^j \in \text{bd}(\mathcal{C})$. That is $\mathsf{G}(r_x^j \eta_j + \tilde{x}) = \mathsf{G}(\tilde{x}) + r_t^j \eta_j'$, i.e., $\zeta^j(\eta_j') = 0$. Since $s^* \in \partial \mathsf{G}(\tilde{x} + r_x^j \eta_j)$, by the chain rule, $\ell r_t^j - s^* r_x^j$ is a subgradient of ζ^j. By the concavity of ζ^j, its subgradients are non-increasing. $\square$

By Prop. 11, the line search problem (17) is reduced into solving a univariate nonlinear equation:

$$\zeta^j(\eta^j) = 0. \tag{18}$$

For each ray r^j, solving (18) gives the unique zero point of the univariate function ζ^j, or certificates that no such point exists.

To solve the univariate nonlinear equation (18), it is natural to deploy a Newton-like algorithm. Therefore, we need the value and (sub)gradient information of ζ^j. Moreover, the computation of ζ^j can be reduced to the computation of G. A sorting algorithm can compute the value and subgradients of G (see Prop. 3). This means that one can compute ζ^j in a strongly polynomial time.

Previous works [22, 77] use the bisection algorithm, which guarantees finding the zero point within a given tolerance. Our implementation is similar to the *discrete Newton algorithm* in [41], but is combined with the bisection algorithm, so we call our implementation the *hybrid discrete Newton algorithm*. The bisection algorithm helps find a starting point for the Newton algorithm. Thanks to the piece-wise linearity of the univariate function ζ^j, our algorithm can find an exact zero point in a finite time.

Proposition 12. *The hybrid discrete Newton algorithm terminates in a finite number of steps and finds the zero point* η_j^*.

Proof. For all $\eta \in \mathbb{R}_+$, we assume that Algorithm 1 chooses and computes a unique subgradient β at η_j, we denote it $\nabla \zeta^j(\eta_j)$, and call it algorithmic gradient. The concavity of ζ^j implies that its algorithmic gradient is monotone-decreasing w.r.t. η_j. There is a threshold $\eta_j' \geq 0$ such that, for

---

Algorithm 1: Hybrid discrete Newton algorithm

---

1 Input: The univariate function ζ^j, (scalar) starting point $\Delta > 0$ (default: 0.2), a numeric η_∞ representing $+\infty$, and the maximum number I of search steps (default: 500);

2 Output: $\eta_j > 0$ such that $\zeta^j(\eta_j) = 0$;

3 Let $i = 0$, $\eta_j = \Delta$;

4 if $\zeta^j(\eta_\infty) > 0$ **then**

5 $\quad$ $\eta_j = \eta_\infty$; $\hfill \triangleright$ safeguard

6 else

7 $\quad$ **while** $i < I$ **do**

8 $\quad\quad$ Let $s^* \in \operatorname{argmax}_{s \in \operatorname{ext}(EPM_g)} s(\tilde{x} + r_x^j \eta_j)$;

9 $\quad\quad$ Compute a subgradient $\beta = r_t^j - s^* r_x^j$;

10 $\quad\quad$ **if** $\zeta^j(\eta_j) = 0$ **then**

11 $\quad\quad\quad$ **break**;

12 $\quad\quad$ **else if** $\beta < 0$ **then**

13 $\quad\quad\quad$ $\eta_j = \eta_j - \frac{\zeta^j(\eta_j)}{\beta}$; $\hfill \triangleright$ Newton step

14 $\quad\quad$ **else**

15 $\quad\quad\quad$ $\eta_j = 2\eta_j$; $\hfill \triangleright$ bisection step

16 $\quad\quad$ $i = i + 1$;

---

all $\eta_j \in [0, \eta_j')$, the algorithmic gradient $\nabla \zeta^j(\eta_j) > 0$; for all $\eta_j \in [\eta_j', +\infty]$ (called the Newton step region), the algorithmic gradient $\nabla \zeta^j(\eta_j) \leq 0$.

After a finite number of bisection steps (at most $\lceil \log(\eta_j'/\Delta) \rceil$), the algorithm enters the Newton step region $[\eta_j', +\infty]$, where the algorithmic gradient is always negative. Then, we prove that the algorithmic gradient $\nabla \zeta^j(\eta_j)$ at step i is different from that at step $i-1$, and the algorithm stays in the Newton step region. Since ζ^j is piece-wise linear (the number of its distinct algorithmic gradients is finite), the algorithm must terminate in a finite number of steps.

If at step $i-1$, $\zeta^j(\eta_j - \frac{\zeta^j(\eta_j)}{\nabla \zeta^j(\eta_j)}) = 0$, then the algorithm terminates at this step and finds the zero point. If at step $i-1$, $\zeta^j(\eta_j - \frac{\zeta^j(\eta_j)}{\nabla \zeta^j(\eta_j)}) < 0$, then we prove that $\nabla \zeta^j(\eta_j - \frac{\zeta^j(\eta_j)}{\nabla \zeta^j(\eta_j)}) \neq \nabla \zeta^j(\eta_j)$ and $\nabla \zeta^j(\eta_j - \frac{\zeta^j(\eta_j)}{\nabla \zeta^j(\eta_j)}) \leq 0$.

First, assume, to aim at a contradiction, that $\nabla \zeta^j(\eta_j - \frac{\zeta^j(\eta_j)}{\nabla \zeta^j(\eta_j)}) = \nabla \zeta^j(\eta_j)$. Knowing that the algorithmic gradient is monotone-decreasing, the piece-wise linearity of ζ^j implies that this algorithmic gradient is constant in the range $[\eta_j - \frac{\zeta^j(\eta_j)}{\nabla \zeta^j(\eta_j)}, \eta_j]$. It follows that for all $\delta \in [0, \frac{\zeta^j(\eta_j)}{\nabla \zeta^j(\eta_j)}]$, $\zeta^j(\eta_j - \delta) = \zeta^j(\eta_j) - \delta \nabla \zeta^j(\eta_j)$. Hence, $\zeta^j(\eta_j - \frac{\zeta^j(\eta_j)}{\nabla \zeta^j(\eta_j)}) = 0$, which leads to a contradiction.

Second, we show that $\nabla \zeta^j(\eta_j - \frac{\zeta^j(\eta_j)}{\nabla \zeta^j(\eta_j)}) \leq 0$. When $\frac{\zeta^j(\eta_j)}{\nabla \zeta^j(\eta_j)} \leq 0$, by the monoticity of $\nabla \zeta^j$, $\nabla \zeta^j(\eta_j - \frac{\zeta^j(\eta_j)}{\nabla \zeta^j(\eta_j)}) \leq \nabla \zeta^j(\eta_j) < 0$. When $\frac{\zeta^j(\eta_j)}{\nabla \zeta^j(\eta_j)} > 0$, as by assumption that $\nabla \zeta^j(\eta_j) < 0$, $\zeta^j(\eta_j)$ must be negative. Then, by the concavity of ζ^j, $\zeta^j(\eta_j - \frac{\zeta^j(\eta_j)}{\nabla \zeta^j(\eta_j)}) \leq \zeta^j(\eta_j) - \nabla \zeta^j(\eta_j) \frac{\zeta^j(\eta_j)}{\nabla \zeta^j(\eta_j)} = 0$. This implies that $\nabla \zeta^j(\eta_j - \frac{\zeta^j(\eta_j)}{\nabla \zeta^j(\eta_j)}) \leq 0$.

$\hfill \square$

From Prop. 12, the hybrid discrete Newton algorithm first executes bisection steps with in-

creasing η_j and $\zeta^j(\eta_j)$. Then it enters into the Newton step region. After a single Newton step, $\zeta^j(\eta_j)$ becomes negative, and then monotonically increases to zero in a finite number of steps. The discrete Newton algorithm in [41] is applied to the line search problem for submodular polyhedra, which are polars of extended polymatroids. The algorithm runs in a strongly polynomial time. In our case, $\mathcal{C}$ includes the extended polymatroid and is unbounded. The corresponding line search problem may have no solutions, and this is a usual case in intersection cut computation [21]. Therefore, Algorithm 1 needs a safeguard step, where we evaluate ζ^j at an user-defined infinity. One may also prove that Algorithm 1 runs in a strongly polynomial time, but a careful analysis for the unbounded case is needed. Owing to the limitation of pages, we do not expand this topic here.

8   Computational results

In this section, we conduct computational experiments to test the proposed cuts.

Setup and performance metrics. The experiments are conducted on a server with Intel Xeon W-2245 CPU @ 3.90GHz and 126GB main memory. We use `SCIP 8.0` [14] as a MINLP framework to solve the natural formulations of test problems. `SCIP` is equipped with `CPLEX 22.1` as an LP solver, and `IPOPT 3.14` as an NLP solver.

By Thm. 2, the simple lifted split $H_j := \{x \in \mathbb{R}^n : 0 \leq x_j \leq 1\} \times \mathbb{R}$ is a maximal $\mathrm{hypo}_{\mathcal{B}}(f)$-free set, where the splitting variable x_j is chosen as the most fractional entry of the relaxation solution. In the *standalone* (resp. the *embedded*) configuration, we deactivate (resp. activate) `SCIP`'s internal cut separators. Under each configuration, the *submodular cut* (resp. the *split cut*) setting adds intersection cuts derived from EE_f (resp. H_j), and the *default* setting does not add any intersection cuts.

We focus on the root node performance and measure the *closed root gap*. Let d_1 be the value of the first LP relaxation (without cuts added), let d_2 be the dual bound after all the cuts are added, and let p be a reference primal bound. The closed root gap $(d_2 - d_1)/(p - d_1)$ is the closed gap improvement of d_2 with respect to d_1. We also record the number of added cuts, the relative improvement to the default setting, and the total running time. For each configuration and setting, we compute these statistics' shifted geometric mean (shift value: 1) within a test problem benchmark.

Experiment 1: MAX CUT. Consider an undirected graph $G = (V, E, w)$, where V is the set of nodes, E is the set of edges, and w is a weight function over E. For a subset S of V, its associated cut capacity is the sum of the weights of edges with one adjacent node in S and the other in $V \smallsetminus S$. The MAX CUT problem aims at finding a subset $S \subseteq V$ with maximum cut capacity. Using a binary variable vector $x \in \mathcal{B}$ indicating whether vertices belong to S, then the problem can be formulated as the following quadratic unconstrained binary optimization (QUBO) problem:

$$\max_{x \in \mathcal{B}} \sum_{\{i,j\} \in E} w_{ij}((1 - x_i)x_j + x_i(1 - x_j)).$$

When w is nonnegative, the cut capacity function (the objective function) is submodular. Our benchmark contains 30 "g05" and 30 "pw" instances with nonnegative weights from Biq Mac [76]. The reference primal bounds are also from Biq Mac. The number of vertices is up to 100, and the number of edges is up to 4455. We encode the hypograph reformulation (1) of the QUBO. `SCIP` will automatically reformulate the problem into a MILP via the reformulation-linearization technique (RLT) [1]. This MILP formulation is a special case of the extended formulation (16) of a degree-2 BMP with $m = 0$.

In Table 1, we report the computational results, where "closed" denotes the average closed root gap, and "relative" denotes its relative value to the default setting. For the standalone (resp. the embedded) configuration, the relative improvement of submodular cuts is 340% (resp. 22%) compared to 193% (resp. 21%) of split cuts. In the standalone configuration, we can compare the "clean" strengths of intersection cuts derived from different hypograph-free sets. Although split cuts are derived from maximal hypograph-free sets and submodular cuts are derived from non-maximal ones, the performance of split cuts is worse.

We observe that fewer split cuts are generated than submodular cuts. This means that the efficiency of some split cuts does not satisfy `SCIP`'s internal criteria, so `SCIP` abandons more split cuts than submodular cuts. As two types of cuts are derived using the same principle but from different hypograph-free sets, the distances between the relaxation points to the boundary of hypograph-free sets determine the cut efficiency. This observation suggests that relaxation points are further to the boundary of the extended envelope epigraph than to the splits. Under the embedded configuration, the difference in relative improvements between the two types of cuts is 1%, so they perform almost equally. However, the separation time of split cuts is shorter than that of submodular cuts. This is because separating submodular cuts requires solving nonlinear equations, while the split cuts can be computed in a closed form.

Configuration	Default		Submodular cut				Split cut			
	closed	time	closed	relative	time	cuts	closed	relative	time	cuts
standalone	0.04	5.13	0.16	4.40	85.40	207.59	0.12	2.93	17.92	92.53
embedded	0.22	12.62	0.27	1.22	104.02	70.68	0.27	1.21	34.62	45.15

Table 1: Summary of MAX CUT results

Experiment 2: PSEUDO BOOLEAN MAXIMIZATION. As mentioned, PSEUDO BOOLEAN MAXIMIZATION is a MUBO problem, a generalization of QUBO. We can use techniques from Sect. 5 to generate intersection cuts.

Our benchmark contains 44 highly dense "autocorr_bern" MUBO instances from MINLPLib [19, 75]. These instances arise in theoretical physics, and the problem is to minimize a degree-four polynomial energy function. The problem is a degree-4 BMP with $m = 0$. `SCIP` constructs the extended formulation (16). The benchmark contains instances with up to 60 binary variables and 3540 Boolean multilinear terms. We use the best-known primal bound from MINLPLib as the reference primal bound.

In Table 2, we report the computational results. For the standalone (resp. the embedded) configuration, the relative improvement of submodular cuts is 381% (resp. 13%) compared to 131% (resp. 1%) of split cuts. In both configurations, the submodular cuts are better than the split cuts in terms of the closed root gap. Moreover, under the embedded configuration, the difference in the relative improvements between the two types of cuts is around 10%. This is larger than 1% of MAX CUT benchmark under the same configuration. This divergence between degree-2 and degree-4 MUBO suggests that the submodular cuts are suitable for high-degree Boolean multilinear constraints.

We recall that to solve the nonlinear equations, the hybrid discrete Newton algorithm needs oracle access to the value of the Boolean multilinear function. For some instances, a Boolean multilinear function may consist of thousands of multilinear terms. After a code timing analysis, we find that the separation of submodular cuts spends the most time computing the function value. Therefore, this is the main time performance bottleneck, which needs to be optimized in the future. An counterintuitive finding is that non-maximal $\mathcal{S}$-free sets may yield stronger cuts. This because

the geometrical relation between the $\mathcal{S}$-free sets and corner polyhedron matters.

Configuration	Default		Submodular cut				Split cut			
	closed	time	closed	relative	time	cuts	closed	relative	time	cuts
standalone	0.01	9.49	0.05	4.81	43.54	43.17	0.03	2.31	14.64	20.94
embedded	0.105	22.52	0.11	1.13	49.61	13.80	0.106	1.01	25.58	28.21

Table 2: Summary of PSEUDO BOOLEAN MAXIMIZATION

Experiment 3: BAYESIAN D-OPTIMAL DESIGN. As mentioned, the BAYESIAN D-OPTIMAL DESIGN problem has a submodular maximization form (15). In particular, we can encode it as an extended formulation (16) in `SCIP`. `SCIP` generates gradient cuts for this convex MINLP. Therefore, we can obtain LP relaxations and corner polyhedra.

Our benchmark consists of two classes of instances. We let parameters $M_j \in \mathbb{R}^{m \times 1}$ be single-column matrices. The first class of instances are block design problems [64], where M_j are sparse 0-1 matrices. The exact designs correspond to the graphs with a given number of edges and nodes that have a maximum number of spanning trees. Recall that n is the variable dimension, m is the matrix dimension, and k is the cardinality. We generate 15 block design instances with $(n, m, k) \in \{(45, 10, 9), (55, 11, 10), (66, 12, 11)\}$. The second class of instances are random Gaussian instances, where M_j are dense real matrices. The entries of matrices M_j are drawn from a Gaussian distribution with zero mean and $1/\sqrt{n}$ variance. We generate 30 random Gaussian instances with $(n, m) \in \{(50, 20), (50, 30), (60, 24), (60, 36), (70, 28), (70, 42)\}$ and $k \in \{m, m+1, m+2, m+3, m+4\}$. We set the regularization constant ϵ to $1e - 6$. We use the best primal bound from all settings as the reference primal bound. Since `SCIP`'s internal gradient cuts are important for linearizing convex nonlinear constraints, we keep the gradient cuts but disable all integer-oriented cuts (GMI cuts and mixed-integer rounding cuts etc.) in the standalone configuration.

In Table 3, we report the computational results. We divide the results of block design and Gaussian random instances, since the density of matrices are different. Looking at the default setting in different benchmarks, there is no difference between the standalone and embedded configurations in terms of the closed root gap. This means that integer-oriented cuts do not improve the root node LP relaxations. We see the same problem for intersection cuts, which do not close the root gap but increase the computing time. In particular, the number of separated cuts is around one. Thereby, many intersection cuts are too weak to add in the cut pool.

We recall that intersection cuts and many integer-oriented cuts are LP-based cuts, i.e., derived from an LP relaxation of the extended formulation (16). Therefore, their strengths depend on the LP relaxation. A flat corner polyhedron, which usually arises from an LP relaxation with many constraints, may yield weak intersection cuts. Based on types of MINLPs, there are two basic ways to construct initial LP relaxations. For nonconvex MINLPs, one way usually uses the factorable programming and term-wise envelopes [53]. Notable examples are Boolean multilinear constraints and continuous quadratic constraints [57]. The McCormick envelopes or Boolean linearization techniques are used to construct their LP relaxations, which have a finite number of constraints. For convex MINLPs, the other way linearizes nonlinear constraints, and the number of constraints in the LP relaxation can grow to infinite. This is because a convex nonlinear constraint is equivalent to an infinite number of linear constraints. Since `SCIP` may add many gradient cuts for approximating the convex MINLP (16), this yields flat corner polyhedrons and weak intersection cuts. In summary, the weakness of intersection cuts is due to the flatness of the corner polyhedron.

Benchmark	Configuration	Default		Submodular cut				Split cut			
		closed	time	closed	relative	time	cuts	closed	relative	time	cuts
Block design	standalone	0.59	20.46	0.59	1.0	18.71	1.84	0.59	1.0	11.62	1.77
	embedded	0.59	21.44	0.59	1.0	19.0	1.84	0.59	1.0	12.41	1.77
Gaussian	standalone	0.83	213.13	0.83	1.0	415.07	1.45	0.83	1.0	214.17	1.45
	embedded	0.83	214.77	0.83	1.0	426.33	1.45	0.83	1.0	214.14	1.45
All	standalone	0.75	98.47	0.75	1.0	149.54	1.57	0.75	1.0	82.6	1.55
	embedded	0.75	100.47	0.75	1.0	153.01	1.57	0.75	1.0	84.31	1.55

Table 3: Summary of Bayesian D-optimal design results

9   Conclusion

We construct hypograph-free sets for submodular functions. Our construction relies on a new continuous extension of submodular functions. We characterize maximal hypograph-free sets, generalize our results to sets involving submodular-supermodular functions. These yield intersection cuts for Boolean multilinear constraints. We exploit the submodular structure in an extended formulation of the D-optimal design problem. We propose a hybrid discrete Newton algorithm that can compute intersection cuts efficiently and exactly. The computational results show that intersection cuts derived from the submodularity are stronger than those derived from split cuts for max cut and pseudo Boolean maximization problems. For convex MINLPs, our computational results on the Bayesian D-optimal design problem suggest that corner polyhedra can be flat, which makes intersection cuts weak.

Statements and Declarations

Non conflicts of interest with the journal or the funding agencies.

References

[1] Warren P Adams and Hanif D Sherali. A tight linearization and an algorithm for zero-one quadratic programming problems. *Management Science*, 32(10):1274–1290, 1986.

[2] Shabbir Ahmed and Alper Atamtürk. Maximizing a class of submodular utility functions. *Mathematical programming*, 128(1):149–169, 2011.

[3] Kent Andersen, Quentin Louveaux, and Robert Weismantel. An analysis of mixed integer linear sets based on lattice point free convex sets. *Mathematics of Operations Research*, 35(1):233–256, feb 2010.

[4] Kent Andersen, Quentin Louveaux, Robert Weismantel, and Laurence A. Wolsey. Inequalities from two rows of a simplex tableau. In Matteo Fischetti and David P. Williamson, editors, *Integer Programming and Combinatorial Optimization*, pages 1–15, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[5] Mosek ApS. Mosek modeling cookbook, 2020.

[6] Alper Atamtürk and Andrés Gómez. Submodularity in conic quadratic mixed 0–1 optimization. *Operations Research*, 68(2):609–630, 2020.

[7] Alper Atamtürk and Andrés Gómez. Supermodularity and valid inequalities for quadratic optimization with indicators. *Mathematical Programming*, pages 1–44, 2022.

[8] Alper Atamtürk and Vishnu Narayanan. Submodular function minimization and polarity. *Mathematical Programming*, 2021.

[9] Egon Balas. Intersection cuts—a new type of cutting planes for integer programming. *Operations Research*, 19(1):19–39, 1971.

[10] Amitabh Basu, Michele Conforti, Gérard Cornuéjols, and Giacomo Zambelli. Maximal lattice-free convex sets in linear subspaces. *Mathematics of Operations Research*, 35(3):704–720, 2010.

[11] Amitabh Basu, Michele Conforti, Gérard Cornuéjols, and Giacomo Zambelli. Maximal lattice-free convex sets in linear subspaces. *Mathematics of Operations Research*, 35(3):704–720, 2010.

[12] Amitabh Basu, Santanu S. Dey, and Joseph Paat. Nonunique lifting of integer variables in minimal inequalities. *SIAM Journal on Discrete Mathematics*, 33(2):755–783, 2019.

[13] Pietro Belotti, Julio C Góez, Imre Pólik, Ted K Ralphs, and Tamás Terlaky. A conic representation of the convex hull of disjunctive sets and conic cuts for integer second order cone optimization. In *Numerical Analysis and Optimization*, pages 1–35. Springer, 2015.

[14] Ksenia Bestuzheva, Mathieu Besançon, Wei-Kun Chen, Antonia Chmiela, Tim Donkiewicz, Jasper van Doornmalen, Leon Eifler, Oliver Gaul, Gerald Gamrath, Ambros Gleixner, et al. The scip optimization suite 8.0. *arXiv preprint arXiv:2112.08872*, 2021.

[15] Ksenia Bestuzheva, Antonia Chmiela, Benjamin Müller, Felipe Serrano, Stefan Vigerske, and Fabian Wegscheider. Global optimization of mixed-integer nonlinear programs with scip 8. *arXiv preprint arXiv:2301.00587*, 2023.

[16] Daniel Bienstock, Chen Chen, and Gonzalo Munoz. Outer-product-free sets for polynomial optimization and oracle-based cuts. *Mathematical Programming*, 183(1):105–148, 2020.

[17] Alain Billionnet and Michel Minoux. Maximizing a supermodular pseudoboolean function: A polynomial algorithm for supermodular cubic functions. *Discrete Applied Mathematics*, 12(1):1–11, 1985.

[18] Mustapha Bouhtou, Stephane Gaubert, and Guillaume Sagnol. Submodularity and randomized rounding techniques for optimal experimental design. *Electronic Notes in Discrete Mathematics*, 36:679–686, 2010.

[19] Michael R Bussieck, Arne Stolbjerg Drud, and Alexander Meeraus. Minlplib—a collection of test models for mixed-integer nonlinear programming. *INFORMS Journal on Computing*, 15(1):114–119, 2003.

[20] Rui Chen, Sanjeeb Dash, and Oktay Günlük. Multilinear sets with two monomials and cardinality constraints. *Discrete Applied Mathematics*, 324:67–79, 2023.

[21] Antonia Chmiela, Gonzalo Muñoz, and Felipe Serrano. On the implementation and strengthening of intersection cuts for qcqps. *Mathematical Programming*, pages 1–38, 2022.

[22] Antonia Chmiela, Gonzalo Muñoz, and Felipe Serrano. Monoidal strengthening and unique lifting in miqcps. In *Integer Programming and Combinatorial Optimization: 24th International Conference, IPCO 2023*, 2023. accepted for publication.

[23] Chris Coey, Miles Lubin, and Juan Pablo Vielma. Outer approximation with conic certificates for mixed-integer convex problems. *Mathematical Programming Computation*, 12(2):249–293, 2020.

[24] Michele Conforti, Gérard Cornuéjols, and Giacomo Zambelli. *Integer programming*. Springer International Publishing, Cham, 2014.

[25] Michele Conforti, Gérard Cornuéjols, Aris Daniilidis, Claude Lemaréchal, and Jérôme Malick. Cut-Generating Functions and S-Free Sets. *Mathematics of Operations Research*, 40(2):276–391, 2015.

[26] Michele Conforti, Gérard Cornuéjols, and Giacomo Zambelli. Corner polyhedron and intersection cuts. *Surveys in Operations Research and Management Science*, 16(2):105–120, 2011.

[27] Stefano Coniglio, Fabio Furini, and Ivana Ljubić. Submodular maximization of concave utility functions composed with a set-union operator with applications to maximal covering location problems. *Mathematical Programming*, pages 1–48, 2022.

[28] Gérard Cornuéjols, Laurence Wolsey, and Sercan Yıldız. Sufficiency of cut-generating functions. *Mathematical Programming*, 152(1):643–651, 2015.

[29] Yves Crama. Concave extensions for nonlinear 0–1 maximization problems. *Mathematical Programming*, 61(1):53–60, 1993.

[30] Alberto Del Pia and Aida Khajavirad. A polyhedral study of binary polynomial programs. *Mathematics of Operations Research*, 42(2):389–410, 2017.

[31] Alberto Del Pia and Aida Khajavirad. The multilinear polytope for acyclic hypergraphs. *SIAM Journal on Optimization*, 28(2):1049–1076, 2018.

[32] Alberto Del Pia, Aida Khajavirad, and Nikolaos V Sahinidis. On the impact of running intersection inequalities for globally solving polynomial optimization problems. *Mathematical programming computation*, 12(2):165–191, 2020.

[33] Alberto Del Pia and Matthias Walter. Simple odd-cycle inequalities for binary polynomial optimization. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 181–194. Springer, 2022.

[34] Alberto Del Pia and Robert Weismantel. Relaxations of mixed integer sets from lattice-free polyhedra. *4OR*, 10(3):221–244, 2012.

[35] Santanu S. Dey and Laurence A. Wolsey. Lifting integer variables in minimal inequalities corresponding to lattice-free triangles. In Andrea Lodi, Alessandro Panconesi, and Giovanni Rinaldi, editors, *Integer Programming and Combinatorial Optimization*, pages 463–475, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[36] Jack Edmonds. Submodular functions, matroids, and certain polyhedra. In *Combinatorial Optimization—Eureka, You Shrink!*, pages 11–26. Springer, 2003.

[37] Matteo Fischetti, Ivana Ljubić, Michele Monaci, and Markus Sinnl. On the use of intersection cuts for bilevel optimization. *Mathematical Programming*, 172(1):77–103, 2018.

[38] Matteo Fischetti and Michele Monaci. A branch-and-cut algorithm for mixed-integer bilinear programming. *European Journal of Operational Research*, 282(2):506–514, 2020.

[39] Robert Fortet. Applications de l'algebre de boole en recherche opérationelle. *Revue Française de Recherche Opérationelle*, 4(14):17–26, 1960.

[40] Fred Glover. Convexity cuts and cut search. *Operations Research*, 21(1):123–134, 1973.

[41] Michel X. Goemans, Swati Gupta, and Patrick Jaillet. Discrete newton's algorithm for parametric submodular function minimization. In Friedrich Eisenbrand and Jochen Koenemann, editors, *Integer Programming and Combinatorial Optimization*, pages 212–227, Cham, 2017. Springer International Publishing.

[42] Ralph E Gomory. Some polyhedra related to combinatorial problems. *Linear algebra and its applications*, 2(4):451–558, 1969.

[43] Ralph E. Gomory. Outline of an algorithm for integer solutions to linear programs and an algorithm for the mixed integer problem. In Michael Jünger, Thomas M. Liebling, Denis Naddef, George L. Nemhauser, William R. Pulleyblank, Gerhard Reinelt, Giovanni Rinaldi, and Laurence A. Wolsey, editors, *50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-Art*, pages 77–103. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[44] Shaoning Han, Andrés Gómez, and Oleg A Prokopyev. Fractional 0–1 programming and submodularity. *Journal of Global Optimization*, pages 1–17, 2022.

[45] Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal. *Fundamentals of convex analysis*. Springer Science & Business Media, 2004.

[46] R. Horst and H. Tuy. *Global Optimization: Deterministic Approaches*. Springer, Berlin, 1990.

[47] Oleg Khamisov. On optimization properties of functions, with a concave minorant. *Journal of Global Optimization*, 14(1):79–101, 1999.

[48] Fatma Kılınç-Karzan, Simge Küçükyavuz, and Dabeen Lee. Joint chance-constrained programs and the intersection of mixing sets through a submodularity lens. *Mathematical Programming*, pages 1–44, 2021.

[49] Fatma Kılınç-Karzan and Sercan Yıldız. Two-term disjunctions on the second-order cone. *Mathematical Programming*, 154(1-2):463–491, 2015.

[50] Fatma Kılınç-Karzan. On minimal valid inequalities for mixed integer conic programs. *Mathematics of Operations Research*, 41(2):477–510, 2016.

[51] Leo Liberti. Spherical cuts for integer programming problems. *International Transactions in Operational Research*, 15(3):283–294, 2008.

[52] László Lovász. Submodular functions and convexity. In *Mathematical programming the state of the art*, pages 235–257. Springer, 1983.

[53] Garth P McCormick. Computability of global solutions to factorable nonconvex programs: Part i—convex underestimating problems. *Mathematical programming*, 10(1):147–175, 1976.

[54] Sina Modaresi, Mustafa R. Kilinç, and Juan Pablo Vielma. Split cuts and extended formulations for Mixed Integer Conic Quadratic Programming. *Operations Research Letters*, 43(1):10–15, 2015.

[55] Sina Modaresi, Mustafa R. Kılınç, and Juan Pablo Vielma. Intersection cuts for nonlinear integer programming: convexification techniques for structured sets. *Mathematical Programming*, 155(1-2):575–611, 2016.

[56] Gonzalo Muñoz, Joseph Paat, and Felipe Serrano. Towards a characterization of maximal quadratic-free sets. *arXiv preprint arXiv:2211.05185*, 2022.

[57] Gonzalo Muñoz and Felipe Serrano. Maximal quadratic-free sets. *Mathematical Programming*, 192(1):229–270, 2022.

[58] Kazuo Murota. Discrete convex analysis. *Mathematical Programming*, 83(1):313–371, 1998.

[59] George Nemhauser and Laurence Wolsey. Matroid and Submodular Function Optimization, jun 1988.

[60] 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.

[61] John MW Rhys. A selection problem of shared fixed costs and network flows. *Management Science*, 17(3):200–207, 1970.

[62] Jean-Philippe P Richard and Santanu S Dey. The group-theoretic approach in mixed integer programming. In *50 Years of Integer Programming 1958-2008*, pages 727–801. Springer, 2010.

[63] Guillaume Sagnol. Approximation of a maximum-submodular-coverage problem involving spectral functions, with application to experimental designs. *Discrete Applied Mathematics*, 161(1-2):258–276, 2013.

[64] Guillaume Sagnol and Radoslav Harman. Computing exact d-optimal designs by mixed integer second-order cone programming. *The Annals of Statistics*, 43(5):2198–2224, 2015.

[65] Anureet Saxena, Pierre Bonami, and Jon Lee. Convex relaxations of non-convex mixed integer quadratically constrained programs: projected formulations. *Mathematical programming*, 130(2):359–413, 2011.

[66] Alexander Schrijver et al. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer, 2003.

[67] Felipe Serrano. Intersection cuts for factorable MINLP. In Andrea Lodi and Viswanath Nagarajan, editors, *Integer Programming and Combinatorial Optimization*, pages 385–398, Cham, 2019. Springer International Publishing.

[68] Manohar Shamaiah, Siddhartha Banerjee, and Haris Vikalo. Greedy sensor selection: Leveraging submodularity. In *49th IEEE conference on decision and control (CDC)*, pages 2572–2577. IEEE, 2010.

[69] Xueyu Shi, Oleg A Prokopyev, and Bo Zeng. Sequence independent lifting for a set of submodular maximization problems. *Mathematical Programming*, pages 1–46, 2022.

[70] Mohit Tawarmalani and Nikolaos V Sahinidis. A polyhedral branch-and-cut approach to global optimization. *Mathematical programming*, 103(2):225–249, 2005.

[71] Donald M Topkis. *Supermodularity and complementarity*. Princeton university press, Princeton, 2011.

[72] Eli Towle and James Luedtke. Intersection disjunctions for reverse convex sets. *Mathematics of Operations Research*, 47(1):297–319, 2022.

[73] Hoang Tuy. Concave programming under linear constraints. *Soviet Mathematics*, 5:1437–1440, 1964.

[74] Hoang Tuy. Concave programming with linear constraints. In *Doklady Akademii Nauk*, volume 159, pages 32–35. Russian Academy of Sciences, 1964.

[75] Stefan Vigerske. MINLPLib: A Library of Mixed-Integer and Continuous Nonlinear Programming Instances, Feb 2022. [Online; accessed 1. Feb. 2022].

[76] Angelika Wiegele. Biq mac library—a collection of max-cut and quadratic 0-1 programming instances of medium size. *Preprint*, 51, 2007.

[77] Liding Xu, Claudia D'Ambrosio, Leo Liberti, and Sonia Haddad Vanier. On cutting planes for extended formulation of signomial programming, 2022.

[78] Qimeng Yu and Simge Küçükyavuz. Strong valid inequalities for a class of concave submodular minimization problems under cardinality constraints. *Mathematical Programming*, pages 1–59, 2023.