Investigating Polyhedra by Oracles

Matthias Walter

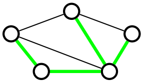
Joint work with
Volker Kaibel (Otto-von-Guericke Universität Magdeburg)

International Conference on Mathematical Software,
Berlin 2016

Operations Research
RWTH AACHEN UNIVERSITY

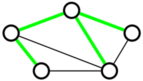
Polyhedral Method

Graph $G = (V, E)$
$T \subseteq E$ span. tree
Edge costs $c \in \mathbb{R}^E$



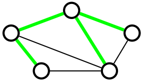
Polyhedral Method

Graph $G = (V, E)$
$T \subseteq E$ span. tree
Edge costs $c \in \mathbb{R}^E$



Polyhedral Method

Graph $G = (V, E)$
$T \subseteq E$ span. tree
Edge costs $c \in \mathbb{R}^E$

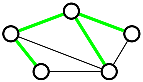


$$\chi(T)_e := \begin{cases} 1 & \text{if } e \in T \\ 0 & \text{if } e \notin T \end{cases}$$

1. Identify feasible objects with (integral) vectors in $\mathbb{R}^n$ s.t. objective is linear.

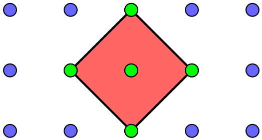
Polyhedral Method

Graph $G = (V, E)$
$T \subseteq E$ span. tree
Edge costs $c \in \mathbb{R}^E$



$$\chi(T)_e := \begin{cases} 1 & \text{if } e \in T \\ 0 & \text{if } e \notin T \end{cases}$$

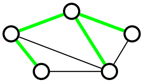
$$P_{\text{sp.tree}}(G) := \text{conv}\{\chi(T) \mid T \text{ span. tree}\}$$



1. Identify feasible objects with (integral) vectors in $\mathbb{R}^n$ s.t. objective is linear.
2. Object of interest: Convex hull of all these vectors, a polytope.

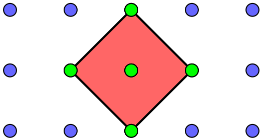
Polyhedral Method

Graph $G = (V, E)$
$T \subseteq E$ span. tree
Edge costs $c \in \mathbb{R}^E$



$$\chi(T)_e := \begin{cases} 1 & \text{if } e \in T \\ 0 & \text{if } e \notin T \end{cases}$$

$$P_{\text{sp.tree}}(G) := \text{conv}\{\chi(T) \mid T \text{ span. tree}\}$$

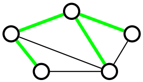


$$\{x \mid Ax \leq b\} = P_{\text{sp.tree}}(G)$$

1. Identify feasible objects with (integral) vectors in $\mathbb{R}^n$ s.t. objective is linear.
2. Object of interest: Convex hull of all these vectors, a polytope.
3. Search for outer description: inequalities & equations.

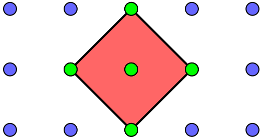
Polyhedral Method

Graph $G = (V, E)$
$T \subseteq E$ span. tree
Edge costs $c \in \mathbb{R}^E$



$$\chi(T)_e := \begin{cases} 1 & \text{if } e \in T \\ 0 & \text{if } e \notin T \end{cases}$$

$$P_{\text{sp.tree}}(G) := \text{conv}\{\chi(T) \mid T \text{ span. tree}\}$$

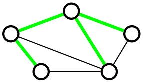


$$\{x \mid Ax \leq b\} = P_{\text{sp.tree}}(G)$$

1. Identify feasible objects with (integral) vectors in $\mathbb{R}^n$ s.t. objective is linear.
2. Object of interest: Convex hull of all these vectors, a polytope.
3. Search for outer description: inequalities & equations.
4. Solve optimization problems with LP solvers.

Polyhedral Method

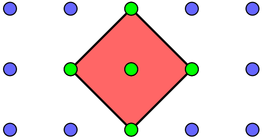
Graph $G = (V, E)$
$T \subseteq E$ span. tree
Edge costs $c \in \mathbb{R}^E$



Graph $G = (V, E)$
$T \subseteq E$ tree, $V(T)$ its nodes
Edge costs $c \in \mathbb{R}^E$, node costs $d \in \mathbb{R}^V$

$$\chi(T)_e := \begin{cases} 1 & \text{if } e \in T \\ 0 & \text{if } e \notin T \end{cases}$$

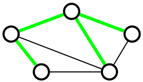
$$P_{\text{sp.tree}}(G) := \text{conv}\{\chi(T) \mid T \text{ span. tree}\}$$



$$\{x \mid Ax \le b\} = P_{\text{sp.tree}}(G)$$

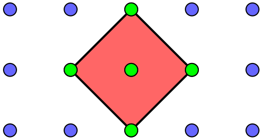
Polyhedral Method

Graph $G = (V, E)$
$T \subseteq E$ span. tree
Edge costs $c \in \mathbb{R}^E$



$$\chi(T)_e := \begin{cases} 1 & \text{if } e \in T \\ 0 & \text{if } e \notin T \end{cases}$$

$$P_{\text{sp.tree}}(G) := \text{conv}\{\chi(T) \mid T \text{ span. tree}\}$$



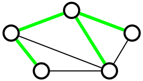
$$\{x \mid Ax \leq b\} = P_{\text{sp.tree}}(G)$$

Graph $G = (V, E)$
$T \subseteq E$ tree, $V(T)$ its nodes
Edge costs $c \in \mathbb{R}^E$, node costs $d \in \mathbb{R}^V$

1. Identify feasible objects with (integral) vectors in $\mathbb{R}^n$ s.t. objective is linear.
2. Object of interest: Convex hull of all these vectors, a polytope.
3. Search for outer description: inequalities & equations.
4. Solve optimization problems with LP solvers.

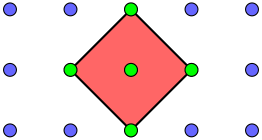
Polyhedral Method

Graph $G = (V, E)$
$T \subseteq E$ span. tree
Edge costs $c \in \mathbb{R}^E$



$$\chi(T)_e := \begin{cases} 1 & \text{if } e \in T \\ 0 & \text{if } e \notin T \end{cases}$$

$$P_{\text{sp.tree}}(G) := \text{conv}\{\chi(T) \mid T \text{ span. tree}\}$$



$$\{x \mid Ax \leq b\} = P_{\text{sp.tree}}(G)$$

Graph $G = (V, E)$
$T \subseteq E$ tree, $V(T)$ its nodes
Edge costs $c \in \mathbb{R}^E$, node costs $d \in \mathbb{R}^V$

$$(\chi(T), \chi(V(T))) \in \{0,1\}^E \times \{0,1\}^V$$

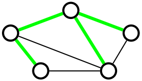
2. Object of interest: Convex hull of all these vectors, a polytope.

3. Search for outer description: inequalities & equations.

4. Solve optimization problems with LP solvers.

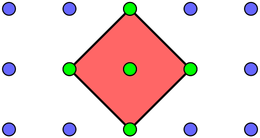
Polyhedral Method

Graph $G = (V, E)$
$T \subseteq E$ span. tree
Edge costs $c \in \mathbb{R}^E$



$$\chi(T)_e := \begin{cases} 1 & \text{if } e \in T \\ 0 & \text{if } e \notin T \end{cases}$$

$$P_{\text{sp.tree}}(G) := \text{conv}\{\chi(T) \mid T \text{ span. tree}\}$$

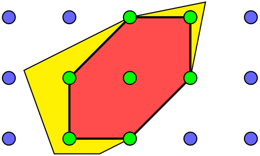


$$\{x \mid Ax \leq b\} = P_{\text{sp.tree}}(G)$$

Graph $G = (V, E)$
$T \subseteq E$ tree, $V(T)$ its nodes
Edge costs $c \in \mathbb{R}^E$, node costs $d \in \mathbb{R}^V$

$$(\chi(T), \chi(V(T))) \in \{0,1\}^E \times \{0,1\}^V$$

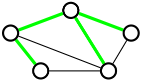
$$P_{\text{tree}}(G) := \text{conv}\{(\chi(T), \chi(V(T))) \mid T \text{ tree }\}$$



3. Search for outer description: inequalities & equations.
4. Solve optimization problems with LP solvers.

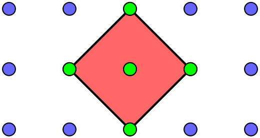
Polyhedral Method

Graph $G = (V, E)$
$T \subseteq E$ span. tree
Edge costs $c \in \mathbb{R}^E$



$$\chi(T)_e := \begin{cases} 1 & \text{if } e \in T \\ 0 & \text{if } e \notin T \end{cases}$$

$$P_{\text{sp.tree}}(G) := \text{conv}\{\chi(T) \mid T \text{ span. tree}\}$$

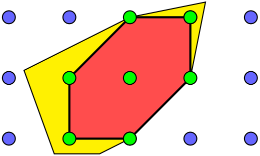


$$\{x \mid Ax \leq b\} = P_{\text{sp.tree}}(G)$$

Graph $G = (V, E)$
$T \subseteq E$ tree, $V(T)$ its nodes
Edge costs $c \in \mathbb{R}^E$, node costs $d \in \mathbb{R}^V$

$$(\chi(T), \chi(V(T))) \in \{0,1\}^E \times \{0,1\}^V$$

$$P_{\text{tree}}(G) := \text{conv}\{(\chi(T), \chi(V(T))) \mid T \text{ tree }\}$$

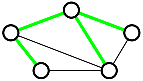


$$\{(x,y) \mid Ax + By \leq b\} \cap \mathbb{Z}^E \times \mathbb{Z}^V$$
$$= P_{\text{tree}}(G) \cap \mathbb{Z}^E \times \mathbb{Z}^V$$

4. Solve optimization problems with LP solvers.

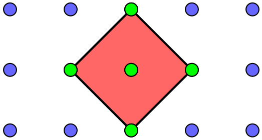
Polyhedral Method

Graph $G = (V, E)$
$T \subseteq E$ span. tree
Edge costs $c \in \mathbb{R}^E$



$$\chi(T)_e := \begin{cases} 1 & \text{if } e \in T \\ 0 & \text{if } e \notin T \end{cases}$$

$$P_{\text{sp.tree}}(G) := \text{conv}\{\chi(T) \mid T \text{ span. tree}\}$$

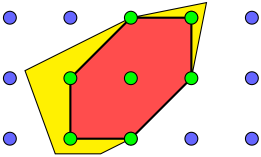


$$\{x \mid Ax \le b\} = P_{\text{sp.tree}}(G)$$

Graph $G = (V, E)$
$T \subseteq E$ tree, $V(T)$ its nodes
Edge costs $c \in \mathbb{R}^E$, node costs $d \in \mathbb{R}^V$

$$(\chi(T), \chi(V(T))) \in \{0,1\}^E \times \{0,1\}^V$$

$$P_{\text{tree}}(G) := \text{conv}\{(\chi(T), \chi(V(T))) \mid T \text{ tree }\}$$



$$\{(x,y) \mid Ax + By \le b\} \cap \mathbb{Z}^E \times \mathbb{Z}^V$$
$$= P_{\text{tree}}(G) \cap \mathbb{Z}^E \times \mathbb{Z}^V$$

4. Solve optimization problems with MILP solvers.

Hunting Facets: Traditional Approach

Mixed-integer set:
$Ax + By \le d$
$x_i \in \mathbb{Z}, y_j \in \mathbb{R}$

Recognized class of facets:

$a^\top x + b^\top y \le \beta$ for all $(a, b, \beta) : \ldots$

Hunting Facets: Traditional Approach

Mixed-integer set:
$Ax + By \leq d$
$x_i \in \mathbb{Z}, y_j \in \mathbb{R}$

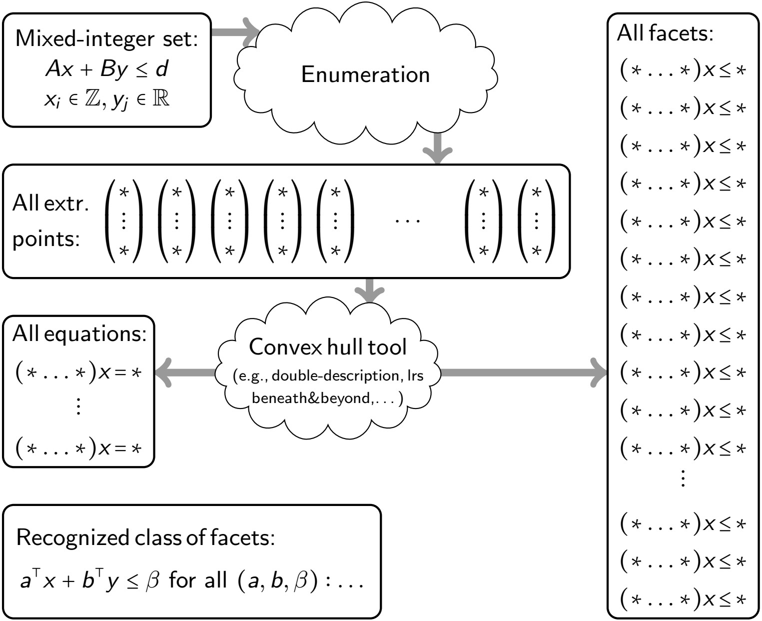
→ Enumeration ↓

All extr. points: $\begin{pmatrix} * \\ \vdots \\ * \end{pmatrix} \begin{pmatrix} * \\ \vdots \\ * \end{pmatrix} \begin{pmatrix} * \\ \vdots \\ * \end{pmatrix} \begin{pmatrix} * \\ \vdots \\ * \end{pmatrix} \begin{pmatrix} * \\ \vdots \\ * \end{pmatrix} \cdots \begin{pmatrix} * \\ \vdots \\ * \end{pmatrix} \begin{pmatrix} * \\ \vdots \\ * \end{pmatrix}$

Recognized class of facets:

$a^\top x + b^\top y \leq \beta$ for all $(a, b, \beta) : \ldots$

Hunting Facets: Traditional Approach



Hunting Facets: Traditional Approach

Mixed-integer set:
$Ax + By \leq d$
$x_i \in \mathbb{Z}, y_j \in \mathbb{R}$

→ Enumeration

All extr. points: $\begin{pmatrix} * \\ \vdots \\ * \end{pmatrix} \begin{pmatrix} * \\ \vdots \\ * \end{pmatrix} \begin{pmatrix} * \\ \vdots \\ * \end{pmatrix} \begin{pmatrix} * \\ \vdots \\ * \end{pmatrix} \begin{pmatrix} * \\ \vdots \\ * \end{pmatrix} \quad \cdots \quad \begin{pmatrix} * \\ \vdots \\ * \end{pmatrix} \begin{pmatrix} * \\ \vdots \\ * \end{pmatrix}$

→ Convex hull tool
(e.g., double-description, lrs beneath&beyond, ...)

All equations:
$(* \ldots *)x = *$
$\vdots$
$(* \ldots *)x = *$

All facets:
$(* \ldots *)x \leq *$
$(* \ldots *)x \leq *$
$(* \ldots *)x \leq *$
$(* \ldots *)x \leq *$
$(* \ldots *)x \leq *$
$(* \ldots *)x \leq *$
$(* \ldots *)x \leq *$
$(* \ldots *)x \leq *$
$(* \ldots *)x \leq *$
$(* \ldots *)x \leq *$
$(* \ldots *)x \leq *$
$\vdots$
$(* \ldots *)x \leq *$
$(* \ldots *)x \leq *$
$(* \ldots *)x \leq *$

Recognized class of facets:
$a^\top x + b^\top y \leq \beta$ for all $(a, b, \beta) : \ldots$

Hunting Facets: New Approach

Mixed-integer set:

$$Ax + By \leq d$$
$$x_i \in \mathbb{Z}, y_j \in \mathbb{R}$$

Recognized class of facets:

$$a^\top x + b^\top y \leq \beta \text{ for all } (a, b, \beta) : \ldots$$

Hunting Facets: New Approach

Mixed-integer set:

$$Ax + By \leq d$$
$$x_i \in \mathbb{Z}, y_j \in \mathbb{R}$$

All equations:

$$(* \dots *)x = *$$
$$\vdots$$
$$(* \dots *)x = *$$

Recognized class of facets:

$$a^\top x + b^\top y \leq \beta \text{ for all } (a, b, \beta) : \dots$$

Hunting Facets: New Approach

Mixed-integer set:

$$Ax + By \le d$$
$$x_i \in \mathbb{Z}, y_j \in \mathbb{R}$$

All equations:

$$(* \ldots *)x = *$$
$$\vdots$$
$$(* \ldots *)x = *$$

Recognized class of facets:

$a^\top x + b^\top y \le \beta$ for all $(a, b, \beta) : \ldots$

Only some useful facets:

$$(* \ldots *)x \le *$$
$$(* \ldots *)x \le *$$
$$\vdots$$
$$(* \ldots *)x \le *$$
$$(* \ldots *)x \le *$$

Hunting Facets: New Approach

Mixed-integer set:

$$Ax + By \leq d$$
$$x_i \in \mathbb{Z}, y_j \in \mathbb{R}$$

MIP solver

All equations:

$$(* \dots *)x = *$$
$$\vdots$$
$$(* \dots *)x = *$$

Recognized class of facets:

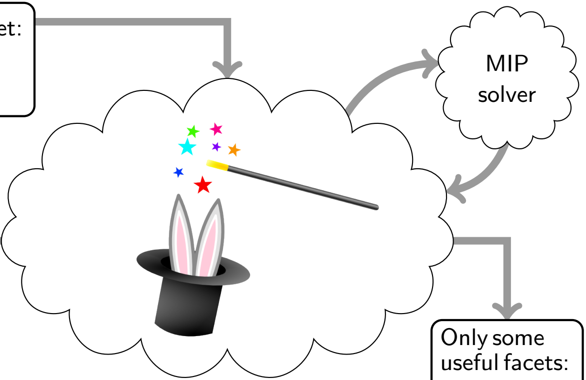
$$a^\top x + b^\top y \leq \beta \text{ for all } (a, b, \beta) : \dots$$

Only some useful facets:

$$(* \dots *)x \leq *$$
$$(* \dots *)x \leq *$$
$$\vdots$$
$$(* \dots *)x \leq *$$
$$(* \dots *)x \leq *$$

Hunting Facets: New Approach

Mixed-integer set:
$Ax + By \leq d$
$x_i \in \mathbb{Z}, y_j \in \mathbb{R}$



All equations:
$(* \dots *)x = *$
$\vdots$
$(* \dots *)x = *$

Only some useful facets:
$(* \dots *)x \leq *$
$(* \dots *)x \leq *$
$\vdots$
$(* \dots *)x \leq *$
$(* \dots *)x \leq *$

Recognized class of facets:
$a^\top x + b^\top y \leq \beta$ for all $(a, b, \beta) : \dots$

Agenda

1. Introduction
   - Polyhedral Method
   - Facet Hunting
2. IPO
   - Oracles
   - Capabilities
   - Details: Facets
   - Details: Affine Hull
3. Studies
   - Matching Polytopes with One Quadratic Term
   - MIPLIB Dimensions
   - Constraint Dimensions
   - TSP Polytopes

The IPO Framework

Oracle Zoo

Base oracles:

- Instance & MIP solver $\rightsquigarrow$ oracle for (mixed-) integer hull.
- Instance & LP solver $\rightsquigarrow$ oracle for LP relaxation.

Oracle Zoo

Base oracles:

- Instance & MIP solver $\rightsquigarrow$ oracle for (mixed-) integer hull.
- Instance & LP solver $\rightsquigarrow$ oracle for LP relaxation.

Auxiliary oracles:

- Caching of oracle answers.
- Heuristics: Feasibility of returned solutions may be sufficient for the algorithms to make progress. $\rightsquigarrow$ speed-up!

Oracle Zoo

Base oracles:

- Instance & MIP solver $\rightsquigarrow$ oracle for (mixed-) integer hull.
- Instance & LP solver $\rightsquigarrow$ oracle for LP relaxation.

Auxiliary oracles:

- Caching of oracle answers.
- Heuristics: Feasibility of returned solutions may be sufficient for the algorithms to make progress. $\rightsquigarrow$ speed-up!

Oracles for related polytopes:

- Restriction to face.
- Affine projection.
- Recession cone.

Capabilities

Facets:

- Given a point $\hat{x}$, compute a facet-defining inequality $a^\top x \le \beta$ of P that is violated by $\hat{x}$.
- Given an objective $c \in \mathbb{R}^n$, compute facet-defining inequalities until optimization of the LP results in an integer point.

Capabilities

Facets:

- Given a point $\hat{x}$, compute a facet-defining inequality $a^\top x \leq \beta$ of P that is violated by $\hat{x}$.
- Given an objective $c \in \mathbb{R}^n$, compute facet-defining inequalities until optimization of the LP results in an integer point.

Affine hull:

- Compute the dimension d of P.
- Find a system of $n - d$ (independent) equations $a^\top x = \beta$ valid for P.
- Find a set of $d + 1$ affinely independent points in P.

Capabilities

Facets:

- Given a point $\hat{x}$, compute a facet-defining inequality $a^\intercal x \leq \beta$ of P that is violated by $\hat{x}$.
- Given an objective $c \in \mathbb{R}^n$, compute facet-defining inequalities until optimization of the LP results in an integer point.

Affine hull:

- Compute the dimension d of P.
- Find a system of $n - d$ (independent) equations $a^\intercal x = \beta$ valid for P.
- Find a set of $d + 1$ affinely independent points in P.

Smallest Face:

- Compute the smallest face that contains a point $\hat{x} \in P$.
- Is $\hat{x}$ a vertex of P?
- Are the vertices u and v of P connected by an edge of P?

Details: Facets

- Let $d := \dim P$ and let $o \in \text{relint}(P)$.
- $\hat{x}$ is to be separated by a facet $\langle a, x \rangle \leq \beta$.
- $S \subseteq P$ contains P's vertices and $R \subseteq \text{recc}(P)$ contains all extreme rays.

Details: Facets

- Let $d := \dim P$ and let $o \in \text{relint}(P)$.
- $\hat{x}$ is to be separated by a facet $\langle a, x \rangle \leq \beta$.
- $S \subseteq P$ contains P's vertices and $R \subseteq \text{recc}(P)$ contains all extreme rays.
- We solve the following LP, adding constraints for S and R lazily.

$$\begin{aligned}
\max \quad & \langle \hat{x}, a \rangle - \beta & & \\
\text{s.t.} \quad & \langle s, a \rangle - \beta \leq 0 & & \text{for all } s \in S \\
& \langle r, a \rangle \leq 0 & & \text{for all } r \in R \\
& \langle \hat{x} - o, a \rangle \leq 1 & & \\
& a \in \mathbb{R}^n, \ \beta \in \mathbb{R} & &
\end{aligned}$$

Details: Facets

- Let $d := \dim P$ and let $o \in \operatorname{relint}(P)$.
- $\hat{x}$ is to be separated by a facet $\langle a, x \rangle \leq \beta$.
- $S \subseteq P$ contains P's vertices and $R \subseteq \operatorname{recc}(P)$ contains all extreme rays.
- We solve the following LP, adding constraints for S and R lazily.

$$\begin{aligned} \max \quad & \langle \hat{x}, a \rangle - \beta & & \\ \text{s.t.} \quad & \langle s, a \rangle - \beta \leq 0 & & \text{for all } s \in S \\ & \langle r, a \rangle \leq 0 & & \text{for all } r \in R \\ & \langle \hat{x} - o, a \rangle \leq 1 & & \\ & a \in \mathbb{R}^n, \quad \beta \in \mathbb{R} & & \end{aligned}$$

Theorem (W., 2016)

Let (a^*, β^*) be an optimum that lies in a minimal face of the feasible set. If $\langle a^*, o \rangle < \beta^*$, then $\langle a^*, x \rangle \leq \beta^*$ is valid and facet-defining for P. Otherwise, $\langle a^*, x \rangle = \beta^*$ is valid for P.

Details: Affine Hull

Input:

- Oracle optimizing any rational objective over P.

Output:

- Dimension d of P
- $(d+1)$-many affinely independent points in P
- $(n-d)$-many irredundant equations $Cx = d$ valid for P

Details: Affine Hull

Input:

- Oracle optimizing any rational objective over P.

Output:

- Dimension d of P
- $(d+1)$-many affinely independent points in P
- $(n-d)$-many irredundant equations $Cx = d$ valid for P

Ideas:

1. Maintain known equations $Cx = d$ and points $x_1, x_2, \dots, x_\ell \in P$.

Details: Affine Hull

Input:

- Oracle optimizing any rational objective over P.

Output:

- Dimension d of P
- $(d+1)$-many affinely independent points in P
- $(n-d)$-many irredundant equations $Cx = d$ valid for P

Ideas:

1. Maintain known equations $Cx = d$ and points $x_1, x_2, \dots, x_\ell \in P$.
2. Repeatedly find a "useful" direction $c \in \mathbb{R}^n$, and compute $z^+ := \max_{x \in P} \langle c, x \rangle$ and $z^- := \min_{x \in P} \langle c, x \rangle$ **(2 oracle calls)**.

Details: Affine Hull

Input:

- Oracle optimizing any rational objective over P.

Output:

- Dimension d of P
- $(d+1)$-many affinely independent points in P
- $(n-d)$-many irredundant equations $Cx = d$ valid for P

Ideas:

1. Maintain known equations $Cx = d$ and points $x_1, x_2, \dots, x_\ell \in P$.
2. Repeatedly find a "useful" direction $c \in \mathbb{R}^n$, and compute $z^+ := \max_{x \in P} \langle c, x \rangle$ and $z^- := \min_{x \in P} \langle c, x \rangle$ **(2 oracle calls)**.
3. If we are lucky and $z^+ = z^-$ holds, then $\langle c, x \rangle = z^+$ is a valid equation. To make progress, we want c to be linear independent of C's rows.

Details: Affine Hull

Input:

- Oracle optimizing any rational objective over P.

Output:

- Dimension d of P
- $(d+1)$-many affinely independent points in P
- $(n-d)$-many irredundant equations $Cx = d$ valid for P

Ideas:

1. Maintain known equations $Cx = d$ and points $x_1, x_2, \dots, x_\ell \in P$.
2. Repeatedly find a "useful" direction $c \in \mathbb{R}^n$, and compute $z^+ := \max_{x \in P} \langle c, x \rangle$ and $z^- := \min_{x \in P} \langle c, x \rangle$ **(2 oracle calls)**.
3. If we are lucky and $z^+ = z^-$ holds, then $\langle c, x \rangle = z^+$ is a valid equation. To make progress, we want c to be linear independent of C's rows.
4. If we also choose c to be orthogonal to $\text{aff}(x_1, \dots, x_\ell)$ (that is, $\langle c, x_i \rangle = \langle c, x_1 \rangle$ for all $i = 2, \dots, \ell$), the result $z^+ > z^-$ yields a point $x_{\ell+1}$ affinely independent of $x_1, \dots, x_\ell$.

Computing the Affine Hull

Reminder:

- $x_1, \dots, x_\ell \in P \subseteq \mathbb{R}^n$ are affinely independent points found so far.
- $Cx = d$ are the equations found so far.

Computing the Affine Hull

Reminder:

- $x_1, \ldots, x_\ell \in P \subseteq \mathbb{R}^n$ are affinely independent points found so far.
- $Cx = d$ are the equations found so far.

Hidden details:

- How to model linear independence? Orthogonality is a bad choice for exact arithmetic!

Computing the Affine Hull

Reminder:

- $x_1, \ldots, x_\ell \in P \subseteq \mathbb{R}^n$ are affinely independent points found so far.
- $Cx = d$ are the equations found so far.

Hidden details:

- How to model linear independence? Orthogonality is a bad choice for exact arithmetic!
- Better: Try basis vectors of $\text{aff}\,(x_1, \ldots, x_\ell)^\perp$ and skip if it depends on C's rows.

Computing the Affine Hull

Reminder:

- $x_1, \ldots, x_\ell \in P \subseteq \mathbb{R}^n$ are affinely independent points found so far.
- $Cx = d$ are the equations found so far.

Hidden details:

- How to model linear independence? Orthogonality is a bad choice for exact arithmetic!
- Better: Try basis vectors of $\operatorname{aff}(x_1, \ldots, x_\ell)^\perp$ and skip if it depends on C's rows.
- Even better: Compute whole basis approximately to estimate sparsity and then try (exactly computed) sparse candidates.

Computing the Affine Hull

Reminder:

- $x_1, \ldots, x_\ell \in P \subseteq \mathbb{R}^n$ are affinely independent points found so far.
- $Cx = d$ are the equations found so far.

Hidden details:

- How to model linear independence? Orthogonality is a bad choice for exact arithmetic!
- Better: Try basis vectors of $\operatorname{aff}(x_1, \ldots, x_\ell)^\perp$ and skip if it depends on C's rows.
- Even better: Compute whole basis approximately to estimate sparsity and then try (exactly computed) sparse candidates.
- What if P is unbounded? Have to consider unbounded rays as well!

Computing the Affine Hull

Reminder:

- $x_1, \ldots, x_\ell \in P \subseteq \mathbb{R}^n$ are affinely independent points found so far.
- $Cx = d$ are the equations found so far.

Hidden details:

- How to model linear independence? Orthogonality is a bad choice for exact arithmetic!
- Better: Try basis vectors of $\operatorname{aff}(x_1, \ldots, x_\ell)^\perp$ and skip if it depends on C's rows.
- Even better: Compute whole basis approximately to estimate sparsity and then try (exactly computed) sparse candidates.
- What if P is unbounded? Have to consider unbounded rays as well!
- Naïve way: $2n + 1$ oracle calls.

Computing the Affine Hull

Reminder:

- $x_1, \ldots, x_\ell \in P \subseteq \mathbb{R}^n$ are affinely independent points found so far.
- $Cx = d$ are the equations found so far.

Hidden details:

- How to model linear independence? Orthogonality is a bad choice for exact arithmetic!
- Better: Try basis vectors of $\text{aff}\,(x_1, \ldots, x_\ell)^\perp$ and skip if it depends on C's rows.
- Even better: Compute whole basis approximately to estimate sparsity and then try (exactly computed) sparse candidates.
- What if P is unbounded? Have to consider unbounded rays as well!
- Naïve way: $2n + 1$ oracle calls.
- With some more tricks: $2n$ oracle calls. $\leftarrow$ great result!

Theorem (W., 2016)

Every algorithm which computes the affine hull of polyhedra $P \subseteq \mathbb{R}^n$ specified only by an optimization oracle needs at least $2n$ oracle calls in the worst case.

Computational Studies

Study: Quadratic Matching Polytopes

Consider the quadratic matching polytope of order n with one quadratic term:

$$P_n := \operatorname{conv}\left\{ (\chi(M), y) \in \{0,1\}^{|E_n|+1} \mid M \text{ matching in } K_n,\ y = x_{1,2}x_{3,4} \right\}$$

Study: Quadratic Matching Polytopes

Consider the quadratic matching polytope of order n with one quadratic term:

$$P_n := \text{conv}\left\{(\chi(M), y) \in \{0,1\}^{|E_n|+1} \mid M \text{ matching in } K_n,\ y = x_{1,2}x_{3,4}\ \right\}$$

Hupp, Klein & Liers, '15 obtained a bunch of facets:

- $x(\delta(v)) \leq 1$ for all $v \in V_n$.
- $x_e \geq 0$ for all $e \in E_n$.
- $y \leq x_{1,2}$ and $y \leq x_{3,4}$. (Note that $y \geq x_{1,2} + x_{3,4} - 1$ is no facet.)

Study: Quadratic Matching Polytopes

Consider the quadratic matching polytope of order n with one quadratic term:

$$P_n := \operatorname{conv}\left\{(\chi(M), y) \in \{0,1\}^{|E_n|+1} \mid M \text{ matching in } K_n,\ y = x_{1,2}x_{3,4}\right\}$$

Hupp, Klein & Liers, '15 obtained a bunch of facets:

- $x(\delta(v)) \leq 1$ for all $v \in V_n$.
- $x_e \geq 0$ for all $e \in E_n$.
- $y \leq x_{1,2}$ and $y \leq x_{3,4}$. (Note that $y \geq x_{1,2} + x_{3,4} - 1$ is no facet.)
- $x(E[S]) + y \leq \frac{|S|-1}{2}$ for certain odd S.
- $x(E[S]) \leq \frac{|S|-1}{2}$ for certain odd S.
- $x(E[S]) + x(E[S \setminus \{1,2\}]) + x_{3,4} - y \leq |S| - 2$ for certain odd S.
- $x(E[S]) + x_{2,a} + x_{3,a} + x_{4,a} + y \leq \frac{|S|}{2}$ for certain even S and nodes a.
- $x_{1,2} + x_{1,a} + x_{2,a} + x(E[S]) + x_{3,4} + x_{3,b} + x_{4,b} - y \leq \frac{|S|}{2} + 1$ for certain even S and certain nodes a, b.

Study: Some are Missing!

Excerpt from their paper:

- $x_{\mathring{u}\mathring{v}} = x_{\mathring{w}\mathring{z}} = 0.5, y = 0.3,$
- $x_{\mathring{u}\mathring{w}} = x_{\mathring{v}\mathring{z}} = 0.3,$
- $x_{\mathring{u}a} = x_{\mathring{v}a} = x_{\mathring{w}b} = x_{\mathring{z}b} = 0.2,$
- $x_e = 0$ otherwise.

This fractional solution satisfies all introduced valid and face
the nonnegativity and the linearisation constraints, all blosso
inequalities (4) and the hourglass inequalities (5).

Study: Some are Missing!

Excerpt from their paper:

- $x_{\mathring{u}\mathring{v}} = x_{\mathring{w}\mathring{z}} = 0.5, y = 0.3,$
- $x_{\mathring{u}\mathring{w}} = x_{\mathring{v}\mathring{z}} = 0.3,$
- $x_{\mathring{u}a} = x_{\mathring{v}a} = x_{\mathring{w}b} = x_{\mathring{z}b} = 0.2,$
- $x_e = 0$ otherwise.

This fractional solution satisfies all introduced valid and face
the nonnegativity and the linearisation constraints, all blosso
inequalities (4) and the hourglass inequalities (5).

Study: An IP Model

```
param n := 6;

set V := { 1 to n };
set E := { <u,v> in V*V with u < v };
set F := { <1,2>,<3,4>,<1,5>,<2,5>,<3,6>,<4,6>,<1,3>,<2,4> };

var x[E] binary;
var y binary;

maximize weights:
  10*x[1,2] + 10*x[3,4] + 2*x[1,5] + 2*x[2,5] + 2*x[3,6]
  + 2*x[4,6] + 4*x[1,3] + 4*x[2,4] -10*y
  + sum <u,v> in E-F: -1000*x[u,v];

subto degree:   forall <w> in V:
    (sum <u,v> in E with u == w or v == w: x[u,v]) <= 1;
subto product1: y <= x[1,2];
subto product2: y <= x[3,4];
subto product3: y >= x[1,2] + x[3,4] - 1;
```

Study: Running IPO

```
% ipo --dimension --facets product-matching-missing.zpl
Computing the affine hull:
 Dimension: 29
Objective <instance> 10 x#1#2 + 4 x#1#3 ...
 Facet: 2 x#1#2 + x#1#3 + x#1#4 + x#1#5 + x#2#3 + x#2#4 + x#2#5
       + x#3#4 - y <= 2

  Certifying point: (y=1, x#5#6=1, x#3#4=1, x#1#2=1)
  Certifying point: (y=1, x#5#8=1, x#3#4=1, x#1#2=1)
  Certifying point: (y=1, x#6#7=1, x#3#4=1, x#1#2=1)
  Certifying point: (x#2#5=1, x#1#3=1)
  Certifying point: (x#3#4=1, x#2#5=1)
...
```

Study: Running IPO

```
% ipo --dimension --facets product-matching-missing.zpl
Computing the affine hull:
 Dimension: 29
Objective <instance> 10 x#1#2 + 4 x#1#3 ...
 Facet: 2 x#1#2 + x#1#3 + x#1#4 + x#1#5 + x#2#3 + x#2#4 + x#2#5
        + x#3#4 - y <= 2

  Certifying point: (y=1, x#5#6=1, x#3#4=1, x#1#2=1)
  Certifying point: (y=1, x#5#8=1, x#3#4=1, x#1#2=1)
  Certifying point: (y=1, x#6#7=1, x#3#4=1, x#1#2=1)
  Certifying point: (x#2#5=1, x#1#3=1)
  Certifying point: (x#3#4=1, x#2#5=1)
...
```

This facet does not belong to the previous types!

Study: MIPLIB 2 Dimensions

Oracles

- Oracle: SCIP-3.0.0-ex
- Heuristic: SCIP-3.1.1 with postprocessing

Study: MIPLIB 2 Dimensions

Oracles

- Oracle: SCIP-3.0.0-ex
- Heuristic: SCIP-3.1.1 with postprocessing

Postprocessing of solutions

Let $I \subseteq [n]$ be the set of integral variables.

1. For $x \in \mathbb{Q}^n$, obtain $\overline{x}$ from x by rounding x_i for all $i \in I$.
2. Compute optimal choice for $x_{[n] \setminus I}$ using an exact LP solver, e.g., SoPlex.

Study: MIPLIB 2 Dimensions

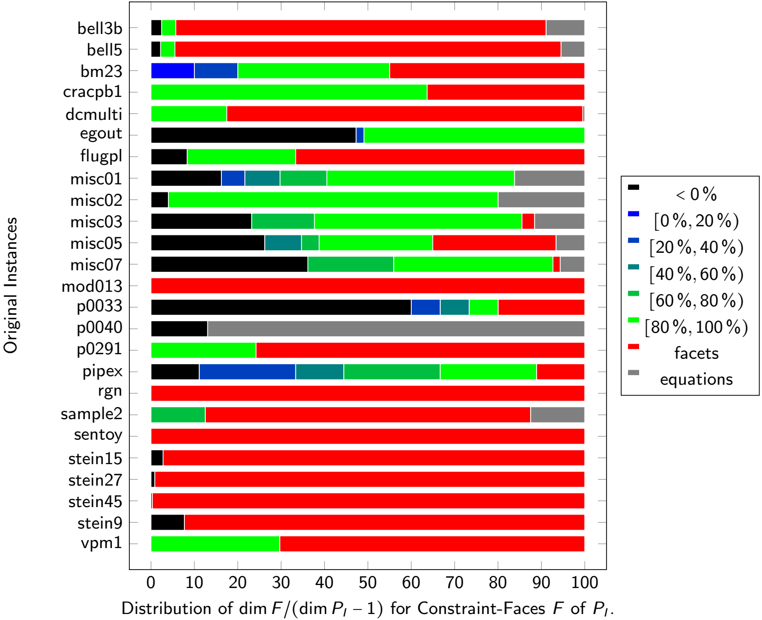
For the 34 instances with $n \leq 1000$ solved within 5 mins, we considered the original P, and the presolved instances Q, and their resp. hulls P_I and Q_I:

Instance	n	n'	dim P	dim P_I
air01	771	750	732	617
bell3b	133	133	133	115
bell5	104	104	104	97
bm23	27	27	27	27
cracpb1	572	484	484	478
dcmulti	548	470	470	467
diamond	2	2	2	-1
egout	141	68	68	41
enigma	100	79	79	3
flugpl	18	12	12	9
gen	870	720	720	540
lseu	89	89	89	89
misc01	83	68	60	44
misc02	59	47	41	37
misc03	160	136	121	116
misc05	136	108	100	98
misc07	260	228	207	204
p0033	33	33	33	27
p0040	40	40	30	30
p0201	201	201	145	139
p0548	548	548	545	
pipex	48	32	32	31
rgn	180	160	160	160
sample2	67	44	44	32
vpm1	378	336	288	288

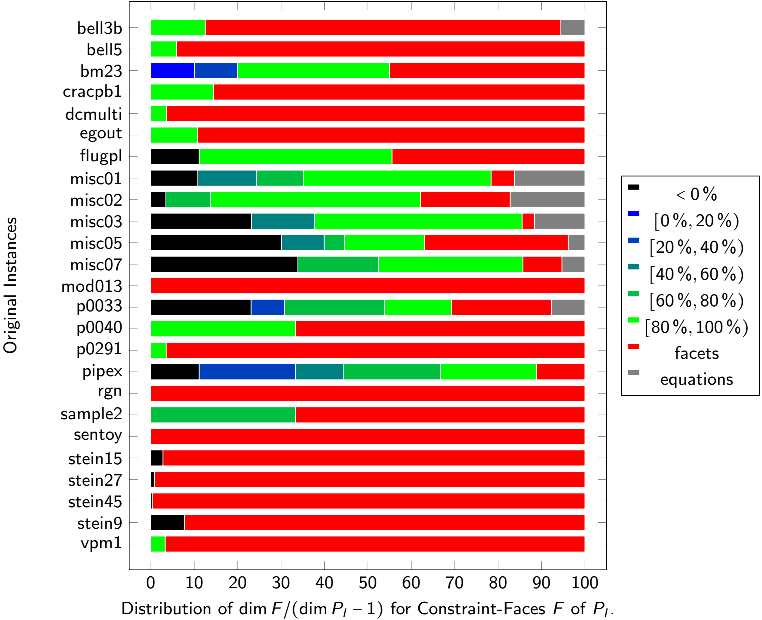
Instance	n	n'	dim Q	dim Q_I
air01	760	363	363	361
bell3b	113	91	91	86
bell5	87	56	56	56
bm23	27	27	27	27
cracpb1	518	478	478	478
dcmulti	548	469	469	467
diamond	2	0	-1	−1
egout	118	41	41	41
enigma	100	79	79	3
flugpl	16	10	10	7
gen	699	509	411	411
lseu	89	85	85	85
misc01	82	56	56	44
misc02	58	37	37	33
misc03	159	115	115	110
misc05	128	100	100	98
misc07	259	201	201	198
p0033	29	26	26	20
p0040	40	20	20	20
p0201	201	163	127	127
p0548	527	362	362	357
pipex	48	32	32	31
rgn	175	160	160	160
sample2	55	32	32	32
vpm1	362	168	168	168

Full-dim. instances: `mod008`, `mod013`, `p0282`, `p0291`, `sentoy`, `stein15`, `stein27`, `stein45`, `stein9`

Study: Constraint Dimensions (Original)



Study: Constraint Dimensions (Presolved)



Study: TSP Polytopes

Oracles

- Oracle: Concorde (famous TSP solver)
- Heuristic: Nearest neighbor plus 2-opt, searching once from each node.

Study: TSP Polytopes

Oracles

- Oracle: Concorde (famous TSP solver)
- Heuristic: Nearest neighbor plus 2-opt, searching once from each node.

Results for $10,000$ **random tests:**

Nodes	Adjacent	Time/pair	LP	Heuristics	Oracles	Cache	Tours	Vertices
5	91.23 %	0.3 s	0.5 %	0.1 %	97.9 %	0.1 %	12	$1.2 \cdot 10^1$
6	69.32 %	0.4 s	0.7 %	0.1 %	97.5 %	0.1 %	45	$6.0 \cdot 10^1$
7	46.16 %	0.6 s	1.1 %	0.1 %	96.1 %	0.5 %	207	$3.6 \cdot 10^2$
8	28.07 %	0.8 s	1.5 %	0.1 %	93.1 %	2.5 %	1,189	$2.5 \cdot 10^3$
9	17.46 %	1.0 s	2.0 %	0.2 %	86.1 %	8.7 %	5,759	$2.0 \cdot 10^4$
10	10.52 %	1.5 s	2.3 %	0.2 %	77.5 %	17.3 %	15,472	$1.8 \cdot 10^5$
11	6.53 %	2.1 s	2.7 %	0.2 %	67.4 %	26.9 %	33,935	$1.8 \cdot 10^6$
12	3.67 %	3.0 s	3.8 %	0.3 %	54.2 %	38.8 %	66,510	$2.0 \cdot 10^7$
13	2.20 %	4.9 s	5.1 %	0.3 %	39.6 %	52.0 %	125,298	$2.4 \cdot 10^8$
14	1.13 %	10.1 s	7.7 %	0.3 %	22.9 %	65.8 %	232,995	$3.1 \cdot 10^9$
15	0.59 %	24.3 s	12.9 %	0.2 %	11.0 %	71.6 %	406,315	$4.4 \cdot 10^{10}$

Summary

Tool:

- Approach allows to handle much higher dimensions than usual convex hull codes.
- But: not useful for checking whether formulation is complete.
- Exact LP solver and exactly represented points / rays are essential.
- For $n > 100$, exact MIP solver is also essential.
- For $n > 500$, linear algebra starts to be time-consuming.

Summary

Tool:

- Approach allows to handle much higher dimensions than usual convex hull codes.
- But: not useful for checking whether formulation is complete.
- Exact LP solver and exactly represented points / rays are essential.
- For $n > 100$, exact MIP solver is also essential.
- For $n > 500$, linear algebra starts to be time-consuming.

Computational Study:

- Some MIPLIB 2.0 models seem to be not so well-posed.
- Larger problems / concrete models would be interesting to check.

Summary

Tool:

- Approach allows to handle much higher dimensions than usual convex hull codes.
- But: not useful for checking whether formulation is complete.
- Exact LP solver and exactly represented points / rays are essential.
- For $n > 100$, exact MIP solver is also essential.
- For $n > 500$, linear algebra starts to be time-consuming.

Computational Study:

- Some MIPLIB 2.0 models seem to be not so well-posed.
- Larger problems / concrete models would be interesting to check.

polyhedra-oracles.bitbucket.org