Primal and dual approaches for the enumeration of hyperplane arrangements

Baptiste Plaquevent-Jourdain, with
Jean-Pierre Dussault, Université de Sherbrooke
Jean Charles Gilbert, INRIA Paris

October, 9th 2025

Outline

Hyperplanes

Hyperplane H := affine (linear) subspace of dimension $n-1$ in $\mathbb{R}^n$.
For $v \in \mathbb{R}^n$ and $t \in \mathbb{R}$, $H_{(v,t)} := \{x \in \mathbb{R}^n : v^\top x = \sum_{i=1}^n x_i v_i = t\}$.

Hyperplanes

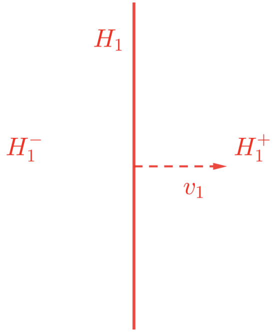
Hyperplane H := affine (linear) subspace of dimension $n-1$ in $\mathbb{R}^n$.
For $v \in \mathbb{R}^n$ and $t \in \mathbb{R}$, $H_{(v,t)} := \{x \in \mathbb{R}^n : v^\mathsf{T} x = \sum_{i=1}^n x_i v_i = t\}$.

$$H^+ := \{x \in \mathbb{R}^n : v^\mathsf{T} x > t\} = \{x \in \mathbb{R}^n : +(v^\mathsf{T} x - t) > 0\},$$

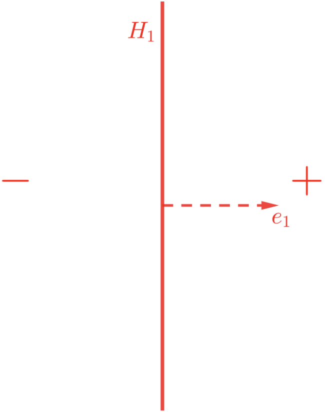
Hyperplanes

Hyperplane H := affine (linear) subspace of dimension $n-1$ in $\mathbb{R}^n$.
For $v \in \mathbb{R}^n$ and $t \in \mathbb{R}$, $H_{(v,t)} := \{x \in \mathbb{R}^n : v^\top x = \sum_{i=1}^n x_i v_i = t\}$.

$$H^+ := \{x \in \mathbb{R}^n : v^\top x > t\} = \{x \in \mathbb{R}^n : +(v^\top x - t) > 0\},$$
$$H^- := \{x \in \mathbb{R}^n : v^\top x < t\} = \{x \in \mathbb{R}^n : -(v^\top x - t) > 0\}.$$

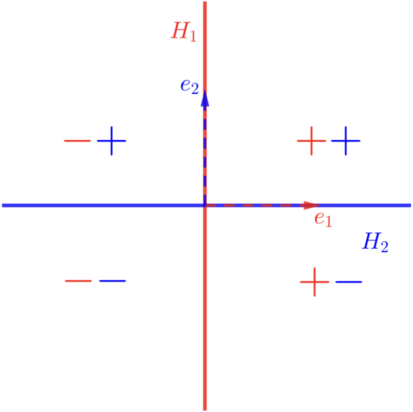


Several hyperplanes: geometric aspect



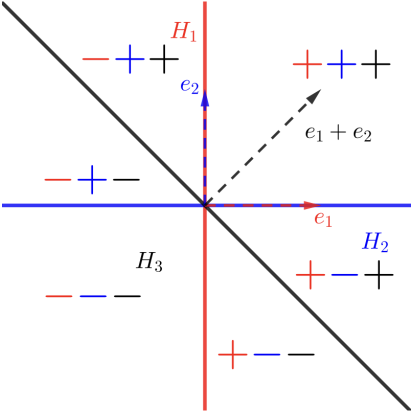
Example with a few hyperplanes and the *signs* of the halfspaces.

Several hyperplanes: geometric aspect



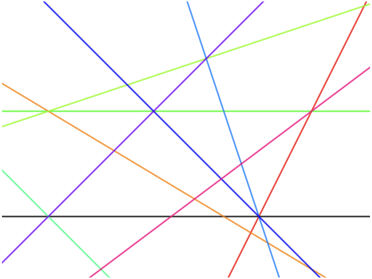
Example with a few hyperplanes and the *signs* of the halfspaces.

Several hyperplanes: geometric aspect



Example with a few hyperplanes and the *signs* of the halfspaces. The combinations of signs are **chambers** of the arrangement.

Less trivial example



More chaotic arrangement in dimension 2.

Already studied in the 19th century [Ste26; Rob87; Sch50].

Notation

Dimension $n \in \mathbb{N}^*$, $p \in \mathbb{N}^*$ hyperplanes, $v_i \in \mathbb{R}^n, \tau_i \in \mathbb{R}$ $1 \leqslant i \leqslant p$.

$$H_i := \{x \in \mathbb{R}^n : v_i^\mathsf{T} x = \tau_i\}, \quad V = [v_1 \ \ldots \ v_p], \quad \tau = [\tau_1; \ldots; \tau_p]$$

Notation

Dimension $n \in \mathbb{N}^*$, $p \in \mathbb{N}^*$ hyperplanes, $v_i \in \mathbb{R}^n, \tau_i \in \mathbb{R}$ $1 \leqslant i \leqslant p$.

$$H_i := \{x \in \mathbb{R}^n : v_i^\mathsf{T} x = \tau_i\}, \quad V = [v_1 \; \ldots \; v_p], \quad \tau = [\tau_1; \ldots; \tau_p]$$
$$H_i^+ := \{x : +(v_i^\mathsf{T} x - \tau_i) > 0\}, H_i^- := \{x \in \mathbb{R}^n : -(v_i^\mathsf{T} x - \tau_i) > 0\}$$

Chambers: **subset** of the $\bigcap_{i=1}^{p} (H_i^+ \text{ or } H_i^-)$, the nonempty ones.

Notation

Dimension $n \in \mathbb{N}^*$, $p \in \mathbb{N}^*$ hyperplanes, $v_i \in \mathbb{R}^n, \tau_i \in \mathbb{R}$ $1 \leqslant i \leqslant p$.

$$H_i := \{x \in \mathbb{R}^n : v_i^{\mathsf{T}} x = \tau_i\}, \quad V = [v_1 \ \ldots \ v_p], \quad \tau = [\tau_1; \ldots; \tau_p]$$
$$H_i^+ := \{x : +(v_i^{\mathsf{T}} x - \tau_i) > 0\}, H_i^- := \{x \in \mathbb{R}^n : -(v_i^{\mathsf{T}} x - \tau_i) > 0\}$$

Chambers: **subset** of the $\bigcap_{i=1}^{p} (H_i^+ \text{ or } H_i^-)$, the nonempty ones.

Geometric to analytic: sign vectors

$$\text{find } \mathcal{S}(V, \tau) := \{s = (s_1, \ldots, s_p) \in \{\pm 1\}^p,$$
$$\text{s.t. } \exists\ x^s \in \mathbb{R}^n, \quad \forall\ i \in [1:p], \quad s_i(v_i^{\mathsf{T}} x^s - \tau_i) > 0\}.$$

Notation

Dimension $n \in \mathbb{N}^*$, $p \in \mathbb{N}^*$ hyperplanes, $v_i \in \mathbb{R}^n, \tau_i \in \mathbb{R}$ $1 \leqslant i \leqslant p$.

$$H_i := \{x \in \mathbb{R}^n : v_i^\mathsf{T} x = \tau_i\}, \quad V = [v_1 \ \ldots \ v_p], \quad \tau = [\tau_1; \ldots; \tau_p]$$
$$H_i^+ := \{x : +(v_i^\mathsf{T} x - \tau_i) > 0\}, H_i^- := \{x \in \mathbb{R}^n : -(v_i^\mathsf{T} x - \tau_i) > 0\}$$

Chambers: **subset** of the $\bigcap_{i=1}^p (H_i^+ \text{ or } H_i^-)$, the nonempty ones.

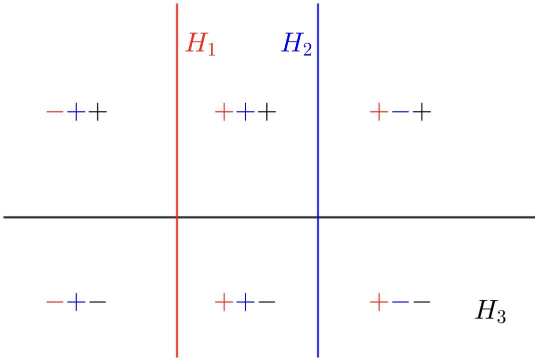
Geometric to analytic: sign vectors

$$\text{find } \mathcal{S}(V,\tau) := \{s = (s_1, \ldots, s_p) \in \{\pm 1\}^p,$$
$$\text{s.t. } \exists\ x^s \in \mathbb{R}^n, \quad \forall\ i \in [1:p], \quad s_i(v_i^\mathsf{T} x^s - \tau_i) > 0\}.$$

Subset of $\{\pm 1\}^p$; up to 2^p objects to identify.

Extension

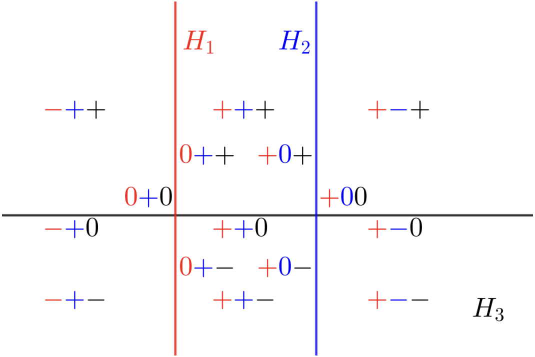
The "whole arrangement" $\{\pm 1\} \to \{-1, 0, +1\}$: $\overline{\mathcal{S}}(V, \tau)$



Toy example with three hyperplanes.

Extension

The "whole arrangement" $\{\pm 1\} \to \{-1, 0, +1\}$: $\overline{\mathcal{S}}(V, \tau)$



Toy example with three hyperplanes.

Outline

Starting points

Basic POVs

- geometric: hyperplane arrangements,
- algebraic: systems of affine inequalities.

- other algebra/geometry questions,
- nonsmooth analysis,
- convex analysis,
- computer science,
- some "applications".

Each viewpoint: new insights / tools / ...
Mostly $\tau = 0$: hyperplanes intersect in $0 \in \mathbb{R}^n$.

Starting points

Basic POVs

- geometric: hyperplane arrangements,
- algebraic: systems of affine inequalities.

- other algebra/geometry questions,
- nonsmooth analysis,
- convex analysis,
- computer science,
- some "applications".

Each viewpoint: new insights / tools / ...
Mostly $\tau = 0$: hyperplanes intersect in $0 \in \mathbb{R}^n$.

Our initial motivation

Nonsmooth analysis/optimization

Not a single gradient (∇) but a set of *generalized gradients*.

- For a specific method in complementarity problems, the generalized gradient := the chambers of an arrangement.
- See [DGP25a], additional uses in [Pla25].

Our initial motivation

Nonsmooth analysis/optimization

Not a single gradient (∇) but a set of *generalized gradients*.

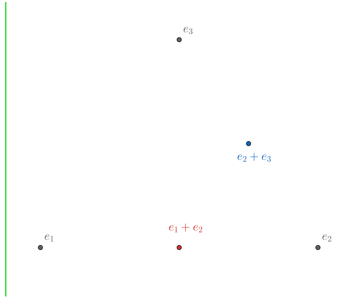
- For a specific method in complementarity problems, the generalized gradient := the chambers of an arrangement.
- See [DGP25a], additional uses in [Pla25].

Ways to linearly separate points [Zas75, section 6.A]

Set of points $v_1, \ldots, v_p \subseteq \mathbb{R}^n$.
Up to 2^p ways to linearly separate them in two groups.

Ways to linearly separate points [Zas75, section 6.A]

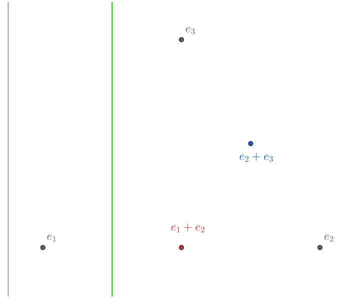
Set of points $v_1, \ldots, v_p \subseteq \mathbb{R}^n$.
Up to 2^p ways to linearly separate them in two groups.



Different linear separations.

Ways to linearly separate points [Zas75, section 6.A]

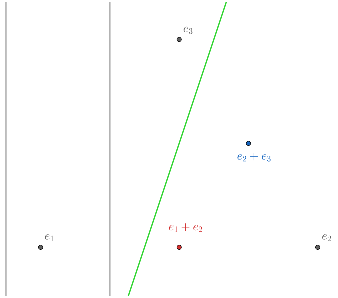
Set of points $v_1, \ldots, v_p \subseteq \mathbb{R}^n$.
Up to 2^p ways to linearly separate them in two groups.



Different linear separations.

Ways to linearly separate points [Zas75, section 6.A]

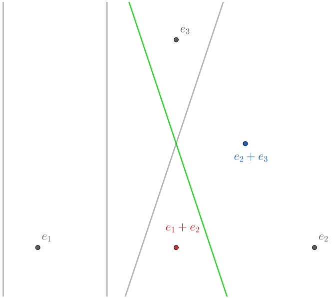
Set of points $v_1, \ldots, v_p \subseteq \mathbb{R}^n$.
Up to 2^p ways to linearly separate them in two groups.



Different linear separations.

Ways to linearly separate points [Zas75, section 6.A]

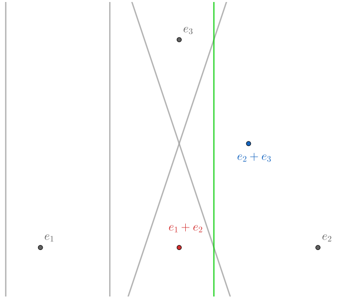
Set of points $v_1, \ldots, v_p \subseteq \mathbb{R}^n$.
Up to 2^p ways to linearly separate them in two groups.



Different linear separations.

Ways to linearly separate points [Zas75, section 6.A]

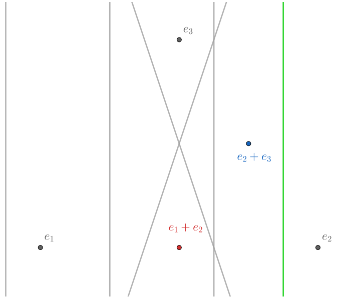
Set of points $v_1, \dots, v_p \subseteq \mathbb{R}^n$.
Up to 2^p ways to linearly separate them in two groups.



Different linear separations. Not all linear separations are possible.

Ways to linearly separate points [Zas75, section 6.A]

Set of points $v_1, \ldots, v_p \subseteq \mathbb{R}^n$.
Up to 2^p ways to linearly separate them in two groups.

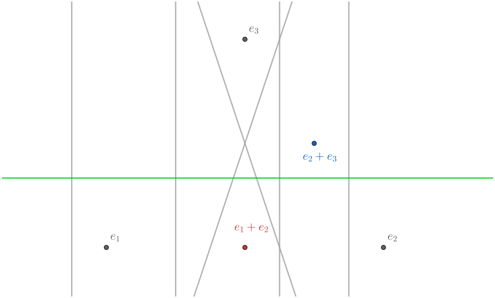


Different linear separations. Not all linear separations are possible.

Ways to linearly separate points [Zas75, section 6.A]

Set of points $v_1, \ldots, v_p \subseteq \mathbb{R}^n$.

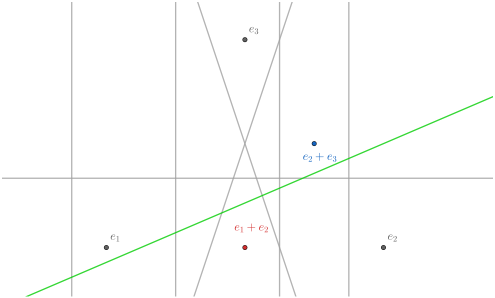
Up to 2^p ways to linearly separate them in two groups.



Different linear separations. Not all linear separations are possible.

Ways to linearly separate points [Zas75, section 6.A]

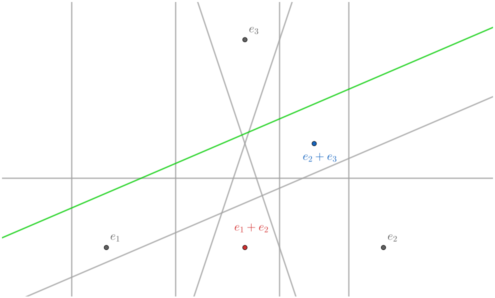
Set of points $v_1, \ldots, v_p \subseteq \mathbb{R}^n$.
Up to 2^p ways to linearly separate them in two groups.



Different linear separations. Not all linear separations are possible.

Ways to linearly separate points [Zas75, section 6.A]

Set of points $v_1, \ldots, v_p \subseteq \mathbb{R}^n$.
Up to 2^p ways to linearly separate them in two groups.



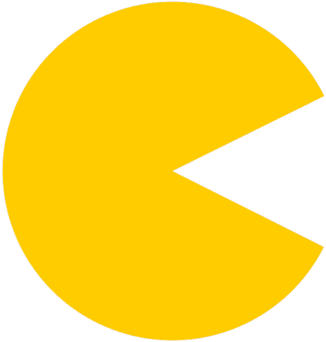
Different linear separations. Not all linear separations are possible.

Orientations of vectors forming cones (1) (see [DP22])

Let $v_1, \dots, v_p \subseteq \mathbb{R}^n$, $\mathrm{cone}\{v_1, \dots, v_p\} = \{\sum_{i=1}^{p} t_i v_i : t_i \geqslant 0\}$.

Orientations of vectors forming cones (1) (see [DP22])

Let $v_1, \ldots, v_p \subseteq \mathbb{R}^n$, $\mathrm{cone}\{v_1, \ldots, v_p\} = \{\sum_{i=1}^{p} t_i v_i : t_i \geqslant 0\}$.



Pacman is **not pointed**: does not look like a cone. His mouth $\checkmark$.

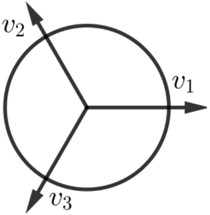
Orientations of vectors forming cones (2)

Consider the 2^p cones $\mathrm{cone}\{s_i v_i : s_i \in \{-1, +1\}\}$:
which are pointed cones = look like a cone?

Orientations of vectors forming cones (2)

Consider the 2^p cones $\mathrm{cone}\{s_i v_i : s_i \in \{-1, +1\}\}$:
which are pointed cones = look like a cone?

Example: $v_1 = (1, 0)$, $v_2 = (-1/2, \sqrt{3}/2)$, $v_3 = (-1/2, -\sqrt{3}/2)$.

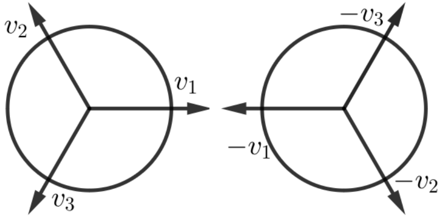


The cone of these vectors generate $\mathbb{R}^2$.

Orientations of vectors forming cones (2)

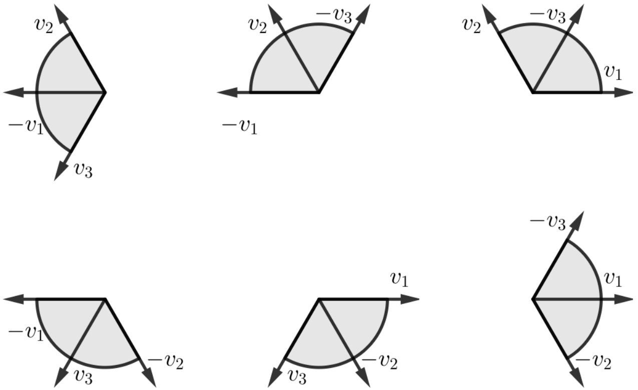
Consider the 2^p cones $\text{cone}\{s_i v_i : s_i \in \{-1, +1\}\}$:
which are pointed cones = look like a cone?

Example: $v_1 = (1, 0)$, $v_2 = (-1/2, \sqrt{3}/2)$, $v_3 = (-1/2, -\sqrt{3}/2)$.



The cone of these vectors generate $\mathbb{R}^2$. The left cone $(+v_1, +v_2, +v_3)$ and the right cone $(-v_1, -v_2, -v_3)$ are not pointed.

Orientations of vectors forming cones (3)



Examples with pointed cones (swaps by opposing an extremal vector).
Here, $(+, +, +)$ and $(-, -, -)$ are incorrect, others are correct.

Orthants and null space

Orthant: the signs of $y \in \mathbb{R}^p$ remain constant.
Positive orthant $\mathbb{R}^p_{++} = \{y \in \mathbb{R}^p : y > 0\}$... 2^p orthants in total.

Duality

$$\mathcal{S}(V, 0) \iff \text{orthants of } \mathbb{R}^p \textbf{ not } \text{intersecting } \mathcal{N}(V).$$

$$V = \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & +\sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix}, \quad \mathcal{N}(V) = \text{vect}[1;1;1]$$

Orthants and null space

Orthant: the signs of $y \in \mathbb{R}^p$ remain constant.
Positive orthant $\mathbb{R}^p_{++} = \{y \in \mathbb{R}^p : y > 0\}$... 2^p orthants in total.

Duality

$$\mathcal{S}(V, 0) \iff \text{orthants of } \mathbb{R}^p \textbf{ not} \text{ intersecting } \mathcal{N}(V).$$

$$V = \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & +\sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix}, \quad \mathcal{N}(V) = \text{vect}[1;1;1]$$

Orthants and null space

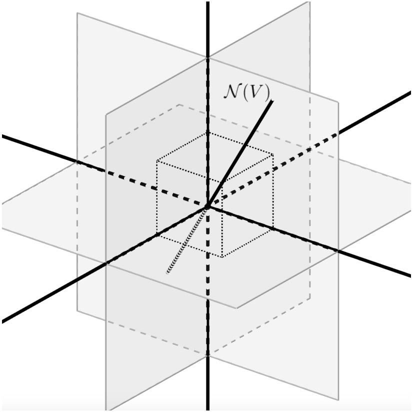
Orthant: the signs of $y \in \mathbb{R}^p$ remain constant.
Positive orthant $\mathbb{R}^p_{++} = \{y \in \mathbb{R}^p : y > 0\}\dots$ 2^p orthants in total.

Duality

$$\mathcal{S}(V,0) \iff \text{orthants of } \mathbb{R}^p \textbf{ not} \text{ intersecting } \mathcal{N}(V).$$

$$V = \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & +\sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix}, \quad \mathcal{N}(V) = \text{vect}[1;1;1]$$

Orthants and null space, example



$\mathcal{N}(V)$ has nonempty intersection with orthants $\mathbb{R}^3_+$ and $\mathbb{R}^3_-$, corresponding to infeasible $(+,+,+)$ and $(-,-,-)$.

Zonotopes

$V \in \mathbb{R}^{n\times p}$, $Z(V) := V[-1,+1]^p = \{V\eta : -1_p \leqslant \eta \leqslant 1_p\} \subseteq \mathbb{R}^n$.
Centrally symmetric polytope, [McM71; Zie07; Alt22; KA21; ST19]

Vertices: subset of the 2^p points $V\{-1,+1\}^p$: $Vs = \sum_{i=1}^p v_i s_i$, $s \in \{\pm 1\}^p$; some Vs are inside $Z(V)$: not vertices.

One main combinatorial properties of zonotopes

- $\mathcal{S}(V,0) \Leftrightarrow$ vertices of $Z(V)$;
- $\overline{\mathcal{S}}(V,0) \Leftrightarrow$ faces of $Z(V)$.

$\{\pm 1\}^p$: faces of dimension 0, $\{0,\pm 1\}^p$: faces of all dimensions.

Zonotopes

$V \in \mathbb{R}^{n\times p}$, $Z(V) := V[-1,+1]^p = \{V\eta : -1_p \leqslant \eta \leqslant 1_p\} \subseteq \mathbb{R}^n$.
Centrally symmetric polytope, [McM71; Zie07; Alt22; KA21; ST19]

Vertices: subset of the 2^p points $V\{-1,+1\}^p$: $Vs = \sum_{i=1}^p v_i s_i$, $s \in \{\pm 1\}^p$; some Vs are inside $Z(V)$: not vertices.

One main combinatorial properties of zonotopes

- $\mathcal{S}(V,0) \Leftrightarrow$ vertices of $Z(V)$;
- $\overline{\mathcal{S}}(V,0) \Leftrightarrow$ faces of $Z(V)$.

$\{\pm 1\}^p$: faces of dimension 0, $\{0,\pm 1\}^p$: faces of all dimensions.

Zonotopes

$V \in \mathbb{R}^{n \times p}$, $Z(V) := V[-1,+1]^p = \{V\eta : -1_p \leqslant \eta \leqslant 1_p\} \subseteq \mathbb{R}^n$.
Centrally symmetric polytope, [McM71; Zie07; Alt22; KA21; ST19]

Vertices: subset of the 2^p points $V\{-1,+1\}^p$: $Vs = \sum_{i=1}^p v_i s_i$, $s \in \{\pm 1\}^p$; some Vs are inside $Z(V)$: not vertices.

One main combinatorial properties of zonotopes

- $\mathcal{S}(V,0) \Leftrightarrow$ vertices of $Z(V)$;
- $\overline{\mathcal{S}}(V,0) \Leftrightarrow$ faces of $Z(V)$.

$\{\pm 1\}^p$: faces of dimension 0, $\{0, \pm 1\}^p$: faces of all dimensions.

Zonotopes

$V \in \mathbb{R}^{n \times p}$, $Z(V) := V[-1,+1]^p = \{V\eta : -1_p \leqslant \eta \leqslant 1_p\} \subseteq \mathbb{R}^n$.
Centrally symmetric polytope, [McM71; Zie07; Alt22; KA21; ST19]

Vertices: subset of the 2^p points $V\{-1,+1\}^p$: $Vs = \sum_{i=1}^p v_i s_i$, $s \in \{\pm 1\}^p$; some Vs are inside $Z(V)$: not vertices.

One main combinatorial properties of zonotopes

- $\mathcal{S}(V,0) \Leftrightarrow$ vertices of $Z(V)$;
- $\overline{\mathcal{S}}(V,0) \Leftrightarrow$ faces of $Z(V)$.

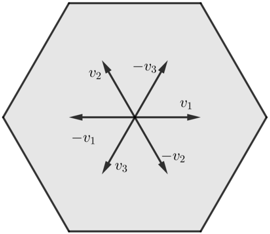
$\{\pm 1\}^p$: faces of dimension 0, $\{0,\pm 1\}^p$: faces of all dimensions.

Zonotopes illustrated

With $V = \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & +\sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix}$, $Z(V)$ and vertices:

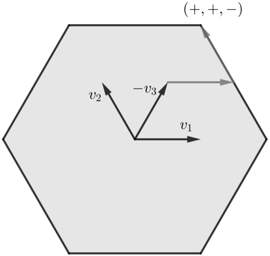
Zonotopes illustrated

With $V = \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & +\sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix}$, $Z(V)$ and vertices:



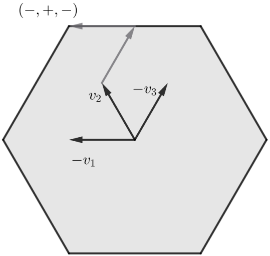
Zonotopes illustrated

With $V = \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & +\sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix}$, $Z(V)$ and vertices:



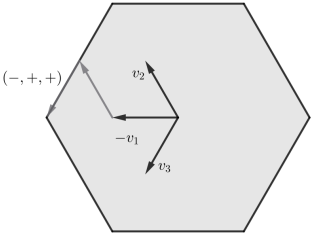
Zonotopes illustrated

With $V = \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & +\sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix}$, $Z(V)$ and vertices:



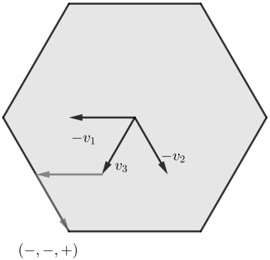
Zonotopes illustrated

With $V = \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & +\sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix}$, $Z(V)$ and vertices:



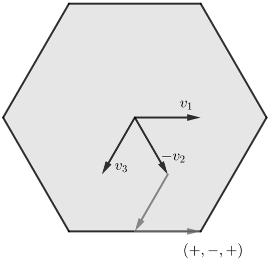
Zonotopes illustrated

With $V = \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & +\sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix}$, $Z(V)$ and vertices:



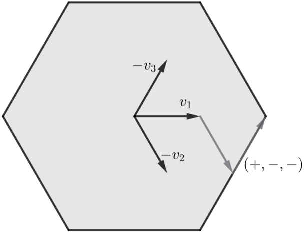
Zonotopes illustrated

With $V = \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & +\sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix}$, $Z(V)$ and vertices:



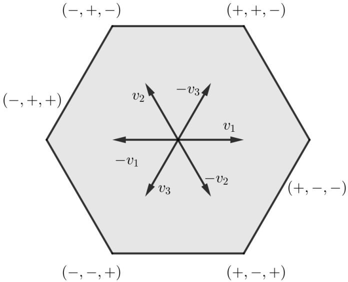
Zonotopes illustrated

With $V = \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & +\sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix}$, $Z(V)$ and vertices:



Zonotopes illustrated

With $V = \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & +\sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix}$, $Z(V)$ and vertices:



$(+,+,+)$ and $(-,-,-)$ do not correspond to vertices.

Arrangements and graphs [Sta07]

Graph G with vertices $= [1:n]$, p edges.
If $\{i,j\}$ is an edge, $H_{ij} := \{x : x_i - x_j = 0\}$ in the arrangement.

Relation with chambers

- An orientation of G is choosing $i \to j$ or $i \leftarrow j$ for each edge $\{i,j\}$: 2^p orientations.
- The **acyclic** orientations are in bijection with the chambers.

Also a relation involving the (proper) colorings of graphs.

Arrangements and graphs [Sta07]

Graph G with vertices $= [1 : n]$, p edges.
If $\{i, j\}$ is an edge, $H_{ij} := \{x : x_i - x_j = 0\}$ in the arrangement.

Relation with chambers

- An orientation of G is choosing $i \to j$ or $i \leftarrow j$ for each edge $\{i, j\}$: 2^p orientations.
- The **acyclic** orientations are in bijection with the chambers.

Also a relation involving the (proper) colorings of graphs.

Arrangements and graphs [Sta07]

Graph G with vertices $= [1:n]$, p edges.
If $\{i, j\}$ is an edge, $H_{ij} := \{x : x_i - x_j = 0\}$ in the arrangement.

Relation with chambers

- An orientation of G is choosing $i \to j$ or $i \leftarrow j$ for each edge $\{i, j\}$: 2^p orientations.
- The **acyclic** orientations are in bijection with the chambers.

Also a relation involving the (proper) colorings of graphs.

Illustration (1)

Let $n = 3$ with the $p = 3$ possible edges/planes.
$\{(x, y, z) : x = y\}$, $\{(x, y, z) : y = z\}$, $\{(x, y, z) : z = x\}$.

Illustration (1)

Let $n = 3$ with the $p = 3$ possible edges/planes.
$\{(x, y, z) : x = y\}$, $\{(x, y, z) : y = z\}$, $\{(x, y, z) : z = x\}$.

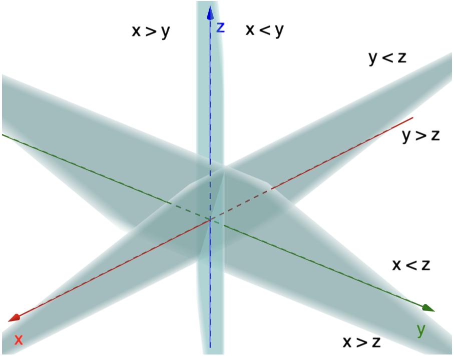
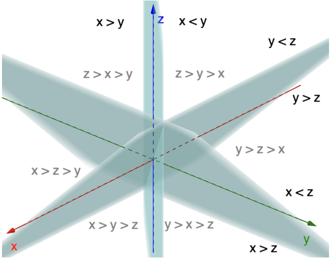
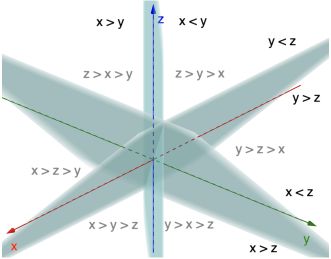


Illustration (2)



Example with the corresponding regions: 6 and not $2^3 = 8$.

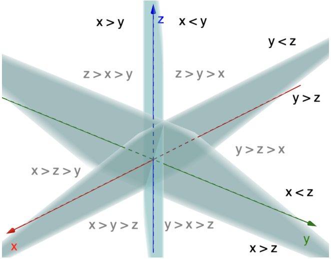
Illustration (2)



Example with the corresponding regions: 6 and not $2^3 = 8$.

The cyclic orientations are
$1 \to 2, 2 \to 3, 3 \to 1 \Leftrightarrow x > y, y > z, z > x \Leftrightarrow (+,+,+)$ and

Illustration (2)



Example with the corresponding regions: 6 and not $2^3 = 8$.

The cyclic orientations are
$1 \to 2, 2 \to 3, 3 \to 1 \Leftrightarrow x > y, y > z, z > x \Leftrightarrow (+,+,+)$ and
$1 \to 3, 3 \to 2, 2 \to 1 \Leftrightarrow x < y, y < z, z < x \Leftrightarrow (-,-,-)$.

Other combinatorial shenanigans [Sta07]

Very Important Property

The set of intersections of hyperplanes form a **poset**.

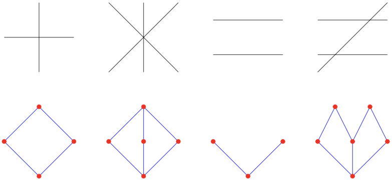
- structure “encoding” the combinatorics of the arrangement
- one level for $\mathbb{R}^n$, one level for planes, . . . , one level for points

Other combinatorial shenanigans [Sta07]

Very Important Property

The set of intersections of hyperplanes form a **poset**.

- structure "encoding" the combinatorics of the arrangement
- one level for $\mathbb{R}^n$, one level for planes, . . . , one level for points



[Sta07, fig.2 p.8] Arrangements and corresponding posets. No signs $\pm$.

Robot path planning [Sle00]

How to help a robot move inside a building?

Robot path planning [Sle00]

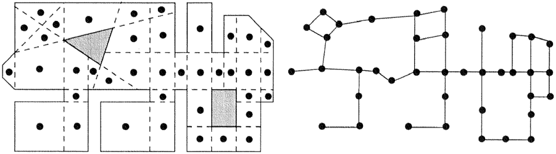
How to help a robot move inside a building?

In dimension 2 or 3, walls are extended to planes and form cells. Those "inside the building" form the chambers of interest.

Robot path planning [Sle00]

How to help a robot move inside a building?

In dimension 2 or 3, walls are extended to planes and form cells.
Those "inside the building" form the chambers of interest.



[Sle00, fig. 8.10, p.85] Chamber decomposition of a building.

One aspect of neural networks

Consider neurons with affine weights: $(v, t) \in \mathbb{R}^n \times \mathbb{R}$

$$\underbrace{x}_{\text{input}} \mapsto \underbrace{v^{\mathsf{T}} x - t}_{\text{action}} > 0?$$

Relation with arrangements

Each neuron creates a hyperplane in $\mathbb{R}^n$: layer = arrangement.

- nonlinear / piecewise neurons?
- with multiple layers?

Maybe a *base tool* for more advanced constructions.

One aspect of neural networks

Consider neurons with affine weights: $(v, t) \in \mathbb{R}^n \times \mathbb{R}$

$$\underbrace{x}_{\text{input}} \mapsto \underbrace{v^{\mathsf{T}} x - t}_{\text{action}} > 0?$$

Relation with arrangements

Each neuron creates a hyperplane in $\mathbb{R}^n$: layer = arrangement.

- nonlinear / piecewise neurons?
- with multiple layers?

Maybe a *base tool* for more advanced constructions.

One aspect of neural networks

Consider neurons with affine weights: $(v, t) \in \mathbb{R}^n \times \mathbb{R}$

$$\underbrace{x}_{\text{input}} \mapsto \underbrace{v^\top x - t}_{\text{action}} > 0?$$

Relation with arrangements

Each neuron creates a hyperplane in $\mathbb{R}^n$: layer = arrangement.

- nonlinear / piecewise neurons?
- with multiple layers?

Maybe a *base tool* for more advanced constructions.

Where some arrangements intervene

See [Win66; BEK23; Sta07; PS00; Ath96].

- specific families of arrangements (up to convention):
- combinatorics / geometry,
- algebraic statistics,
- quantum field theory,
- economics,
- psychometrics. . .

Applications with the whole arrangement $\{-1, 0, +1\}$ [EOS86].

Where some arrangements intervene

See [Win66; BEK23; Sta07; PS00; Ath96].

- specific families of arrangements (up to convention):
- combinatorics / geometry,
- algebraic statistics,
- quantum field theory,
- economics,
- psychometrics...

Applications with the whole arrangement $\{-1, 0, +1\}$ [EOS86].

Outline

Assumptions

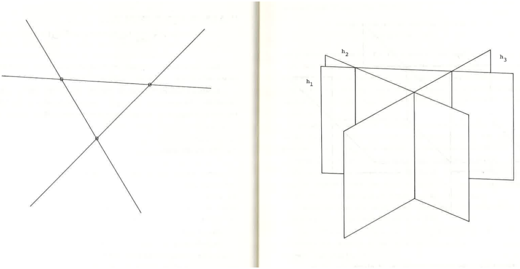
(1) $v_i \neq 0$ (otherwise H_i^-, H_i or H_i^+ is $\mathbb{R}^n$),

Assumptions

(1) $v_i \neq 0$ (otherwise H_i^-, H_i or H_i^+ is $\mathbb{R}^n$),
(2) $H_i \neq H_j$,

Assumptions

(1) $v_i \neq 0$ (otherwise H_i^-, H_i or H_i^+ is $\mathbb{R}^n$),
(2) $H_i \neq H_j$,
(3) $\text{range}(V) = \mathbb{R}^n$.



Projection for arrangements without full dimension [Zas75, fig. 2.1-2.2].

Formulas

$|\mathcal{S}(V,\tau)| \leqslant 2^p$; equality iff $p = \mathrm{rank}(V) = n$.

General upper bound ([Sch50], [Sta07])

$$|\mathcal{S}(V,\tau)| \leqslant \sum_{i=0}^{n} \binom{p}{i} \quad (\leqslant 2^p)$$

$=$ when in *general position*: $\simeq V$, τ random.

Formulas valid all the time (!): [Win66; Zas75]

$$|\mathcal{S}(V,\tau)| = \sum_{J\subseteq[1:p],\tau_J\in\mathcal{R}(V_{:,J}^{\mathsf{T}})} (-1)^{\dim(\mathcal{N}(V_{:,J}))} = (-1)^n \chi(-1)$$

Both computations difficult: #P-hard; don't give $\mathcal{S}(V,\tau)$.

Formulas

$|\mathcal{S}(V,\tau)| \leqslant 2^p$; equality iff $p = \text{rank}(V) = n$.

General upper bound ([Sch50], [Sta07])

$$|\mathcal{S}(V,\tau)| \leqslant \sum_{i=0}^{n} \binom{p}{i} \quad (\leqslant 2^p)$$

$=$ when in *general position*: $\simeq$ V, τ random.

Formulas valid all the time (!): [Win66; Zas75]

$$|\mathcal{S}(V,\tau)| = \sum_{J\subseteq[1:p],\tau_J\in\mathcal{R}(V_{:,J}^{\mathsf{T}})} (-1)^{\dim(\mathcal{N}(V_{:,J}))} = (-1)^n \chi(-1)$$

Both computations difficult: #P-hard; don't give $\mathcal{S}(V,\tau)$.

Formulas

$|\mathcal{S}(V,\tau)| \leqslant 2^p$; equality iff $p = \mathrm{rank}(V) = n$.

General upper bound ([Sch50], [Sta07])

$$|\mathcal{S}(V,\tau)| \leqslant \sum_{i=0}^{n} \binom{p}{i} \quad (\leqslant 2^p)$$

$=$ when in *general position*: $\simeq$ V, τ random.

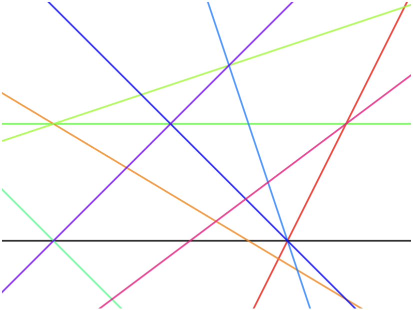
Formulas valid all the time (!): [Win66; Zas75]

$$|\mathcal{S}(V,\tau)| = \sum_{J\subseteq[1:p],\tau_J\in\mathcal{R}(V_{:,J}^{\mathsf{T}})} (-1)^{\dim(\mathcal{N}(V_{:,J}))} = (-1)^n\chi(-1)$$

Both computations difficult: #P-hard; don't give $\mathcal{S}(V,\tau)$.

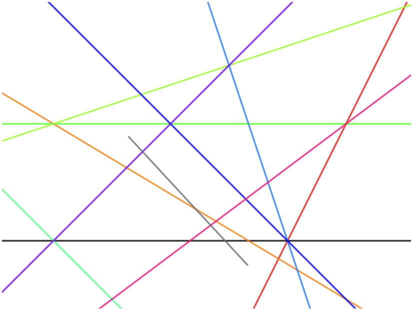
Connectivity properties

The chambers are the nodes of graph, edges = hyperplanes.



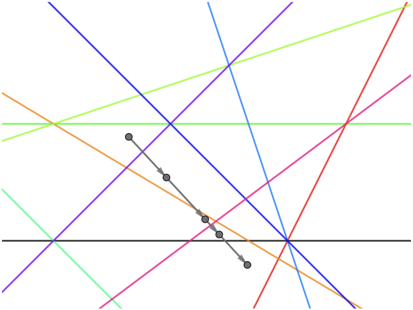
Connectivity properties

The chambers are the nodes of graph, edges = hyperplanes.



Connectivity properties

The chambers are the nodes of graph, edges = hyperplanes.

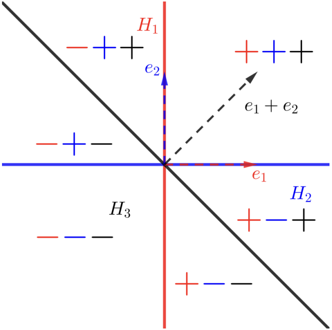


Paramount in some algorithms.
Transposable to vertices of zonotopes, cones. . .

Outline

Symmetric arrangements

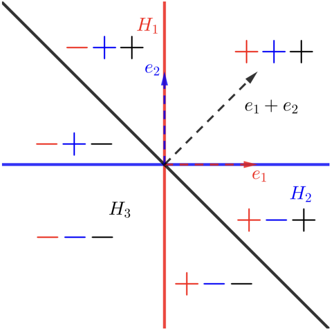
$\mathcal{S}(V, 0)$ is symmetric, $0 \in \mathbb{R}^n$ center of symmetry.



Algorithmically : just compute half of $\mathcal{S}(V, 0)$ or $\mathcal{S}(V, \tau)$.
In general, $\mathcal{S}(V, \tau)$ asymmetric.

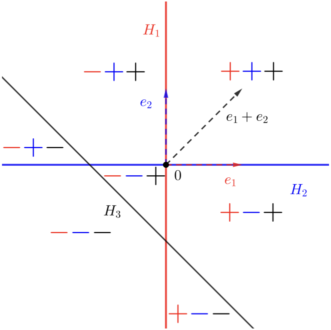
Symmetric arrangements

$\mathcal{S}(V, 0)$ is symmetric, $0 \in \mathbb{R}^n$ center of symmetry.



Algorithmically : just compute half of $\mathcal{S}(V, 0)$ or $\mathcal{S}(V, \tau)$.
In general, $\mathcal{S}(V, \tau)$ asymmetric.

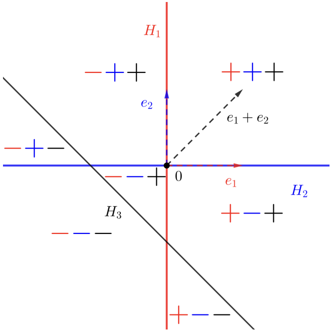
Partial symmetry



$--+$ asymmetric, others symmetric. Symmetric part $= S(V, 0)$.

Idea: compute asymmetric part + half of the symmetric part.

Partial symmetry



$--+$ asymmetric, others symmetric. Symmetric part $= S(V, 0)$.
Idea: compute asymmetric part + half of the symmetric part.

Outline

Some software

- Sagemath (documentation) [Dev24]
- Macaulay2 (see for arrangements or matroids) [GS24]
- polymake [GJ00], see [KP20] for arrangements
- TOPCOM [Ram02; Ram23]
- for matroids: Oid, [KK05]
- see also OSCAR [Dec+24; OSC24] (used in [BEK23])

Warning

- Sometimes, theoretical algos (not always experimentations).
- Some may be lost to time (and/or not reimplemented?).

Two algorithms for the whole arrangement

Bieri-Nef [BN82], the first algorithm:

Dimensional recursion + sweeping planes (H_0 “scans the space”), some unexplained elements.

Two algorithms for the whole arrangement

Bieri-Nef [BN82], the first algorithm:

Dimensional recursion + sweeping planes (H_0 "scans the space"), some unexplained elements.

Edelsbrunner-O'Rourke-Seidel [EOS86]

Asymptotic optimal complexity, *incremental* (H_1 then $H_2 \dots$). Involved algorithm: many definitions / subcases.

Back to the chambers: “simplex-type” algorithm

Chambers: connected graph but with **unknown nodes** and edges. Avis, Fukuda [AF92; AF96] (Sleumer [Sle98]) go through the graph *while* identifying the nodes := reverse search (RS).

Principle of the reverse search:

- start from an arbitrary chamber;
- find the neighbors of this chamber: changing one sign;
- recurse on neighbors (+ rule so no redundancy).

RS is a general principle useful for other identification problems.

Back to the chambers: “simplex-type” algorithm

Chambers: connected graph but with **unknown nodes** and edges. Avis, Fukuda [AF92; AF96] (Sleumer [Sle98]) go through the graph *while* identifying the nodes := reverse search (RS).

Principle of the reverse search:

- start from an arbitrary chamber;
- find the neighbors of this chamber: changing one sign;
- recurse on neighbors (+ rule so no redundancy).

RS is a general principle useful for other identification problems.

Back to the chambers: "simplex-type" algorithm

Chambers: connected graph but with **unknown nodes** and edges. Avis, Fukuda [AF92; AF96] (Sleumer [Sle98]) go through the graph *while* identifying the nodes := reverse search (RS).

Principle of the reverse search:

- start from an arbitrary chamber;
- find the neighbors of this chamber: changing one sign;
- recurse on neighbors (+ rule so no redundancy).

RS is a general principle useful for other identification problems.

Back to the chambers: “simplex-type” algorithm

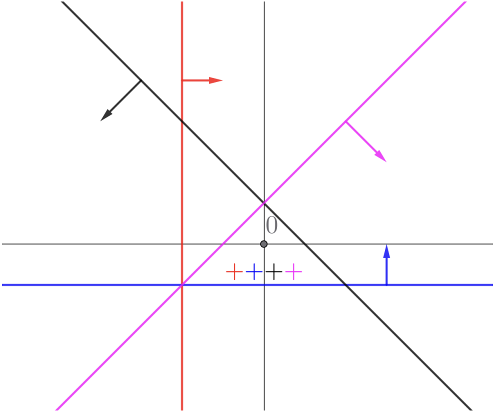
Chambers: connected graph but with **unknown nodes** and edges. Avis, Fukuda [AF92; AF96] (Sleumer [Sle98]) go through the graph *while* identifying the nodes := reverse search (RS).

Principle of the reverse search:

- start from an arbitrary chamber;
- find the neighbors of this chamber: changing one sign;
- recurse on neighbors (+ rule so no redundancy).

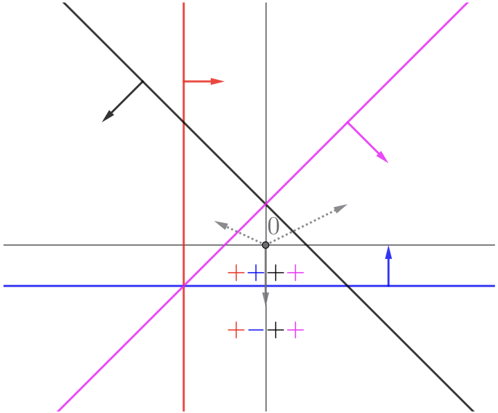
RS is a general principle useful for other identification problems.

Illustration of the reverse search



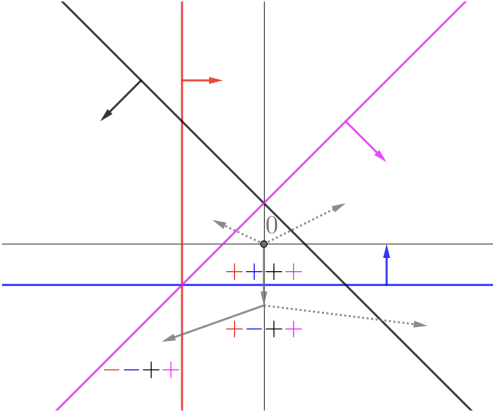
Grey arrows: order of the reverse search. Dotted grey arrows: pending neighbors; smaller ones: neighbors not visited due to the ordering rule(s).

Illustration of the reverse search



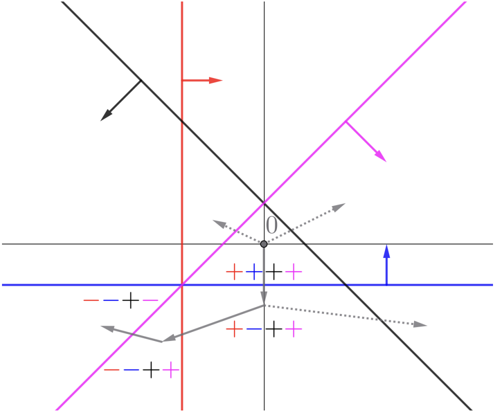
Grey arrows: order of the reverse search. Dotted grey arrows: pending neighbors; smaller ones: neighbors not visited due to the ordering rule(s).

Illustration of the reverse search



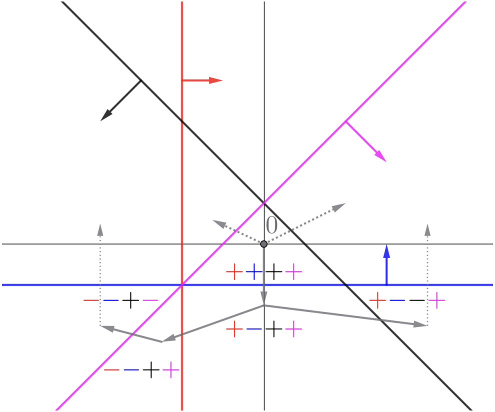
Grey arrows: order of the reverse search. Dotted grey arrows: pending neighbors; smaller ones: neighbors not visited due to the ordering rule(s).

Illustration of the reverse search



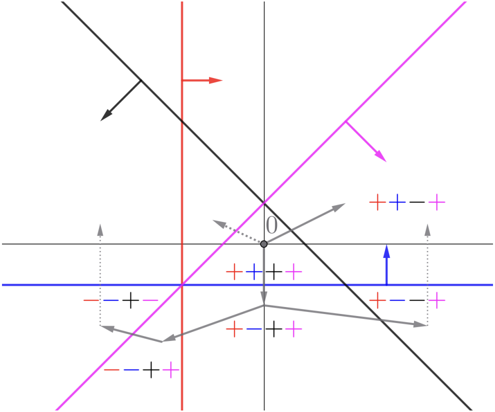
Grey arrows: order of the reverse search. Dotted grey arrows: pending neighbors; smaller ones: neighbors not visited due to the ordering rule(s).

Illustration of the reverse search



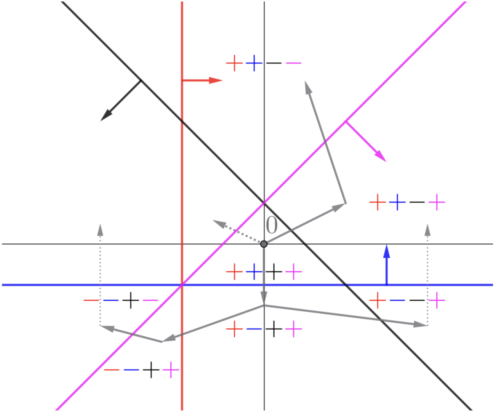
Grey arrows: order of the reverse search. Dotted grey arrows: pending neighbors; smaller ones: neighbors not visited due to the ordering rule(s).

Illustration of the reverse search



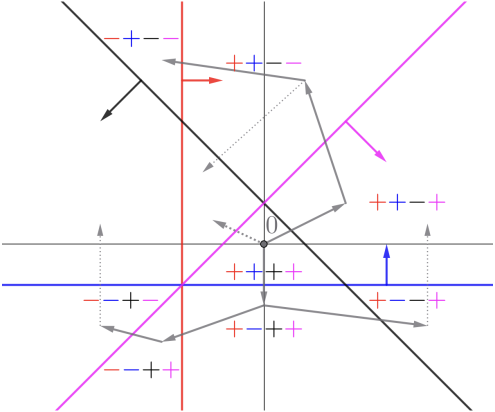
Grey arrows: order of the reverse search. Dotted grey arrows: pending neighbors; smaller ones: neighbors not visited due to the ordering rule(s).

Illustration of the reverse search



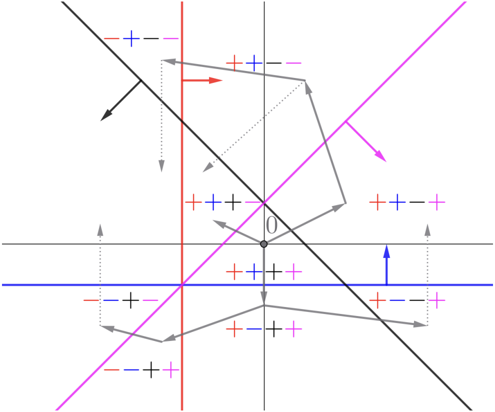
Grey arrows: order of the reverse search. Dotted grey arrows: pending neighbors; smaller ones: neighbors not visited due to the ordering rule(s).

Illustration of the reverse search



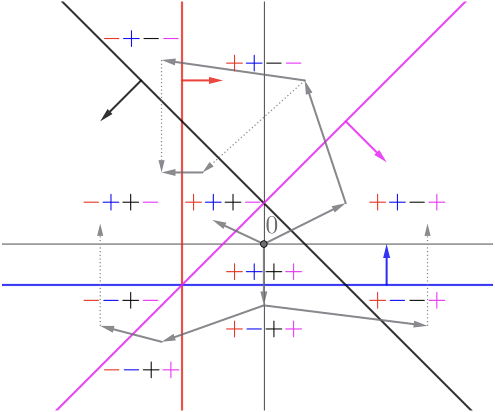
Grey arrows: order of the reverse search. Dotted grey arrows: pending neighbors; smaller ones: neighbors not visited due to the ordering rule(s).

Illustration of the reverse search



Grey arrows: order of the reverse search. Dotted grey arrows: pending neighbors; smaller ones: neighbors not visited due to the ordering rule(s).

Illustration of the reverse search



Grey arrows: order of the reverse search. Dotted grey arrows: pending neighbors; smaller ones: neighbors not visited due to the ordering rule(s).

For the zonotopes [DP22]

Recent method for the vertices of a zonotope.

Sort of revisited RS

Uses the framework of pointed cones.
Rule to select only some potential neighbors.

- Clear and short article (actually not their main goal);
- complexity result, no implementation / experiments.

For the zotopes [DP22]

Recent method for the vertices of a zonotope.

Sort of revisited RS

Uses the framework of pointed cones.
Rule to select only some potential neighbors.

- Clear and short article (actually not their main goal);
- complexity result, no implementation / experiments.

Hamilton(ian) path on the graph? [MM24]

Such path would be quite practical if it exists!

Answer

- Theoretically yes, even for more general problems.
- Practically, not sure (personal opinion, especially for $\mathcal{S}(V,\tau)$).

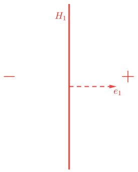
Hamilton(ian) path on the graph? [MM24]

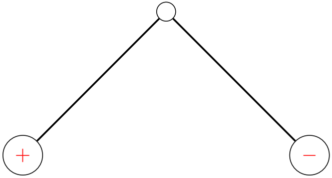
Such path would be quite practical if it exists!

Answer

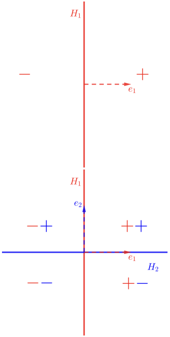
- Theoretically yes, even for more general problems.
- Practically, not sure (personal opinion, especially for $\mathcal{S}(V, \tau)$).

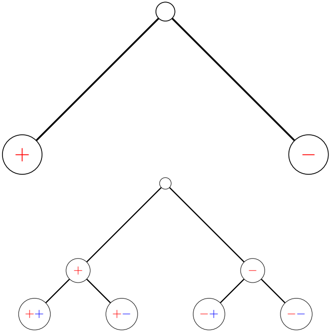
Rada-Černý algorithm: incremental tree



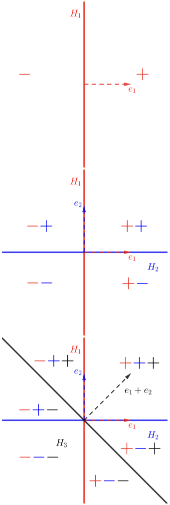


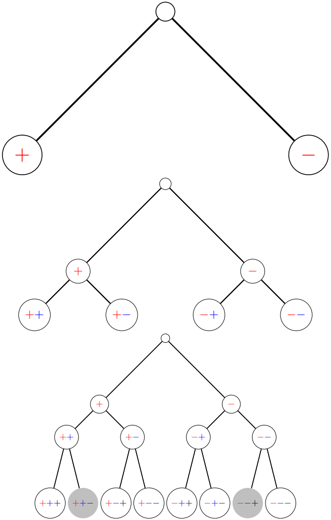
Rada-Černý algorithm: incremental tree





Rada-Černý algorithm: incremental tree





Main step at inner nodes

- node $s = (s_1, \ldots, s_k) \in \mathcal{S}(V_{:,[1:k]}, \tau_{[1:k]})$,
- with $x^s \in \mathbb{R}^n$: $s_i(v_i^\top x^s - \tau_i) > 0, \quad 1 \leqslant i \leqslant k$,
- $s_{k+1} := \text{sgn}(v_{k+1}^\top x^s - \tau_{k+1})$, $x^s \in H_{k+1}^{s_{k+1}}$: one descendant ✓
- for $(s, -s_{k+1})$: search for a solution x to

$$\exists\, x : \quad \begin{array}{r} s_i(v_i^\top x - \tau_i) > 0, \quad 1 \leqslant i \leqslant k \\ -s_{k+1}(v_{k+1}^\top x - \tau_{k+1}) > 0. \end{array} \qquad (1)$$

  Done by linear optimization.
- Better theoretically/practically than RS algorithms.

Main step at inner nodes

- node $s = (s_1, \ldots, s_k) \in \mathcal{S}(V_{:,[1:k]}, \tau_{[1:k]})$,
- with $x^s \in \mathbb{R}^n$: $s_i(v_i^\top x^s - \tau_i) > 0, \quad 1 \leqslant i \leqslant k$,
- $s_{k+1} := \operatorname{sgn}(v_{k+1}^\top x^s - \tau_{k+1})$, $x^s \in H_{k+1}^{s_{k+1}}$: one descendant ✓
- for $(s, -s_{k+1})$: search for a solution x to

$$\exists\, x : \quad \begin{aligned} s_i(v_i^\top x - \tau_i) &> 0, \quad 1 \leqslant i \leqslant k \\ -s_{k+1}(v_{k+1}^\top x - \tau_{k+1}) &> 0. \end{aligned} \tag{1}$$

  Done by linear optimization.
- Better theoretically/practically than RS algorithms.

Main step at inner nodes

- node $s = (s_1, \ldots, s_k) \in \mathcal{S}(V_{:,[1:k]}, \tau_{[1:k]})$,
- with $x^s \in \mathbb{R}^n$: $s_i(v_i^\top x^s - \tau_i) > 0, \quad 1 \leqslant i \leqslant k$,
- $s_{k+1} := \operatorname{sgn}(v_{k+1}^\top x^s - \tau_{k+1})$, $x^s \in H_{k+1}^{s_{k+1}}$: one descendant ✓
- for $(s, -s_{k+1})$: search for a solution x to

$$\exists\, x : \begin{array}{r} s_i(v_i^\top x - \tau_i) > 0, \quad 1 \leqslant i \leqslant k \\ -s_{k+1}(v_{k+1}^\top x - \tau_{k+1}) > 0. \end{array} \tag{1}$$

  Done by linear optimization.
- Better theoretically/practically than RS algorithms.

Main step at inner nodes

- node $s = (s_1, \ldots, s_k) \in \mathcal{S}(V_{:,[1:k]}, \tau_{[1:k]})$,
- with $x^s \in \mathbb{R}^n$: $s_i(v_i^\top x^s - \tau_i) > 0, \quad 1 \leqslant i \leqslant k$,
- $s_{k+1} := \operatorname{sgn}(v_{k+1}^\top x^s - \tau_{k+1})$, $x^s \in H_{k+1}^{s_{k+1}}$: one descendant ✓
- for $(s, -s_{k+1})$: search for a solution x to

$$\exists\, x : \begin{array}{r} s_i(v_i^\top x - \tau_i) > 0, \quad 1 \leqslant i \leqslant k \\ -s_{k+1}(v_{k+1}^\top x - \tau_{k+1}) > 0. \end{array} \tag{1}$$

Done by linear optimization.

- Better theoretically/practically than RS algorithms.

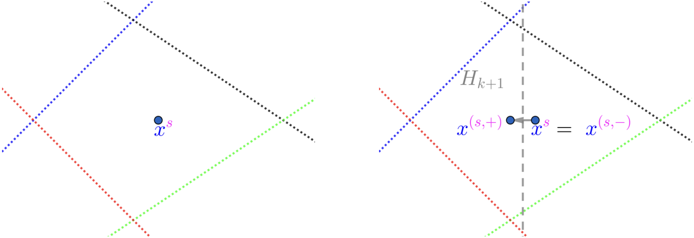
Main step at inner nodes

- node $s = (s_1, \dots, s_k) \in \mathcal{S}(V_{:,[1:k]}, \tau_{[1:k]})$,
- with $x^s \in \mathbb{R}^n$: $s_i(v_i^\mathsf{T} x^s - \tau_i) > 0, \quad 1 \leqslant i \leqslant k$,
- $s_{k+1} := \operatorname{sgn}(v_{k+1}^\mathsf{T} x^s - \tau_{k+1})$, $x^s \in H_{k+1}^{s_{k+1}}$: one descendant ✓
- for $(s, -s_{k+1})$: search for a solution x to

$$\exists\, x : \quad \begin{aligned} s_i(v_i^\mathsf{T} x - \tau_i) &> 0, \quad 1 \leqslant i \leqslant k \\ -s_{k+1}(v_{k+1}^\mathsf{T} x - \tau_{k+1}) &> 0. \end{aligned} \tag{1}$$

  Done by linear optimization.
- Better theoretically/practically than RS algorithms.

Closeness

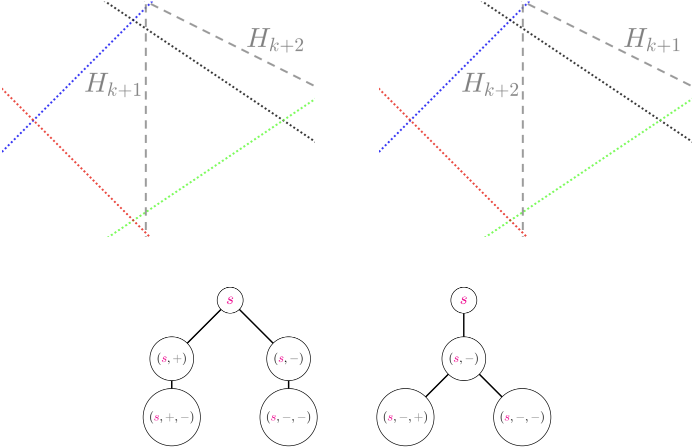


Left: level k. Right: shift of x^s when $x^s \underset{\sim}{\in} H_{k+1}$.

Details

For $s \in \{\pm 1\}^k$ with x^s, if $x^s \underset{\sim}{\in} H_{k+1} \Leftrightarrow v_{k+1}^\top x^s - \tau_{k+1} \simeq 0$, $(s, +1)$ and $(s, -1)$ in level $k+1$ without LOP.

Sequencing – which order to choose?



Changes inner levels – level p is always $\mathcal{S}(V,\tau)$.

A different approach

So far, we verify if descendants exist $\Leftrightarrow$ not [not exist].

A different approach

So far, we verify if descendants exist $\Leftrightarrow$ not [not exist].

Dual method: new viewpoint via convex duality

- Detect **infeasibilities**: when to stop the tree.
- Verify if a descendant contains an infeasibility.

A different approach

So far, we verify if descendants exist $\Leftrightarrow$ not [not exist].

Dual method: new viewpoint via convex duality

- Detect **infeasibilities**: when to stop the tree.
- Verify if a descendant contains an infeasibility.
- Precomputation of the *minimal* infeasibilities (sufficient).

A different approach

So far, we verify if descendants exist $\Leftrightarrow$ not [not exist].

Dual method: new viewpoint via convex duality

- Detect **infeasibilities**: when to stop the tree.
- Verify if a descendant contains an infeasibility.
- Precomputation of the *minimal* infeasibilities (sufficient).
- Equivalent to computethe *matroid circuits* of V [Oxl11].
- Which is *another* combinatorial problem.

A different approach

So far, we verify if descendants exist $\Leftrightarrow$ not [not exist].

Dual method: new viewpoint via convex duality

- Detect **infeasibilities**: when to stop the tree.
- Verify if a descendant contains an infeasibility.
- Precomputation of the *minimal* infeasibilities (sufficient).
- Equivalent to computethe *matroid circuits* of V [Oxl11].
- Which is *another* combinatorial problem.

That dual method is thus not that practical.
$\rightarrow$ primal-dual version, learns some infeasible combinations.
With everything, $\simeq$ 8 times faster [DGP25b].

Conclusion

Main take-aways

- relations/applications with many other topics
- various techniques can be employed for computations or to improve existing algorithms.

Thank you for your attention! Any question?

Bibliographic elements I

[AF92] David Avis and Komei Fukuda. "A Pivoting Algorithm for Convex Hulls and Vertex Enumeration of Arrangements and Polyhedra". In: *Discrete Computational Geometry* 8 (1992), pp. 295–31. ISSN: 0179-5376,1432-0444. DOI: 10.1007/BF02293050.

[AF96] David Avis and Komei Fukuda. "Reverse Search for Enumeration". In: *Discrete Applied Mathematics* 65 (Mar. 1996), pp. 21–46. DOI: 10.1016/0166-218X(95)00026-N.

[Alt22] Matthias Althoff. *On Computing the Minkowski Difference of Zonotopes*. Aug. 2022. arXiv: 1512.02794 [cs].

[Ath96] Christos A. Athanasiadis. "Characteristic Polynomials of Subspace Arrangements and Finite Fields". In: *Advances in Mathematics* 122.2 (Sept. 1996), pp. 193–233. ISSN: 00018708. DOI: 10.1006/aima.1996.0059.

[BEK23] Taylor Brysiewicz, Holger Eble, and Lukas Kühne. "Computing Characteristic Polynomials of Hyperplane Arrangements with Symmetries". In: *Discrete & Computational Geometry* 70.4 (Dec. 2023), pp. 1356–1377. ISSN: 0179-5376, 1432-0444. DOI: 10.1007/s00454-023-00557-2.

[BN82] Hanspeter Bieri and Walter Nef. "A Recursive Sweep-Plane Algorithm, Determining All Cells of a Finite Division of Rˆd". In: *Computing* 28 (1982), pp. 189–198.

[Dec+24] Wolfram Decker et al. *The Computer Algebra System OSCAR: Algorithms and Examples*. 1st ed. Vol. 32. Algorithms and {C}omputation in {M}athematics. Springer, 2024. ISBN: 1431-1550 (issn).

[Dev24] The Sage Developers. *Sagemath, the Sage Mathematics*. 2024.

[DGP25a] Jean-Pierre Dussault, Jean Charles Gilbert, and Baptiste Plaquevent-Jourdain. "On the B-differential of the Componentwise Minimum of Two Affine Vector Functions". In: *Mathematical Programming Computation* (2025).

Bibliographic elements II

[DGP25b] Jean-Pierre Dussault, Jean Charles Gilbert, and Baptiste Plaquevent-Jourdain. "Primal and Dual Approaches for the Chamber Enumeration of Real Hyperplane Arrangements". In: *(submitted)* (2025).

[DP22] Antoine Deza and Lionel Pournin. "A Linear Optimization Oracle for Zonotope Computation". In: *Computational Geometry* 100 (Jan. 2022), p. 101809. ISSN: 09257721. DOI: 10.1016/j.comgeo.2021.101809.

[DSL06] György Dósa, István Szalkai, and Claude Laflamme. "The Maximum and Minimum Number of Circuits and Bases of Matroids.". In: *Pure Mathematics and Applications*. Mathematics of Optimization 15.4 (Sept. 2006), pp. 383–392.

[EOS86] Herbert Edelsbrunner, Joseph O'ROURKE, and Raimund Seidel. "CONSTRUCTING ARRANGEMENTS OF LINES AND HYPERPLANES WITH APPLICATIONS". In: *SIAM Journal on Computation* 15.2 (1986), pp. 341–363.

[GJ00] Ewgenij Gawrilow and Michael Joswig. "Polymake: A Framework for Analyzing Convex Polytopes.". In: *Polytopes—Combinatorics and Computation (Oberwolfach, 1997)*. DMV Sem. 29. Basel: Birkhäuser, 2000, pp. 43–73. ISBN: 3-7643-6351-7.

[GS24] Daniel R. Grayson and Michael E. Stillman. *Macaulay2, a Software System for Research in Algebraic Geometry*. 2024.

[KA21] Adrian Kulmburg and Matthias Althoff. "On the Co-NP-completeness of the Zonotope Containment Problem". In: *European Journal of Control* 62 (Nov. 2021), pp. 84–91. ISSN: 09473580. DOI: 10.1016/j.ejcon.2021.06.028.

[KK05] Robert John Kingan and Sandra Reuben Kingan. "A Software System for Matroids". In: *Graphs and Discovery*. DIMACS Ser. Discrete Math. Theoret. Comput. Sci. 69. Providence, Rhode Island: American Mathematical Society, 2005, pp. 287–295. ISBN: 0-8218-3761-3.

Bibliographic elements III

[KP20] Lars Kastner and Marta Panizzut. "Hyperplane Arrangements in Polymake". In: *Mathematical Software – ICMS 2020*. Ed. by Anna Maria Bigatti et al. Vol. 12097. Cham: Springer International Publishing, 2020, pp. 232–240. ISBN: 978-3-030-52199-8 978-3-030-52200-1. DOI: 10.1007/978-3-030-52200-1_23.

[McM71] Peter McMullen. "On Zonotopes". In: *Transactions of the American Mathematical Society* 159 (Sept. 1971), pp. 91–109. DOI: 10.2307/1996000.

[MM24] Arturo Merino and Torsten Mütze. "Traversing Combinatorial 0/1-Polytopes via Optimization". In: *SIAM Journal on Computing* 53.5 (Oct. 2024), pp. 1257–1292. ISSN: 0097-5397, 1095-7111. DOI: 10.1137/23M1612019.

[Mot36] Theodore S. Motzkin. *Beiträge zur Theorie der linearen Ungleichungen*. Tech. rep. Jerusalem, Israel: University Basel, 1936.

[OSC24] OSCAR. *OSCAR – Open Source Computer Algebra Research System, Version 1.0.0*. The OSCAR Team. 2024.

[OT92] Peter Orlik and Hiroaki Terao. *Arrangement of Hyperplanes*. Vol. 300. Grundlehren Der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]. Madison: Springer-Verlag Berlin Heidelberg GmbH, 1992. ISBN: 3-540-55259-6.

[Oxl11] James G. Oxley. *Matroid Theory*. Second edition. Oxford Graduate Texts in Mathematics 21. Oxford New York, NY: Oxford University Press, 2011. ISBN: 978-0-19-856694-6 978-0-19-960339-8.

[Pla25] Baptiste Plaquevent-Jourdain. "A Robust Linearization Method for Complementarity Problem – A Detour Through Hyperplane Arrangements". PhD thesis. Sorbonne Université, Université de Sherbrooke, 2025.

Bibliographic elements IV

[PS00] Alexander Postnikov and Richard P. Stanley. “Deformations of Coxeter Hyperplane Arrangements”. In: *Journal of Combinatorial Theory, Series A* 91.1-2 (Mar. 2000), pp. 544–597. ISSN: 00973165. DOI: 10.1006/jcta.2000.3106.

[Ram02] Jörg Rambau. “TOPCOM: TRIANGULATIONS OF POINT CONFIGURATIONS AND ORIENTED MATROIDS”. In: *Mathematical Software*. Beijing, China: WORLD SCIENTIFIC, July 2002, pp. 330–340. ISBN: 978-981-238-048-7 978-981-277-717-1. DOI: 10.1142/9789812777171_0035.

[Ram23] Jörg Rambau. “Symmetric Lexicographic Subset Reverse Search for the Enumeration of Circuits, Cocircuits, and Triangulations up to Symmetry”. In: (2023), pp. 1–41.

[RČ18] Miroslav Rada and Michal Černý. “A New Algorithm for Enumeration of Cells of Hyperplane Arrangements and a Comparison with Avis and Fukuda’s Reverse Search”. In: *SIAM Journal on Discrete Mathematics* 32.1 (Jan. 2018), pp. 455–473. ISSN: 0895-4801, 1095-7146. DOI: 10.1137/15M1027930.

[Rob87] Samuel Roberts. “On the Figures Formed by the Intercepts of a System of Straight Lines in a, Plane, and on Analogous Relations in Space of Three Dimensions”. In: *Proceedings of the London Mathematical Society* s1-19.1 (Nov. 1887), pp. 405–422. ISSN: 00246115. DOI: 10.1112/plms/s1-19.1.405.

[Sch50] Ludwig Schläfli. *Gesammelte mathematische Abhandlugen*. Springer, Basel: Birkhäuser, 1950.

[Sle00] Nora Helena Sleumer. “Hyperplane Arrangements: Construction, Visualization and Applications”. PhD thesis. Zurich, Switzerland: Swiss Federal Institute of Technology, 2000.

Bibliographic elements V

[Sle98] Nora Helena Sleumer. "Output-Sensitive Cell Enumeration in Hyperplane Arrangements". In: *Algorithm Theory — SWAT'98*. Ed. by Gerhard Goos et al. Vol. 1432. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 300–309. ISBN: 978-3-540-64682-2 978-3-540-69106-8. DOI: 10.1007/BFb0054377.

[ST19] Sadra Sadraddini and Russ Tedrake. *Linear Encodings for Polytope Containment Problems*. Mar. 2019. arXiv: 1903.05214 [math].

[Sta07] Richard P. Stanley. "An Introduction to Hyperplane Arrangements". In: *Geometric Combinatorics*. 1st ed. Vol. 13. IAS/Park City Math. Ser. Providence, Rhode Island: Amer. Math. Soc., 2007, pp. 389–496. ISBN: 978-0-8218-3736-8 0-8218-3736-2.

[Ste26] Jakob Steiner. "Einige Gesetze über die Theilung der Ebene und des Raumes.". In: *J. Reine Angew. Math* (1826), pp. 349–364.

[Win66] Robert Owen Winder. "Partitions of N -Space by Hyperplanes". In: *SIAM Journal on Applied Mathematics* 14.4 (July 1966), pp. 811–818. ISSN: 0036-1399, 1095-712X. DOI: 10.1137/0114068.

[Zas75] Thomas Zaslavsky. "Facing up to Arrangements: Face-Count Formulas for Partitions of Space by Hyperplanes". In: *Memoirs of the American Mathematical Society* 1.154 (1975), 1–109 (?)

[Zie07] Günter M. Ziegler. *Lectures on Polytopes*. 7th. Vol. 152. Graduate Texts in Mathematics. New York, NY: Springer New York, 2007. ISBN: 978-0-387-94365-7 978-1-4613-8431-1. DOI: 10.1007/978-1-4613-8431-1.

[Zie99] Günter M. Ziegler. *Lectures on 0/1-Polytopes*. Sept. 1999. arXiv: math/9909177.

General position expressions

All $\forall\ I \subseteq [1:p]$:

$$\begin{cases} \cap_{i\in I} H_i \neq \varnothing \text{ and } \dim(\cap_{i\in I} H_i) = n - |I| & \text{if } |I| \leqslant r \\ \cap_{i\in I} H_i = \varnothing & \text{if } |I| \geqslant r+1 \end{cases}$$

$$\begin{cases} \mathrm{rank}(V_{:,I}) = |I| & \text{if } |I| \leqslant r \\ \mathrm{rank}([V;\tau^{\mathsf{T}}]_{:,I}) = r+1 & \text{if } |I| \geqslant r+1, \end{cases}$$

$$\begin{cases} \mathrm{rank}(V_{:,I}) = \min(|I|, r) \\ \mathrm{rank}([V;\tau^{\mathsf{T}}]_{:,I}) = \min(|I|, r+1). \end{cases}$$

Possible to slightly specify (simplify) when $\tau = 0$.

Affine $\leftrightarrow$ linear (1)

Main property (for instance [OT92])

Affine arrangements are “half” of linear arrangements.

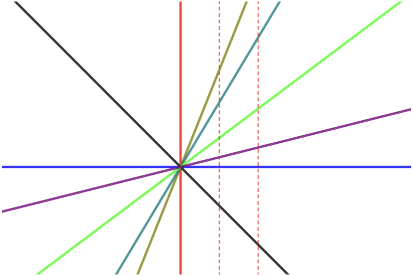
Half of linear arrangement: half-space of *one* of the hyperplanes: $\mathbb{R}^n \to \{x \in \mathbb{R}^n : v_i^\top x > 0\}$.

Affine $\leftrightarrow$ linear (1)

Main property (for instance [OT92])

Affine arrangements are "half" of linear arrangements.

Half of linear arrangement: half-space of *one* of the hyperplanes: $\mathbb{R}^n \to \{x \in \mathbb{R}^n : v_i^T x > 0\}$.



By homogeneity, translating H_i: dimension $n-1$, $p-1$ hyperplanes.

Affine ↔ linear (2)

One can to the converse to go from affine to linear by adding a dimension: $(V, \tau) \to (\mathcal{V}, 0)$

$$\mathcal{V} := \begin{bmatrix} V & 0 \\ \tau^\intercal & (\pm)1 \end{bmatrix}$$

$\mathcal{S}(V, \tau) :=$ affine$(n, p) \simeq$ linear$(n+1, p+1)$ (half of);
$\mathcal{S}(V, 0) :=$ linear$(n, p) \simeq$ affine$(n-1, p-1)$ (two opposite).

Affine arrangements are slightly more general.

Affine ↔ linear (2)

One can to the converse to go from affine to linear by adding a dimension: $(V, \tau) \to (\mathcal{V}, 0)$

$$\mathcal{V} := \begin{bmatrix} V & 0 \\ \tau^\intercal & (\pm)1 \end{bmatrix}$$

$\mathcal{S}(V, \tau) :=$ affine$(n, p) \simeq$ linear$(n+1, p+1)$ (half of);
$\mathcal{S}(V, 0) :=$ linear$(n, p) \simeq$ affine$(n-1, p-1)$ (two opposite).

Affine arrangements are slightly more general.

The (other) augmented matrix

$\mathcal{V} = \begin{bmatrix} V & 0 \\ \tau^\top & -1 \end{bmatrix}$: to swap linear $\leftrightarrow$ affine, but useless "numerically".

However, $[V; \tau^\top]$ can help:

$$\mathcal{S}([V;\tau^\top],0) = \mathcal{S}(V,\tau) \cup \mathcal{S}(V,-\tau)$$
$$= \mathcal{S}(V,0) \cup \mathcal{S}_a(V,\tau) \cup \mathcal{S}_a(V,-\tau)$$

$$\underbrace{\mathcal{S}(V,0)}_{\text{symmetric}} \overset{\cup\, \mathcal{S}_a(V,\tau)}{\longrightarrow} \underbrace{\mathcal{S}(V,\tau)}_{\text{asymmetric}} \overset{\cup\, -\mathcal{S}_a(V,\tau)}{\longrightarrow} \underbrace{\mathcal{S}([V;\tau^\top],0)}_{\text{symmetric}}$$

computing $\mathcal{S}(V,\tau)$ can be partially symmetrized (see later).

The (other) augmented matrix

$\mathcal{V} = \begin{bmatrix} V & 0 \\ \tau^{\mathsf{T}} & -1 \end{bmatrix}$: to swap linear ↔ affine, but useless "numerically".

However, $[V; \tau^{\mathsf{T}}]$ can help:

$$\begin{aligned} \mathcal{S}([V;\tau^{\mathsf{T}}],0) &= \mathcal{S}(V,\tau) \;\cup\; \mathcal{S}(V,-\tau) \\ &= \mathcal{S}(V,0) \;\cup\; \mathcal{S}_a(V,\tau) \;\cup\; \mathcal{S}_a(V,-\tau) \end{aligned}$$

$$\underbrace{\mathcal{S}(V,0)}_{\text{symmetric}} \xrightarrow{\cup\ \mathcal{S}_a(V,\tau)} \underbrace{\mathcal{S}(V,\tau)}_{\text{asymmetric}} \xrightarrow{\cup\ -\mathcal{S}_a(V,\tau)} \underbrace{\mathcal{S}([V;\tau^{\mathsf{T}}],0)}_{\text{symmetric}}$$

computing $\mathcal{S}(V,\tau)$ can be partially symmetrized (see later).

The (other) augmented matrix

$\mathcal{V} = \begin{bmatrix} V & 0 \\ \tau^{\mathsf{T}} & -1 \end{bmatrix}$: to swap linear $\leftrightarrow$ affine, but useless "numerically".

However, $[V; \tau^{\mathsf{T}}]$ can help:

$$\begin{aligned} \mathcal{S}([V;\tau^{\mathsf{T}}],0) &= \mathcal{S}(V,\tau) \ \cup \ \mathcal{S}(V,-\tau) \\ &= \mathcal{S}(V,0) \ \cup \ \mathcal{S}_a(V,\tau) \ \cup \ \mathcal{S}_a(V,-\tau) \end{aligned}$$

$$\underbrace{\mathcal{S}(V,0)}_{\text{symmetric}} \overset{\cup\ \mathcal{S}_a(V,\tau)}{\longrightarrow} \underbrace{\mathcal{S}(V,\tau)}_{\text{asymmetric}} \overset{\cup\ -\mathcal{S}_a(V,\tau)}{\longrightarrow} \underbrace{\mathcal{S}([V;\tau^{\mathsf{T}}],0)}_{\text{symmetric}}$$

computing $\mathcal{S}(V,\tau)$ can be partially symmetrized (see later).

Dual approach: avoid LOPs

- $s \in \{\pm 1\}^p$ is incompatible if $s \notin \mathcal{S}(V, \tau)$ ($s \in \mathcal{S}(V, \tau)^c$):

$$\nexists\, x \in \mathbb{R}^n : \qquad s \cdot (V^\mathsf{T} x - \tau) > 0,$$
$$\Leftrightarrow s \cdot V^\mathsf{T} x > s \cdot \tau.$$

- For $s \in \{\pm 1\}^p$ and $I \subseteq [1:p]$, s_I incompatible $\Rightarrow s$ is incompatible (more inequalities).
- For $s \in \{\pm 1\}^p$ incompatible, $\exists$ minimal incompatible s_I.
- With all incompatible s_I, no need for LO in the tree: check
- Find all the smallest I with s_I incompatible.

Dual approach: avoid LOPs

- $s \in \{\pm 1\}^p$ is incompatible if $s \notin \mathcal{S}(V, \tau)$ ($s \in \mathcal{S}(V, \tau)^c$):

$$\nexists\, x \in \mathbb{R}^n : \quad s \bullet (V^\top x - \tau) > 0,$$
$$\Leftrightarrow s \bullet V^\top x > s \bullet \tau.$$

- For $s \in \{\pm 1\}^p$ and $I \subseteq [1:p]$, s_I incompatible $\Rightarrow$ s is incompatible (more inequalities).
- For $s \in \{\pm 1\}^p$ incompatible, $\exists$ minimal incompatible s_I.
- With all incompatible s_I, no need for LO in the tree: check
- Find all the smallest I with s_I incompatible.

Dual approach: avoid LOPs

- $s \in \{\pm 1\}^p$ is incompatible if $s \notin \mathcal{S}(V, \tau)$ ($s \in \mathcal{S}(V, \tau)^c$):

$$\nexists\, x \in \mathbb{R}^n : \quad s \bullet (V^\top x - \tau) > 0, \\ \Leftrightarrow s \bullet V^\top x > s \bullet \tau.$$

- For $s \in \{\pm 1\}^p$ and $I \subseteq [1:p]$, s_I incompatible $\Rightarrow$ s is incompatible (more inequalities).
- For $s \in \{\pm 1\}^p$ incompatible, $\exists$ minimal incompatible s_I.
- With all incompatible s_I, no need for LO in the tree: check
- Find all the smallest I with s_I incompatible.

Dual approach: avoid LOPs

- $s \in \{\pm 1\}^p$ is incompatible if $s \notin \mathcal{S}(V, \tau)$ ($s \in \mathcal{S}(V, \tau)^c$):

$$\nexists\, x \in \mathbb{R}^n : \qquad s \bullet (V^\top x - \tau) > 0, \\ \Leftrightarrow s \bullet V^\top x > s \bullet \tau.$$

- For $s \in \{\pm 1\}^p$ and $I \subseteq [1:p]$, s_I incompatible $\Rightarrow$ s is incompatible (more inequalities).
- For $s \in \{\pm 1\}^p$ incompatible, $\exists$ minimal incompatible s_I.
- With all incompatible s_I, no need for LO in the tree: check
- Find all the smallest I with s_I incompatible.

Dual approach: avoid LOPs

- $s \in \{\pm 1\}^p$ is incompatible if $s \notin \mathcal{S}(V, \tau)$ ($s \in \mathcal{S}(V, \tau)^c$):

$$\nexists\, x \in \mathbb{R}^n : \quad s \bullet (V^\mathsf{T} x - \tau) > 0, \\ \Leftrightarrow s \bullet V^\mathsf{T} x > s \bullet \tau.$$

- For $s \in \{\pm 1\}^p$ and $I \subseteq [1:p]$, s_I incompatible $\Rightarrow$ s is incompatible (more inequalities).
- For $s \in \{\pm 1\}^p$ incompatible, $\exists$ minimal incompatible s_I.
- With all incompatible s_I, no need for LO in the tree: check
- Find all the smallest I with s_I incompatible.

Circuits and stem vectors – 1

A convex analysis tool: duality via Motzkin's alternative [Mot36]

$$\nexists\, x : Mx > m \quad \Longleftrightarrow \quad \exists\, \alpha \in \mathbb{R}^p_+ \setminus \{0\} : M^\mathsf{T}\alpha = 0, m^\mathsf{T}\alpha \geqslant 0.$$

$$s_I \text{ incompatible} \iff \nexists\, x \in \mathbb{R}^n : s_I \bullet V_{:,I}^\mathsf{T} x > s_I \bullet \tau_I$$

$$\iff \exists\, \alpha \in \mathbb{R}^I_+ \setminus \{0\} : V_{:,I}(\underbrace{s_I \bullet \alpha}_{=\eta \in \mathbb{R}^I}) = 0,\ \tau_I^\mathsf{T}(\underbrace{s_I \bullet \alpha}_{=\eta \in \mathbb{R}^I}) \geqslant 0.$$

The η is in $\mathcal{N}(V_{:,I}) \setminus \{0\}$, and oriented: $\tau_I^\mathsf{T}\eta \geqslant 0$ (otherwise: $-\eta$).

Circuits and stem vectors – 1

A convex analysis tool: duality via Motzkin's alternative [Mot36]

$$\nexists\, x : Mx > m \iff \exists\, \alpha \in \mathbb{R}_+^p \setminus \{0\} : M^\mathsf{T}\alpha = 0, m^\mathsf{T}\alpha \geqslant 0.$$

$$s_I \text{ incompatible} \iff \nexists\, x \in \mathbb{R}^n : s_I \bullet V_{:,I}^\mathsf{T} x > s_I \bullet \tau_I$$

$$\iff \exists\, \alpha \in \mathbb{R}_+^I \setminus \{0\} : V_{:,I}(\underbrace{s_I \bullet \alpha}_{=\eta\in\mathbb{R}^I}) = 0,\ \tau_I^\mathsf{T}(\underbrace{s_I \bullet \alpha}_{=\eta\in\mathbb{R}^I}) \geqslant 0.$$

The η is in $\mathcal{N}(V_{:,I}) \setminus \{0\}$, and oriented: $\tau_I^\mathsf{T}\eta \geqslant 0$ (otherwise: $-\eta$).

Circuits and stem vectors – 1

A convex analysis tool: duality via Motzkin's alternative [Mot36]

$$\nexists\, x : Mx > m \quad \Longleftrightarrow \quad \exists\, \alpha \in \mathbb{R}^p_+ \setminus \{0\} : M^\mathsf{T}\alpha = 0, m^\mathsf{T}\alpha \geqslant 0.$$

$$s_I \text{ incompatible } \Longleftrightarrow \nexists\, x \in \mathbb{R}^n : s_I \bullet V_{:,I}^\mathsf{T} x > s_I \bullet \tau_I$$
$$\Longleftrightarrow \exists\, \alpha \in \mathbb{R}^I_+ \setminus \{0\} : V_{:,I}(\underbrace{s_I \bullet \alpha}_{=\eta \in \mathbb{R}^I}) = 0,\ \tau_I^\mathsf{T}(\underbrace{s_I \bullet \alpha}_{=\eta \in \mathbb{R}^I}) \geqslant 0.$$

The η is in $\mathcal{N}(V_{:,I}) \setminus \{0\}$, and oriented: $\tau_I^\mathsf{T}\eta \geqslant 0$ (otherwise: $-\eta$).

Circuits and stem vectors – 2

$$s_I \text{ incompatible } \iff \exists\, \eta \in \mathbb{R}^I \setminus \{0\} : V_{:,I} \underbrace{\eta}_{s_I \bullet \alpha} = 0,\; \tau_I^\mathsf{T} \underbrace{\eta}_{s_I \bullet \alpha} \geqslant 0.$$

- Smallest I's, $\eta \in \mathcal{N}(V_{:,I}) \setminus \{0\} \Rightarrow$ *matroid circuits* of V [Oxl11]:

$$\mathcal{C}(V) := \{I \subseteq [1:p] : \underbrace{\operatorname{null}(V_{:,I})}_{\dim(\mathcal{N}(V_{:,I}))} = 1, \operatorname{null}(V_{:,I_0}) = 0\ \forall\, I_0 \subsetneq I\}$$

- *Stem vectors* $\mathfrak{S}(V,\tau) := \{\sigma \in \{\pm 1\}^I : I \in \mathcal{C}(V)$ and

$$\sigma = \operatorname{sgn}(\eta) \text{ for } \eta \in \mathcal{N}(V_{:,I}) \setminus \{0\} \text{ s.t. } \tau_I^\mathsf{T} \eta \geqslant 0\}.$$

Circuits and stem vectors – 2

$$s_I \text{ incompatible} \iff \exists\, \eta \in \mathbb{R}^I \setminus \{0\} : V_{:,I} \underbrace{\eta}_{s_I \bullet \alpha} = 0,\ \tau_I^\top \underbrace{\eta}_{s_I \bullet \alpha} \geqslant 0.$$

- Smallest I's, $\eta \in \mathcal{N}(V_{:,I}) \setminus \{0\} \Rightarrow$ *matroid circuits* of V [Oxl11]:

$$\mathcal{C}(V) := \{I \subseteq [1:p] : \underbrace{\text{null}(V_{:,I})}_{\dim(\mathcal{N}(V_{:,I}))} = 1, \text{null}(V_{:,I_0}) = 0\ \forall\ I_0 \subsetneq I\}$$

- *Stem vectors* $\mathfrak{S}(V,\tau) := \{\sigma \in \{\pm 1\}^I : I \in \mathcal{C}(V)$ and

$$\sigma = \text{sgn}(\eta) \text{ for } \eta \in \mathcal{N}(V_{:,I}) \setminus \{0\} \text{ s.t. } \tau_I^\top \eta \geqslant 0\}.$$

Circuits and stem vectors – 2

$$s_I \text{ incompatible} \iff \exists\, \eta \in \mathbb{R}^I \setminus \{0\} : V_{:,I} \underbrace{\eta}_{s_I \bullet \alpha} = 0,\ \tau_I^{\mathsf{T}} \underbrace{\eta}_{s_I \bullet \alpha} \geqslant 0.$$

- Smallest I's, $\eta \in \mathcal{N}(V_{:,I}) \setminus \{0\} \Rightarrow$ *matroid circuits* of V [Oxl11]:

$$\mathcal{C}(V) := \{I \subseteq [1:p] : \underbrace{\text{null}(V_{:,I})}_{\dim(\mathcal{N}(V_{:,I}))} = 1, \text{null}(V_{:,I_0}) = 0\ \forall\ I_0 \subsetneq I\}$$

- *Stem vectors* $\mathfrak{S}(V,\tau) := \{\sigma \in \{\pm 1\}^I : I \in \mathcal{C}(V)$ and

$$\sigma = \text{sgn}(\eta) \text{ for } \eta \in \mathcal{N}(V_{:,I}) \setminus \{0\} \text{ s.t. } \tau_I^{\mathsf{T}} \eta \geqslant 0\}.$$

Circuits and stem vectors – 3

Covering test

$$s \in \mathcal{S}(V,\tau)^c \quad \Longleftrightarrow \quad s_I \in \mathfrak{S}(V,\tau) \text{ for some } I \subseteq [1:p].$$

$$(\mathrm{sgn}(\eta) = \mathrm{sgn}(s_I \cdot \alpha) = \mathrm{sgn}(s_I) = s_I)$$

Dual algorithm: tree with covering tests

- Compute $\mathfrak{S}(V,\tau)$ (via $\mathcal{C}(V)$).
- Test if $(s, +1)$ covers a stem vector.
- If yes, stop; if no, recursion on $(s, +1)$.
- Same for $(s, -1)$.

Circuits and stem vectors – 3

Covering test

$$s \in \mathcal{S}(V,\tau)^c \quad \Longleftrightarrow \quad s_I \in \mathfrak{S}(V,\tau) \text{ for some } I \subseteq [1:p].$$

$$(\mathrm{sgn}(\eta) = \mathrm{sgn}(s_I \bullet \alpha) = \mathrm{sgn}(s_I) = s_I)$$

Dual algorithm: tree with covering tests

- Compute $\mathfrak{S}(V,\tau)$ (via $\mathcal{C}(V)$).
- Test if $(s,+1)$ covers a stem vector.
- If yes, stop; if no, recursion on $(s,+1)$.
- Same for $(s,-1)$.

Circuits and stem vectors – 3

Covering test

$$s \in \mathcal{S}(V,\tau)^c \iff s_I \in \mathfrak{S}(V,\tau) \text{ for some } I \subseteq [1:p].$$

$$(\text{sgn}(\eta) = \text{sgn}(s_I \cdot \alpha) = \text{sgn}(s_I) = s_I)$$

Dual algorithm: tree with covering tests

- Compute $\mathfrak{S}(V,\tau)$ (via $\mathcal{C}(V)$).
- Test if $(s,+1)$ covers a stem vector.
- If yes, stop; if no, recursion on $(s,+1)$.
- Same for $(s,-1)$.

Circuits and stem vectors – 3

Covering test

$$s \in \mathcal{S}(V,\tau)^c \quad \Longleftrightarrow \quad s_I \in \mathfrak{S}(V,\tau) \text{ for some } I \subseteq [1:p].$$

$$(\mathrm{sgn}(\eta) = \mathrm{sgn}(s_I \cdot \alpha) = \mathrm{sgn}(s_I) = s_I)$$

Dual algorithm: tree with covering tests

- Compute $\mathfrak{S}(V,\tau)$ (via $\mathcal{C}(V)$).
- Test if $(s,+1)$ covers a stem vector.
- If yes, stop; if no, recursion on $(s,+1)$.
- Same for $(s,-1)$.

Circuits and stem vectors – 3

Covering test

$$s \in \mathcal{S}(V,\tau)^c \quad \Longleftrightarrow \quad s_I \in \mathfrak{S}(V,\tau) \text{ for some } I \subseteq [1:p].$$

$$(\text{sgn}(\eta) = \text{sgn}(s_I \bullet \alpha) = \text{sgn}(s_I) = s_I)$$

Dual algorithm: tree with covering tests

- Compute $\mathfrak{S}(V,\tau)$ (via $\mathcal{C}(V)$).
- Test if $(s,+1)$ covers a stem vector.
- If yes, stop; if no, recursion on $(s,+1)$.
- Same for $(s,-1)$.

Comparison

each inner node	Primal	Dual
verification concretely	1 LOP: low-dimension	1-2 covering test(s): array operations

Computing $\mathfrak{S}(V, \tau)$ is a combinatorial problem.
If $|\mathfrak{S}(V, \tau)|$ large, long computation and covering tests.

Intermediate: Primal-Dual, only *some* duality

- launch the primal tree;
- $(s, \pm)$ incompatible $\overset{\text{Motzkin}}{\Longrightarrow}$ stem vector;
- use the *acquired* stem vectors (and still LOPs).

Comparison

each inner node	Primal	Dual
verification concretely	1 LOP: low-dimension	1-2 covering test(s): array operations

Computing $\mathfrak{S}(V, \tau)$ is a combinatorial problem.
If $|\mathfrak{S}(V, \tau)|$ large, long computation and covering tests.

Intermediate: Primal-Dual, only *some* duality

- launch the primal tree;
- $(s, \pm)$ incompatible $\overset{\text{Motzkin}}{\Longrightarrow}$ stem vector;
- use the *acquired* stem vectors (and still LOPs).

Comparison

each inner node	Primal	Dual
verification concretely	1 LOP: low-dimension	1-2 covering test(s): array operations

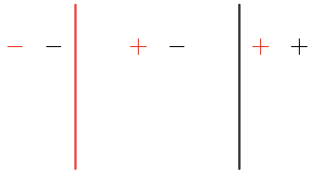
Computing $\mathfrak{S}(V, \tau)$ is a combinatorial problem.
If $|\mathfrak{S}(V, \tau)|$ large, long computation and covering tests.

Intermediate: Primal-Dual, only *some* duality

- launch the primal tree;
- $(s, \pm)$ incompatible $\overset{\text{Motzkin}}{\Longrightarrow}$ stem vector;
- use the *acquired* stem vectors (and still LOPs).

Illustration of duality

$M = s \cdot V^\mathsf{T}$, $m = s \cdot \tau$: $s \cdot (V^\mathsf{T} x - \tau) > 0 \Leftrightarrow s \cdot V^\mathsf{T} x > s \cdot \tau$



With $V = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}$ and $\tau = [-1; 1]$, $\{x : x_1 = -1\}$ and $\{x : x_1 = +1\}$.
No $-+$ since (geometrically) $-$: left to the red hyperplane and $+$ right to the black hyperplane. Algebraically, $-$ means $x_1 < -1$ and $+$ $x_1 > 1$.

$$\alpha = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \; (V \cdot [- \; +])\alpha = \begin{bmatrix} - & + \\ 0 & 0 \end{bmatrix} \alpha = 0, \; ([- \; +] \cdot \tau)\alpha = 2 \geqslant 0$$

About circuits/stem vectors

$$\mathcal{C}(V) := \{I \subseteq [1:p] : \mathrm{null}(V_{:,I}) = 1, \mathrm{null}(V_{:,I_0}) = 0 \ \forall\ I_0 \subsetneq I\}$$

No "good" algo (Rambau [Ram23]); adaptable for symmetries.
Upper bound $\binom{p}{r+1}$ [DSL06], $=$ under general position.
"Double punishment" for fully dual method.

For degenerate arrangements, short circuits so less susbets explored, but maybe lots of circuits (p large).
Ex: parallel hyperplanes – circuits of size 2 (so no larger subsets).

About circuits/stem vectors

$$\mathcal{C}(V) := \{I \subseteq [1:p] : \text{null}(V_{:,I}) = 1, \text{null}(V_{:,I_0}) = 0 \ \forall \ I_0 \subsetneq I\}$$

No "good" algo (Rambau [Ram23]); adaptable for symmetries.
Upper bound $\binom{p}{r+1}$ [DSL06], = under general position.
"Double punishment" for fully dual method.

For degenerate arrangements, short circuits so less susbets explored, but maybe lots of circuits (p large).
Ex: parallel hyperplanes – circuits of size 2 (so no larger subsets).

Affine or linear?

coning/homogeneization/embedding/lifting/...

$$\mathcal{S}\left(\begin{bmatrix} V & 0 \\ \tau & -1 \end{bmatrix}, 0\right) = [\mathcal{S}(V,\tau) \times \{+1\}] \cup [-\mathcal{S}(V,\tau) \times \{-1\}],$$

i.e., “an affine arrangement in dimension n is the upper [or lower] half of a centered arrangement in dimension $n+1$”.

Natural way so swap between affine and linear arrangements
$\mathcal{S}(V,\tau) :=$ affine$(n, p) \simeq$ linear$(n+1, p+1)$ (half of);
$\mathcal{S}(V,0) :=$ linear$(n, p) \simeq$ affine$(n-1, p-1)$ (two opposite).

General improvement: “compaction”

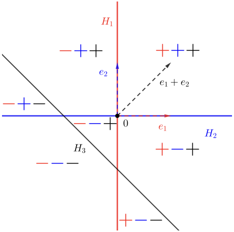
Principle

- $\mathcal{S}(V, \tau)$ (and tree) asymmetric, we can “symmetrize”.
- For all variants (RČ, P, D, PD).

General improvement: "compaction"

Principle

- $\mathcal{S}(V, \tau)$ (and tree) asymmetric, we can "symmetrize".
- For all variants (RČ, P, D, PD).

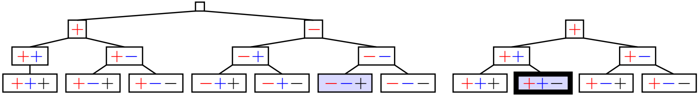


Asymmetric arrangement

$$\mathcal{S}(V, \tau) = \{(+++), (-++), (+-+), (--+), (---), (+--), (-+-)\}$$

except $(--+)$, rest symmetric

Compaction illustrated



Classic tree. Compact tree.

Blued nodes: asymmetric nodes, correction in the right tree. At the end, the other nodes are multipled by -1 to recover all nodes.

Details on compaction

$$\begin{cases} \mathcal{S}(V,0) & := \{s \in \{\pm 1\}^p : \exists\, x^s \in \mathbb{R}^n : s \bullet V^\mathsf{T} x^s > 0\} \\ \mathcal{S}(V,\tau) & := \{s \in \{\pm 1\}^p : \exists\, x^s \in \mathbb{R}^n : s \bullet (V^\mathsf{T} x^s - \tau) > 0\} \\ \mathcal{S}([V;\tau^\mathsf{T}],0) & := \{s \in \{\pm 1\}^p : \exists\, d^s \in \mathbb{R}^{n+1} : s \bullet [V^\mathsf{T}\ \tau] d^s > 0\} \end{cases}$$

$\mathcal{S}(V,\tau)$ has a *symmetric part* (not perfectly geometrically).

$\mathcal{S}(V,\tau)$ exactly between $\mathcal{S}(V,0)$ and $\mathcal{S}([V;\tau^\mathsf{T}],0)$ (symmetric).

Possible to quantify the difference in # of LOPs.

Compute less than $|\mathcal{S}(V,\tau)|$ chambers.

Details on compaction

$$\begin{cases} \mathcal{S}(V,0) & := \{s \in \{\pm 1\}^p : \exists\ x^s \in \mathbb{R}^n : s \bullet V^\top x^s > 0\} \\ \mathcal{S}(V,\tau) & := \{s \in \{\pm 1\}^p : \exists\ x^s \in \mathbb{R}^n : s \bullet (V^\top x^s - \tau) > 0\} \\ \mathcal{S}([V;\tau^\top],0) & := \{s \in \{\pm 1\}^p : \exists\ d^s \in \mathbb{R}^{n+1} : s \bullet [V^\top\ \tau] d^s > 0\} \end{cases}$$

$\mathcal{S}(V,\tau)$ has a *symmetric part* (not perfectly geometrically).

$\mathcal{S}(V,\tau)$ exactly between $\mathcal{S}(V,0)$ and $\mathcal{S}([V;\tau^\top],0)$ (symmetric).

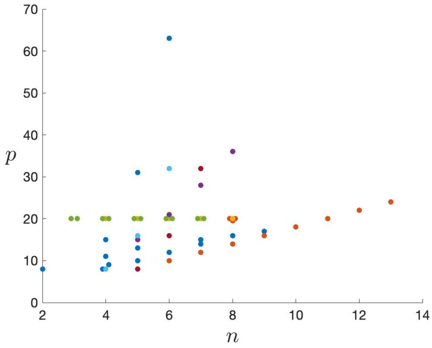
Possible to quantify the difference in # of LOPs.

Compute less than $|\mathcal{S}(V,\tau)|$ chambers.

Algorithms and instances

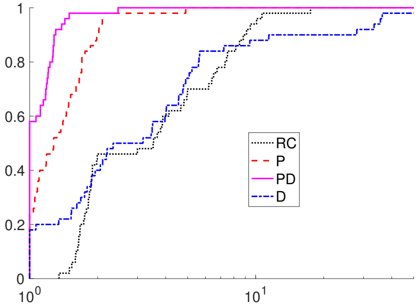
- Basic: [RČ18] – "RČ" (Rada Černý).
- With heuristics – "P" (Primal).
- Without LOPs, just stem vectors – "D" (Dual).
- LOPs and some stem vectors – "PD" (Primal-Dual).
- Relevance of compaction (/C).

Details on instances



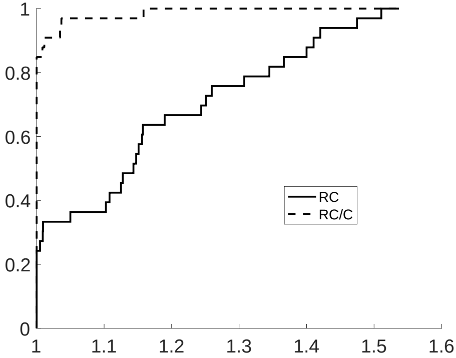
Pairs (n, p) for some linear and affine instances, grouped by colors. Instances up to 10^6 chambers/circuits (to run on a laptop). Example: $n = 7, p = 20$, up to 137980 chambers, 125970 stem vectors.

Comparison of the main variants



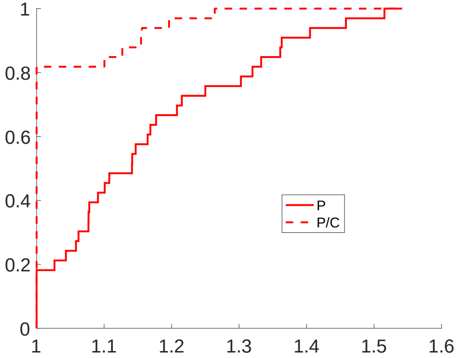
x-axis: relative efficiency (on time), y-axis: % of problems; above/left means being better. One has: primal-dual (PD) > primal (P) on some instances, both > Rada-Černý (RČ) and dual (D), which are quite close.

Variant vs compact variant



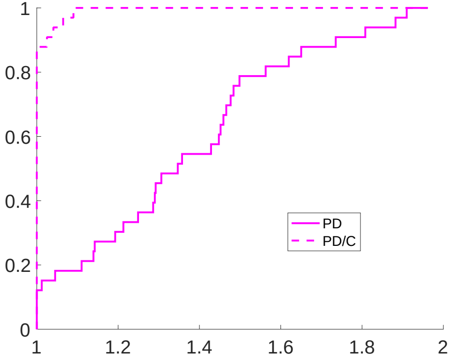
Compaction improves Rada-Černý RC, primal P and primal-dual PD (axis up to 2), but not really dual (D): less tests but more stems.

Variant vs compact variant



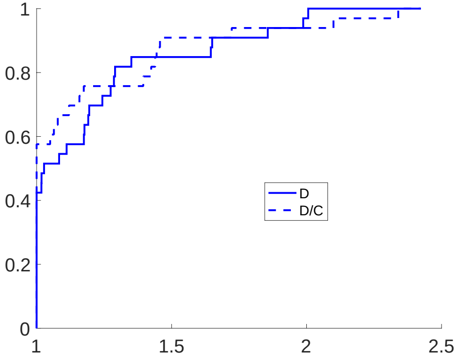
Compaction improves Rada-Černý RC, primal P and primal-dual PD (axis up to 2), but not really dual (D): less tests but more stems.

Variant vs compact variant



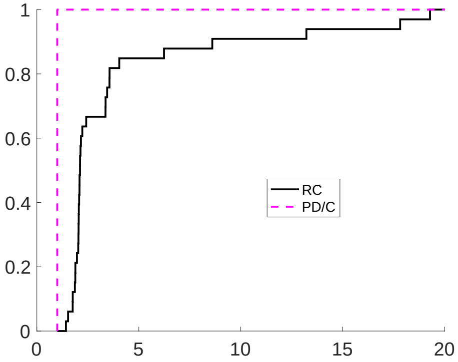
Compaction improves Rada-Černý RC, primal P and primal-dual PD (axis up to 2), but not really dual (D): less tests but more stems.

Variant vs compact variant



Compaction improves Rada-Černý RC, primal P and primal-dual PD (axis up to 2), but not really dual (D): less tests but more stems.

Initial vs best algorithm [not updated results]



Larger x-axis: average $\simeq 4$. Especially better on "structured" instances.

Possible code improvements: data structures, parallelism. . .

One last technique

Combinatorial symmetries

For instances where “all dimension are equivalent”, inspired from [BEK23] (just $|\mathcal{S}(V,\tau)|$) and [Ram23] ($\mathcal{C}(V)$ and other stuff).

$$V = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}, \tau = 0$$

Dimensions (rows) can be interchanged.

One last technique

Combinatorial symmetries

For instances where "all dimension are equivalent", inspired from [BEK23] (just $|\mathcal{S}(V,\tau)|$) and [Ram23] ($\mathcal{C}(V)$ and other stuff).

$$V = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}, \tau = 0$$

Dimensions (rows) can be interchanged.
Idea: just consider a part of the tree (a part of the space), obtain the rest by combinatorial symmetry.

Such instances have interest for combinatoricians.

Illustration

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}, p = 2^n - 1$$

Consider $\mathcal{O}_1 := \{x \in \mathbb{R}^n : x_1 < 0, x_{[2:n]} > 0\}$ corresponding to $(-, +, \ldots, +) \in \{-, +\}^n$.

Illustration

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}, p = 2^n - 1$$

Consider $\mathcal{O}_1 := \{x \in \mathbb{R}^n : x_1 < 0, x_{[2:n]} > 0\}$ corresponding to $(-, +, \ldots, +) \in \{-, +\}^n$.

Launch the RČ subtree and compute this part.

Illustration

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}, p = 2^n - 1$$

Consider $\mathcal{O}_1 := \{x \in \mathbb{R}^n : x_1 < 0, x_{[2:n]} > 0\}$ corresponding to $(-, +, \ldots, +) \in \{-, +\}^n$.

Launch the RČ subtree and compute this part.

The orthants $\mathcal{O}_2 := \{x \in \mathbb{R}^n : x_1 > 0, x_2 < 0, x_{[3:n]} > 0\}, \ldots,$ $\mathcal{O}_n := \{x \in \mathbb{R}^n : x_{[1:n-1]} > 0, x_n < 0\}$ are all equivalent to $\mathcal{O}_1$.

Illustration

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}, p = 2^n - 1$$

Consider $\mathcal{O}_1 := \{x \in \mathbb{R}^n : x_1 < 0, x_{[2:n]} > 0\}$ corresponding to $(-, +, \dots, +) \in \{-, +\}^n$.

Launch the RČ subtree and compute this part.

The orthants $\mathcal{O}_2 := \{x \in \mathbb{R}^n : x_1 > 0, x_2 < 0, x_{[3:n]} > 0\}, \dots,$ $\mathcal{O}_n := \{x \in \mathbb{R}^n : x_{[1:n-1]} > 0, x_n < 0\}$ are all equivalent to $\mathcal{O}_1$.

Same with 2 components $x_i < 0$, rest > 0, then 3 < 0...

The unifying method, Merino Mütze [MM24]?

$\{\pm 1\} \to \{0, 1\}$, *connected* vertices X of the hypercube.
A priori: the path may not be connected in $\mathbb{R}^n$;
To next chamber: binary variable, not LO

$$\min_{y,z} w^\top (y - x), \quad y_{P_0} = 0, \quad y_{P_1} = 1, \quad (2y - 1) \bullet (V^\top z - \tau) > 0?$$

For vertices of $P = \{z : Az \leqslant b\}$ **assumes it is a** $\mathrm{conv}(X)$ from A and b. (Not obvious according to Ziegler [Zie99]?)

For circuits? $x(C)_i := \mathbb{1}(i \in C)$, $x(C) \in \{0, 1\}^n$, $C(x) = \cup_{x_j = 1}\{j\}$.
No "swaps" (flips) for circuits. The exchange axiom: 3 circuits...

$$\min_{y} w^\top (y - x), \; y_{P_0} = 0, \; y_{P_1} = 1, \; \begin{cases} \mathrm{null}(V_{:,C(y)}) = 1, \\ \mathrm{null}(V_{:,C'}) = 0, C' \subsetneq C(y)? \end{cases}$$

The unifying method, Merino Mütze [MM24]?

$\{\pm 1\} \to \{0,1\}$, *connected* vertices X of the hypercube.
A priori: the path may not be connected in $\mathbb{R}^n$;
To next chamber: binary variable, not LO

$$\min_{y,z} w^\top (y - x), \quad y_{P_0} = 0, \quad y_{P_1} = 1, \quad (2y - 1) \bullet (V^\top z - \tau) > 0?$$

For vertices of $P = \{z : Az \leqslant b\}$ **assumes it is a** $\mathrm{conv}(X)$ from A and b. (Not obvious according to Ziegler [Zie99]?)

For circuits? $x(C)_i := \mathbb{1}(i \in C)$, $x(C) \in \{0,1\}^n$, $C(x) = \cup_{x_j=1}\{j\}$.
No "swaps" (flips) for circuits. The exchange axiom: 3 circuits...

$$\min_{y} w^\top (y - x), \quad y_{P_0} = 0, \quad y_{P_1} = 1, \quad \begin{cases} \mathrm{null}(V_{:,C(y)}) = 1, \\ \mathrm{null}(V_{:,C'}) = 0, C' \subsetneq C(y)? \end{cases}$$

The unifying method, Merino Mütze [MM24]?

$\{\pm 1\} \to \{0,1\}$, *connected* vertices X of the hypercube.
A priori: the path may not be connected in $\mathbb{R}^n$;
To next chamber: binary variable, not LO

$$\min_{y,z} w^\top (y-x), \quad y_{P_0} = 0, \quad y_{P_1} = 1, \quad (2y-1) \bullet (V^\top z - \tau) > 0?$$

For vertices of $P = \{z : Az \leqslant b\}$ **assumes it is a** $\mathrm{conv}(X)$ from A and b. (Not obvious according to Ziegler [Zie99]?)

For circuits? $x(C)_i := \mathbb{1}(i \in C)$, $x(C) \in \{0,1\}^n$, $C(x) = \cup_{x_j=1}\{j\}$.
No "swaps" (flips) for circuits. The exchange axiom: 3 circuits...

$$\min_y w^\top (y-x), \quad y_{P_0} = 0, \quad y_{P_1} = 1, \quad \begin{cases} \mathrm{null}(V_{:,C(y)}) = 1, \\ \mathrm{null}(V_{:,C'}) = 0, C' \subsetneq C(y)? \end{cases}$$

Full arrangements: not only halfspaces

With $\{-1, 0, +1\}^p$, what changes? $2^p \to 3^p$,
known bounds (general position), RC algorithm with ternary tree.

Some things need to be adapted: especially compaction (relations).
Main issue: equalities ($s_i = 0$) are not maintained if $\tau \neq 0$.

For σ's, no changes? "chamber infeasible has no boundary":
so stem vectors $\sigma \in \{\pm 1\}^I$ mean every "$s^I \in [0, \sigma]$" infeasible too.

Algorithmically? Tree has 1 / 3 descendants (two ⇒ third).
Compute chambers, join neighbors for $n - 1$ subchambers, $n - 2$...
Compute intersections of H_i's and binary trees on them, project in the subspaces...

Full arrangements: not only halfspaces

With $\{-1, 0, +1\}^p$, what changes? $2^p \to 3^p$,
known bounds (general position), RC algorithm with ternary tree.

Some things need to be adapted: especially compaction (relations).
Main issue: equalities ($s_i = 0$) are not maintained if $\tau \neq 0$.

For σ's, no changes? "chamber infeasible has no boundary":
so stem vectors $\sigma \in \{\pm 1\}^I$ mean every "$s^I \in [0, \sigma]$" infeasible too.

Algorithmically? Tree has 1 / 3 descendants (two ⇒ third).
Compute chambers, join neighbors for $n-1$ subchambers, $n-2$...
Compute intersections of H_i's and binary trees on them, project in the subspaces...

Full arrangements: not only halfspaces

With $\{-1, 0, +1\}^p$, what changes? $2^p \to 3^p$,
known bounds (general position), RC algorithm with ternary tree.

Some things need to be adapted: especially compaction (relations).
Main issue: equalities ($s_i = 0$) are not maintained if $\tau \neq 0$.

For σ's, no changes? "chamber infeasible has no boundary":
so stem vectors $\sigma \in \{\pm 1\}^I$ mean every "$s^I \in [0, \sigma]$" infeasible too.

Algorithmically? Tree has 1 / 3 descendants (two $\Rightarrow$ third).
Compute chambers, join neighbors for $n-1$ subchambers, $n-2$...
Compute intersections of H_i's and binary trees on them, project in the subspaces...

Full arrangements: not only halfspaces

With $\{-1, 0, +1\}^p$, what changes? $2^p \to 3^p$,
known bounds (general position), RC algorithm with ternary tree.

Some things need to be adapted: especially compaction (relations).
Main issue: equalities ($s_i = 0$) are not maintained if $\tau \neq 0$.

For σ's, no changes? "chamber infeasible has no boundary":
so stem vectors $\sigma \in \{\pm 1\}^I$ mean every "$s^I \in [0, \sigma]$" infeasible too.

Algorithmically? Tree has 1 / 3 descendants (two ⇒ third).
Compute chambers, join neighbors for $n-1$ subchambers, $n-2$...
Compute intersections of H_i's and binary trees on them, project in the subspaces...

About duality

- one stem vector ⇔ empty region $s \in \{-1, +1\}^J$ in the subarrangement with J;
- that empty region has no boundary (or empty boundary);
- putting some s_i to zeros: infeasible since it corresponds to the boundary of the chamber of s.

Good surprise

Only minor changes are required to the dual notions.

About duality

- one stem vector $\Leftrightarrow$ empty region $s \in \{-1, +1\}^J$ in the subarrangement with J;
- that empty region has no boundary (or empty boundary);
- putting some s_i to zeros: infeasible since it corresponds to the boundary of the chamber of s.

Good surprise

Only minor changes are required to the dual notions.

About duality

- one stem vector ⇔ empty region $s \in \{-1, +1\}^J$ in the subarrangement with J;
- that empty region has no boundary (or empty boundary);
- putting some s_i to zeros: infeasible since it corresponds to the boundary of the chamber of s.

Good surprise

Only minor changes are required to the dual notions.

About duality

- one stem vector $\Leftrightarrow$ empty region $s \in \{-1, +1\}^J$ in the subarrangement with J;
- that empty region has no boundary (or empty boundary);
- putting some s_i to zeros: infeasible since it corresponds to the boundary of the chamber of s.

Good surprise

Only minor changes are required to the dual notions.

Applications for the whole arrangement [EOS86]

theoretical / for complexity results, lots of “we present in dimension 2/3 and generalizations are clearly straightforward”

- “n-dimensional sorting”, “λ-matrices”
- K-Voronoi diagrams (points closest to a subsets of K points instead of just one)
- Testing degeneracy / general position (see later).
- Among p points, find the $n+1$ forming the smallest (in measure) simplex.

Applications for the whole arrangement [EOS86]

theoretical / for complexity results, lots of "we present in dimension 2/3 and generalizations are clearly straightforward"

- "n-dimensional sorting", "λ-matrices"
- K-Voronoi diagrams (points closest to a subsets of K points instead of just one)
- Testing degeneracy / general position (see later).
- Among p points, find the $n+1$ forming the smallest (in measure) simplex.

Applications for the whole arrangement [EOS86]

theoretical / for complexity results, lots of "we present in dimension 2/3 and generalizations are clearly straightforward"

- "n-dimensional sorting", "λ-matrices"
- K-Voronoi diagrams (points closest to a subsets of K points instead of just one)
- Testing degeneracy / general position (see later).
- Among p points, find the $n+1$ forming the smallest (in measure) simplex.

Applications for the whole arrangement [EOS86]

theoretical / for complexity results, lots of "we present in dimension 2/3 and generalizations are clearly straightforward"

- "n-dimensional sorting", "λ-matrices"
- K-Voronoi diagrams (points closest to a subsets of K points instead of just one)
- Testing degeneracy / general position (see later).
- Among p points, find the $n+1$ forming the smallest (in measure) simplex.

Some properties

- up to 3^p sign vectors (objects) to identify,
- similar bounds in general position: for cells of dimension $k \in [0:n]$, $\binom{p}{n-k}\sum_{i=0}^{k}\binom{p-n+k}{i}$,
- formulas exists but more complicated,
- some symmetry properties hold,
- but not all: $\mathcal{S}_s(V,\tau) \neq \mathcal{S}(V,0)$: $\mathcal{S}(V,0)$ is centered so contains $(0,\ldots,0)$, which isn't $\mathcal{S}(V,\tau)$ unless centered,
- good surprise: dual aspect may be kept with nearly no change.

Some properties

- up to 3^p sign vectors (objects) to identify,
- similar bounds in general position: for cells of dimension $k \in [0:n]$, $\binom{p}{n-k} \sum_{i=0}^{k} \binom{p-n+k}{i}$,
- formulas exists but more complicated,
- some symmetry properties hold,
- but not all: $\mathcal{S}_s(V,\tau) \neq \mathcal{S}(V,0)$: $\mathcal{S}(V,0)$ is centered so contains $(0,\ldots,0)$, which isn't $\mathcal{S}(V,\tau)$ unless centered,
- good surprise: dual aspect may be kept with nearly no change.

Some properties

- up to 3^p sign vectors (objects) to identify,
- similar bounds in general position: for cells of dimension $k \in [0:n]$, $\binom{p}{n-k} \sum_{i=0}^{k} \binom{p-n+k}{i}$,
- formulas exists but more complicated,
- some symmetry properties hold,
- but not all: $\mathcal{S}_s(V,\tau) \neq \mathcal{S}(V,0)$: $\mathcal{S}(V,0)$ is centered so contains $(0,\ldots,0)$, which isn't $\mathcal{S}(V,\tau)$ unless centered,
- good surprise: dual aspect may be kept with nearly no change.

Some properties

- up to 3^p sign vectors (objects) to identify,
- similar bounds in general position: for cells of dimension $k \in [0 : n]$, $\binom{p}{n-k} \sum_{i=0}^{k} \binom{p-n+k}{i}$,
- formulas exists but more complicated,
- some symmetry properties hold,
- but not all: $\mathcal{S}_s(V, \tau) \neq \mathcal{S}(V, 0)$: $\mathcal{S}(V, 0)$ is centered so contains $(0, \ldots, 0)$, which isn't $\mathcal{S}(V, \tau)$ unless centered,
- good surprise: dual aspect may be kept with nearly no change.

Some properties

- up to 3^p sign vectors (objects) to identify,
- similar bounds in general position: for cells of dimension $k \in [0 : n]$, $\binom{p}{n-k} \sum_{i=0}^{k} \binom{p-n+k}{i}$,
- formulas exists but more complicated,
- some symmetry properties hold,
- but not all: $\mathcal{S}_s(V, \tau) \neq \mathcal{S}(V, 0)$: $\mathcal{S}(V, 0)$ is centered so contains $(0, \dots, 0)$, which isn't $\mathcal{S}(V, \tau)$ unless centered,
- good surprise: dual aspect may be kept with nearly no change.

Some properties

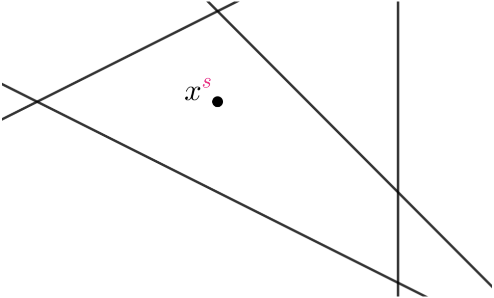
- up to 3^p sign vectors (objects) to identify,
- similar bounds in general position: for cells of dimension $k \in [0:n]$, $\binom{p}{n-k} \sum_{i=0}^{k} \binom{p-n+k}{i}$,
- formulas exists but more complicated,
- some symmetry properties hold,
- but not all: $\mathcal{S}_s(V,\tau) \neq \mathcal{S}(V,0)$: $\mathcal{S}(V,0)$ is centered so contains $(0,\ldots,0)$, which isn't $\mathcal{S}(V,\tau)$ unless centered,
- good surprise: dual aspect may be kept with nearly no change.

For the RČ algorithm

- number of descendants $\in \{1, 3\}$ (not 2),
- from 2 descendants you can obtain a third without “heavy” computation (so still only 1 LOP),

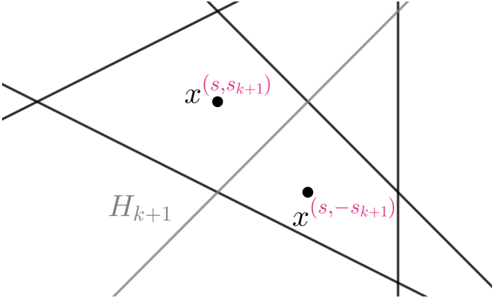
For the RČ algorithm

- number of descendants $\in \{1, 3\}$ (not 2),
- from 2 descendants you can obtain a third without "heavy" computation (so still only 1 LOP),



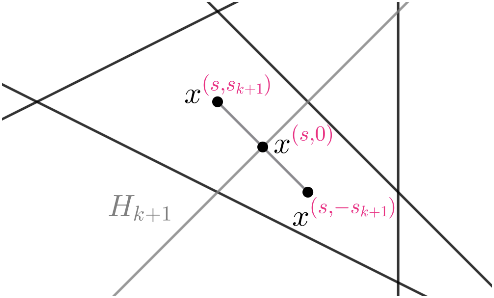
For the RČ algorithm

- number of descendants $\in \{1, 3\}$ (not 2),
- from 2 descendants you can obtain a third without "heavy" computation (so still only 1 LOP),



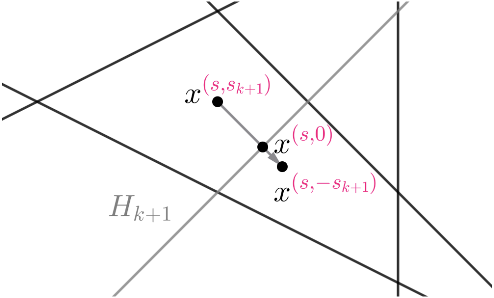
For the RČ algorithm

- number of descendants $\in \{1, 3\}$ (not 2),
- from 2 descendants you can obtain a third without "heavy" computation (so still only 1 LOP),



For the RČ algorithm

- number of descendants $\in \{1, 3\}$ (not 2),
- from 2 descendants you can obtain a third without "heavy" computation (so still only 1 LOP),



Variations of the algorithm

Idea 1: project the data in the subspaces (the $s_i = 0$)

Reduce the size of the LOPs, but chaining projections may be bad for precision / redundancy / ...

Idea 2: compute the "intersections"

Compute the nonempty $H_K := \cap_{k \in K} H_k$, project the hyperplanes H_i, $i \notin K$ in the subspace H_K then launch a smaller RČ in each.

- not sure how different it really is
- requires to compute the intersections ([BN82] does that and suggests a method but nothing is said)
- may require some redundant computations as well (projecting multiple times the data)

Variations of the algorithm

Idea 1: project the data in the subspaces (the $s_i = 0$)

Reduce the size of the LOPs, but chaining projections may be bad for precision / redundancy / ...

Idea 2: compute the "intersections"

Compute the nonempty $H_K := \cap_{k \in K} H_k$, project the hyperplanes H_i, $i \notin K$ in the subspace H_K then launch a smaller RČ in each.

- not sure how different it really is
- requires to compute the intersections ([BN82] does that and suggests a method but nothing is said)
- may require some redundant computations as well (projecting multiple times the data)

Other ideas

Combining the regions

With the n-dimensional chambers, combine the neighbors to get the ones of dimension $n-1$, then $n-2$...

May benefit from the adjacency property (so not RČ?), or from a clever storing of the chambers...

Other ideas

Combining the regions

With the n-dimensional chambers, combine the neighbors to get the ones of dimension $n-1$, then $n-2$...

May benefit from the adjacency property (so not RČ?), or from a clever storing of the chambers...