# $S$-free Sets for Polynomial Optimization and Oracle-Based Cuts

Daniel Bienstock, Chen Chen, Gonzalo Muñoz, IEOR, Columbia University

May, 2017

# Background

Consider a mathematical program of the following form

$$\min c^T x$$

subject to $x \in S \cap P$.

$P := \{x \in \mathbb{R}^n | Ax \leq b\}$ is a polyhedral set, and $S \subset \mathbb{R}^n$ is a closed set.

Can we strengthen $P$ with cuts?

Consider a mathematical program of the following form

$$\min c^T x$$

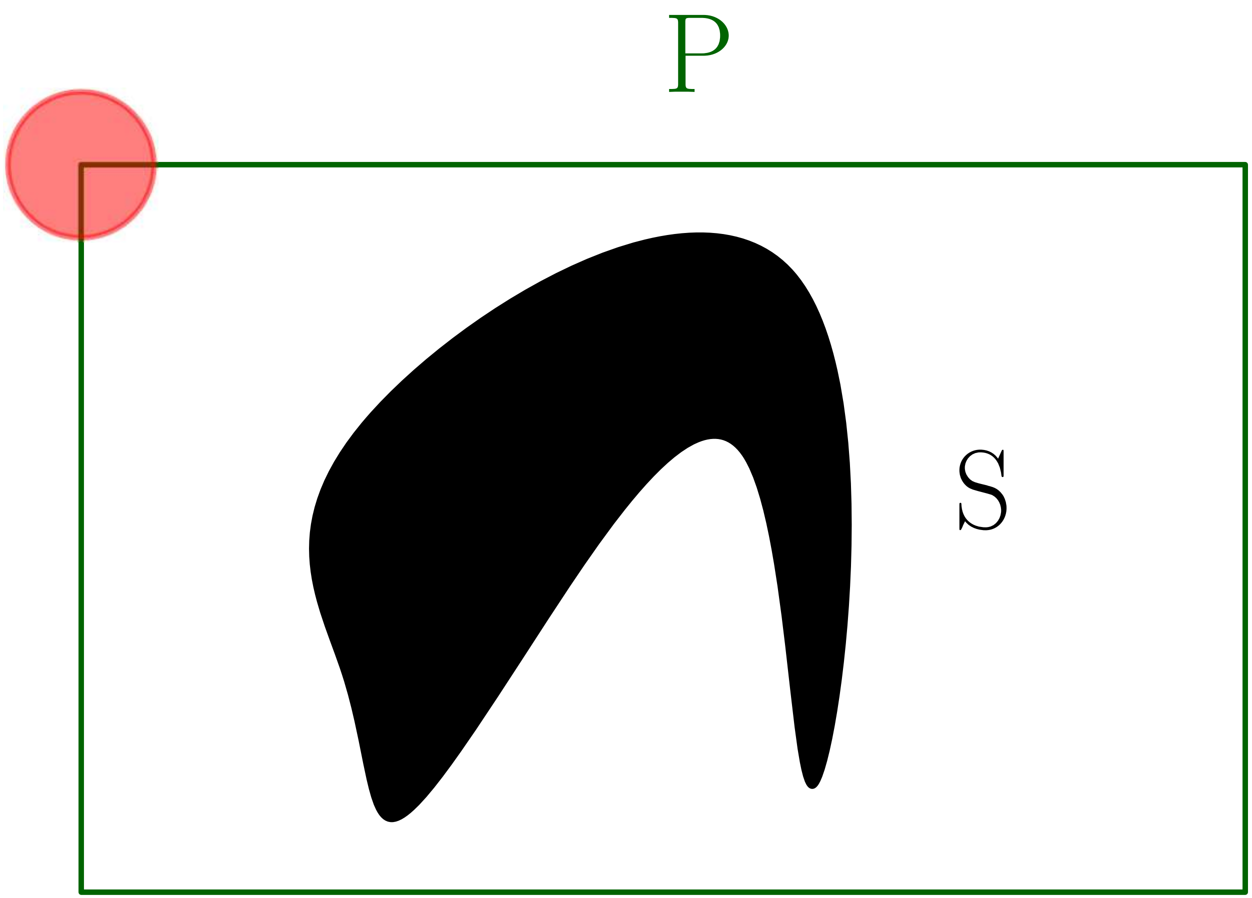
$$\text{subject to } x \in S \cap P.$$

$P := \{x \in \mathbb{R}^n | Ax \leq b\}$ is a polyhedral set, and $S \subset \mathbb{R}^n$ is a closed set.

Can we strengthen $P$ with cuts?

We shall focus on the geometric approach: cuts via $S$-free sets.

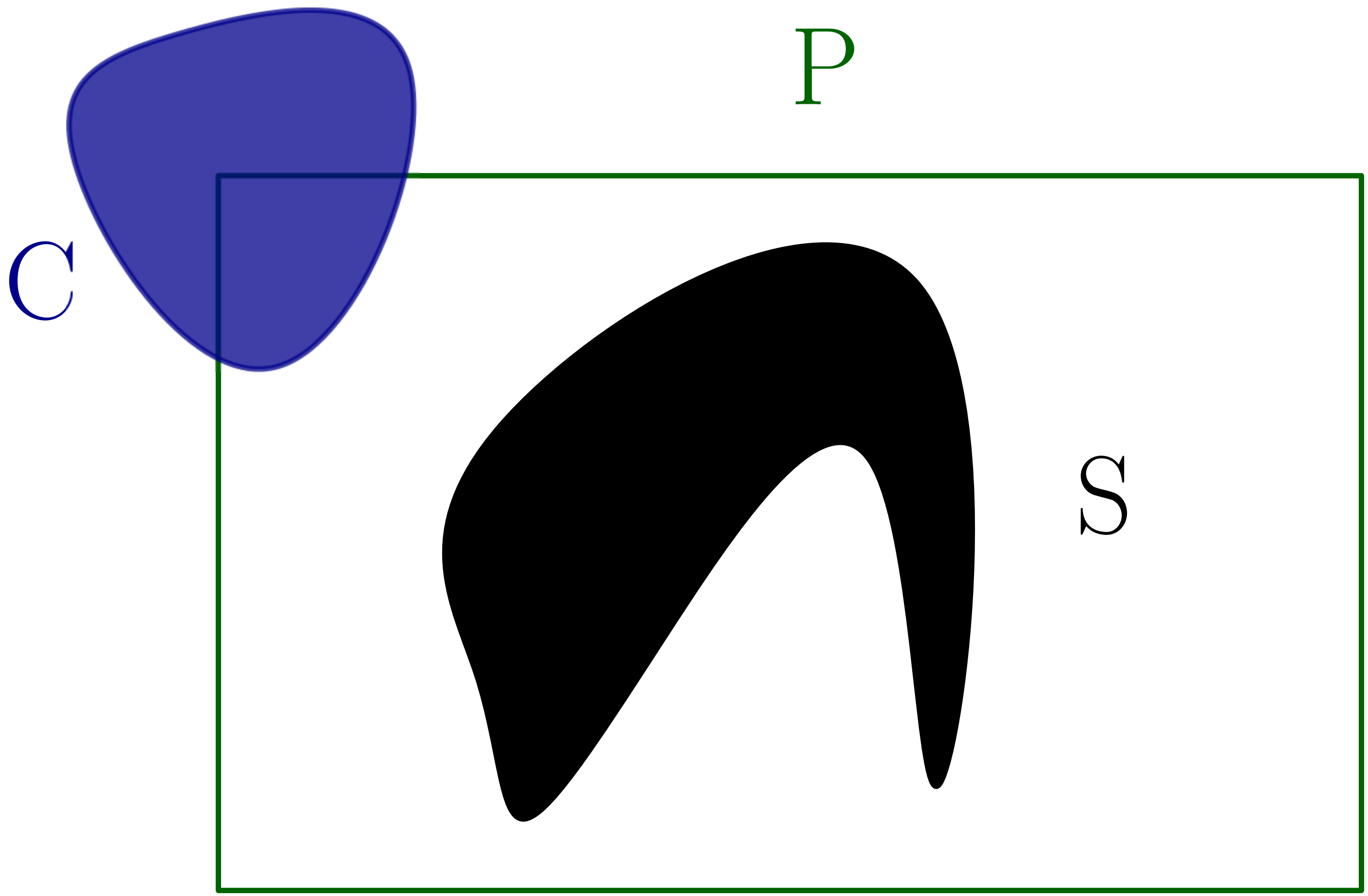
(Many other ways to generate cuts, e.g. disjunctions, algebraic arguments, combinatorics, convex cuts, etc.)



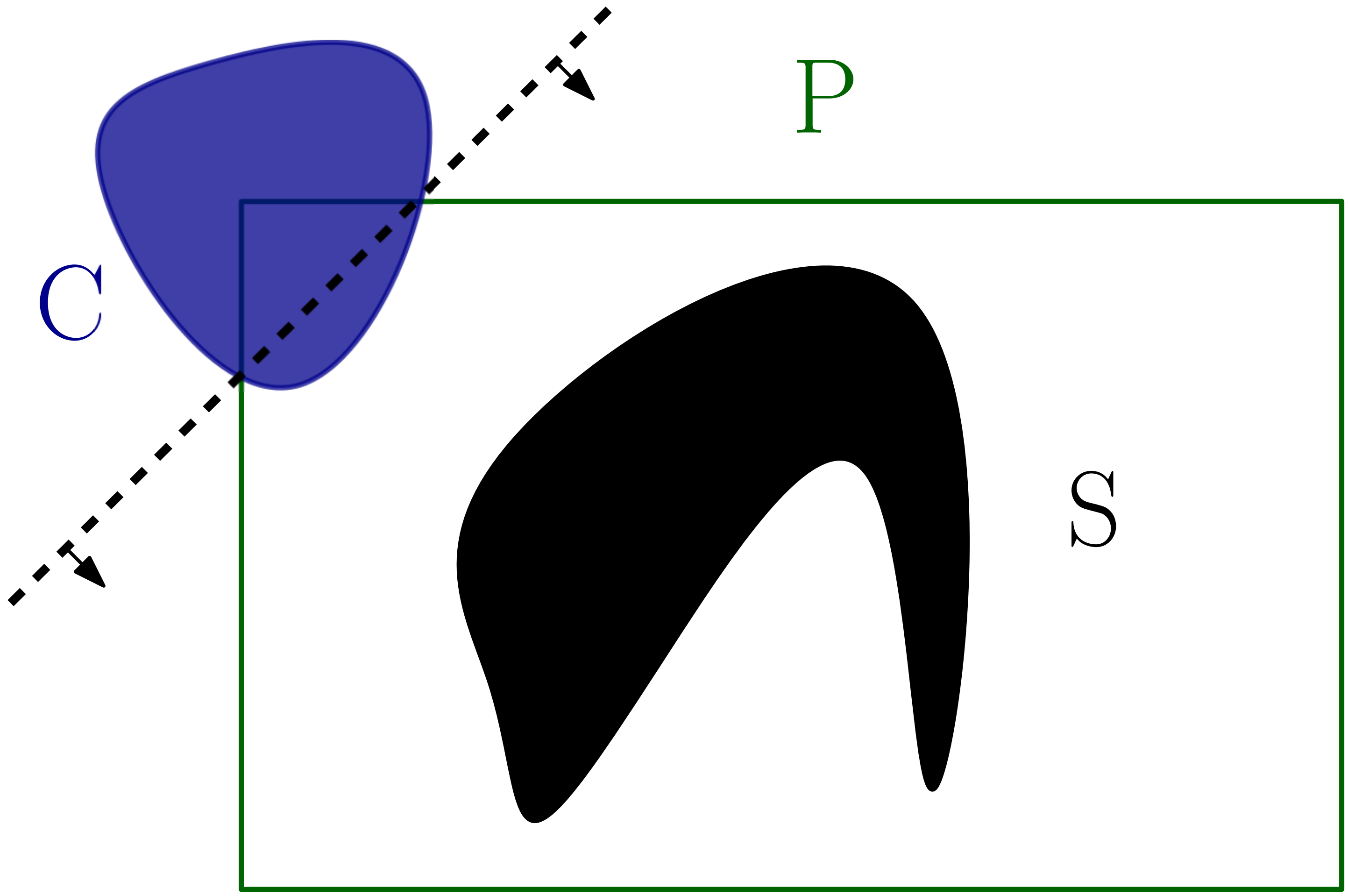

Let $S$ be some closed set. Example: a rectangle $P$. Want to separate the extreme point marked with a red circle from $S$.

# Tightening $P$ with an $S$-free set $C$

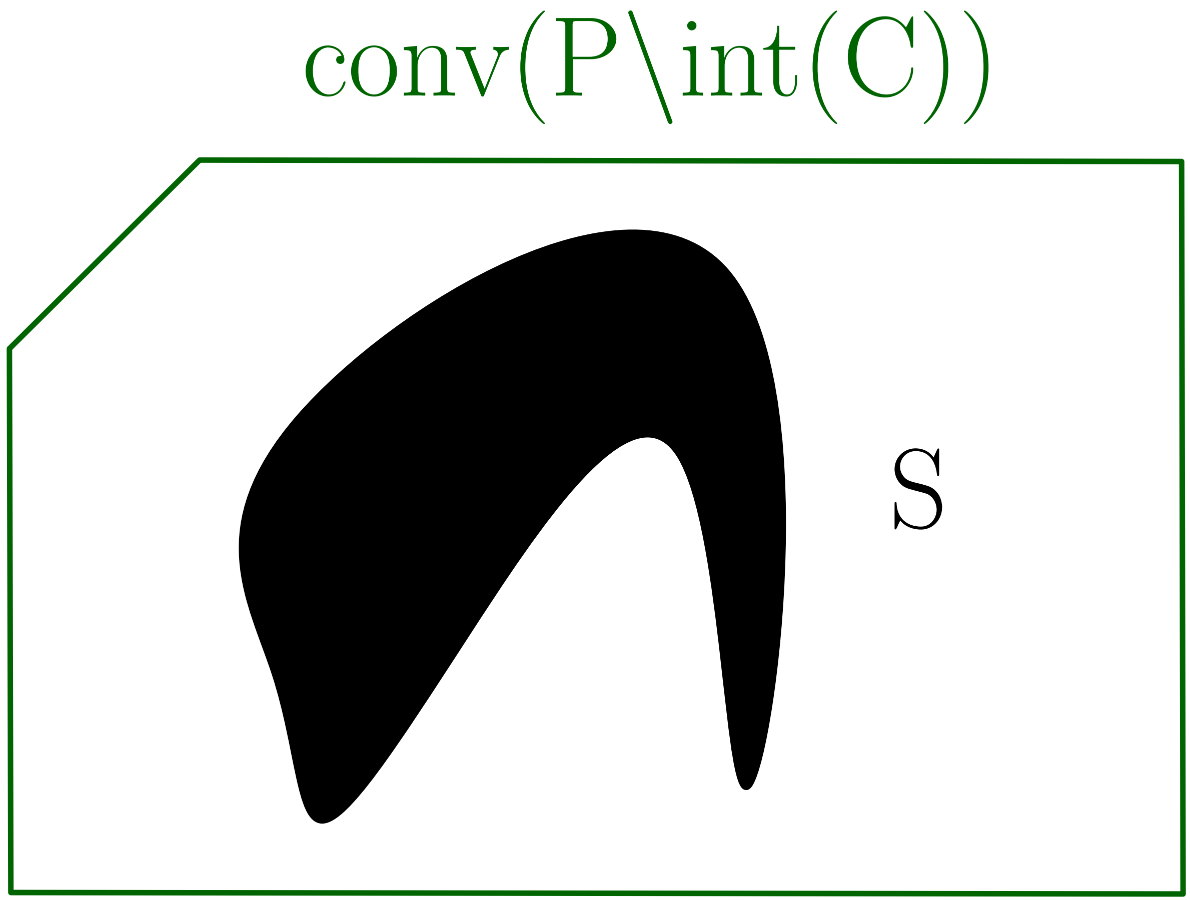




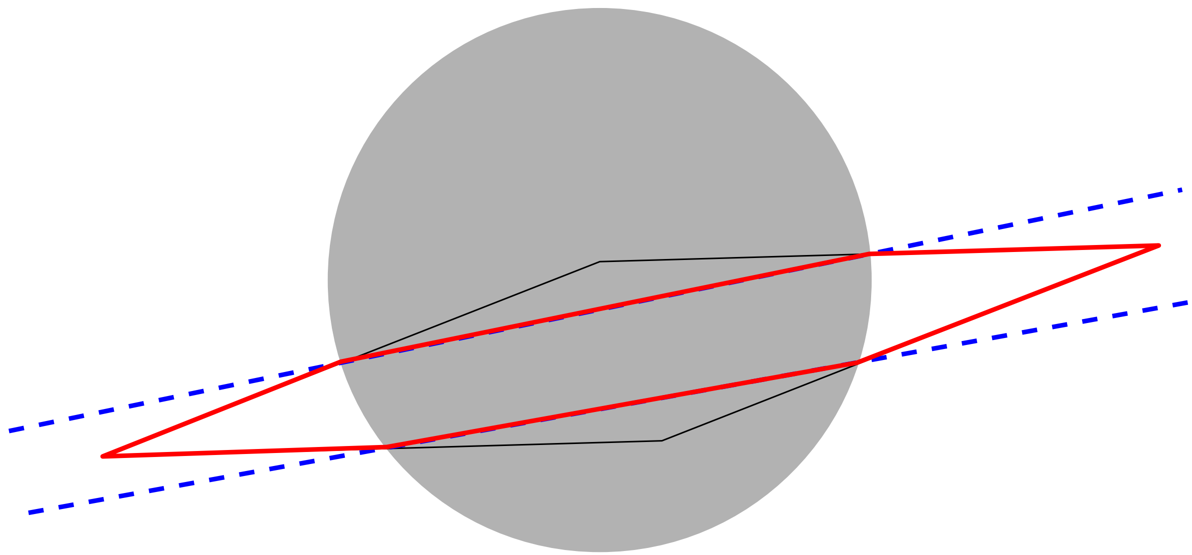

$C$ is an $S$-free set [Dey and Wolsey 2010]: a closed convex set with an interior that does not intersect with $S$.




We obtain a cut by subtracting $C$ from $P$.




Applying the single cut gives us conv($P \setminus C$).



More than one cut, possibly an infinite number are needed.
Separation is generally NP-Hard, e.g. $P$ a polytope, $C$ a ball models an NP-complete set containment problem.

# conv($P \setminus$ int$(C)$) is tricky

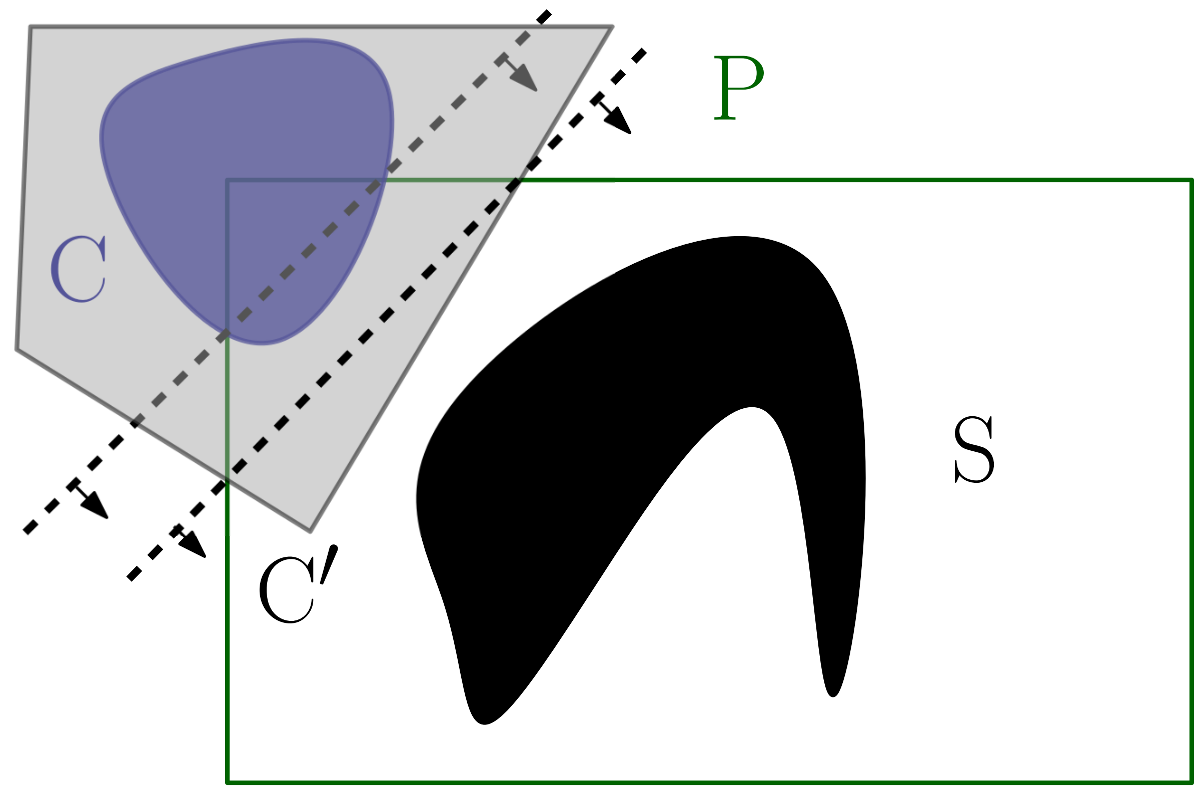


Balas, 1971 (see also Tuy, 1964): If $P$ is a simplicial cone then the intersection cut guarantees separation over conv($P \setminus$ int$(C)$)
Simplicial cone: $n$ linearly independent linear inequalities
Simplicial conic relaxation $P' \supseteq P$ is easily obtained from a basic solution of $P$
With less ambition we go for conv($P' \setminus$ int$(C)$)
Intersection cut is described in closed form $\rightarrow$ fast separation of extreme points of $P$ using $P'$

# Intersection cuts and maximal $S$-free sets



Bigger $C$, deeper cuts.




An $S$-free set $C$ is *maximal* $S$-free if it is not contained in another $S$-free set.

# (Non-Exhaustive) Additional Literature



Maximal $S$-free sets and minimal valid inequalities: [Basu et al. 2010], [Conforti et al. 2014], [Cornuejols, Wolsey, Yildiz, 2015], [Kilinc-Karzan 2015], etc.

Intersection cuts and for mixed-integer conic programs programming: [Atamturk and Narayanan 2010], [Belotti et al., 2013], [Andersen and Jensen, 2013], [Dadush, Dey, Vielma 2011], [Modaresi, Kilinc, Vielma 2015/2016], etc.

Intersection cuts for bilevel optimization: [Fischetti, Monaci, Sinni, 2016].

Generalized intersection cut procedures: [Balas and Margot, 2013], [Balas, Kazachkov, Margot 2016]

# Our Contributions



1) A simple, generic way to generate $S$-free sets that ensures separation. Also, a corresponding cutting plane method for arbitrary closed sets, guaranteed to converge on bounded problems.
2) A study of maximal $S$-free sets for polynomial optimization

# Oracle-Based Cuts

# Distance Oracle



Suppose we have an oracle for a closed set $S$ that gives us the distance $d(x, S)$ from any point $x \in \mathbb{R}^n$ to the nearest point in $S$.

# Distance Oracle



Suppose we have an oracle for a closed set $S$ that gives us the distance $d(x, S)$ from any point $x \in \mathbb{R}^n$ to the nearest point in $S$.
Examples:
Integer programming: if $S$ is the lattice, then one can round.
Polynomial optimization: distance can be calculated to arbitrary accuracy in polynomial time
Cardinality constraints: nearest vector of card $(k)$ can be obtained by rounding.

# Distance Oracle



Suppose we have an oracle for a closed set $S$ that gives us the distance $d(x, S)$ from any point $x \in \mathbb{R}^n$ to the nearest point in $S$.
Examples:
Integer programming: if $S$ is the lattice, then one can round.
Polynomial optimization: distance can be calculated to arbitrary accuracy in polynomial time
Cardinality constraints: nearest vector of card $(k)$ can be obtained by rounding.
Observation. The ball centered around $x$ with radius $d(x, S)$ is $S$-free. Call it $\mathcal{B}(x, d(x, S))$.
We shall call the corresponding intersection cut an *oracle ball cut*.

# Cut Closure Convergence



Start with a polytope $P_0$. Define $P_{k+1}$ as $P_k$ intersected with conv($P_k \setminus$ int($\mathcal{B}(x, d(x, S))$)) for every extreme point $x$. This is the rank $k$ *oracle cut closure*.

# Cut Closure Convergence



Start with a polytope $P_0$. Define $P_{k+1}$ as $P_k$ intersected with conv($P_k \setminus$ int$(\mathcal{B}(x, d(x, S))))$ for every extreme point $x$. This is the rank $k$ *oracle cut closure*.

Theorem: $\lim_{k\to\infty} P_k =$ conv$(S \cap P)$.

# Cut Closure Convergence



Start with a polytope $P_0$. Define $P_{k+1}$ as $P_k$ intersected with conv($P_k \setminus$ int($\mathcal{B}(x, d(x, S)))$) for every extreme point $x$. This is the rank $k$ *oracle cut closure*.

Theorem: $\lim_{k\to\infty} P_k =$ conv$(S \cap P)$.

Corollary: given an inexact but arbitrarily accurate distance oracle, we can obtain arbitrarily close (in terms of Hausdorff distance) polyhedral approximation to conv$(S \cap P)$ in finite time.

Borrows from proof technique used in [Averkov 2011].

# Polynomial Optimization

# Polynomial Optimization



$$z^* := \inf_{x \in S} p_0(x)$$

$$S := \{x \in \mathbb{R}^n | p_1(x) \geq 0, ..., p_m(x) \geq 0\}$$

# Cuts for Polynomial Optimization



[Saxena, Bonami, Lee 2010/2011] Disjunctive cuts from MILP inner-approximation + convex cuts
Applies to bounded polynomial optimization
[Ghaddar, Vera, Anjos 2011] Projections of moment relaxations.
Generalizes Balas, Ceria, Cornuejols lifting. Separation not guaranteed in general.
Older literature on convex envelopes of functions, e.g. multilinear.

# Cuts for Polynomial Optimization



[Saxena, Bonami, Lee 2010/2011] Disjunctive cuts from MILP inner-approximation + convex cuts
Applies to bounded polynomial optimization
[Ghaddar, Vera, Anjos 2011] Projections of moment relaxations.
Generalizes Balas, Ceria, Cornuejols lifting. Separation not guaranteed in general.
Older literature on convex envelopes of functions, e.g. multilinear.
Our intersection cuts (using e.g. the ball) guarantee polynomial-time separation without boundedness assumptions.

# Lifted Polynomial Representation



[Shor 1987], [Lovasz and Schrijver 1991] Define a vector of monomials, $m = [1, x_1, ..., x_n, x_1x_2, x_1x_3, ..., x_n^k]$. Let $M = mm^T$.

Polynomial optimization can be formulated as

$$\min \langle A_0, M \rangle$$
$$\text{s.t. } \langle A_i, M \rangle \leq b_i, i = 1, ..., m.$$

This is a linear programming relaxation with respect to $M$. $\langle A_i, M \rangle := \sum a_{ij} m_{ij}$ is the inner product.

[Shor 1987], [Lovasz and Schrijver 1991] Define a vector of monomials, $m = [1, x_1, ..., x_n, x_1x_2, x_1x_3, ..., x_n^k]$. Let $M = mm^T$.

Polynomial optimization can be formulated as

$$\min \langle A_0, M \rangle$$
$$\text{s.t. } \langle A_i, M \rangle \leq b_i, i = 1, ..., m.$$

This is a linear programming relaxation with respect to $M$.
$\langle A_i, M \rangle := \sum a_{ij} m_{ij}$ is the inner product.
Equivalency when $M \succeq 0$, rank$(M) = 1$ and consistency constraints. Dropping the rank constraint gives the moment relaxation [Lasserre, *SIAMOPT* 2001].

# $S$-free sets for Polynomial Optimization



Geometric notions are with respect to a vectorized space, e.g.

$$M \in \mathbb{S}^{2\times 2} \to \{M_{11}, M_{12}, M_{22}\} \in \mathbb{R}^3$$

A convex set (in the appropriate vectorized space) is *outer-product-free* (OPF) if no point in the interior corresponds to a matrix that can be represented as an outer-product.
A set is maximal OPF if no OPF set strictly contains it.

# $S$-free sets for Polynomial Optimization



Geometric notions are with respect to a vectorized space, e.g.

$$M \in \mathbb{S}^{2\times 2} \to \{M_{11}, M_{12}, M_{22}\} \in \mathbb{R}^3$$

A convex set (in the appropriate vectorized space) is *outer-product-free* (OPF) if no point in the interior corresponds to a matrix that can be represented as an outer-product.

A set is maximal OPF if no OPF set strictly contains it.

It turns out OPF sets can have nice structure and lead to easily generated intersection cuts.

# Violation Ball



Using a modification by Dax (2016) of the Eckart-Young-Mirksy theorem, we can find the nearest (by Frobenius norm) outer-product to a given matrix. This uses eigenvalues, and hence can be generated (to specified precision) in polynomial time.

# Max Full-Dim OPF Sets are Cones



Theorem: Let $C \subset \mathbb{S}^{n \times n}$ be an outer-product-free set with full dimension. Then clcone$(C)$ is outer-product-free.

# Negative Semidefinite $P_i$



$\langle P_i, M \rangle \geq 0$

Theorem: Every such halfspace with $P_i$ NSD is maximal outer-product-free.

These halfspaces yield the standard outer approximation cut for SDP: $M \succeq 0 \iff c^T M c \geq 0 \forall c \in \mathbb{R}^n$.

Define $\mathcal{C}_{i,j} := \{M \in \mathbb{S}^{n\times n} | M_{[i,j]} \succeq 0\}$
Theorem: $\mathcal{C}_{i,j}$ is max OPF for $1 \le i \neq j \le n$.

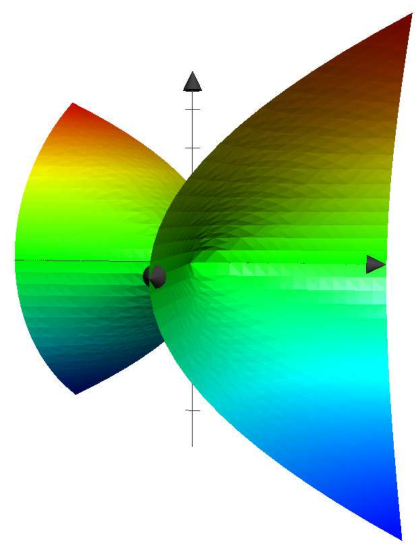
# Characterization for $2 \times 2$



The 3-dimensional case.

Theorem: For $n = 2$, the maximal OPF set are $\mathcal{C}_{1,2}$ and halfspaces of the form $\langle P, M \rangle \geq 0$, where $P$ is NSD.

i.e. the PSD cone and halfspaces with boundaries that support the cone



# Cuts

Recall: we want every extreme point of our relaxation to be an outer-product, i.e. PSD and rank one.
Any non-PSD extreme point can be separated by the outer-approximation cuts $c^T M c \geq 0$.
Any PSD extreme point with rank greater than 1 can be separated by the intersection cut given by $\mathcal{C}_{i,j}$ for some $i, j$ [Chen, Oren, Atamturk 2016].
The oracle cut can be strengthened by recentering the ball and taking the conic hull (complicated expression but computationally fast).

# Numerical Experiments: Setup



Pure cutting plane algorithm implemented in Python using combinations of the following cuts:

OB (Oracle Ball Cuts), SO (Strengthened Oracle Cuts)

OA (Outer Approximation Cuts), 2x2 (2x2 Principal Minor Cuts)

LP solver: Gurobi 7.0.1

Hardware: 20-core server, Intel Xeon 3.10GHz CPU, 264 GB RAM

26 QCQP problems from GLOBALLib (6-63 variables)

99 BoxQP instances (21-126 variables)

# Numerical Experiments: Results



| Cut Family | Initial Gap | End Gap | Closed Gap | # Cuts | Iters | Time (s) | LPTime (%) |
|---|---|---|---|---|---|---|---|
| OB | 1387.92% | 1387.85% | 1.00% | 16.48 | 17.20 | 2.59 | 2.06% |
| SO | | 1387.83% | 8.77% | 18.56 | 19.52 | 4.14 | 2.29% |
| OA | | 1001.81% | 8.61% | 353.40 | 83.76 | 33.25 | 7.51% |
| 2x2 + OA | | 1003.33% | 32.61% | 284.98 | 118.08 | 30.40 | 15.03% |
| SO+2x2+OA | | 1069.59% | 31.91% | 174.79 | 107.16 | 29.55 | 12.56% |

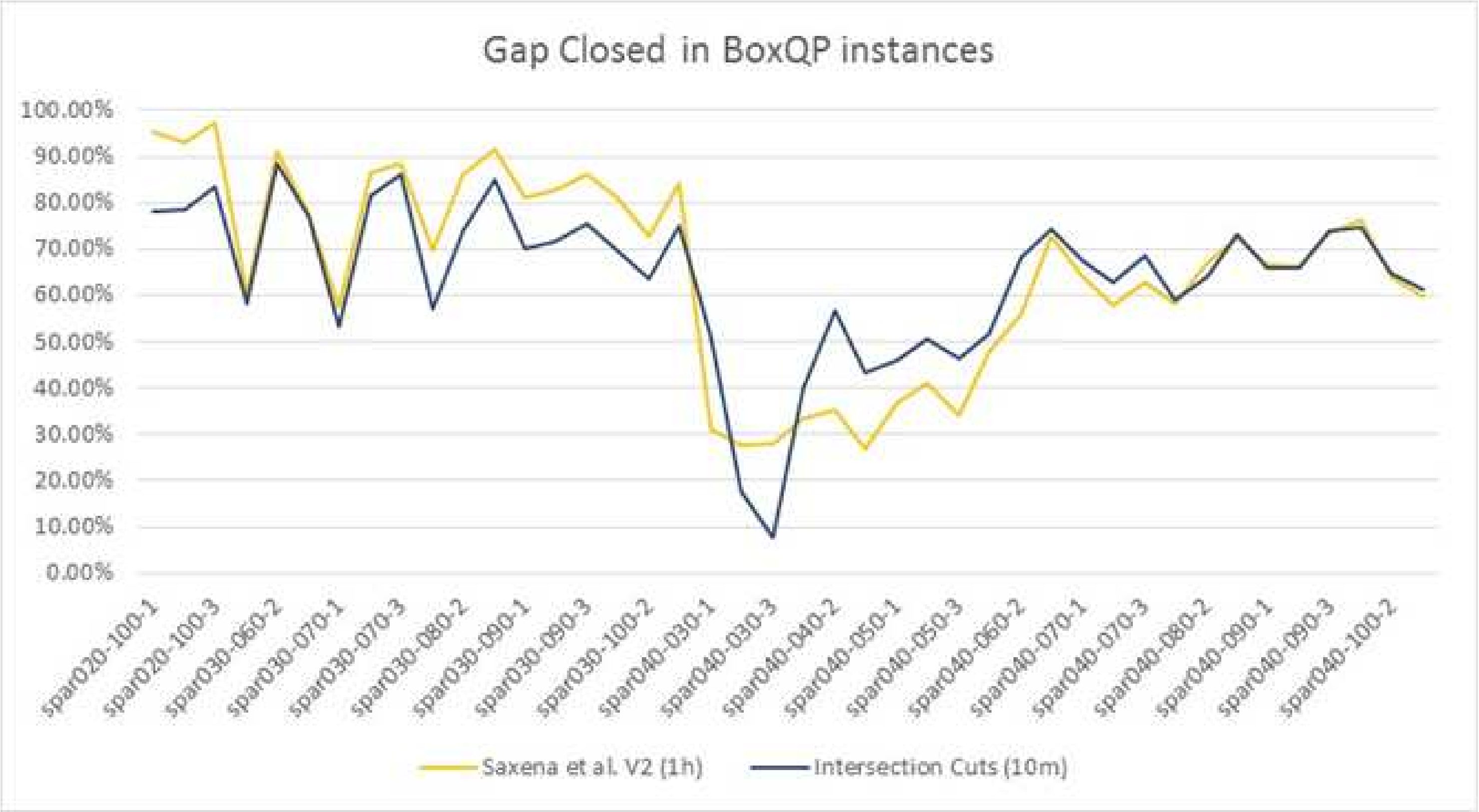
Table 1: Averages for GLOBALLib instances

| Cut Family | Initial Gap | End Gap | Closed Gap | # Cuts | Iters | Time (s) | LPTime (%) |
|---|---|---|---|---|---|---|---|
| OB | 103.59% | 103.56% | 0.04% | 12.84 | 13.62 | 127.15 | 0.40% |
| SO | | 103.33% | 0.34% | 14.34 | 15.45 | 132.07 | 0.49% |
| OA | | 30.88% | 75.55% | 676.90 | 137.52 | 459.28 | 31.80% |
| 2x2 + OA | | 32.84% | 74.52% | 349.21 | 140.40 | 473.18 | 28.76% |
| SO+2x2+OA | | 33.43% | 74.03% | 227.39 | 136.93 | 475.38 | 26.59% |

Table 2: Averages for BoxQP instances

# Comparison with V2: BoxQP

V2: second-order conic outer-approximation of PSD constraint MIP to derive disjunctive cuts



# Comparison with V2: GLOBALLib



| Instance | V2 Gap | V2 Time | Gap Closed | Time |
|---|---|---|---|---|
| Ex2_1_1 | 72.62% | 704.40 | 53.21% | 0.41 |
| Ex2_1_5 | 99.98% | 0.17 | 99.68% | 0.13 |
| Ex2_1_6 | 99.95% | 3397.65 | 93.87% | 0.95 |
| Ex2_1_8 | 84.70% | 3632.28 | 73.23% | 19.13 |
| Ex2_1_9 | 98.79% | 1587.94 | 29.87% | 36.9 |
| Ex3_1_1 | 15.94% | 3600.27 | 0.34% | 0.55 |
| Ex3_1_2 | 99.99% | 0.08 | 99.98% | 0.04 |
| Ex3_1_4 | 86.31% | 21.26 | 29.49% | 0.26 |
| Ex5_2_2_case1 | 0.00% | 0.02 | **2.05%** | 0.47 |
| Ex5_2_2_case2 | 0.00% | 0.05 | 0.00% | 0.26 |
| Ex5_2_2_case3 | 0.36% | 0.36 | 0.00% | 0.16 |
| Ex5_2_4 | 79.31% | 68.93 | 29.04% | 5.69 |

Table 3: Comparison with V2 on GLOBALLib instances

# Comparison with V2: GLOBALLib



| Instance | V2 Gap | V2 Time | Gap Closed | Time |
|---|---|---|---|---|
| Ex5_3_2 | 7.27% | 245.82 | 0.00% | 2.33 |
| Ex5_4_2 | 27.57% | 3614.38 | 0.24% | 0.59 |
| Ex9_1_4 | 0.00% | 0.60 | 0.00% | 0.34 |
| Ex9_2_1 | 60.04% | 2372.64 | 54.17% | 28.37 |
| Ex9_2_2 | 88.29% | 3606.36 | 77.90% | 30.84 |
| Ex9_2_6 | 87.93% | 2619.02 | **90.45%** | 0.12 |
| Ex9_2_8 | - | - | **83.27%** | 0.12 |

# Conclusion

# Take-homes



We introduced an oracle-based intersection cut for closed sets. Furthermore, we constructed a convergent cutting plane algorithm that uses this oracle to 'ping' the set $S$. All of this is done without using any explicit structure about $S$.

Outer-product-free sets provide a new way to generate cuts for polynomial optimization.

# Thanks!

Preprint available:

http://arxiv.org/abs/1610.04604