Hyperplane Arrangements and Matroids for Linear Complementarity Problems

Computing the B-differential of the Componentwise Minimum of Affine Functions

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

May, 29 2023

Outline

Plan

Systems of smooth nonlinear equations

General problem:

Find a point $x_* \in \mathbb{R}^n$: $F(x_*) = 0$, with $F : \mathbb{R}^n \to \mathbb{R}^n$ smooth

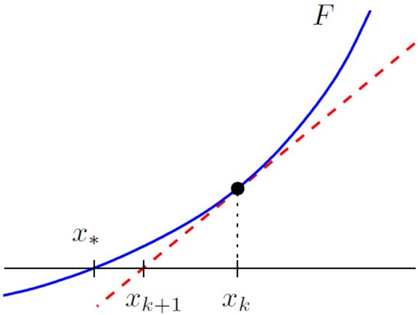


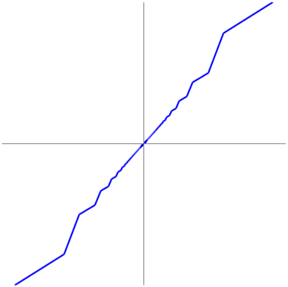
Illustration of Newton's method in 1D

x_0 near x_*,
$F \in \mathcal{C}^{1,1}$,
$F'(x_*)$ non-singular
$\Downarrow$
quadratic convergence

Nonsmooth equations

Harder problem:

Find a poind $x_* \in \mathbb{R}^n$: $F(x_*) = 0$, with $F : \mathbb{R}^n \to \mathbb{R}^n$ nonsmooth



Kummer's counter-example to Newton; in the nonsmooth case the Jacobian might not be defined.

x_0 near x_*,
F semismooth,
all $J \in$ "$F'(x_*)$"
non-singular
$\Downarrow$
quadratic
convergence

Remedy: 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 $F(x_k) + J_k\delta_k = 0$ for δ_k, with $J_k \in \partial_B F(x_k)$: $\partial_B F$ is the Bouligand differential
- then $x_{k+1} = x_k + (\alpha_k)\delta_k$

Remedy: 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 $F(x_k) + J_k\delta_k = 0$ for δ_k, with $J_k \in \partial_B F(x_k)$: $\partial_B F$ is the Bouligand differential
- then $x_{k+1} = x_k + (\alpha_k)\delta_k$

Remedy: 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 $F(x_k) + J_k\delta_k = 0$ for δ_k, with $J_k \in \partial_B F(x_k)$: $\partial_B F$ is the Bouliganda differential
- then $x_{k+1} = x_k + (\alpha_k)\delta_k$

Remedy: 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

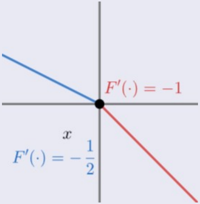
- take $x_0 \in \mathbb{R}^n$ (near x_*)
- for k = 1 2, . . . , solve $F(x_k) + J_k\delta_k = 0$ for δ_k, with $J_k \in \partial_B F(x_k)$: $\partial_B F$ is the Bouligand differential
- then $x_{k+1} = x_k + (\alpha_k)\delta_k$

Generalized derivatives

Bouligand differential

$$\partial_{\mathrm{B}} F(x) = \{J \in \mathbb{R}^{n\times n} : \exists\ (x_k)_k \to x, F'(x_k) \to J\} \tag{1}$$

Example: $F(x) = \begin{cases} -x/2 & \text{if } x \leq 0 \\ -x & \text{if } x > 0 \end{cases}$,
$\partial_B F(0) = \{-1/2, -1\}$.



$\underbrace{\partial_B F(x)}_{=??} \subsetneq \underbrace{\partial_B F_1(x) \times \cdots \times \partial_B F_p(x)}_{\prod \cdots = \text{ easy}}$ (sometimes $=$)

$\exists$ other differentials: Clarke, Mordukhovich, 2nd order...

Linear Complementarity Problems

General form [CPS92; FP03]

$$A, B \in \mathbb{R}^{n\times n}, a, b \in \mathbb{R}^n, \to \ \mathcal{A}(x) = Ax + a, \mathcal{B}(x) = Bx + b$$
$$0 \leq (Ax + a) \perp (Bx + b) \geq 0 \Leftrightarrow \tag{2}$$
$$\forall\ i, A_{i,:}x + a_i \geq 0, B_{i,:}x + b_i \geq 0, (A_{i,:}x + a_i)(B_{i,:}x + b_i) = 0$$

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

$(2) \Leftrightarrow \forall\ i, F_i(x) := \min(\mathcal{A}_i(x), \mathcal{B}_i(x)) = 0 \Leftrightarrow F(x) = 0$

Linear Complementarity Problems

General form [CPS92; FP03]

$$A, B \in \mathbb{R}^{n\times n}, a, b \in \mathbb{R}^n, \rightarrow \ \mathcal{A}(x) = Ax + a, \mathcal{B}(x) = Bx + b$$
$$0 \leq (Ax + a) \perp (Bx + b) \geq 0 \Leftrightarrow \tag{2}$$
$$\forall\ i, A_{i,:}x + a_i \geq 0, B_{i,:}x + b_i \geq 0, (A_{i,:}x + a_i)(B_{i,:}x + b_i) = 0$$

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

$(2) \Leftrightarrow \forall\ i, F_i(x) := \min(\mathcal{A}_i(x), \mathcal{B}_i(x)) = 0 \Leftrightarrow F(x) = 0$

Summary

Minimum function on LCPs $\Rightarrow$ semismooth system

Adapted Newton requires info on $\partial_B \min(\mathcal{A}, \mathcal{B})(\cdot) = \partial_B F(\cdot)$

Or to form $\partial_C \min(\mathcal{A}, \mathcal{B})(\cdot)$.

One $J_B \in \partial_B F$: [Qi93]
One $J_C \in \partial_C F$: [CX11]
But all of them?

Upcoming plan

The main question

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

- their structure
- finite (but exponential) number of elements
- how to compute them

Plan

Computing the B-differential

$f, g \in \mathcal{C}^1$: $\min(f(x), g(x))$ diff $\Leftrightarrow$ $f(x) \neq g(x)$ or $f'(x) = g'(x)$.

Important: F is piecewise affine: F' is piecewise **constant**.

$$\begin{cases} A_{i,:} & = & B_{i,:} \\ A_{i,:}x + a_i & < & B_{i,:}x + b_i \\ A_{i,:}x + a_i & > & B_{i,:}x + b_i \end{cases} \Rightarrow \forall\, J \in \partial_B F(x), J_{i,:} = \begin{cases} A_{i,:} = B_{i,:} \\ A_{i,:} \\ B_{i,:} \end{cases}$$

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

$\min(\mathcal{A}, \mathcal{B})$ non-diff $\Leftrightarrow$ affine terms equal $\Leftrightarrow$ **hyperplanes**

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

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

H_i^- or H_i^+, $\forall\, i \in [1:p]$: $H_i^+ \Leftrightarrow J_{i,:} = A_{i,:}$, $H_i^- \Leftrightarrow J_{i,:} = B_{i,:}$

Computing the B-differential

$f, g \in \mathcal{C}^1$: $\min(f(x), g(x))$ diff $\Leftrightarrow f(x) \neq g(x)$ or $f'(x) = g'(x)$.

Important: F is piecewise affine: F' is piecewise **constant**.

$$\begin{cases} A_{i,:} = B_{i,:} \\ A_{i,:}x + a_i < B_{i,:}x + b_i \\ A_{i,:}x + a_i > B_{i,:}x + b_i \end{cases} \Rightarrow \forall\, J \in \partial_B F(x), J_{i,:} = \begin{cases} A_{i,:} = B_{i,:} \\ A_{i,:} \\ B_{i,:} \end{cases}$$

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

$\min(\mathcal{A}, \mathcal{B})$ non-diff $\Leftrightarrow$ affine terms equal $\Leftrightarrow$ **hyperplanes**

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

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

H_i^- or H_i^+, $\forall\, i \in [1:p]$: $H_i^+ \Leftrightarrow J_{i,:} = A_{i,:}$, $H_i^- \Leftrightarrow J_{i,:} = B_{i,:}$

Computing the B-differential

$f, g \in \mathcal{C}^1$: $\min(f(x), g(x))$ diff $\Leftrightarrow$ $f(x) \neq g(x)$ or $f'(x) = g'(x)$.

Important: F is piecewise affine: F' is piecewise **constant**.

$$\begin{cases} A_{i,:} = B_{i,:} \\ A_{i,:}x + a_i < B_{i,:}x + b_i \\ A_{i,:}x + a_i > B_{i,:}x + b_i \end{cases} \Rightarrow \forall\, J \in \partial_B F(x), J_{i,:} = \begin{cases} A_{i,:} = B_{i,:} \\ A_{i,:} \\ B_{i,:} \end{cases}$$

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

$\min(\mathcal{A}, \mathcal{B})$ non-diff $\Leftrightarrow$ affine terms equal $\Leftrightarrow$ **hyperplanes**

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

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

H_i^- or H_i^+, $\forall\, i \in [1:p]$: $H_i^+ \Leftrightarrow J_{i,:} = A_{i,:}$, $H_i^- \Leftrightarrow J_{i,:} = B_{i,:}$

Computing the B-differential

$f, g \in \mathcal{C}^1$: $\min(f(x), g(x))$ diff $\Leftrightarrow$ $f(x) \neq g(x)$ or $f'(x) = g'(x)$.

Important: F is piecewise affine: F' is piecewise **constant**.

$$\begin{cases} A_{i,:} & = & B_{i,:} \\ A_{i,:}x + a_i & < & B_{i,:}x + b_i \\ A_{i,:}x + a_i & > & B_{i,:}x + b_i \end{cases} \Rightarrow \forall\, J \in \partial_B F(x), J_{i,:} = \begin{cases} A_{i,:} = B_{i,:} \\ A_{i,:} \\ B_{i,:} \end{cases}$$

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

$\min(\mathcal{A}, \mathcal{B})$ non-diff $\Leftrightarrow$ affine terms equal $\Leftrightarrow$ **hyperplanes**

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

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

H_i^- or H_i^+, $\forall\, i \in [1:p]$: $H_i^+ \Leftrightarrow J_{i,:} = A_{i,:}$, $H_i^- \Leftrightarrow J_{i,:} = B_{i,:}$

Computing the B-differential

$f, g \in \mathcal{C}^1$: $\min(f(x), g(x))$ diff $\Leftrightarrow f(x) \neq g(x)$ or $f'(x) = g'(x)$.

Important: F is piecewise affine: F' is piecewise **constant**.

$$\begin{cases} A_{i,:} & = & B_{i,:} \\ A_{i,:}x + a_i & < & B_{i,:}x + b_i \\ A_{i,:}x + a_i & > & B_{i,:}x + b_i \end{cases} \Rightarrow \forall\, J \in \partial_B F(x), J_{i,:} = \begin{cases} A_{i,:} = B_{i,:} \\ A_{i,:} \\ B_{i,:} \end{cases}$$

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

$\min(\mathcal{A}, \mathcal{B})$ non-diff $\Leftrightarrow$ affine terms equal $\Leftrightarrow$ **hyperplanes**

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

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

H_i^- or H_i^+, $\forall\, i \in [1:p]$: $H_i^+ \Leftrightarrow J_{i,:} = A_{i,:}$, $H_i^- \Leftrightarrow J_{i,:} = B_{i,:}$

Computing the B-differential

$f, g \in \mathcal{C}^1$: $\min(f(x), g(x))$ diff $\Leftrightarrow$ $f(x) \neq g(x)$ or $f'(x) = g'(x)$.

Important: F is piecewise affine: F' is piecewise **constant**.

$$\begin{cases} A_{i,:} & = & B_{i,:} \\ A_{i,:}x + a_i & < & B_{i,:}x + b_i \\ A_{i,:}x + a_i & > & B_{i,:}x + b_i \end{cases} \Rightarrow \forall\, J \in \partial_B F(x), J_{i,:} = \begin{cases} A_{i,:} = B_{i,:} \\ A_{i,:} \\ B_{i,:} \end{cases}$$

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

$\min(\mathcal{A}, \mathcal{B})$ non-diff $\Leftrightarrow$ affine terms equal $\Leftrightarrow$ **hyperplanes**

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

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

H_i^- or H_i^+, $\forall\, i \in [1:p]$: $H_i^+ \Leftrightarrow J_{i,:} = A_{i,:}$, $H_i^- \Leftrightarrow J_{i,:} = B_{i,:}$

Computing the B-differential

$f, g \in \mathcal{C}^1$: $\min(f(x), g(x))$ diff $\Leftrightarrow$ $f(x) \neq g(x)$ or $f'(x) = g'(x)$.

Important: F is piecewise affine: F' is piecewise **constant**.

$$\begin{cases} A_{i,:} &= B_{i,:} \\ A_{i,:}x + a_i &< B_{i,:}x + b_i \\ A_{i,:}x + a_i &> B_{i,:}x + b_i \end{cases} \Rightarrow \forall\, J \in \partial_B F(x), J_{i,:} = \begin{cases} A_{i,:} = B_{i,:} \\ A_{i,:} \\ B_{i,:} \end{cases}$$

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

$\min(\mathcal{A}, \mathcal{B})$ non-diff $\Leftrightarrow$ affine terms equal $\Leftrightarrow$ **hyperplanes**

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

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

H_i^- or H_i^+, $\forall\, i \in [1:p]$: $H_i^+ \Leftrightarrow J_{i,:} = A_{i,:}$, $H_i^- \Leftrightarrow J_{i,:} = B_{i,:}$

Computing the B-differential

$f, g \in \mathcal{C}^1$: $\min(f(x), g(x))$ diff $\Leftrightarrow$ $f(x) \neq g(x)$ or $f'(x) = g'(x)$.

Important: F is piecewise affine: F' is piecewise **constant**.

$$\begin{cases} A_{i,:} & = & B_{i,:} \\ A_{i,:}x + a_i & < & B_{i,:}x + b_i \\ A_{i,:}x + a_i & > & B_{i,:}x + b_i \end{cases} \Rightarrow \forall\, J \in \partial_B F(x), J_{i,:} = \begin{cases} A_{i,:} = B_{i,:} \\ A_{i,:} \\ B_{i,:} \end{cases}$$

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

$\min(\mathcal{A}, \mathcal{B})$ non-diff $\Leftrightarrow$ affine terms equal $\Leftrightarrow$ **hyperplanes**

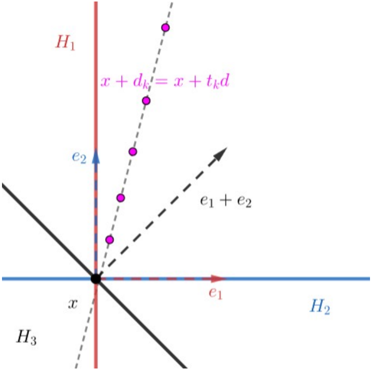
$$\mathbb{R}^n = H_i^- \cup H_i \cup H_i^+, \quad H_i^{-,+} = \{x \in \mathbb{R}^n : v_i^\mathsf{T} x <,> 0\}$$

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

H_i^- or H_i^+, $\forall\, i \in [1:p]$: $H_i^+ \Leftrightarrow J_{i,:} = A_{i,:}$, $H_i^- \Leftrightarrow J_{i,:} = B_{i,:}$

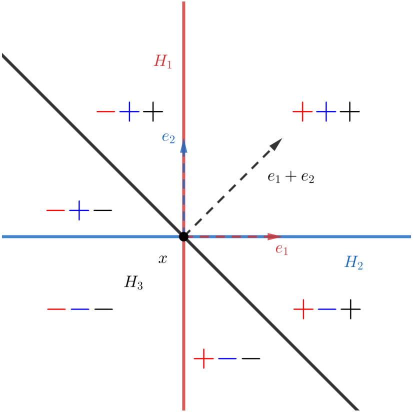
Directions and hyperplanes - 1

$$V = [v_1 \quad v_2 \quad v_3] = [e_1 \quad e_2 \quad e_1 + e_2]$$



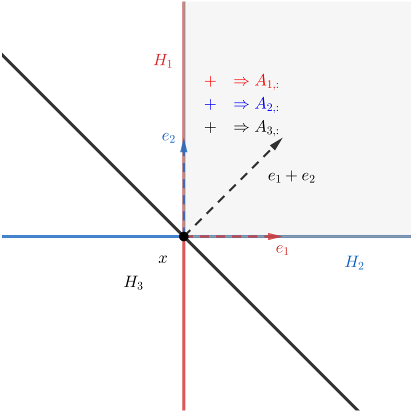
Red, blue, black: H_i and v_i. In magenta, points of the form $x + t_k d$, $t_k \searrow 0$. The points remain on the same sides of the hyperplanes along k: the J is constant; no need for sequences, points are sufficient.

Directions and hyperplanes - 2



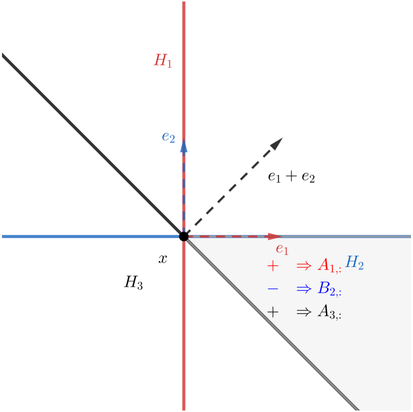
$\forall\ i \in [1:p]$, 2 possibilities: maximum of 2^p Jacobians. Here, 6 among the $2^3 = 8$ possible Jacobians exist in $\partial_B F$

Directions and hyperplanes - 3



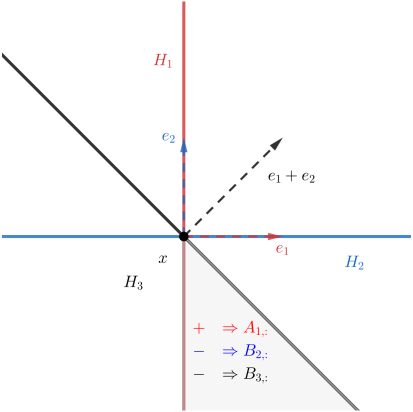
In the grey area, the points remain on the same sides of the hyperplanes: the Jacobian matrix remains constant.

Directions and hyperplanes - 3



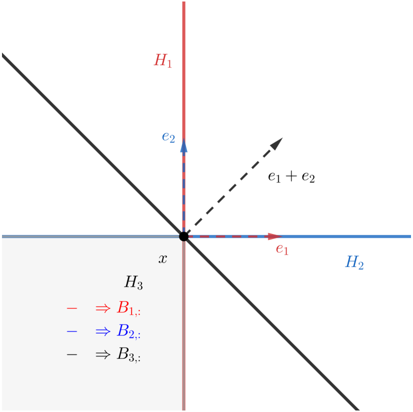
In the grey area, the points remain on the same sides of the hyperplanes: the Jacobian matrix remains constant.

Directions and hyperplanes - 3



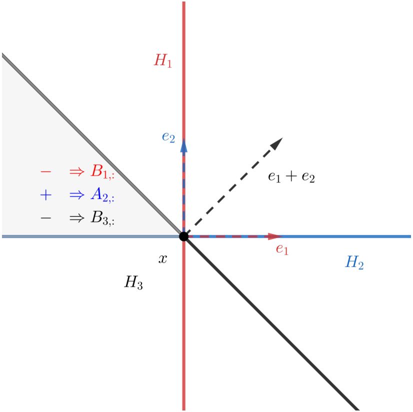
In the grey area, the points remain on the same sides of the hyperplanes: the Jacobian matrix remains constant.

Directions and hyperplanes - 3



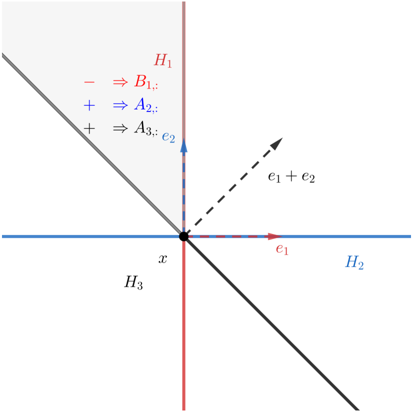
In the grey area, the points remain on the same sides of the hyperplanes: the Jacobian matrix remains constant.

Directions and hyperplanes - 3



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

Directions and hyperplanes - 3



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

Summary

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

Fundamental question

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

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

Summary

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

Fundamental question

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

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

From literature

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

Very impressive results such as $|\partial_B F(x)|$ (Winder [Win66])

$$
\begin{aligned}
|\partial_B F(x)| &= \sum_{T \subset \{H_i, i \in [1:m]\}} (-1)^{|T| - n + \dim(\bigcap H_t, t \in T)} \\
&= \sum_{\mathcal{V} \subset \{v_1, \dots, v_m\}} (-1)^{|\mathcal{V}| - \operatorname{rank}(\mathcal{V})} \\
&= \sum_{\mathcal{V} \subset \{v_1, \dots, v_m\}} (-1)^{\operatorname{null}(\mathcal{V})}
\end{aligned}
$$

or upper bounds but not exactly $\partial_B F(x)$.

From literature

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

Very impressive results such as $|\partial_B F(x)|$ (Winder [Win66])

$$
\begin{aligned}
|\partial_B F(x)| &= \sum_{T \subset \{H_i, i \in [1:m]\}} (-1)^{|T| - n + \dim(\bigcap H_t, t \in T)} \\
&= \sum_{\mathcal{V} \subset \{v_1, \dots, v_m\}} (-1)^{|\mathcal{V}| - \mathrm{rank}(\mathcal{V})} \\
&= \sum_{\mathcal{V} \subset \{v_1, \dots, v_m\}} (-1)^{\mathrm{null}(\mathcal{V})}
\end{aligned}
$$

or upper bounds but not exactly $\partial_B F(x)$.

Plan

Main reasoning

Algorithm from [RČ18]:

- recursive process that adds hyperplanes one at a time
- uses a tree structure
- each node has one or two descendants,
- depending on linear feasibility (optimization) check

Or same tree structure but in a 'dual' way ('dual algorithm').

Main reasoning

Algorithm from [RČ18]:

- recursive process that adds hyperplanes one at a time
- uses a tree structure
- each node has one or two descendants,
- depending on linear feasibility (optimization) check

Or same tree structure but in a 'dual' way ('dual algorithm').

Illustration of the regions and tree on the previous example

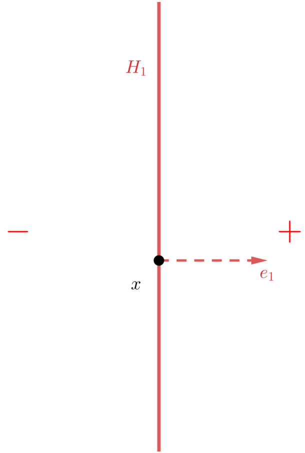


Illustration of the regions and tree on the previous example

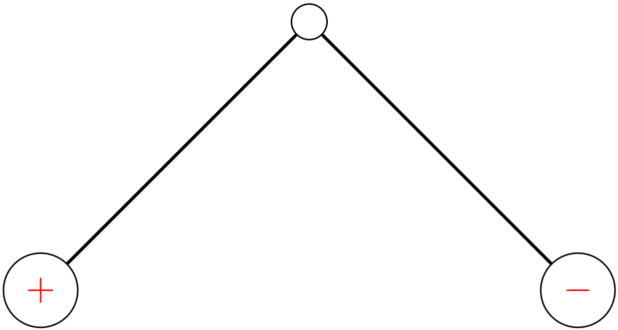


Illustration of the regions and tree on the previous example

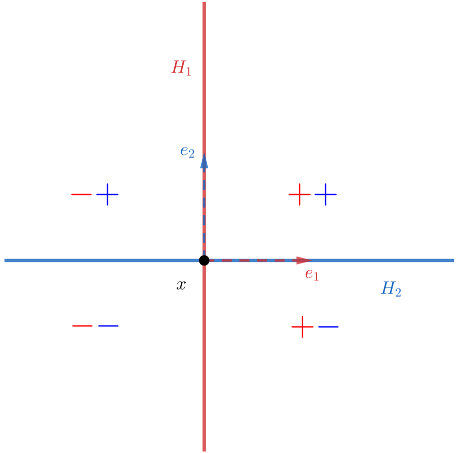


Illustration of the regions and tree on the previous example

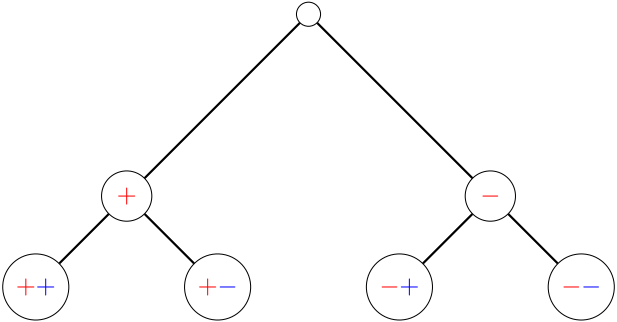


Illustration of the regions and tree on the previous example

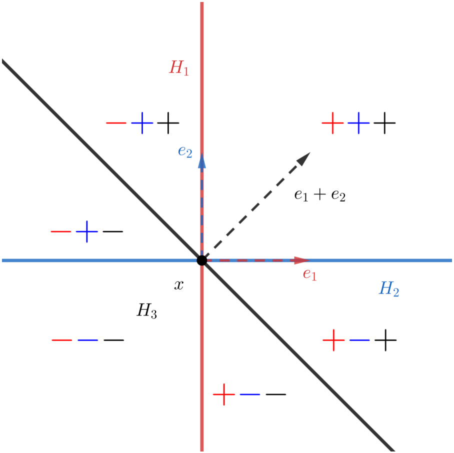
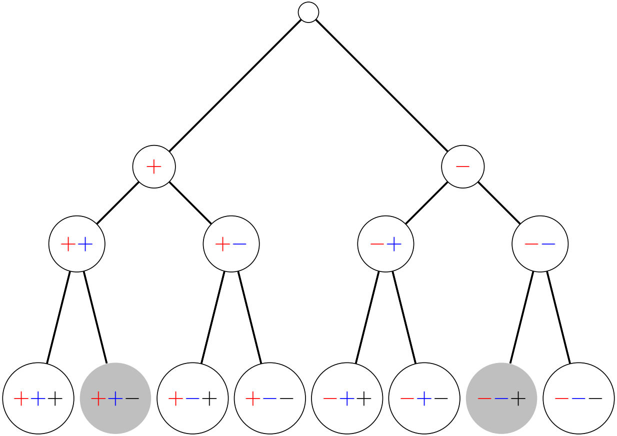
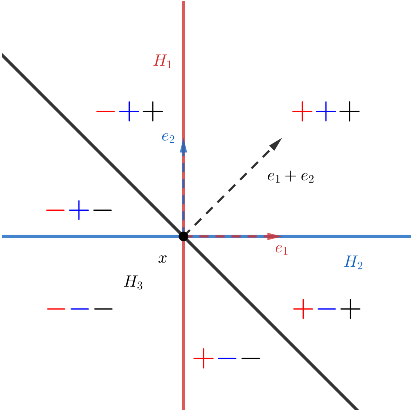


Illustration of the regions and tree on the previous example



Plan

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.

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:

$$\begin{cases} \exists\, d \in \mathbb{R}^n : Md > 0_{\mathbb{R}^p} \\ \exists\, \gamma \in \mathbb{R}^p_+ \backslash \{0\} : M^\top \gamma = 0 \end{cases} \quad (3)$$

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

"Feasibility of a system <u>or</u> 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:
$M = \mathrm{diag}(s) V^\mathsf{T} \to Md = (s_1 v_1^\mathsf{T} d; \ldots; s_p v_p^\mathsf{T} d)$

"Feasibility of a system <u>or</u> 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:

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

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

"Feasibility of a system <u>or</u> 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]]
- slight improvements on the overall tree
- the dual algorithm detecting infeasibilities
- normal tree algorithm but with some infeasibility detection

Quite significant improvements brought by duality!

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

Quite significant improvements brought by duality!

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

Quite significant improvements brought by duality!

Numerical results

	CPU times (in sec)						
	original	Improved tree-1		Improved tree-2		Fully dual tree	
Problem	code	Time	Ratio	Time	Ratio	Time	Ratio
rand-4-8-2	1.06	0.10	10.75	0.02	48.44	0.03	36.67
rand-7-9-4	1.13	0.45	2.51	0.29	3.95	0.02	68.67
rand-7-13-5	11.06	4.29	2.58	2.94	3.76	0.25	44.60
rand-8-15-7	64.79	29.53	2.19	27.59	2.35	4.54	14.29
rand-9-16-8	157.05	78.01	2.01	81.61	1.92	18.87	8.32
rand-10-17-9	352.42	196.09	1.80	213.48	1.65	70.19	5.02
srand-8-20-4	874.01	323.56	2.70	649.61	1.35	705.36	1.24
rc-2d-20-6	12.68	0.35	36.06	0.26	48.78	0.26	49.63
rc-2d-20-7	23.01	0.56	40.87	0.53	43.06	0.45	51.50
rc-perm-6	62.89	0.84	74.44	2.33	27.03	2.46	25.61
rc-perm-8	6589.31	85.70	76.89	1599.53	4.12	5290.13	1.25
rc-ratio-20-5-7	91.57	27.43	3.34	29.70	3.08	20.54	4.46
rc-ratio-20-5-9	88.24	25.21	3.50	27.54	3.20	17.75	4.97
rc-ratio-20-7-7	581.28	241.24	2.41	506.67	1.15	447.83	1.30
rc-ratio-20-7-9	460.64	162.95	2.83	315.67	1.46	234.72	1.96
Mean			16.60		13.90		30.31
Median			3.24		4.12		27.80

Conclusion

Duality and matroids brought considerable improvements

Future work

- adaptations to affine hyperplanes (for themselves)
- Julia code for this in development
- $\partial_B \min(\mathrm{nonlinear})$

Thank you for your attention! Any question?

Conclusion

Duality and matroids brought considerable improvements

Future work

- adaptations to affine hyperplanes (for themselves)
- Julia code for this in development
- $\partial_B \min(\text{nonlinear})$

Thank you for your attention! Any question?

Conclusion

Duality and matroids brought considerable improvements

Future work

- adaptations to affine hyperplanes (for themselves)
- Julia code for this in development
- $\partial_B \min(\text{nonlinear})$

Thank you for your attention! Any question?

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

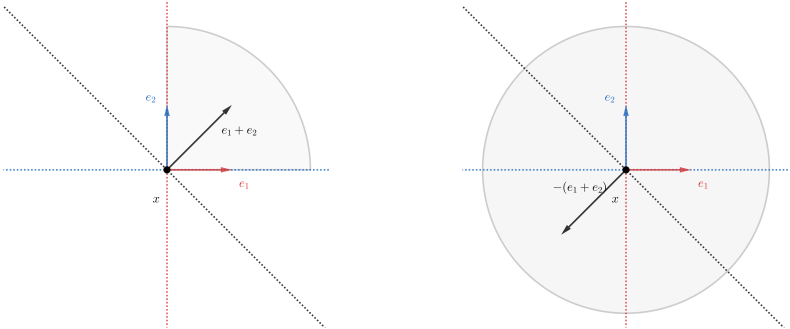
[QS93] L. Qi and J. Sun. “A nonsmooth version of Newton’s method”. In: *Mathematical Programming* 58 (1993). [doi], pp. 353–367.

[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.

[Win66] Robert O. Winder. “Partitions of N-space by hyperplanes”. In: *SIAM Journal on Applied Mathematics* 14.4 (1966). [doi], pp. 811–818.

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 $(v_1, \dots, v_{k-1})$; 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. :
  $\forall\ i \in [1 : k-1],\ s_i v_i^\top d_s^{k-1} > 0$

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 $(v_1, \ldots, v_{k-1})$; 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. :
  $\forall\ i \in [1 : k-1],\ s_i v_i^{\mathsf{T}} d_s^{k-1} > 0$
- $v_k^{\mathsf{T}} d_s^{k-1} > 0 \Rightarrow \begin{cases} +v_k^{\mathsf{T}} d_s^{k-1} > 0 \\ s_i v_i^{\mathsf{T}} d_s^{k-1} > 0 \end{cases} \checkmark, \begin{cases} -v_k^{\mathsf{T}} d > 0 \\ s_i v_i^{\mathsf{T}} d > 0 \end{cases} ? \to$ L.O.
- $v_k^{\mathsf{T}} d_s^{k-1} < 0 \Rightarrow \begin{cases} -v_k^{\mathsf{T}} d_s^{k-1} > 0 \\ s_i v_i^{\mathsf{T}} d_s^{k-1} > 0 \end{cases} \checkmark, \begin{cases} +v_k^{\mathsf{T}} d > 0 \\ s_i v_i^{\mathsf{T}} d > 0 \end{cases} ? \to$ L.O.

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. :
  $\forall\ i \in [1 : k-1],\ s_i v_i^\top d_s^{k-1} > 0$
- $v_k^\top d_s^{k-1} > 0 \Rightarrow \begin{cases} +v_k^\top d_s^{k-1} > 0 \\ s_i v_i^\top d_s^{k-1} > 0 \end{cases} \checkmark, \begin{cases} -v_k^\top d > 0 \\ s_i v_i^\top d > 0 \end{cases} ? \rightarrow$ L.O.
- $v_k^\top d_s^{k-1} < 0 \Rightarrow \begin{cases} -v_k^\top d_s^{k-1} > 0 \\ s_i v_i^\top d_s^{k-1} > 0 \end{cases} \checkmark, \begin{cases} +v_k^\top d > 0 \\ s_i v_i^\top d > 0 \end{cases} ? \rightarrow$ L.O.
- $v_k^\top d_s^{k-1} = 0 \Rightarrow$ both systems $\checkmark$ 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