Intersection cuts meet submodularity

Liding Xu
Joint with Leo Liberti

OptimiX, LIX, École Polytechnique

March 1, 2023

Intersection cuts

The goal of intersection cuts: convexify a hard non-convex set $\mathcal{S}$.
Useful for LP-based global optimization solvers.

History

The history in discrete and continuous optimization:

- Continuous programs (Hoang 1964): $\mathcal{S}$ is the hypograph of a convex function (maximization of a convex function!);
- Integer programs (Balas 1971): $\mathcal{S}$ is a lattice.

Submodular functions

Submodular functions: discrete convex functions defined on Boolean hypercube.

- Let $f : \{0,1\}^n \to \mathbb{R}$ be a submodular function, define $\mathcal{S} := \{(x,t) \in \{0,1\}^n \times \mathbb{R} : f(x) \geq t\}$ the hypograph of f;
- Maximization problem ($\mathcal{NP}$-hard): $\max_{x \in \{0,1\}^n} f(x) = \max_{(x,t) \in \mathrm{conv}(S)} t$.

Outline of this talk

- Construction of $\mathcal{S}$-free sets and their maximality;
- Separation of intersection cuts;
- Generalization and Boolean multilinear (quadratic) constraints;
- Applications and testing;
- Open problems and future direction.

$\mathcal{S}$-free sets

- $\mathcal{S}$-free set: a convex set $\mathcal{C}$ such that $\text{inter}(\mathcal{C}) \cap \mathcal{S} = \varnothing$.

$\mathcal{S}$-free sets: construction 1

Lifting.

Theorem

Let $\mathcal{H}$ be a maximal $\{0,1\}^n$-free set, then $\mathcal{H} \times \mathbb{R}$ is a maximal $\mathcal{S}$-free set.

Remarks:

- Proof similar to mixed-lattice set;
- Examples: $\mathcal{H} = \{x \in \mathbb{R}^n : 0 \leq x \leq 1\}$ defined by a split.

Construct the Lovász extension of f on $[0,1]^n$, and further extend it to a continuous function $\bar{f}$ on $\mathbb{R}^n$.

Lemma

For all $x \in \{0,1\}^n$, $\bar{f}(x) = f(x)$ and $\bar{f} : \mathbb{R}^n \to \mathbb{R}$ is a (continuous) polyhedral convex function.

$\mathcal{S}$-free sets: construction 2 (cont.)

Facts: Linear components of $\bar{f}$ $\equiv$ facets of epi($\bar{f}$) $\equiv$ permutations on $\{1, \cdots, n\}$ $\equiv$ chains in $\{0, 1\}^n$.

Theorem

$\text{conv}(\text{epi}(f)) = \text{epi}(\bar{f}) \cap ([0, 1]^n \times \mathbb{R})$.

Remarks:

- Proof based on polymatroid [Atamtürk et al. 2022].
- Separation of epi($\bar{f}$): Strongly polynomial time.

$\mathcal{S}$-free sets: construction 2 (cont.)

Theorem

The epigraph $\mathrm{epi}(\bar{f})$ *of* $\bar{f}$ *is a (non-maximal)* $\mathcal{S}$*-free set.*

Why not maximal?

Theorem

Let $\mathcal{C}$ include $\text{epi}(\bar{f})$*,* $\mathcal{C}$ *is a maximal* $\mathcal{S}$*-free set if the following two conditions are satsified:*

- *each of its facets contains a point* $(x, f(x))$ *(*$x \in \{0,1\}^n$*) in its relative interior;*
- *The boundary of* $\mathcal{S}$ *contains all points* $(x, f(x))$ *(*$x \in \{0,1\}^n$*).*

Remark: Proof similar to lattice set.

A counter example

$n = 3$, $6 = 3!$ permutations/chains/facets, and $8 = 2^3$ points of $(x, f(x))$.

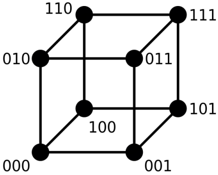


Figure: $\{0,1\}^3$

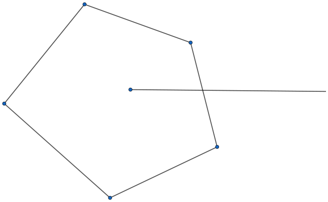
3 facets (each facet supported by 4 points) are enough:

$$\begin{aligned}
&(\,(0,0,0),(0,0,1),(0,1,1),(1,1,1)\,)\,,\\
&(\,(0,0,0),(0,1,0),(1,1,0),(1,1,1)\,)\,,\\
&(\,(0,0,0),(1,0,0),(1,0,1),(1,1,1)\,)\,.
\end{aligned}$$

Dropping remaining facets from $\mathrm{epi}(\bar{f})$ = enlarging $\mathrm{epi}(\bar{f})$.

Cut separation

- Reduction 1: The computation of one coefficient of intersection cut is reduced to a line search problem: from an interior point of epi($\bar{f}$), find the intersection point along a ray to the border of the polyhedron;
- Reduction 2: Equivalent to finding the zero point of a univariate piece-wise linear convex function.



(a) Line search

(b) 1-D zero finding

Zero finding

- Previous results [Chmiela et al. 2022, Xu et al. 2022] are based on binary search.
- New discrete Newton algorithm similar to [Iwata et al. 2008].
- Newton algorithm requires gradient information.
- Separation of epi($\bar{f}$) in a polynomial time implies that gradients can be computed in a polynomial time.

Consider $\mathcal{S} := \{(x,t) \in \{0,1\}^n \times \mathbb{R} : f_1(x) - f_2(x) \geq \ell t\}$, with $\ell \in \{0,1\}$, f_1, f_2 being submodular.

Theorem

Let $\bar{f}_1, \bar{f}_2$ be the continuous extensions of f_1, f_2 on $\mathbb{R}^n$, then $\{(x,t) \in \mathbb{R}^n \times \mathbb{R} : \bar{f}_1(x) - \bar{f}_2(x') - \nabla \bar{f}_2(x')(x - x') \leq \ell t\}$ is $\mathcal{S}$-free.

Key idea: Construct the 'best' submodular over-estimator of the submodular-supermodular function.

Boolean multilinear constraint

Theorem

Given a Boolean multilinear function $f : \{0,1\}^n \to \mathbb{R}$ *defined as* $f(x) := \sum_{k \in [K]} a_k \prod_{j \in A_k} x_j$ *(for index sets* $A_k \subseteq \{1, \cdots, n\}$*) with* K *multilinear terms, 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 on* $\{0,1\}^n$*.*

Remark: Apply the previous theorem to the Boolean multilinear constraint $f(x) \geq \ell t$.

Implementation and test

More insights come from coding and experiments:

- Implementation in SCIP 8.0.
- Result 1: Maximal $\mathcal{S}$-free sets does not imply practically good cuts;
- Result 2: Performance difference between problems with natural MILP and MINLP formulations, monotone and non-monotone.

Problem zoo

Monotone submodular maximization usually has a cardinality or knapsack constraint.

- Max cut with positive weights: non-monotone submodular maximization, natural MILP formulation.
- Exponential utility function maximization: monotone submodular maximization, natural convex MINLP formulation (too easy for SCIP).
- D-optimal design: submodular maximization, natural convex MISDP/MICP formulation (not useful).
- MUBO: submodular-supermodular maximization.

Max cut

30 “g05” and 30 “pw” instances with nonnegative weights from Biq Mac.

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	1.22	34.62	45.15

Table: Summary of MAX CUT results

Multilinear unconstrained Boolean optimization

44 “autocorr_bern” MUBO instances from MINLPLib.

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: Summary of PSEUDO BOOLEAN MAXIMIZATION results

Open problems and future

- Enlarging $\mathcal{S}$-free set;
- Is the discrete Newton algorithm strongly polynomial time? (unbounded test) [Goemans et al, 2017];
- Natural extension to submodular functions over integer lattice (integer quadratic/multilinear constraint);
- Monoid strengthening of intersection cuts similar for quadratic-constraint [Chmiela et al. 2022].
- Using the submodular overestimator for the submodular-supermodular function: better approximation algorithm? DC programming?