On a concept of a generic intersection cut callback

Liding Xu

OptimiX, LIX, École Polytechnique

November 18, 2022

Table of Contents

Intersection cuts

The goal of intersection cuts: convexify hard non-convex sets.

- Given a complex set $\mathcal{S}$, we want to tighten a polyhedral outer approximation $\mathcal{P}$ of $\mathcal{S}$;
- The polyhedral outer approximation (an LP relaxation) should be constructed *a priori*.
- Useful for LP-based solvers.

History and recent development

History:

- Concave programs (Hoang 1964): $\mathcal{S}$ is the epigraph of a concave function;
- Integer programs (Balas 1971): $\mathcal{S}$ is a lattice:
- Linear complementary programs (Ibaraki 1973): $\mathcal{S}$ is a complementary condition $x_i x_j = 0$.

History and recent development

Recent development (in non-convex MINLPs):

- Bilevel programs (Fischetti 2018);
- Factorable Programs (Serrano 2019): $\mathcal{S}$ is a sublevel set of a difference of concave functions;

History and recent development

Recent development (in non-convex MINLPs):

- Bilevel programs (Fischetti 2018);
- Factorable Programs (Serrano 2019): $\mathcal{S}$ is a sublevel set of a difference of concave functions;
- Extended formulation of quadratic/polynomial programs (Bienstock 2020): $\mathcal{S}$ is an outer product set (set of rank-1 matrices):
- Projected formulation of quadratic programs (Muñoz 2022): $\mathcal{S}$ is a sublevel set of a quadratic function (quadratic constraint).

Cut construction methods: phase 1

Preparation phase:

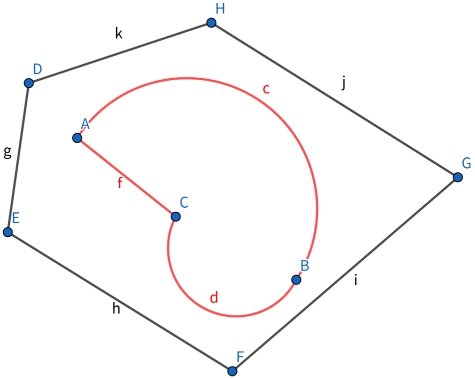
- Assumption: a point $z' \notin \mathcal{S}$, and a corner polyhedron (simplicial cone) $\mathcal{R}$ pointed at z'.

Cut construction methods: phase 1

Preparation phase:

- Assumption: a point $z' \notin \mathcal{S}$, and a corner polyhedron (simplicial cone) $\mathcal{R}$ pointed at z'.
- How to obtain?
  - optimizing a relaxation problem over the polyhedral outer approximation $\mathcal{P}$.
  - z' is the optimal solution at a vertex of $\mathcal{P}$.
  - find edges of $\mathcal{P}$ adjacent to z', these edges' convex hull is $\mathcal{R}$.

Visualization of preparation phase



Nonconvex S is enclosed by red border.
Polyheral outer approximation P is the outer polytope.

Cut construction methods: phase 2

Set construction phase:

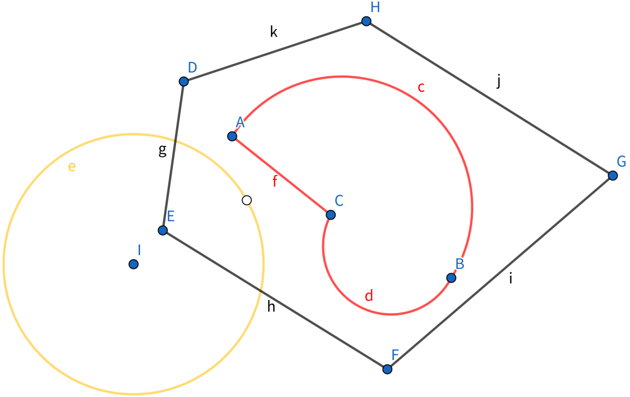
Definition

Given $\mathcal{S} \in \mathbb{R}^p$, a closed set $\mathcal{C}$ is called $\mathcal{S}$-free, if the following conditions are satisfied:

1. $\mathcal{C}$ is convex;
2. $\text{inter}(\mathcal{C}) \cap \mathcal{S} = \emptyset$.

Find an $\mathcal{S}$-free set $\mathcal{C}$ containing z'.

Visualization of set construction phase



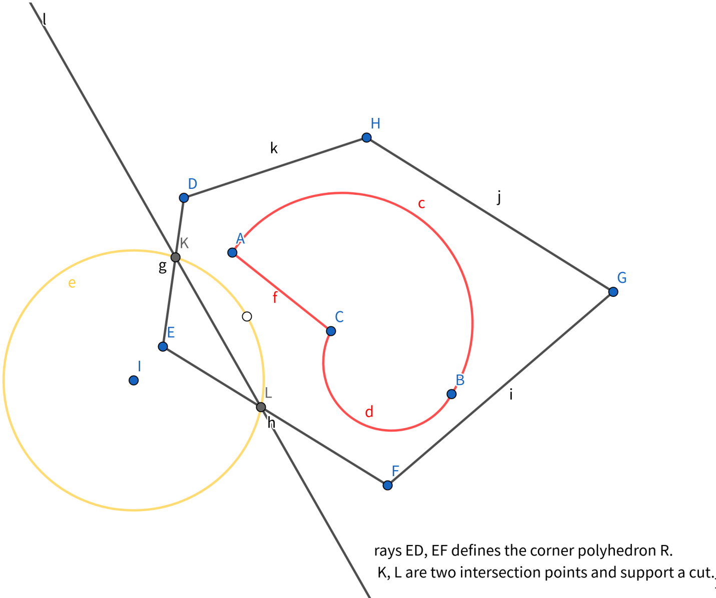
E is the relaxation point,
C is the circle containing it.

Cut construction methods: phase 3

Separation phase:

- Intersect the corner polyhedron $\mathcal{R}$ with the set $\mathcal{C}$.
- Intersection points support a separating hyperplane (an intersection cut).

Visualization of separation phase



rays ED, EF defines the corner polyhedron R.
K, L are two intersection points and support a cut.

Separation problem reduction

- Phase 1 and 3 are standard procedures.
- The only non-standard (non-trivial) procedure is Phase 2.

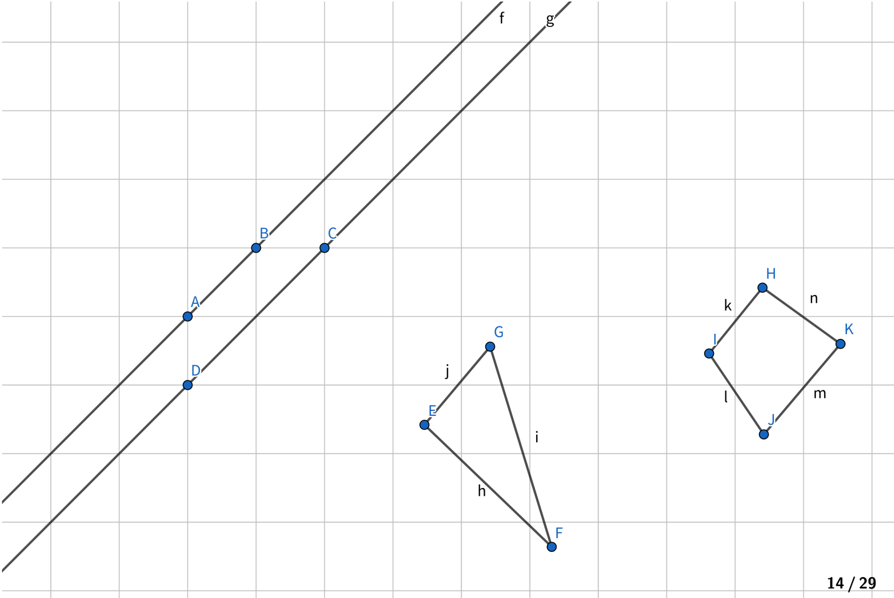
Separation problem reduction

- Phase 1 and 3 are standard procedures.
- The only non-standard (non-trivial) procedure is Phase 2.
- Larger $\mathcal{S}$-fee set gives rise to stronger cuts, so maximal $\mathcal{S}$-free set is good.
- We next review methods to construct $\mathcal{S}$-free sets in Phase 2.

Lattice sets

- Integer Programming: $\mathcal{S}$ is a lattice (the set of integer points).
- Maximal lattice-free sets in $\mathbb{R}^2$:
  - Splits;
  - Triangles;
  - Quadrilaterals;
- Gomory's Mixed Integer Cuts are split intersection cuts.

Visualization of lattice-free sets



Sublevel set of difference of concave (convex) forms

Theorem (Khamisov 1999,Serrano 2019)

Assume $\mathcal{S} := \{z \in \mathbb{R}^p : f_1(z) - f_2(z) \leq 0\}$, *where* f_1, f_2 *are concave functions. Then, for* $z' \in \mathrm{dom}(f_2)$,
$\mathcal{C}_{z'} := \{z \in \mathbb{R}^p : f_1(z) - f_2(z') - \nabla f_2(z')^\top (z - z') \geq 0\}$ *is a* $\mathcal{S}$*-free set.*

Sublevel set of difference of concave (convex) forms

Theorem (Khamisov 1999,Serrano 2019)

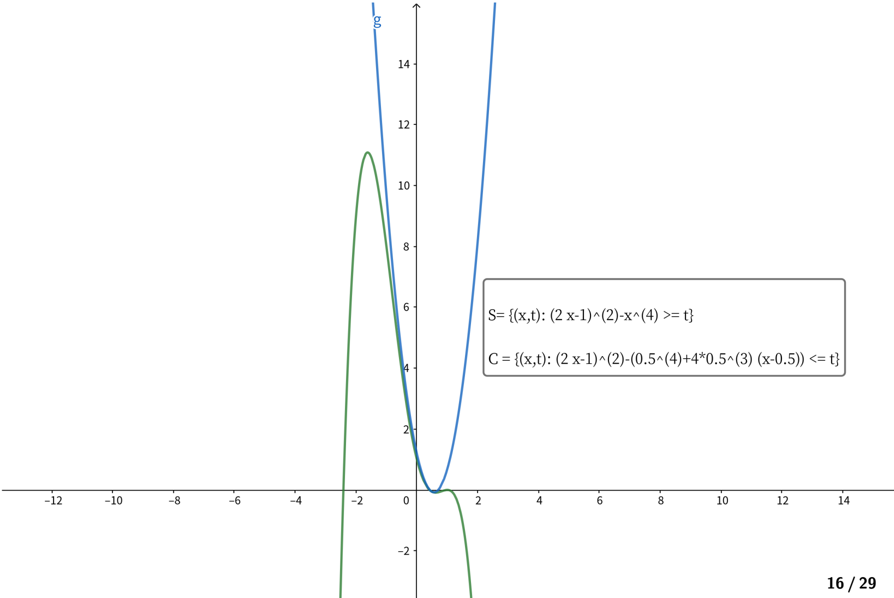
Assume $\mathcal{S} := \{z \in \mathbb{R}^p : f_1(z) - f_2(z) \leq 0\}$*, where* f_1, f_2 *are concave functions. Then, for* $z' \in \mathrm{dom}(f_2)$*,*
$\mathcal{C}_{z'} := \{z \in \mathbb{R}^p : f_1(z) - f_2(z') - \nabla f_2(z')^\top (z - z') \geq 0\}$ *is a* $\mathcal{S}$*-free set.*

Theorem (Serrano 2021)

Assume $\mathcal{S} := \{z \in \mathbb{R}^p : f_1(z) - f_2(z) \leq 0\}$*, where* f_1, f_2 *are concave functions and positive-homogeneous of degree-1. Then, for* $z' \in \mathrm{dom}(f_2)$*,*
$\mathcal{C}_{z'} := \{z \in \mathbb{R}^p : f_1(z) - f_2(z') - \nabla f_2(z')^\top (z - z') \geq 0\}$ *is a maximal* $\mathcal{S}$*-free set.*

Remark: for some case, positive-homogeneity of one concave function can be relaxed.

Visualization of a sublevel-free set



Polynomial/signomial programming

$$
\begin{aligned}
\max \quad & \sum_{k\in\mathcal{K}_0} a_{ik} \prod_{j\in[n]} x_j^{\alpha_{kj}} && \text{(1a)}\\
\forall i \in [m] \quad & \sum_{k\in\mathcal{K}_i} a_{ik} \prod_{j\in[n]} x_j^{\alpha_{kj}} \le 0 && \text{(1b)}
\end{aligned}
$$

where $\mathcal{K}$ is the index set for the whole monomial terms $\{\prod_{j\in[n]} x_j^{\alpha_{kj}}\}_{k\in\mathcal{K}}$, $\mathcal{K}_0$ and $\mathcal{K}_i$ are its subsets.

- Polynomial programming: $\alpha_{kj} \in \mathbb{Z}_+$ (nonegative integer);
- Signomial programming: $\alpha_{kj} \in \mathbb{R}$ (real);

Examples: extended formulation of polynomial programming

Dense lifting: a polynomial program can be lifted to an LP + rank-1 condition on a matrix X (Bienstock 2020).

- X_{ij} represents a product of two monomial terms.
- Theorem: if X is rank one, then the determinants of its 2-by-2 minors are zeros;
- Example of a principle minor: $X_{ii}X_{jj} - X_{ij}^2 = 0$.

Examples: extended formulation of polynomial programming

Dense lifting: a polynomial program can be lifted to an LP + rank-1 condition on a matrix X (Bienstock 2020).

- X_{ij} represents a product of two monomial terms.
- Theorem: if X is rank one, then the determinants of its 2-by-2 minors are zeros;
- Example of a principle minor: $X_{ii}X_{jj} - X_{ij}^2 = 0$.
- Reformulation: $(X_{ii} + X_{jj})^2 - (X_{ii} - X_{jj})^2 = 4X_{ij}^2$;
- DCC equivalence: $(X_{ii} + X_{jj})^2 - (X_{ii} - X_{jj})^2 - 4X_{ij}^2 \leq 0$ and $(X_{ii} + X_{jj})^2 - (X_{ii} - X_{jj})^2 - 4X_{ij}^2 \geq 0$;

Examples: extended formulation of signomial programming

Sparse lifting: a signomial program can be lifted to an LP + condition $y = x^\alpha$ (our working paper).

- Signomial-term-set $\mathcal{S} = \{(x, y) \in \mathbb{R}^n_+ \times \mathbb{R}_+ : y \leq x^\alpha\}$, where α is an exponent vector with negative and/or positive entries;

Examples: extended formulation of signomial programming

Sparse lifting: a signomial program can be lifted to an LP + condition $y = x^\alpha$ (our working paper).

- Signomial-term-set $\mathcal{S} = \{(x, y) \in \mathbb{R}^n_+ \times \mathbb{R}_+ : y \ \leq \ x^\alpha\}$, where α is an exponent vector with negative and/or positive entries;
- After some transformation, $\mathcal{S} = \{(u, v) \in \mathbb{R}^h_+ \times \mathbb{R}^l_+ : \ u^\beta - v^\gamma \leq 0\}$, where $\max(\|\beta\|_1, \|\gamma\|_1) = 1$ and $\beta, \gamma \geq 0$.

Examples: extended formulation of signomial programming

Sparse lifting: a signomial program can be lifted to an LP + condition $y = x^\alpha$ (our working paper).

- Signomial-term-set $\mathcal{S} = \{(x, y) \in \mathbb{R}^n_+ \times \mathbb{R}_+ : y \le x^\alpha\}$, where α is an exponent vector with negative and/or positive entries;
- After some transformation, $\mathcal{S} = \{(u, v) \in \mathbb{R}^h_+ \times \mathbb{R}^l_+ : u^\beta - v^\gamma \le 0\}$, where $\max(\|\beta\|_1, \|\gamma\|_1) = 1$ and $\beta, \gamma \ge 0$.
- Intersection cuts: u^β, v^γ are power functions (whose hypograph are power cone representable) and concave, $\mathcal{S}$ now is in the difference of concave form;

Examples: extended formulation of signomial programming

- Factorable programming: u^{β} is concave, so it under-estimators can be constructed by factorization. For instance, $u_1^{0.5}u_2^{0.3}u_3^{0.2} \leq t$ is reverse convex.
- (conventional) multilinear factorization: $u_1^{0.5} \leq t_1, u_2^{0.3} \leq t_2, u_3^{0.2} \leq t_3, t_1 t_2 t_3 \leq t$.
- (new) power factorization: $u_2^{0.6}u_3^{0.4} \leq t_1, u_1^{0.5}t_1^{0.5} \leq t$. We can give convex envelopes of $u_2^{0.6}u_3^{0.4}$, $u_1^{0.5}t_1^{0.5}$.

Supporting intersection cuts

- In the future, we will find more families of $\mathcal{S}$-free sets.
- Users want to quickly know the performance of cuts from their $\mathcal{S}$-free sets in a real solver, rather than manually constructing polyhedral outer approximation.
- A callback-based solution.

Pipeline of intersection cuts

- Phase 1 deals with simplex tableau and construct corner polyhedron (standard).
- Phase 3 finds intersection points (standard).
- Non-standard: phase 2, defining an $\mathcal{S}$-free set.

Defining $\mathcal{S}$-free set

An $\mathcal{S}$-free set is $\mathcal{C} := \{z \in \mathcal{D} : g(z) \geq 0\}$, $\mathcal{D}$ is a domain, and $g(z') \geq 0$.

- g is concave over $\mathcal{D}$.
- A sublevel-free set $\mathcal{C} := \{z \in \mathcal{D} : g(z) \geq 0\}$.
- Arbitrary set $\mathcal{C}$ (like lattice-free): $g(z) = \begin{cases} 1, & z \in \mathcal{D} \cap \mathcal{C} \\ -\infty, & \text{otherwise.} \end{cases}$

  is an indicator function.

Interface: the user needs to register the defining-variables of $\mathcal{C}$ and domain $\mathcal{D}$.

Oracle access and separation

Defining $\mathcal{C}$ is equivalent to defining 0th-order (function value) access to $g(z)$, optional: 1th-order (gradient value) oracle access to $g(z)$.

- The separation problem: find intersection point of ray $z' + tr$ ($t \geq 0$) with $\mathcal{C}$, where r is an extreme ray of the corner polyhedron $\mathcal{R}$;
- Equivalently, find root of the 1d function $g(z' + tr)$;

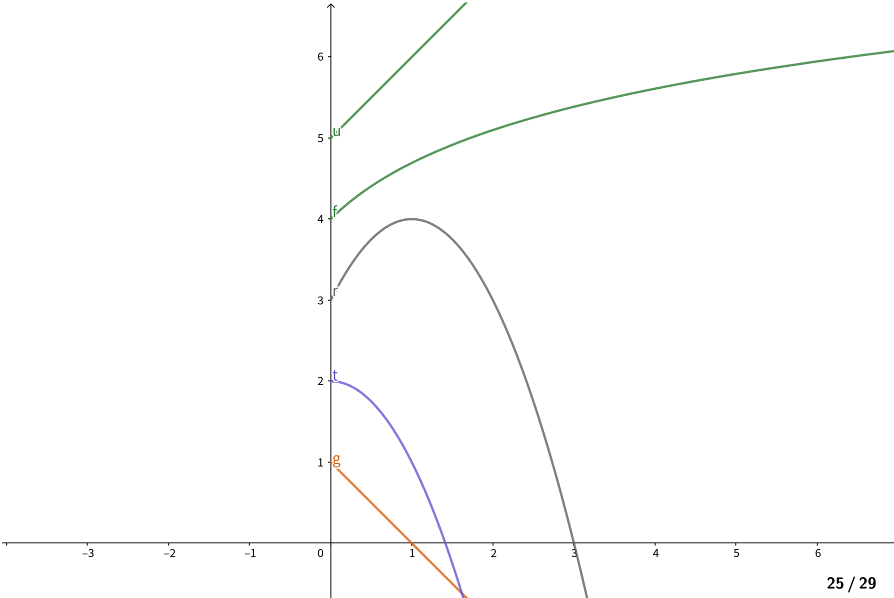
Oracle access and separation

Defining $\mathcal{C}$ is equivalent to defining 0th-order (function value) access to $g(z)$, optional: 1th-order (gradient value) oracle access to $g(z)$.

- The separation problem: find intersection point of ray $z' + tr$ ($t \geq 0$) with $\mathcal{C}$, where r is an extreme ray of the corner polyhedron $\mathcal{R}$;
- Equivalently, find root of the 1d function $g(z' + tr)$;
- Bisection root finding: 0th-order oracle access.
- Newton root finding: 0th-order and 1th-order oracle access.

Interface: user provides 0th-order and 1th-order oracle access.

Root finding



Abstract functions of the callback

Setting:

- BisectionOrNewtion: TRUE or FALSE.

Minimal interface functions

- Register(): register variables and domain for an S-free set.
- ZeroOrderOracle(): 0th-order access.
- FirstOrderOracle(): 1st-order access.

The callback automatically extracts corner polyhedron, finds roots, and checks numerical stability.

Limitations

Intersection cuts can be dense and thus numerically dangerous.

Strengthenning methods

We can at best approximate $\operatorname{conv}(\mathcal{C}^c \cap \mathcal{R})$, and $\mathcal{R}$ is a loose relaxation of $\mathcal{P}$. Balas's original (generalized) intersection cuts definition: $\mathcal{R}$ is $\mathcal{P}$.

- Consider variables' bounds: Chielma 2022.
- Consider bounded simplex paths from a relaxation point, more edges of $\mathcal{P}$ are considered: Balas 2022.

Comparing lift-and-project

When $\mathcal{C}$ is a polyhedron,

- Intersection cuts for $(\text{conv}(\mathcal{C}^c \cap \mathcal{R}))$ is weaker than lift-project cuts $(\text{conv}(\mathcal{C}^c \cap \mathcal{P}))$.
- Assume $\mathcal{P} = \mathcal{R}$, intersection cuts are then equivalent to lift-and-project cuts

When $\mathcal{C}$ is not polyhedron

- Only Intersection cuts works.