From non-differentiability to discrete geometry

B-differential, hyperplanes and matroids

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

October, 05 2023

Outline

Plan

Linear Complementarity Problems

General form [CPS92; FP03]

$$A, B \in \mathbb{R}^{n\times n}, a, b \in \mathbb{R}^n, \rightarrow \ \mathcal{A}(x) = Ax + a, \mathcal{B}(x) = Bx + b$$

$$0 \leq (Ax + a) \perp (Bx + b) \geq 0 \tag{1}$$

$$\forall\ i \in [1:n], \begin{cases} A_{i,:}x + a_i \geq 0 \\ B_{i,:}x + b_i \geq 0 \end{cases}, (A_{i,:}x + a_i)(B_{i,:}x + b_i) = 0$$

Remark: $u \geq 0, v \geq 0, uv = 0 \Leftrightarrow \min(u, v) = 0$

$(1) \Leftrightarrow \forall\ i \in [1:n], F_i(x) := \min(\mathcal{A}_i(x), \mathcal{B}_i(x)) = 0 \Leftrightarrow F(x) = 0$

Linear Complementarity Problems

General form [CPS92; FP03]

$$A, B \in \mathbb{R}^{n\times n}, a, b \in \mathbb{R}^n, \to \ \mathcal{A}(x) = Ax + a, \mathcal{B}(x) = Bx + b$$

$$0 \leq (Ax + a) \perp (Bx + b) \geq 0 \qquad (1)$$

$$\forall\ i \in [1:n], \begin{cases} A_{i,:}x + a_i \geq 0 \\ B_{i,:}x + b_i \geq 0 \end{cases}, (A_{i,:}x + a_i)(B_{i,:}x + b_i) = 0$$

Remark: $u \geq 0, v \geq 0, uv = 0 \Leftrightarrow \min(u, v) = 0$

$(1) \Leftrightarrow \forall\ i \in [1:n], F_i(x) := \min(\mathcal{A}_i(x), \mathcal{B}_i(x)) = 0 \Leftrightarrow F(x) = 0$

Linear Complementarity Problems

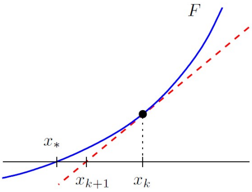
General form [CPS92; FP03]

$$
\begin{gathered}
A, B \in \mathbb{R}^{n\times n}, a, b \in \mathbb{R}^n, \to \ \mathcal{A}(x) = Ax + a, \mathcal{B}(x) = Bx + b \\
0 \leq (Ax + a) \perp (Bx + b) \geq 0 \\
\forall\ i \in [1:n], \begin{cases} A_{i,:}x + a_i \geq 0 \\ B_{i,:}x + b_i \geq 0 \end{cases}, (A_{i,:}x + a_i)(B_{i,:}x + b_i) = 0
\end{gathered}
\quad (1)
$$

Remark: $u \geq 0, v \geq 0, uv = 0 \Leftrightarrow \min(u, v) = 0$

$(1) \Leftrightarrow \forall\ i \in [1:n], F_i(x) := \min(\mathcal{A}_i(x), \mathcal{B}_i(x)) = 0 \Leftrightarrow F(x) = 0$

Nonlinear (nonsmooth) equations - Newton's method



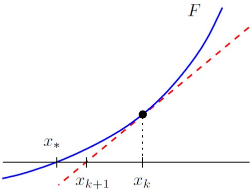
1D Illustration

x^0 near x^*,
$F \in \mathcal{C}^{1,1}$,
$F'(x^*)$ non-singular
$\Rightarrow$ quadratic convergence

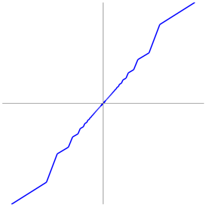
F' not defined everywhere

x^0 near x^*,
F semismooth,
all $J \in$ "$F'(x^*)$" non-singular
$\Rightarrow$ quadratic convergence

Nonlinear (nonsmooth) equations - Newton's method



1D Illustration



F' not defined everywhere

x^0 near x^*,
$F \in \mathcal{C}^{1,1}$,
$F'(x^*)$ non-singular
$\Rightarrow$ quadratic convergence

x^0 near x^*,
F semismooth,
all $J \in$ "$F'(x^*)$" non-singular
$\Rightarrow$ quadratic convergence

Semismooth Newton's method

Adaptation of the usual method for this difficulty ([Qi93; QS93])
Replaces $F'(x_k)$ with a "generalized Jacobian" J_k

Algorithm's sketch

- take $x^0 \in \mathbb{R}^n$ (near x^*)
- for k = 1 2, . . . , solve $F(x^k) + J_k z^k = 0$ for z^k, with $J_k \in \partial_B F(x^k)$: $\partial_B F$ is the Bouligand differential
- then $x^{k+1} = x^k + (\alpha_k) z^k$

Semismooth Newton's method

Adaptation of the usual method for this difficulty ([Qi93; QS93])
Replaces $F'(x_k)$ with a "generalized Jacobian" J_k

Algorithm's sketch

- take $x^0 \in \mathbb{R}^n$ (near x^*)
- for k = 1 2, . . . , solve $F(x^k) + J_k z^k = 0$ for z^k, with $J_k \in \partial_B F(x^k)$: $\partial_B F$ is the Bouligand differential
- then $x^{k+1} = x^k + (\alpha_k) z^k$

Semismooth Newton's method

Adaptation of the usual method for this difficulty ([Qi93; QS93])
Replaces $F'(x_k)$ with a "generalized Jacobian" J_k

Algorithm's sketch

- take $x^0 \in \mathbb{R}^n$ (near x^*)
- for k = 1 2, ..., solve $F(x^k) + J_k z^k = 0$ for z^k, with $J_k \in \partial_B F(x^k)$: $\partial_B F$ is the Bouligand differential
- then $x^{k+1} = x^k + (\alpha_k) z^k$

Semismooth Newton's method

Adaptation of the usual method for this difficulty ([Qi93; QS93])
Replaces $F'(x_k)$ with a "generalized Jacobian" J_k

Algorithm's sketch

- take $x^0 \in \mathbb{R}^n$ (near x^*)
- for k = 1 2, . . . , solve $F(x^k) + J_k z^k = 0$ for z^k, with $J_k \in \partial_B F(x^k)$: $\partial_B F$ is the Bouligand differential
- then $x^{k+1} = x^k + (\alpha_k) z^k$

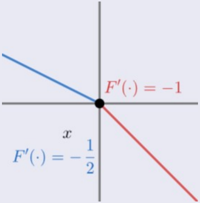
Generalized derivatives

Bouligand differential

$$\partial_{\mathrm{B}} F(x) = \{J \in \mathbb{R}^{n\times n} : \exists\ (x_k)_k \to x, F'(x_k) \to J\} \tag{2}$$

Example: $F(x) = \begin{cases} -x/2 & \text{if } x \leq 0 \\ -x & \text{if } x > 0 \end{cases}$,

$\partial_B F(0) = \{-1/2, -1\}$.



Summary

Minimum(LCPs) $\Rightarrow$ semismooth system, requires info on $\partial_B \min(\mathcal{A}, \mathcal{B})(\cdot) = \partial_B F(\cdot)$

One $J_B \in \partial_B F$: [Qi93]. But all of them?

The main question

Determine generalized Jacobians of $x \mapsto F(x) = \min(Ax + a, Bx + b)$

→ structure?
→ number?
→ computation?

Summary

Minimum(LCPs) $\Rightarrow$ semismooth system, requires info on $\partial_B \min(\mathcal{A}, \mathcal{B})(\cdot) = \partial_B F(\cdot)$

One $J_B \in \partial_B F$: [Qi93]. But all of them?

The main question

Determine generalized Jacobians of $x \mapsto F(x) = \min(Ax + a, Bx + b)$

→ structure?
→ number?
→ computation?

Summary

Minimum(LCPs) $\Rightarrow$ semismooth system, requires info on
$\partial_B \min(\mathcal{A}, \mathcal{B})(\cdot) = \partial_B F(\cdot)$

One $J_B \in \partial_B F$: [Qi93]. But all of them?

The main question

Determine generalized Jacobians of
$x \mapsto F(x) = \min(Ax + a, Bx + b)$

$\rightarrow$ structure?
$\rightarrow$ number?
$\rightarrow$ computation?

Plan

Computing the B-differential - 1

$$\min(\overbrace{f(x), g(x)}^{\in\mathbb{R}}) \text{ NOT diff} \Leftrightarrow f(x) \overset{C1}{=} g(x) \text{ and } f'(x) \overset{C2}{\neq} g'(x).$$

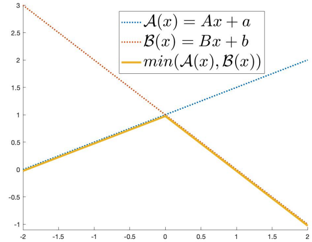


Illustration for 1D affine functions ($\rightarrow$ dimension n)

$$I(x) := \{i \in [1:n] : A_{i,:}x + a_i \overset{C1}{=} B_{i,:}x + b_i, A_{i,:} \overset{C2}{\neq} B_{i,:}\}; |I(x)| = p$$

Computing the B-differential - 1

$$\min(\overbrace{f(x), g(x)}^{\in\mathbb{R}}) \text{ NOT diff} \Leftrightarrow f(x) \overset{C1}{=} g(x) \text{ and } f'(x) \overset{C2}{\neq} g'(x).$$

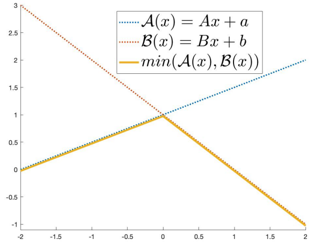


Illustration for 1D affine functions ($\rightarrow$ dimension n)

$$I(x) := \{i \in [1:n] : A_{i,:}x + a_i \overset{C1}{=} B_{i,:}x + b_i, A_{i,:} \overset{C2}{\neq} B_{i,:}\}; |I(x)| = p$$

Computing the B-differential - 2

$\min(\mathcal{A},\mathcal{B})(\overset{\simeq x}{x_k})$ non-diff $\Leftrightarrow \overset{\in I(x)}{\exists\ i}, (Ax_k + a)_i \overset{C1}{=} (Bx_k + b)_i \overset{x_k = x+d}{\Leftrightarrow}$
$(Ax + a)_i + A_{i,:}d = (Bx + b)_i + B_{i,:}d \Leftrightarrow A_{i,:}d = B_{i,:}d \Leftrightarrow d \in v_i^\perp$
$(v_i \neq 0$ by C2)

Hyperplanes $H_i := (B_{i,:} - A_{i,:})^\perp := v_i^\perp$; for ∂_B's def, $\mathbb{R}^n \setminus \cup H_i$

$$\mathbb{R}^n = H_i^- \cup H_i \cup H_i^+, \quad \begin{cases} H_i^- = \{x \in \mathbb{R}^n : v_i^\mathsf{T} x < 0\} \\ H_i^+ = \{x \in \mathbb{R}^n : v_i^\mathsf{T} x > 0\} \end{cases}$$

Convention: $\forall\ i \in [1:p]$,

$$\begin{aligned} H_i^+ &\Leftrightarrow B_{i,:}d - A_{i,:}d > 0 \Leftrightarrow \min(\dots) = \mathcal{A}_i(\dots) \Leftrightarrow J_{i,:} = A_{i,:} \\ H_i^- &\Leftrightarrow B_{i,:}d - A_{i,:}d < 0 \Leftrightarrow \min(\dots) = \mathcal{B}_i(\dots) \Leftrightarrow J_{i,:} = B_{i,:} \end{aligned}$$

so $\forall\ J, \forall\ i, J_{i,:} \in \{A_{i,:}, B_{i,:}\}$

Computing the B-differential - 2

$\min(\mathcal{A},\mathcal{B})(\overset{\simeq x}{x_k})$ non-diff $\Leftrightarrow \overset{\in I(x)}{\exists\ i}, (Ax_k+a)_i \overset{C1}{=} (Bx_k+b)_i \overset{x_k = x+d}{\Leftrightarrow}$
$(Ax+a)_i + A_{i,:}d = (Bx+b)_i + B_{i,:}d \Leftrightarrow A_{i,:}d = B_{i,:}d \Leftrightarrow d \in v_i^\perp$
($v_i \neq 0$ by C2)

Hyperplanes $H_i := (B_{i,:} - A_{i,:})^\perp := v_i^\perp$; for ∂_B's def, $\mathbb{R}^n \backslash \cup H_i$

$$\mathbb{R}^n = H_i^- \cup H_i \cup H_i^+, \quad \begin{cases} H_i^- = \{x \in \mathbb{R}^n : v_i^\top x < 0\} \\ H_i^+ = \{x \in \mathbb{R}^n : v_i^\top x > 0\} \end{cases}$$

Convention: $\forall\ i \in [1:p]$,

$$H_i^+ \Leftrightarrow B_{i,:}d - A_{i,:}d > 0 \Leftrightarrow \min(\dots) = \mathcal{A}_i(\dots) \Leftrightarrow J_{i,:} = A_{i,:}$$
$$H_i^- \Leftrightarrow B_{i,:}d - A_{i,:}d < 0 \Leftrightarrow \min(\dots) = \mathcal{B}_i(\dots) \Leftrightarrow J_{i,:} = B_{i,:}$$

so $\forall\ J, \forall\ i, J_{i,:} \in \{A_{i,:}, B_{i,:}\}$

Computing the B-differential - 2

$\min(\mathcal{A},\mathcal{B})(\overset{\simeq x}{x_k})$ non-diff $\Leftrightarrow \overset{\in I(x)}{\exists\ i}, (Ax_k + a)_i \overset{C1}{=} (Bx_k + b)_i \overset{x_k = x+d}{\Leftrightarrow}$
$(Ax + a)_i + A_{i,:}d = (Bx + b)_i + B_{i,:}d \Leftrightarrow A_{i,:}d = B_{i,:}d \Leftrightarrow d \in v_i^\perp$
($v_i \neq 0$ by C2)

Hyperplanes $H_i := (B_{i,:} - A_{i,:})^\perp := v_i^\perp$; for ∂_B's def, $\mathbb{R}^n \setminus \cup H_i$

$$\mathbb{R}^n = H_i^- \cup H_i \cup H_i^+, \quad \begin{cases} H_i^- = \{x \in \mathbb{R}^n : v_i^\mathsf{T} x < 0\} \\ H_i^+ = \{x \in \mathbb{R}^n : v_i^\mathsf{T} x > 0\} \end{cases}$$

Convention: $\forall\ i \in [1:p]$,

$$H_i^+ \Leftrightarrow B_{i,:}d - A_{i,:}d > 0 \Leftrightarrow \min(\dots) = \mathcal{A}_i(\dots) \Leftrightarrow J_{i,:} = A_{i,:}$$
$$H_i^- \Leftrightarrow B_{i,:}d - A_{i,:}d < 0 \Leftrightarrow \min(\dots) = \mathcal{B}_i(\dots) \Leftrightarrow J_{i,:} = B_{i,:}$$

so $\forall\ J, \forall\ i, J_{i,:} \in \{A_{i,:}, B_{i,:}\}$

Computing the B-differential - 3

So far:

- vectorial problem in dimension n: derivatives $J \in \mathbb{R}^{n \times n}$
- function is piecewise affine, derivative is piecewise constant
- the matrices J are composed of lines of A and B
- $\forall\ i \in [1:p]$, 2 possibilities: 2^p total, combinatorial nature

Computing the B-differential - 3

So far:

- vectorial problem in dimension n: derivatives $J \in \mathbb{R}^{n\times n}$
- function is piecewise affine, derivative is piecewise constant
- the matrices J are composed of lines of A and B
- $\forall\ i \in [1:p]$, 2 possibilities: 2^p total, combinatorial nature

Computing the B-differential - 3

So far:

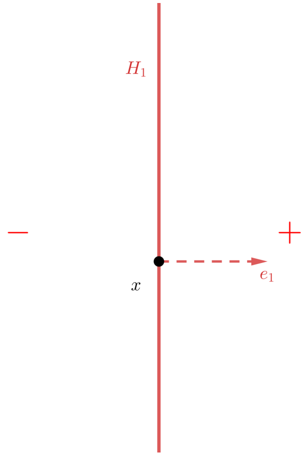
- vectorial problem in dimension n: derivatives $J \in \mathbb{R}^{n \times n}$
- function is piecewise affine, derivative is piecewise constant
- the matrices J are composed of lines of A and B
- $\forall\ i \in [1:p]$, 2 possibilities: 2^p total, combinatorial nature

Computing the B-differential - 3

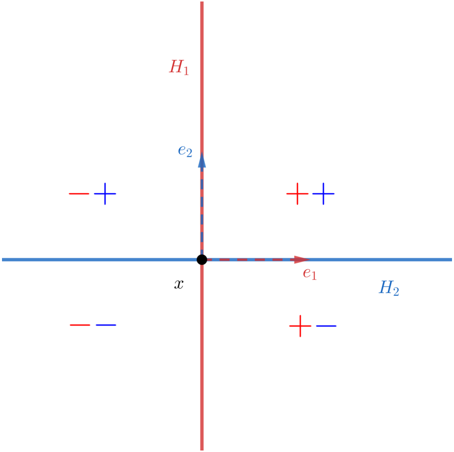
So far:

- vectorial problem in dimension n: derivatives $J \in \mathbb{R}^{n \times n}$
- function is piecewise affine, derivative is piecewise constant
- the matrices J are composed of lines of A and B
- $\forall\ i \in [1 : p]$, 2 possibilities: 2^p total, combinatorial nature

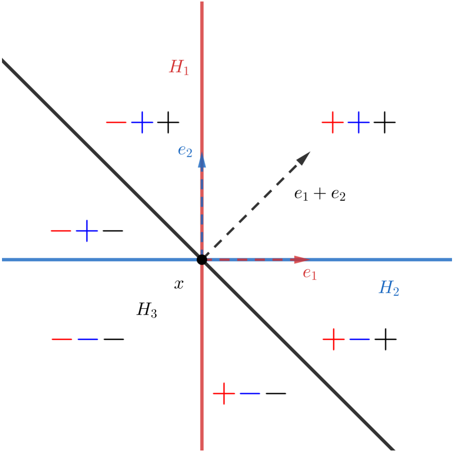
Computing the B-differential - 4



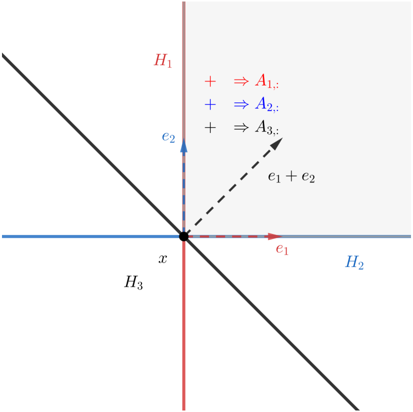
Computing the B-differential - 4



Computing the B-differential - 4

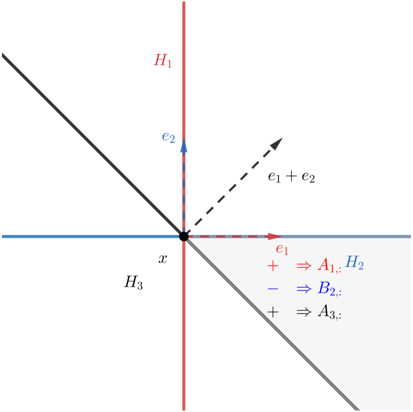


B-differential and hyperplanes



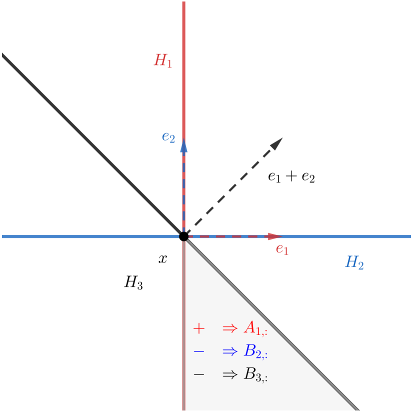
In the grey area, the points remain on the same sides of the hyperplanes: the Jacobian matrix remains constant.

B-differential and hyperplanes



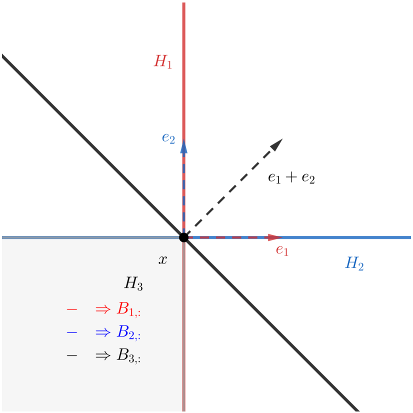
In the grey area, the points remain on the same sides of the hyperplanes: the Jacobian matrix remains constant.

B-differential and hyperplanes



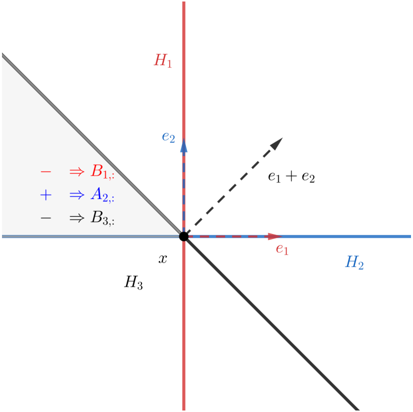
In the grey area, the points remain on the same sides of the hyperplanes: the Jacobian matrix remains constant.

B-differential and hyperplanes



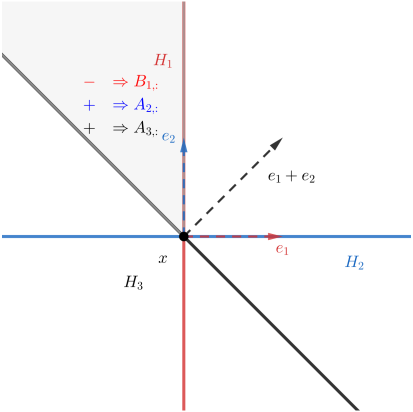
In the grey area, the points remain on the same sides of the hyperplanes: the Jacobian matrix remains constant.

B-differential and hyperplanes



In the grey area, the points remain on the same sides of the hyperplanes: the Jacobian matrix remains constant.

B-differential and hyperplanes



In the grey area, the points remain on the same sides of the hyperplanes: the Jacobian matrix remains constant.

Summary

$|I(x)| = p$ hyperplanes, $H_i = v_i^{\perp}, v_i = B_{i,:} - A_{i,:}$ [data]
$\mathbb{R}^n \setminus \bigcup H_i$ = differentiable points, on the $+$ or $-$ side of every H_i.
By convention: the $\pm$ becomes the sign s

Fundamental question

$$\text{given } V = [v_1 \ \dots \ v_p]$$
$$\text{find all } s = (s_1, \dots, s_p) \in \{\pm 1\}^p,$$
$$\text{s.t. } \exists\, d_s, \forall\, i \in [1:p], s_i v_i^{\top} d_s > 0$$

2^p linear feasibility problems to solve... How to improve?

Summary

$|I(x)| = p$ hyperplanes, $H_i = v_i^\perp, v_i = B_{i,:} - A_{i,:}$ [data]
$\mathbb{R}^n \backslash \bigcup H_i$ = differentiable points, on the $+$ or $-$ side of every H_i.
By convention: the $\pm$ becomes the sign s

Fundamental question

$$\text{given } V = [v_1 \ \dots \ v_p]$$
$$\text{find all } s = (s_1, \dots, s_p) \in \{\pm 1\}^p,$$
$$\text{s.t. } \exists\, d_s, \forall\, i \in [1:p], s_i v_i^\mathsf{T} d_s > 0$$

2^p linear feasibility problems to solve... How to improve?

Plan

1. From a nonsmooth question...
2. ... to combinatorics
3. Algorithmic details
   - Main principle
   - Improving on the structure
   - Dual approach: LO-free method
4. Some results

Main reasoning

Algorithm from [RČ18]:

- recursive tree that adds hyperplanes one at a time
- each node has one or two descendants,
- checked through Linear Optimization Problem (LOP)

at level k, with $s \in \{\pm 1\}^k$,

$$\forall\, i \in [1:k], \exists\, d_s, s_i v_i^\top d_s > 0 \Rightarrow \begin{cases} \forall\, i \in [1:k], s_i v_i^\top d > 0 \\ \qquad +v_{k+1}^\top d > 0 \end{cases} ? \quad \begin{cases} \forall\, i \in [1:k], s_i v_i^\top d > 0 \\ \qquad -v_{k+1}^\top d > 0 \end{cases} ?$$

The LOPs represent the main computational effort

Main reasoning

Algorithm from [RČ18]:

- recursive tree that adds hyperplanes one at a time
- each node has one or two descendants,
- checked through Linear Optimization Problem (LOP)

at level k, with $s \in \{\pm 1\}^k$,

$$\forall\, i \in [1:k], \exists\, d_s, s_i v_i^\mathsf{T} d_s > 0 \Rightarrow \begin{cases} \forall\, i \in [1:k], s_i v_i^\mathsf{T} d > 0 \\ \qquad +v_{k+1}^\mathsf{T} d > 0 & ? \\ \forall\, i \in [1:k], s_i v_i^\mathsf{T} d > 0 \\ \qquad -v_{k+1}^\mathsf{T} d > 0 & ? \end{cases}$$

The LOPs represent the main computational effort

Illustration of the regions and tree on the previous example

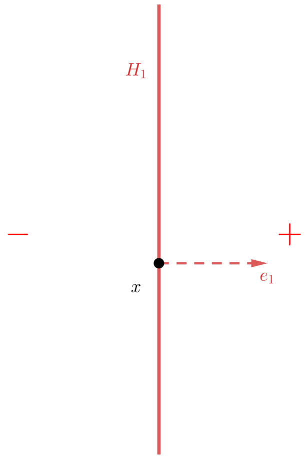


Illustration of the regions and tree on the previous example

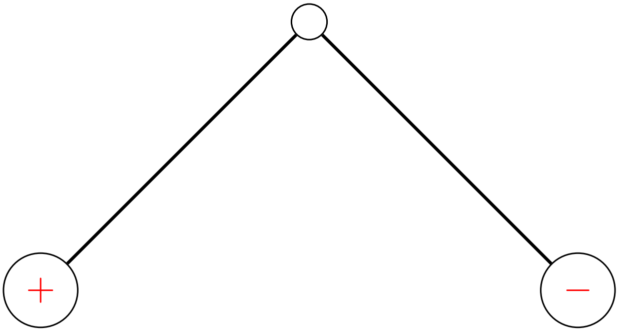


Illustration of the regions and tree on the previous example

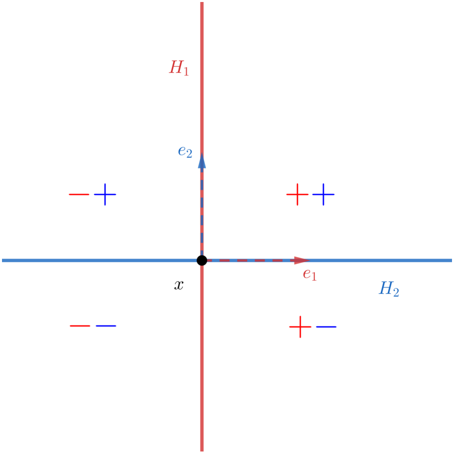


Illustration of the regions and tree on the previous example

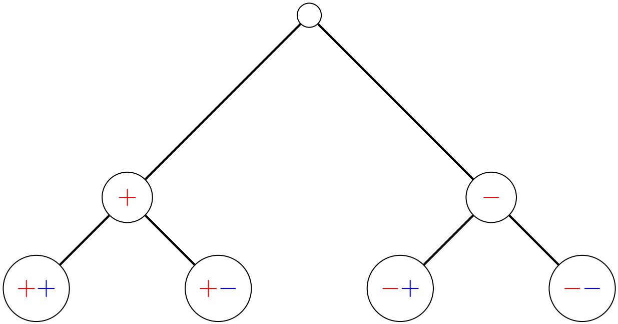


Illustration of the regions and tree on the previous example

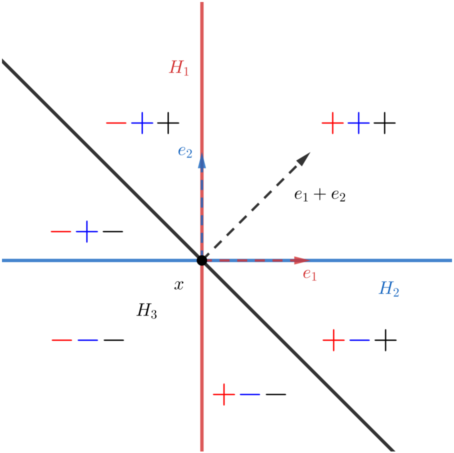
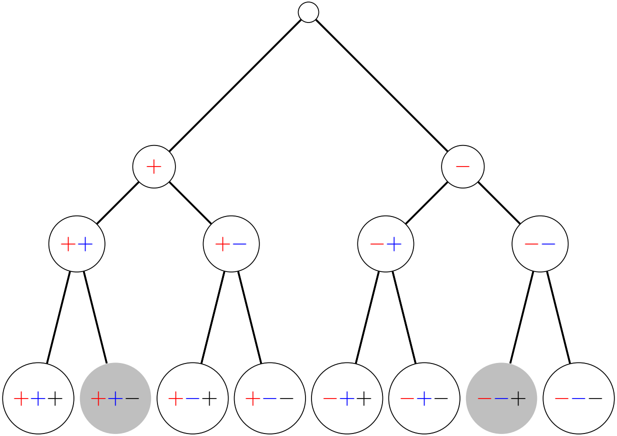


Illustration of the regions and tree on the previous example



Reducing the LOP count - 1

- each tree node: a Linear Optimization Problem; small dimension, but the only 'real' task
- goal is to avoid solving LOPs

Each node is associated to a $s \in \{\pm 1\}^k$ and its $d_s \in \mathbb{R}^n$.
Between levels k and $k+1$, when hyperplane $k+1$ is added, d_s can belong to $H_{k+1} \Leftrightarrow v_{k+1}^\top d_s = 0$: $(i \in [1:k])$

$$\begin{cases} s_i v_i^\top d_s > 0 \\ v_{k+1}^\top d_s = 0 \end{cases} \Rightarrow \exists\, (d^+, d^-), \begin{cases} s_i v_i^\top d^\pm > 0 \\ \pm v_{k+1}^\top d^\pm > 0 \end{cases}$$

$v_{k+1}^\top d_s = 0$ is utopic, but formalized for $|v_{k+1}^\top d_s|$ small enough

Reducing the LOP count - 1

- each tree node: a Linear Optimization Problem; small dimension, but the only 'real' task
- goal is to avoid solving LOPs

Each node is associated to a $s \in \{\pm 1\}^k$ and its $d_s \in \mathbb{R}^n$.
Between levels k and $k+1$, when hyperplane $k+1$ is added, d_s can belong to $H_{k+1} \Leftrightarrow v_{k+1}^\mathsf{T} d_s = 0$: $(i \in [1:k])$

$$\begin{cases} s_i v_i^\mathsf{T} d_s > 0 \\ v_{k+1}^\mathsf{T} d_s = 0 \end{cases} \Rightarrow \exists \, (d^+, d^-), \begin{cases} s_i v_i^\mathsf{T} d^\pm > 0 \\ \pm v_{k+1}^\mathsf{T} d^\pm > 0 \end{cases}$$

$v_{k+1}^\mathsf{T} d_s = 0$ is utopic, but formalized for $|v_{k+1}^\mathsf{T} d_s|$ small enough

Reducing the LOP count - 1

- each tree node: a Linear Optimization Problem; small dimension, but the only 'real' task
- goal is to avoid solving LOPs

Each node is associated to a $s \in \{\pm 1\}^k$ and its $d_s \in \mathbb{R}^n$.
Between levels k and $k+1$, when hyperplane $k+1$ is added, d_s can belong to $H_{k+1} \Leftrightarrow v_{k+1}^\mathsf{T} d_s = 0$: $(i \in [1:k])$

$$\begin{cases} s_i v_i^\mathsf{T} d_s > 0 \\ v_{k+1}^\mathsf{T} d_s = 0 \end{cases} \Rightarrow \exists\ (d^+, d^-), \begin{cases} s_i v_i^\mathsf{T} d^\pm > 0 \\ \pm v_{k+1}^\mathsf{T} d^\pm > 0 \end{cases}$$

$v_{k+1}^\mathsf{T} d_s = 0$ is utopic, but formalized for $|v_{k+1}^\mathsf{T} d_s|$ small enough

Reducing the LOP count - 2

Using the "contrapositive"

$|v_{k+1}^T d_s|$ 'large' $\rightarrow$ less chance of both $(s, +1)$ and $(s, -1)$.

In s, hyperplanes $\overbrace{\{i_1, \dots, i_k\}}^{=I^s}$; $i_{k+1} = \arg\max_j \ |v_j^T d_s|, j \in [1:p] \setminus I^s$

Only a heuristic, but reasonably efficient.
Also, this order change is local - for each s it can change.

Reducing the LOP count - 2

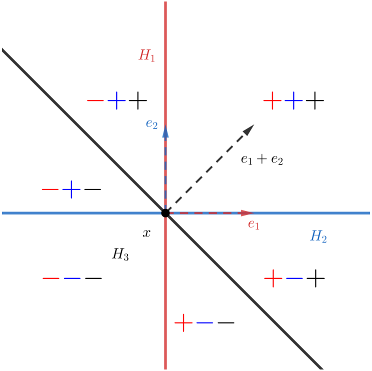
Using the "contrapositive"

$|v_{k+1}^T d_s|$ 'large' $\rightarrow$ less chance of both $(s, +1)$ and $(s, -1)$.

In s, hyperplanes $\overbrace{\{i_1, \ldots, i_k\}}^{=I^s}$; $i_{k+1} = \arg\max_j \; |v_j^T d_s|, j \in [1:p] \setminus I^s$

Only a heuristic, but reasonably efficient.
Also, this order change is local - for each s it can change.

Infeasibility, matroids and circuits - 1



$++-$ (and $--+$) corresponds to an empty region: $+$ means right to H_1, $+$ over H_2, $-$ down left H_3: such a point does not exist. The system is $+ : d_1 > 0, + : d_2 > 0, - : -d_1 - d_2 > 0$

Infeasibility, matroids and circuits - 2

With $p > 3$, $++- \bullet \bullet \ldots \bullet \bullet$ always infeasible.

Gordan's alternative

$M \in \mathbb{R}^{p \times n}$, exactly one is true:

$$\begin{cases} \exists\, d \in \mathbb{R}^n : Md > 0_{\mathbb{R}^p} \\ \exists\, \gamma \in \mathbb{R}^p_+ \setminus \{0\} : M^\mathsf{T} \gamma = 0 \end{cases} \tag{3}$$

$s \in \{\pm 1\}^p$ arbitrary:
$M = \mathrm{diag}(s) V^\mathsf{T} \rightarrow Md = (s_1 v_1^\mathsf{T} d; \ldots; s_p v_p^\mathsf{T} d)$

"Feasibility of a system or element in the null space of the matrix"

Infeasibility, matroids and circuits - 2

With $p > 3$, $++- \bullet \bullet \ldots \bullet \bullet$ always infeasible.

Gordan's alternative

$M \in \mathbb{R}^{p \times n}$, exactly one is true:

$$\begin{cases} \exists\, d \in \mathbb{R}^n : Md > 0_{\mathbb{R}^p} \\ \exists\, \gamma \in \mathbb{R}^p_+ \setminus \{0\} : M^\mathsf{T}\gamma = 0 \end{cases} \tag{3}$$

$s \in \{\pm 1\}^p$ arbitrary:
$M = \operatorname{diag}(s) V^\mathsf{T} \to Md = (s_1 v_1^\mathsf{T} d; \ldots; s_p v_p^\mathsf{T} d)$

"Feasibility of a system or element in the null space of the matrix"

Infeasibility, matroids and circuits - 2

With $p > 3$, $++-\bullet\ \bullet \ldots \bullet\ \bullet$ always infeasible.

Gordan's alternative

$M \in \mathbb{R}^{p \times n}$, exactly one is true:

$$\begin{cases} \exists\ d \in \mathbb{R}^n : Md > 0_{\mathbb{R}^p} \\ \exists\ \gamma \in \mathbb{R}^p_+ \setminus \{0\} : M^\mathsf{T} \gamma = 0 \end{cases} \tag{3}$$

$s \in \{\pm 1\}^p$ arbitrary:
$M = \mathrm{diag}(s) V^\mathsf{T} \rightarrow Md = (s_1 v_1^\mathsf{T} d; \ldots ; s_p v_p^\mathsf{T} d)$

"Feasibility of a system or element in the null space of the matrix"

Infeasibility, matroids and circuits - 3

Instead: search for $\Gamma = \{\gamma\}$ and prune/stop the tree when an infeasibility is detected.
The γ's represent the "**circuits**" of the "**matroid**" defined by V

- the tree from [RČ18]
- some improvements on the overall tree
- the dual algorithm detecting infeasibilities
- normal tree algorithm but with some infeasibility detection

Infeasibility, matroids and circuits - 3

Instead: search for $\Gamma = \{\gamma\}$ and prune/stop the tree when an infeasibility is detected.
The γ's represent the "**circuits**" of the "**matroid**" defined by V

- the tree from [RČ18]
- some improvements on the overall tree
- the dual algorithm detecting infeasibilities
- normal tree algorithm but with some infeasibility detection

Technical details

LO solver

Gurobi chosen (as [RČ18]), practical & easy to use through JuMP.
To compare with others (small dimension LOPs)

For the circuits: several ways to implement/compute - mostly matricial manipulations.
Inspiration: compute at each level the information to know if the current sign vector will cover a circuit.

Technical details

LO solver

Gurobi chosen (as [RČ18]), practical & easy to use through JuMP.
To compare with others (small dimension LOPs)

For the circuits: several ways to implement/compute - mostly matricial manipulations.
Inspiration: compute at each level the information to know if the current sign vector will cover a circuit.

Plan

Summary

- LO only = ABC
- LO + a bit of duality = ABCD2
- LO + a lot of duality = ABCD3
- only duality = AD4

Not clear which is better: dual computations are sometimes not useful

Summary

- LO only = ABC
- LO + a bit of duality = ABCD2
- LO + a lot of duality = ABCD3
- only duality = AD4

Not clear which is better: dual computations are sometimes not useful

Summary

- LO only = ABC
- LO + a bit of duality = ABCD2
- LO + a lot of duality = ABCD3
- only duality = AD4

Not clear which is better: dual computations are sometimes not useful

Summary

- LO only = ABC
- LO + a bit of duality = ABCD2
- LO + a lot of duality = ABCD3
- only duality = AD4

Not clear which is better: dual computations are sometimes not useful

Results; blue = times, black = improvement factor

Name	RC	ABC		ABCD2		ABCD3		AD4	
R-4-8-2	1.70 10^{-2}	7.20 10^{-3}	2.36	6.53 10^{-3}	2.60	3.13 10^{-3}	**5.43**	8.03 10^{-3}	2.12
R-7-8-4	5.70 10^{-2}	3.38 10^{-2}	1.69	3.15 10^{-2}	1.81	2.24 10^{-2}	**2.54**	2.79 10^{-2}	2.04
R-7-9-4	9.97 10^{-2}	4.98 10^{-2}	2.00	4.96 10^{-2}	2.01	3.43 10^{-2}	**2.91**	5.16 10^{-2}	1.93
R-7-10-5	2.33 10^{-1}	1.16 10^{-1}	2.01	1.29 10^{-1}	1.81	1.05 10^{-1}	**2.22**	1.22 10^{-1}	1.91
R-7-11-4	2.36 10^{-1}	1.22 10^{-1}	1.93	1.20 10^{-1}	1.97	8.49 10^{-2}	**2.78**	1.32 10^{-1}	1.79
R-7-12-6	9.35 10^{-1}	5.05 10^{-1}	**1.85**	5.74 10^{-1}	1.63	5.13 10^{-1}	1.82	5.65 10^{-1}	1.65
R-7-13-5	9.11 10^{-1}	4.70 10^{-1}	**1.94**	5.41 10^{-1}	1.68	4.71 10^{-1}	1.93	5.33 10^{-1}	1.71
R-7-14-7	3.69	2.15	**1.72**	2.39	1.54	2.42	1.52	2.42	1.52
R-8-15-7	6.43	3.56	**1.81**	3.92	1.64	4.30	1.50	4.57	1.41
R-9-16-8	1.51 10^{+1}	8.88	**1.70**	1.03 10^{+1}	1.47	1.34 10^{+1}	1.13	1.41 10^{+1}	1.07
R-10-17-9	3.45 10^{+1}	2.08 10^{+1}	**1.66**	2.50 10^{+1}	1.38	4.04 10^{+1}	0.85	3.53 10^{+1}	0.98
2d-20-4	3.48 10^{-1}	1.76 10^{-1}	1.98	8.03 10^{-2}	4.33	6.96 10^{-2}	**5.00**	1.73 10^{-1}	2.01
2d-20-5	6.74 10^{-1}	3.54 10^{-1}	1.90	1.29 10^{-1}	**5.22**	1.32 10^{-1}	5.11	3.59 10^{-1}	1.88
2d-20-6	1.19	6.04 10^{-1}	1.97	2.23 10^{-1}	**5.34**	2.70 10^{-1}	4.41	6.52 10^{-1}	1.83
2d-20-7	2.08	1.45	1.43	5.40 10^{-1}	**3.85**	6.21 10^{-1}	3.35	1.11	1.87
2d-20-8	3.69	1.85	1.99	6.36 10^{-1}	**5.80**	7.95 10^{-1}	4.64	1.92	1.92
sR-2	1.71 10^{+1}	4.26	4.01	3.11	**5.50**	4.14	4.13	1.05 10^{+1}	1.63
sR-4	8.03 10^{+1}	3.68 10^{+1}	2.18	4.40 10^{+1}	**1.83**	1.41 10^{+2}	0.57	2.02 10^{+2}	0.40
sR-6	1.08 10^{+2}	1.54 10^{+2}	0.70	7.01 10^{+1}	**1.54**	2.58 10^{+2}	0.42	4.04 10^{+2}	0.27
perm-5	6.64 10^{-1}	1.89 10^{-1}	3.51	6.87 10^{-2}	**9.67**	8.53 10^{-2}	7.78	3.75 10^{-1}	1.77
perm-6	5.80	1.32	4.39	5.19 10^{-1}	**11.18**	1.03	5.63	3.81	1.52
perm-7	5.70 10^{+1}	1.10 10^{+1}	5.18	4.16	**13.70**	2.12 10^{+1}	2.69	6.37 10^{+1}	0.89
perm-8	5.98 10^{+2}	1.08 10^{+2}	5.54	4.41 10^{+1}	**13.56**	6.46 10^{+2}	0.93	1.59 10^{+3}	0.38
r-3-7	5.83 10^{-1}	3.16 10^{-1}	1.84	2.79 10^{-1}	2.09	2.27 10^{-1}	**2.57**	3.64 10^{-1}	1.60
r-3-9	3.31 10^{-1}	2.92 10^{-1}	1.13	1.96 10^{-1}	1.69	1.41 10^{-1}	**2.35**	1.77 10^{-1}	1.87
r-4-7	3.13	1.62	1.93	1.37	**2.28**	2.21	1.42	3.01	1.04
r-4-9	2.76	1.36	2.03	1.13	**2.44**	1.85	1.49	2.87	0.96
r-5-7	8.92	4.72	1.89	3.94	**2.26**	8.64	1.03	1.26 10^{+1}	0.71
r-5-9	9.02	4.47	2.02	3.72	**2.42**	7.92	1.14	1.06 10^{+1}	0.85
r-6-7	2.18 10^{+1}	1.20 10^{+1}	1.82	1.14 10^{+1}	**1.91**	2.89 10^{+1}	0.75	4.03 10^{+1}	0.54
r-6-9	2.63 10^{+1}	1.45 10^{+1}	1.81	1.17 10^{+1}	**2.25**	3.39 10^{+1}	0.78	4.89 10^{+1}	0.54
r-7-7	5.72 10^{+1}	3.30 10^{+1}	**1.73**	3.49 10^{+1}	1.64	1.17 10^{+2}	0.49	1.60 10^{+2}	0.36
r-7-9	4.68 10^{+1}	2.58 10^{+1}	1.81	2.45 10^{+1}	**1.91**	7.30 10^{+1}	0.64	8.74 10^{+1}	0.54
median/mean			1.93/2.23		2.05/3.70		1.93/2.48		1.52/1.32

Conclusion

- pretty far from the differentiability; but relevant in itself
- various elements: LO, LA, recursivity, implementation choices...
- also: affine hyperplanes ✓, version for rational data $\simeq$✓, dedicated package...

Thanks for your attention! Some questions?

Conclusion

- pretty far from the differentiability; but relevant in itself
- various elements: LO, LA, recursivity, implementation choices...
- also: affine hyperplanes ✓, version for rational data ≃✓, dedicated package...

Thanks for your attention! Some questions?

Conclusion

- pretty far from the differentiability; but relevant in itself
- various elements: LO, LA, recursivity, implementation choices...
- also: affine hyperplanes ✓, version for rational data $\simeq$✓, dedicated package...

Thanks for your attention! Some questions?

Conclusion

- pretty far from the differentiability; but relevant in itself
- various elements: LO, LA, recursivity, implementation choices...
- also: affine hyperplanes ✓, version for rational data $\simeq$✓, dedicated package...

Thanks for your attention! Some questions?

Bibliographic elements I

[CPS92] R.W. Cottle, J.-S. Pang, and R.E. Stone. *The Linear Complementarity Problem*. Academic Press, Boston, 1992.

[CX11] X. Chen and S. Xiang. “Computation of generalized differentials in nonlinear complementarity problems”. In: *Computational Optimization and Applications* 50 (2011). [doi], pp. 403–423.

[FP03] F. Facchinei and J.-S. Pang. *Finite-Dimensional Variational Inequalities and Complementarity Problems*. Springer Series in Operations Research. Springer, 2003.

[Qi93] L. Qi. “Convergence Analysis of Some Algorithms for Solving Nonsmooth Equations”. In: *Mathematics of Operations Research* 18 (Feb. 1993). [doi], pp. 227–244.

Bibliographic elements II

[QS93] L. Qi and J. Sun. "A nonsmooth version of Newton's method". In: *Mathematical Programming* 58 (1993). [doi], pp. 353–367.

[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 (Jan. 2018), pp. 455–473. DOI: 10.1137/15M1027930.

Theoretical detour

Very well-known in algebra / combinatorics...
... but very theoretically: Möbius function, lattices, matroids.

Very impressive results / algorithms for the cardinal (number of feasible systems, number of $J \in \partial_B$)
Upper bound, formula (also combinatorial)...

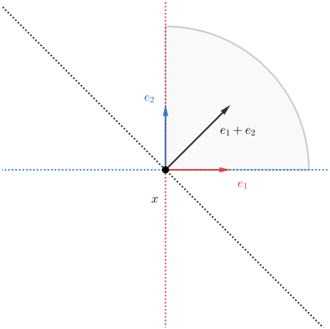
Theoretical detour

Very well-known in algebra / combinatorics...
... but very theoretically: Möbius function, lattices, matroids.

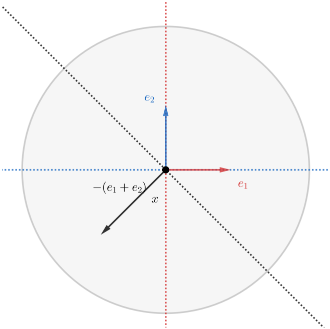
Very impressive results / algorithms for the cardinal (number of feasible systems, number of $J \in \partial_B$)
Upper bound, formula (also combinatorial)...

Various properties

-origin = center of symmetry ; $s_i v_i^\top d > 0 \Leftrightarrow (-s_i) v_i^\top (-d) > 0$



Feasible $\Leftrightarrow$ pointed cone



Infeasible $\Leftrightarrow$ non-pointed

-"connectedness" property (vertices = J's, edges = hyperplanes)

Method - adding vectors one at a time

With one more vector

- Given $(v_1, \ldots, v_{k-1})$; v_k ; $\mathcal{S}_{k-1} \subseteq \{\pm 1\}^{k-1}$

Method - adding vectors one at a time

With one more vector

- Given $(v_1, \ldots, v_{k-1})$; v_k ; $\mathcal{S}_{k-1} \subseteq \{\pm 1\}^{k-1}$
- $\forall\ s = (s_1, \ldots, s_{k-1}) \in \mathcal{S}_{k-1}$, we know d_s^{k-1} s.t. :
  $\forall\ i \in [1 : k-1],\ s_i v_i^\top d_s^{k-1} > 0$

Method - adding vectors one at a time

With one more vector

- Given $(v_1, \dots, v_{k-1})$; v_k ; $\mathcal{S}_{k-1} \subseteq \{\pm 1\}^{k-1}$
- $\forall\, s = (s_1, \dots, s_{k-1}) \in \mathcal{S}_{k-1}$, we know d_s^{k-1} s.t. :
  $\forall\, i \in [1 : k-1],\ s_i v_i^\top d_s^{k-1} > 0$
- $v_k^\top d_s^{k-1} > 0 \Rightarrow \begin{cases} +v_k^\top d_s^{k-1} > 0 \\ s_i v_i^\top d_s^{k-1} > 0 \end{cases} \checkmark, \begin{cases} -v_k^\top d > 0 \\ s_i v_i^\top d > 0 \end{cases} ? \rightarrow$ L.O.

Method - adding vectors one at a time

With one more vector

- Given $(v_1, \dots, v_{k-1})$; v_k ; $\mathcal{S}_{k-1} \subseteq \{\pm 1\}^{k-1}$
- $\forall\ s = (s_1, \dots, s_{k-1}) \in \mathcal{S}_{k-1}$, we know d_s^{k-1} s.t. :
  $\forall\ i \in [1 : k-1]$, $s_i v_i^\mathsf{T} d_s^{k-1} > 0$
- $v_k^\mathsf{T} d_s^{k-1} > 0 \Rightarrow \begin{cases} +v_k^\mathsf{T} d_s^{k-1} > 0 \\ s_i v_i^\mathsf{T} d_s^{k-1} > 0 \end{cases} \checkmark, \begin{cases} -v_k^\mathsf{T} d > 0 \\ s_i v_i^\mathsf{T} d > 0 \end{cases} ? \rightarrow$ L.O.
- $v_k^\mathsf{T} d_s^{k-1} < 0 \Rightarrow \begin{cases} -v_k^\mathsf{T} d_s^{k-1} > 0 \\ s_i v_i^\mathsf{T} d_s^{k-1} > 0 \end{cases} \checkmark, \begin{cases} +v_k^\mathsf{T} d > 0 \\ s_i v_i^\mathsf{T} d > 0 \end{cases} ? \rightarrow$ L.O.

Method - adding vectors one at a time

With one more vector

- Given $(v_1, \dots, v_{k-1})$; v_k ; $\mathcal{S}_{k-1} \subseteq \{\pm 1\}^{k-1}$
- $\forall\ s = (s_1, \dots, s_{k-1}) \in \mathcal{S}_{k-1}$, we know d_s^{k-1} s.t. :
  $\forall\ i \in [1:k-1],\ s_i v_i^\top d_s^{k-1} > 0$
- $v_k^\top d_s^{k-1} > 0 \Rightarrow \begin{cases} +v_k^\top d_s^{k-1} > 0 \\ s_i v_i^\top d_s^{k-1} > 0 \end{cases} \checkmark, \begin{cases} -v_k^\top d > 0 \\ s_i v_i^\top d > 0 \end{cases} ? \to$ L.O.
- $v_k^\top d_s^{k-1} < 0 \Rightarrow \begin{cases} -v_k^\top d_s^{k-1} > 0 \\ s_i v_i^\top d_s^{k-1} > 0 \end{cases} \checkmark, \begin{cases} +v_k^\top d > 0 \\ s_i v_i^\top d > 0 \end{cases} ? \to$ L.O.
- $v_k^\top d_s^{k-1} = 0 \Rightarrow$ both systems ✓ by perturbation

Circuits of matroids

We look at subsets $I \subset [1:p]$, $\dim(\mathcal{N}(V_{:,I})) = \mathbf{1}$
and $\forall\ I' \subsetneq I$, $\dim(\mathcal{N}(V_{:,I'})) = 0$

$$\dim(\mathcal{N}(V_{:,I})) = 1 \Rightarrow \mathcal{N}(V_{:,I}) = \mathrm{Vect}(\eta)$$
$$\Rightarrow V_{:,I}\eta = 0 \Leftrightarrow \underbrace{V_{:,I}\mathrm{sign}(\eta)}_{V_{(:,I)}S_{(I)}}\underbrace{\mathrm{sign}(\eta)\eta}_{=\gamma_{(I)}\geq 0} = 0$$

$\mathcal{N}(V_{:,I})$ gives 'unsigned' η's which define the sign $s_J = 1$ because

if ≥ 2, smaller subsets are of $\dim(\mathcal{N}) = 1$

2^p LO feasibility $\leftrightarrow$ 2^p $\mathcal{N}$ searches; subsets of size $\leq 1 + \mathrm{rank}(V)$

Issue (unresolved): "optimal" way to compute efficiently: if I s.t. $\dim(\mathcal{N}(V_{:,I})) = 1$, $I' \supsetneq I$ useless to check

Circuits of matroids

We look at subsets $I \subset [1:p]$, $\dim(\mathcal{N}(V_{:,I})) = \mathbf{1}$
and $\forall\ I' \subsetneq I$, $\dim(\mathcal{N}(V_{:,I'})) = 0$

$$\dim(\mathcal{N}(V_{:,I})) = 1 \Rightarrow \mathcal{N}(V_{:,I}) = \mathrm{Vect}(\eta)$$
$$\Rightarrow V_{:,I}\eta = 0 \Leftrightarrow \underbrace{V_{:,I}\mathrm{sign}(\eta)}_{V_{(:,I)}S_{(I)}}\underbrace{\mathrm{sign}(\eta)\eta}_{=\gamma_{(I)}\geq 0} = 0$$

$\mathcal{N}(V_{:,I})$ gives 'unsigned' η's which define the sign $s_J = 1$ because

if ≥ 2, smaller subsets are of $\dim(\mathcal{N}) = 1$

2^p LO feasibility $\leftrightarrow$ 2^p $\mathcal{N}$ searches; subsets of size $\leq 1 + \mathrm{rank}(V)$

Issue (unresolved): "optimal" way to compute efficiently: if I s.t. $\dim(\mathcal{N}(V_{:,I})) = 1$, $I' \supsetneq I$ useless to check

Circuits of matroids

We look at subsets $I \subset [1:p]$, $\dim(\mathcal{N}(V_{:,I})) = \mathbf{1}$
and $\forall\ I' \subsetneq I$, $\dim(\mathcal{N}(V_{:,I'})) = 0$

$$\begin{aligned}\dim(\mathcal{N}(V_{:,I})) = 1 &\Rightarrow \mathcal{N}(V_{:,I}) = \mathrm{Vect}(\eta)\\ &\Rightarrow V_{:,I}\eta = 0 \Leftrightarrow \underbrace{V_{:,I}\mathrm{sign}(\eta)}_{V_{(:,I)}S_{(I)}}\underbrace{\mathrm{sign}(\eta)\eta}_{=\gamma_{(I)}\geq 0} = 0\end{aligned}$$

$\mathcal{N}(V_{:,I})$ gives 'unsigned' η's which define the sign $s_J = 1$ because

if ≥ 2, smaller subsets are of $\dim(\mathcal{N}) = 1$

2^p LO feasibility $\leftrightarrow$ 2^p $\mathcal{N}$ searches; subsets of size $\leq 1 + \mathrm{rank}(V)$

Issue (unresolved): "optimal" way to compute efficiently: if I s.t. $\dim(\mathcal{N}(V_{:,I})) = 1$, $I' \supsetneq I$ useless to check

Circuits of matroids

We look at subsets $I \subset [1:p]$, $\dim(\mathcal{N}(V_{:,I})) = \mathbf{1}$
and $\forall\ I' \subsetneq I$, $\dim(\mathcal{N}(V_{:,I'})) = 0$

$$\begin{aligned}\dim(\mathcal{N}(V_{:,I})) = 1 &\Rightarrow \mathcal{N}(V_{:,I}) = \mathrm{Vect}(\eta) \\ &\Rightarrow V_{:,I}\eta = 0 \Leftrightarrow \underbrace{V_{:,I}\mathrm{sign}(\eta)}_{V_{(:,I)}S_{(I)}} \underbrace{\mathrm{sign}(\eta)\eta}_{=\gamma_{(I)}\geq 0} = 0\end{aligned}$$

$\mathcal{N}(V_{:,I})$ gives 'unsigned' η's which define the sign $s_J = 1$ because

if ≥ 2, smaller subsets are of $\dim(\mathcal{N}) = 1$

2^p LO feasibility $\leftrightarrow$ 2^p $\mathcal{N}$ searches; subsets of size $\leq 1 + \mathrm{rank}(V)$

Issue (unresolved): "optimal" way to compute efficiently: if I s.t. $\dim(\mathcal{N}(V_{:,I})) = 1$, $I' \supsetneq I$ useless to check

Circuits of matroids

We look at subsets $I \subset [1:p]$, $\dim(\mathcal{N}(V_{:,I})) = \mathbf{1}$
and $\forall\ I' \subsetneq I$, $\dim(\mathcal{N}(V_{:,I'})) = 0$

$$\dim(\mathcal{N}(V_{:,I})) = 1 \Rightarrow \mathcal{N}(V_{:,I}) = \mathrm{Vect}(\eta)$$
$$\Rightarrow V_{:,I}\eta = 0 \Leftrightarrow \underbrace{V_{:,I}\mathrm{sign}(\eta)}_{V_{(:,I)}S_{(I)}}\underbrace{\mathrm{sign}(\eta)\eta}_{=\gamma_{(I)}\geq 0} = 0$$

$\mathcal{N}(V_{:,I})$ gives 'unsigned' η's which define the sign $s_J = 1$ because

if ≥ 2, smaller subsets are of $\dim(\mathcal{N}) = 1$

2^p LO feasibility $\leftrightarrow$ 2^p $\mathcal{N}$ searches; subsets of size $\leq 1 + \mathrm{rank}(V)$

Issue (unresolved): "optimal" way to compute efficiently: if I s.t. $\dim(\mathcal{N}(V_{:,I})) = 1$, $I' \supsetneq I$ useless to check

Circuits of matroids

We look at subsets $I \subset [1:p]$, $\dim(\mathcal{N}(V_{:,I})) = \mathbf{1}$
and $\forall\ I' \subsetneq I$, $\dim(\mathcal{N}(V_{:,I'})) = 0$

$$\begin{aligned}\dim(\mathcal{N}(V_{:,I})) = 1 \Rightarrow & \mathcal{N}(V_{:,I}) = \mathrm{Vect}(\eta) \\ & \Rightarrow V_{:,I}\eta = 0 \Leftrightarrow \underbrace{V_{:,I}\mathrm{sign}(\eta)}_{V_{(:,I)}S_{(I)}} \underbrace{\mathrm{sign}(\eta)\eta}_{=\gamma_{(I)}\geq 0} = 0\end{aligned}$$

$\mathcal{N}(V_{:,I})$ gives 'unsigned' η's which define the sign $s_J = 1$ because

if ≥ 2, smaller subsets are of $\dim(\mathcal{N}) = 1$

2^p LO feasibility $\leftrightarrow$ 2^p $\mathcal{N}$ searches; subsets of size $\leq 1 + \mathrm{rank}(V)$

Issue (unresolved): "optimal" way to compute efficiently: if I s.t.
$\dim(\mathcal{N}(V_{:,I})) = 1$, $I' \supsetneq I$ useless to check