On the B-differential of the componentwise minimum of two affine vector functions

Jean-Pierre Dussault · Jean Charles Gilbert · Baptiste Plaquevent-Jourdain

March 28, 2023

Abstract This paper focuses on the description and computation of the B-differential of the componentwise minimum of two affine vector functions. This issue arises in the reformulation of the linear complementarity problem with the Min C-function. The question has many equivalent formulations and we identify some of them in linear algebra, convex analysis and discrete geometry. These formulations are used to state some properties of the B-differential, like its symmetry, condition for its completeness, its connectivity, bounds on its cardinal, *etc.* The set to specify has a finite number of elements, which may grow exponentially with the range space dimension of the functions, so that its description is most often algorithmic. We present first an incremental-recursive approach avoiding to solve any optimization subproblem, which is based on the notion of matroid circuit and the related introduced concept of stem vector. Next, we propose modifications, adapted to the problem at stake, of an algorithm introduced by Rada and Černý in 2018 for determining the cells of an arrangement in the space of hyperplanes having a point in common. Measured in CPU time on the considered test-problems, the mean acceleration ratios of the proposed algorithms, with respect to the one of Rada and Černý, are in the range 7..15, and this speed-up can exceed 30, depending on the problem and the approach.

Keywords B-differential · bipartition of a finite set · C-differential · complementarity problem · componentwise minimum of functions · connectivity · dual approach · Gordan's alternative · hyperplane arrangement · matroid circuit · pointed cone · stem vector · strict linear inequalities · symmetry · Winder's formula.

Mathematics Subject Classification (2020) 05A18 · 05C40 · 26A24 · 26A27 · 46N10 · 47A50 · 47A63 · 49J52 · 49N15 · 52C35 · 65Y20 · 65K15 · 90C33 · 90C46.

J.-P. Dussault
Département d'Informatique, Fac. des Sciences, Univ. de Sherbrooke, Québec, Canada
E-mail: Jean-Pierre.Dussault@Usherbrooke.ca, ORCID 0000-0001-7253-7462

J.Ch. Gilbert
Inria Paris, 2 rue Simone Iff, CS 42112, 75589 Paris Cedex 12, France
Département de Mathématiques, Fac. des Sciences, Univ. de Sherbrooke, Québec, Canada
E-mail: Jean-Charles.Gilbert@inria.fr, ORCID 0000-0002-0375-4663

B. Plaquevent-Jourdain
Département de Mathématiques, Fac. des Sciences, Univ. de Sherbrooke, Québec, Canada
Inria Paris, 2 rue Simone Iff, CS 42112, 75589 Paris Cedex 12, France
E-mail: Baptiste.Plaquevent-Jourdain@USherbrooke.ca, ORCID 0000-0001-7055-4568

1 Introduction

Let $\mathbb{E}$ and $\mathbb{F}$ be two real vector spaces of finite dimension $n := \dim \mathbb{E}$ and $m := \dim \mathbb{F}$. The *B-differential at* $x \in \mathbb{E}$ of a function $H : \mathbb{E} \to \mathbb{F}$ is the set denoted and defined by

$$\partial_B H(x) := \{J \in \mathcal{L}(\mathbb{E}, \mathbb{F}) : H'(x_k) \to J \text{ for a sequence } \{x_k\} \subseteq \mathcal{D}_H \text{ converging to } x\},$$

where $\mathcal{L}(\mathbb{E}, \mathbb{F})$ is the set of linear (continuous) maps from $\mathbb{E}$ to $\mathbb{F}$ and $\mathcal{D}_H$ is the set of points at which H is (Fréchet) differentiable (its derivative at x is denoted by $H'(x)$). Recall that a locally Lipschitz continuous function is differentiable almost everywhere in the sense of the Lebesgue measure (Rademacher's theorem [55]) and this property has the consequence that the B-differential of a locally Lipschitz function is nonempty everywhere [19]. The B-differential is an intermediate set used to define the C-differential (C for Clarke [19]) of H at x, which is denoted and defined by

$$\partial_C H(x) := \operatorname{co} \partial_B H(x), \tag{1.1}$$

where $\operatorname{co} S$ denotes the convex hull of a set S [57,41,15]. Both intervene in the specification of conditions ensuring the local convergence of the semismooth Newton algorithm [52,53,61], which can be a motivation for being interested in that concept.

In this paper, we focus on the description of the B-differential of H at x when $H : \mathbb{R}^n \to \mathbb{R}^m$ is the componentwise minimum of two affine functions $x \mapsto Ax + a$ and $x \mapsto Bx + b$, where A, $B \in \mathbb{R}^{m\times n}$ and a, $b \in \mathbb{R}^m$. Hence, H is defined at x by

$$H(x) = \min(Ax + a, Bx + b), \tag{1.2}$$

where the minimum operator "min" acts componentwise (for two vectors u, $v \in \mathbb{R}^m$ and $i \in [1:m] := \{1, \ldots, m\}$: $[\min(u, v)]_i := \min(u_i, v_i)$). A motivation to look at the B-differential of that function H comes from the fact that, when $m = n$ and H is given by (1.2), as explained below, the equation

$$H(x) = 0 \tag{1.3}$$

is a reformulation of the *balanced* [27] *Linear Complementarity Problem* (LCP)

$$0 \leqslant (Ax + a) \perp (Bx + b) \geqslant 0. \tag{1.4}$$

This system expresses the fact that a point $x \in \mathbb{R}^n$ is sought such that $Ax + a \geqslant 0$, $Bx + b \geqslant 0$ and $(Ax + a)^{\mathsf{T}}(Bx + b) = 0$ (the superscript "$^{\mathsf{T}}$" is used here and below to denote vector or matrix transposition). Problem (1.4) is a special case of the so-called (*extended*) *vertical LCP*, which uses more than two matrices and vectors in its formulation [21,64,68]. In the *standard LCP*, A is the identity matrix and $a = 0$ [47,22].

The reformulation (1.3) of (1.4) is based on the fact that, for two real numbers α and β, $\min(\alpha, \beta) = 0$ if and only if $\alpha \geqslant 0$, $\beta \geqslant 0$ and $\alpha\beta = 0$ [1,50]. This reformulation serves as the basis for a number of solving methods and investigations [1,44,49,50,51,33,7,8,42,9,25,26,27]. If (1.4) stands alone, it is appropriate to have $m = n$, but (1.4) may be part of a system with other constraints to satisfy [45,46,10], in which case $m \leqslant n$. In the computation of the B-differential of the Min function (1.2), m and n may be unrelated.

Occasionally, we shall refer to the nonlinear version of the above problem, in which a function $\tilde{H} : \mathbb{E} \to \mathbb{R}^m$ is defined at $x \in \mathbb{E}$ by

$$\tilde{H}(x) := \min(F(x), G(x)), \tag{1.5}$$

where F and $G : \mathbb{E} \to \mathbb{R}^m$ are two functions and the "min" operator still acts componentwise. The equation $\tilde{H}(x) = 0$ is then a reformulation of the complementarity problem "$0 \leqslant F(x) \perp G(x) \geqslant 0$".

As a first general remark, let us quote the fact that the B-differential of H cannot be deduced from the knowledge of the B-differential of its scalar components $H_i : x \in \mathbb{E} \to H_i(x) \in \mathbb{R}$, for $i \in [1:m]$, which is trivial in the present context. Indeed, if [19; proposition 2.6.2(e)]

$$\partial_B H(x) \subseteq \partial_B H_1(x) \times \cdots \times \partial_B H_m(x), \tag{1.6}$$

equality in this inclusion may not always hold (see [33; § 7.1.15] and almost all the examples and test-cases below). Therefore, all the components of H must be taken into account simultaneously.

The B-differential of H at x is a finite set, made of Jacobians J whose ith row is $A_{i,:}$ or $B_{i,:}$ (proposition 2.2). Consequently, its cardinal can be exponential in m and it occurs that its full mathematical description is a tricky task, essentially when there are many indices i for which $(Ax+a)_i = (Bx+b)_i$ and $A_{i,:} \neq B_{i,:}$, a situation that makes H nondifferentiable (lemma 2.1). Then, a rich panorama of configurations appears, which is barely glimpsed in this contribution.

The paper starts with a background section (section 2), which recalls a basic property of the minimum of two functions (lemma 2.1) and gives us a first perception of the structure of the B-differential of the function H, in particular its finite nature (proposition 2.2). A useful technical lemma is also presented (lemma 2.5).

In section 3, it is shown that the problem of computing $\partial_B H(x)$ has a rich panel of equivalent formulations, related to various areas of mathematics. We have quoted two forms of the problem in *linear algebra*, which are dual to each other (section 3.2), two equivalent problems in *convex analysis* (section 3.3) and a last equivalent problem, which arises in *computational discrete geometry* and deals with the arrangement of hyperplanes having the zero point in common (section 3.4).

Section 4 gives some properties of the B-differential of H, recalls Winder's formula of its cardinal, provides some lower and upper bounds on this one, proves necessary and sufficient conditions so that two extreme configurations occur and highlights two links between the B-differential and C-differential.

Section 5 presents algorithms for computing one (section 5.1) or all (section 5.2) the Jacobians of $\partial_B H(x)$. In the latter case, the algorithms construct a tree incrementally and recursively (section 5.2.1), as proposed by Rada and Černý [54]. On the one hand (section 5.2.2), an algorithm based on the notion of matroid circuit of the matrix V expressing the gap of differentiability is proposed; it has the nice feature of requiring no linear optimization (LO) problem to solve. On the other hand (section 5.2.4), various modifications of the algorithm of Rada and Černý [54] are proposed with the goal of decreasing the number of LO problems to solve. Numerical experiments are reported (section 5.2.6), showing that the proposed algorithms significantly improve the performance of the Rada and Černý method, with mean (resp. median) acceleration ratios in the range 7..15 (resp. 3..14), measured by the computing time. This speed-up exceeds 30, for some algorithms and test-problems.

This paper is an abridged version of the more detailed report [28].

Notation. We denote by $|S|$ the number of elements of a set S (i.e., its *cardinal*). The *power set* of a set S is denoted by $\mathfrak{P}(S)$. The set of bipartitions (I, J) of a set K is denoted by $\mathfrak{B}(K)$: $I \cup J = K$ and $I \cap J = \varnothing$. The sets of nonzero natural and real numbers are denoted by $\mathbb{N}^*$ and $\mathbb{R}^*$, respectively. The *sign of a real number* is the multifunction $\operatorname{sgn} : \mathbb{R} \multimap \mathbb{R}$ defined by $\operatorname{sgn}(t) = \{1\}$ if $t > 0$, $\operatorname{sgn}(t) = \{-1\}$ if $t < 0$ and $\operatorname{sgn}(0) = [-1, 1]$. We note $\mathbb{R}^n_+ := \{x \in \mathbb{R}^n : x \geqslant 0\}$ and $\mathbb{R}^n_{++} := \{x \in \mathbb{R}^n : x > 0\}$ (strict inequalities must also be understood componentwise; hence $x > 0$ means $x_i > 0$ for all indices i). For a subset S of a vector space, we denote by $\operatorname{vect}(S)$ the subspace spanned by S. The vector of all one's, in a real space whose dimension is given by the context, is denoted by e. The Hadamard product of u and $v \in \mathbb{R}^n$ is the vector $u \cdot v \in \mathbb{R}^n$ whose ith component is $u_i v_i$. The range space of an $m \times n$ matrix A is denoted by $\mathcal{R}(A)$, its null space by $\mathcal{N}(A)$, its rank is $\operatorname{rank}(A) := \dim \mathcal{R}(A)$ and its nullity is $\operatorname{null}(A) := \dim \mathcal{N}(A) = n - \operatorname{rank}(A)$ by the rank-nullity theorem. The ith row (resp. column) of A is denoted by $A_{i,:}$ (resp. $A_{:,i}$). Transposition operates after a row/column selection: $A^{\mathsf{T}}_{i,:}$ is a short notation for the column vector $(A_{i,:})^{\mathsf{T}}$ and $A^{\mathsf{T}}_{:,i}$ is a short notation for the row vector $(A_{:,i})^{\mathsf{T}}$. For a vector α, $\operatorname{Diag}(\alpha)$ is the square diagonal matrix with the α_i's on its diagonal.

2 Background

Recall that $F : \mathbb{E} \to \mathbb{F}$ is said to be (*Fréchet*) *differentiable* at x if $F(x+d) = F(x) + Ld + o(\|d\|)$ for some $L \in \mathcal{L}(\mathbb{E}, \mathbb{F})$, in which case one denotes by $F'(x) = L$ the *derivative* of F at x. We say below that F is *continuously differentiable at* x if it is differentiable near x (like in [19], "near" means here and below "in a neighborhood of" in the topological sense) and if its derivative is continuous at x.

The next famous lemma recalls a necessary and sufficient condition guaranteeing the differentiability of the minimum of two scalar functions (see [52; 1993, final remarks (1)], [17; 2011, theorem 2.1] and [28]).

Lemma 2.1 (differentiability of the Min function) *Let f and $g : \mathbb{E} \to \mathbb{R}$ be two functions and $h : \mathbb{E} \to \mathbb{R}$ be defined by $h(\cdot) := \min(f(\cdot), g(\cdot))$. Suppose that f and g are differentiable at a point $x \in \mathbb{E}$.*

1) *If $f(x) < g(x)$, then h is differentiable at x and $h'(x) = f'(x)$.*
2) *If $f(x) > g(x)$, then h is differentiable at x and $h'(x) = g'(x)$.*
3) *If $f(x) = g(x)$, then h is differentiable at x if and only if $f'(x) = g'(x)$. In this case, $h'(x) = f'(x) = g'(x)$.*

The previous lemma shows the relevance of the following index sets:

$$\mathcal{A}(x) := \{i \in [1:m] : (Ax+a)_i < (Bx+b)_i\}, \tag{2.1a}$$
$$\mathcal{B}(x) := \{i \in [1:m] : (Ax+a)_i > (Bx+b)_i\}, \tag{2.1b}$$
$$\mathcal{E}(x) := \{i \in [1:m] : (Ax+a)_i = (Bx+b)_i\}, \tag{2.1c}$$
$$\mathcal{E}^{=}(x) := \{i \in \mathcal{E}(x) : A_{i,:} = B_{i,:}\}, \tag{2.1d}$$
$$\mathcal{E}^{\neq}(x) := \{i \in \mathcal{E}(x) : A_{i,:} \neq B_{i,:}\}. \tag{2.1e}$$

To simplify the presentation, we assume in the sequel that

$$\mathcal{E}^{\neq}(x) = [1:p],$$

for some $p \in [0:m]$ ($p = 0$ if and only if $\mathcal{E}^{\neq}(x) = \varnothing$).

The next proposition describes the superset $\overline{\partial}_B H(x)$ of $\partial_B H(x)$ given in the right-hand side of (1.6) (see [43; 1998, § 2] in a somehow different context, [24; 2000, before (8)] and [28] for a meticulous proof). This Cartesian product actually reads

$$\begin{aligned} \overline{\partial}_B H(x) := \{J \in \mathcal{L}(\mathbb{E}, \mathbb{R}^m) : & J_{i,:} = A_{i,:}, \text{ if } i \in \mathcal{A}(x), \\ & J_{i,:} = A_{i,:} = B_{i,:}, \text{ if } i \in \mathcal{E}^{=}(x), \\ & J_{i,:} \in \{A_{i,:}, B_{i,:}\}, \text{ if } i \in \mathcal{E}^{\neq}(x), \\ & J_{i,:} = B_{i,:}, \text{ if } i \in \mathcal{B}(x)\}. \end{aligned} \tag{2.2}$$

Note that $\partial_B H(x)$ may differ from $\overline{\partial}_B H(x)$: if $n = 1$, $m = 2$, $F(x) \equiv 0$ and $G(x) \equiv xe$, one has $\partial_B H(0) = \{\binom{0}{0}, \binom{1}{1}\}$, while $\overline{\partial}_B H(0) = \{\binom{0}{0}, \binom{1}{0}, \binom{0}{1}, \binom{1}{1}\}$.

Proposition 2.2 (superset of $\partial_B H(x)$) *One has* $\partial_B H(x) \subseteq \overline{\partial}_B H(x) = \partial_B H_1(x) \times \cdots \times \partial_B H_m(x)$. *In particular,* $|\partial_B H(x)| \leqslant 2^p$.

The previous proposition shows that $\partial_B H(x)$ is a finite set. It also naturally leads to the next definition.

Definition 2.3 (complete B-differential) We say that the B-differential of H at $x \in \mathbb{R}^n$ is *complete* if $\partial_B H(x) = \overline{\partial}_B H(x)$ or, equivalently, if $|\partial_B H(x)| = 2^p$. □

Definitions 2.4 (symmetry in $\partial_B H(x)$) For $x \in \mathbb{E}$, we say that the Jacobian $\tilde{J} \in \overline{\partial}_B H(x)$ is *symmetric* to the Jacobian $J \in \overline{\partial}_B H(x)$ if

$$\tilde{J}_{i,:} = \begin{cases} A_{i,:} & \text{if } i \in \mathcal{E}^{\neq}(x) \text{ and } J_{i,:} = B_{i,:}, \\ B_{i,:} & \text{if } i \in \mathcal{E}^{\neq}(x) \text{ and } J_{i,:} = A_{i,:}. \end{cases}$$

The B-differential $\partial_B H(x)$ itself is said to be *symmetric* if each Jacobian $J \in \partial_B H(x)$ has its symmetric Jacobian $\tilde{J}$ in $\partial_B H(x)$. □

We shall use the following lemma, which, for the sake of generality, is written in a slightly more abstract formalism than the one we need below (one could take $\mathbb{E} = \mathbb{R}^n$ and the Euclidean scalar product for $\langle \cdot, \cdot \rangle$). It is a refinement of [17; lemma 2.1].

Lemma 2.5 (discriminating covectors) *Suppose that* $(\mathbb{E}, \langle \cdot, \cdot \rangle)$ *is a Euclidean vector space,* $p \in \mathbb{N}^*$ *and* $v_1, \ldots, v_p$ *are* p *distinct vectors of* $\mathbb{E}$. *Then, the set of vectors* $\xi \in \mathbb{E}$ *such that* $|\{\langle \xi, v_i \rangle\}_{i \in [1:p]}| = p$ *is dense in* $\mathbb{E}$.

Proof Denote by Ξ the set of vectors $\xi \in \mathbb{E}$ such that $|\{\langle \xi, v_i \rangle\}_{i \in [1:p]}| = p$ (i.e., $\{\langle \xi, v_i \rangle\}_{i \in [1:p]}$ contains p distinct values in $\mathbb{R}$). We have to show that Ξ is dense in $\mathbb{E}$.

Take $\xi_0 \notin \Xi$, so that $\langle \xi_0, v_i \rangle = \langle \xi_0, v_j \rangle$ for some $i \neq j$ in $[1:p]$. By continuity of the scalar product, for any $\varepsilon_0 > 0$ sufficiently small, the vector $\xi_1 := \xi_0 - \varepsilon_0 (v_i - v_j)$ guarantees

$$\langle \xi_1, v_{i_1} \rangle < \langle \xi_1, v_{i_2} \rangle$$

for all i_1 and $i_2 \in [1:p]$ such that $\langle \xi_0, v_{i_1} \rangle < \langle \xi_0, v_{i_2} \rangle$ (in other words, ξ_1 maintains strict the inequalities that are strict with ξ_0). In addition

$$\langle \xi_1, v_i \rangle - \langle \xi_1, v_j \rangle = \underbrace{\langle \xi_0, v_i - v_j \rangle}_{=0} - \underbrace{\varepsilon_0 \|v_i - v_j\|^2}_{>0} < 0.$$

Therefore, one gets one more strict inequality with ξ_1 than with ξ_0. Pursuing like this, one can finally obtain a vector ξ in Ξ. This vector is arbitrarily close to ξ_0 by taking the ε_i's positive and sufficienty small. □

3 Equivalent problems

The problem of determining the B-differential of the piecewise affine function, that is the minimum (1.2) of two *affine* functions, appears in various contexts, sometimes with non straightforward connections with it (this one is recalled in section 3.1). We review some equivalent formulations in this section (see also [66,5,6] and the references therein) and give a few properties of the B-differential in this piecewise affine case. As suggested by proposition 2.2, these problems have an enumeration nature, since a finite list of mathematical objects has to be determined. This list may have a number of elements exponential in p, which makes its content difficult to specify (in this respect, the particular case where the B-differential is complete is a trivial exception). Some formulations, such as the one related to the arrangement of hyperplanes containing the origin (section 3.4), have been extensively explored, others much less. Each formulation sheds a particular light on the problem and is therefore, as such, interesting to mention and keep in mind. It also offers the possibility of introducing new algorithmic approaches to describe the B-differential.

3.1 B-differential of the minimum of two affine functions

The problem of this section was already presented in the introduction and is sometimes referred to, in this paper, as the *original problem*.

Problem 3.1 (B-differential of the minimum of two affine functions) Let be given two positive integers n and $m \in \mathbb{N}^*$, two matrices A, $B \in \mathbb{R}^{m\times n}$ and two vectors a, $b \in \mathbb{R}^m$. It is requested to compute the B-differential at some $x \in \mathbb{R}^n$ of the function $H : \mathbb{R}^n \to \mathbb{R}^m$ defined by (1.2). □

When $\mathcal{E}^{\neq}(x) \neq \varnothing$, the rows of $B - A$ with indices in $\mathcal{E}^{\neq}(x)$ will play a key role below. We denote its transpose by

$$V := (B - A)^{\mathsf{T}}_{\mathcal{E}^{\neq}(x),:} \in \mathbb{R}^{n\times p}. \tag{3.1}$$

Note that, due to their indices in $\mathcal{E}^{\neq}(x) = [1:p]$ and the definition of this index set, the columns of V are nonzero. This matrix may not always have full rank, however.

3.2 Linear algebra problems

3.2.1 Signed feasibility of strict inequality systems

Many proofs below leverage the equivalence between the original problem 3.1 and the following one. The reason is that working on problem 3.2 often allows us to propose shorter proofs.

Problem 3.2 (signed feasibility of strict inequality systems) Let be given two positive integers n and $p \in \mathbb{N}^*$ and a matrix V in $\mathbb{R}^{n\times p}$ with nonzero columns. It is requested to determine the set

$$\mathcal{S} := \{s \in \{\pm 1\}^p : \ s \bullet V^{\mathsf{T}} d > 0 \text{ is feasible for } d \in \mathbb{R}^n\}. \tag{3.2}$$

□

The link between the two problems is established by the following map

$$\sigma : J \in \overline{\partial}_B H(x) \mapsto s \in \{\pm 1\}^p, \text{ where } s_i = \begin{cases} +1 \text{ if } i \in \mathcal{E}^{\neq}(x),\ J_{i,:} = A_{i,:}, \\ -1 \text{ if } i \in \mathcal{E}^{\neq}(x),\ J_{i,:} = B_{i,:}. \end{cases} \tag{3.3a}$$

The map is well defined since $A_{i,:} \neq B_{i,:}$ when $i \in \mathcal{E}^{\neq}(x)$. Furthermore, σ is bijective since two Jacobians in $\overline{\partial}_B H(x)$ only differ by their rows with index in $\mathcal{E}^{\neq}(x)$ and that these rows can take any of the values $A_{i,:}$ or $B_{i,:}$. Actually, its reverse map is

$$\sigma^{-1} : s \in \{\pm 1\}^p \mapsto J \in \overline{\partial}_B H(x), \text{ where } J_{i,:} = \begin{cases} A_{i,:} \text{ if } i \in \mathcal{E}^{\neq}(x),\ s_i = +1, \\ B_{i,:} \text{ if } i \in \mathcal{E}^{\neq}(x),\ s_i = -1. \end{cases} \tag{3.3b}$$

The question that arises is whether σ is also a bijection between $\partial_B H(x)$ and $\mathcal{S}$.

Proposition 3.3 (bijection $\partial_B H(x) \leftrightarrow \mathcal{S}$) *Let $H : \mathbb{R}^n \to \mathbb{R}^m$ be given by (1.2), x be a point in $\mathbb{R}^n$ such that $p \neq 0$ and V be given by (3.1). Then, the map σ is a bijection from $\partial_B H(x)$ onto $\mathcal{S}$. In particular, the following properties hold.*

1) *If $J \in \partial_B H(x)$, then $\exists\, d \in \mathbb{R}^n$ such that $\sigma(J) \bullet V^\mathsf{T} d > 0$.*
2) *If $s \in \{\pm 1\}^p$ and $\exists\, d \in \mathbb{R}^n$ is such that $s \bullet V^\mathsf{T} d > 0$, then $\sigma^{-1}(s) \in \partial_B H(x)$.*
3) *Let $J \in \overline{\partial}_B H(x)$. Then, $J \in \partial_B H(x) \iff \sigma(J) \bullet V^\mathsf{T} d > 0$ is feasible for $d \in \mathbb{R}^n$.*

Proof The properties 1, 2 and 3 in the statement of the proposition are straightforward consequences of the bijectivity of $\sigma : \partial_B H(x) \to \mathcal{S}$. Now, the discussion before the proposition has shown that $\sigma : \overline{\partial}_B H(x) \mapsto \{\pm 1\}^p$ is a bijection. Therefore, $\sigma : \partial_B H(x) \mapsto \{\pm 1\}^p$ is injective and it suffices to prove that

$$\sigma(\partial_B H(x)) = \mathcal{S}. \tag{3.4a}$$

[$\subseteq$ or point 1] Let $J \in \partial_B H(x)$. We have to show that $\sigma(J) \in \mathcal{S}$. By $J \in \partial_B H(x)$, there exists a sequence $\{x_k\} \subseteq \mathcal{D}_H$ converging to x such that

$$H'(x_k) \to J. \tag{3.4b}$$

For $i \in \mathcal{E}^{\neq}(x)$, one cannot have $(Ax_k + a)_i = (Bx_k + b)_i$, since $A_{i,:} \neq B_{i,:}$ would imply that $x_k \notin \mathcal{D}_H$ (lemma 2.1). Therefore, one can find a subsequence $\mathcal{K}$ of indices k and a partition $(\mathcal{A}_0, \mathcal{B}_0)$ of $\mathcal{E}^{\neq}(x)$ such that for all $k \in \mathcal{K}$:

$$(Ax_k + a)_{\mathcal{A}_0} < (Bx_k + b)_{\mathcal{A}_0} \qquad \text{and} \qquad (Ax_k + a)_{\mathcal{B}_0} > (Bx_k + b)_{\mathcal{B}_0}. \tag{3.4c}$$

Now, fix $k \in \mathcal{K}$ and set $d := x_k - x$. Since $(Ax + a)_i = (Bx + b)_i$ for $i \in \mathcal{E}^{\neq}(x)$, one deduces from (3.4c) that

$$(B - A)_{\mathcal{A}_0,:}\, d > 0 \qquad \text{and} \qquad (B - A)_{\mathcal{B}_0,:}\, d < 0.$$

Recalling the definitions of V in (3.1) and of $\mathcal{S}$ in (3.2), we see that, to conclude the proof of the membership $\sigma(J) \in \mathcal{S}$, it suffices to show that $[\sigma(J)]_{\mathcal{A}_0} = +1$ and $[\sigma(J)]_{\mathcal{B}_0} = -1$ or, equivalently, by the definition of σ, ($J_{i,:} = A_{i,:}$ for $i \in \mathcal{A}_0$) and ($J_{i,:} = B_{i,:}$ for $i \in \mathcal{B}_0$). This is indeed the case, since by (3.4c), for all $k \in \mathcal{K}$, one has ($H_i'(x_k) = A_{i,:}$ for $i \in \mathcal{A}_0$) and ($H_i'(x_k) = B_{i,:}$ for $i \in \mathcal{B}_0$); now, use the convergence (3.4b) to conclude.

[$\supseteq$ or point 2] Let $s \in \mathcal{S}$. We have to find a $J \in \partial_B H(x)$ such that $\sigma(J) = s$, that is, which satisfies

$$(J_{i,:} = A_{i,:} \text{ if } s_i = +1) \quad \text{and} \quad (J_{i,:} = B_{i,:} \text{ if } s_i = -1). \tag{3.4d}$$

Since $s \in \mathcal{S}$, there is a $d \in \mathbb{R}^n$ such that

$$s \bullet V^\mathsf{T} d > 0. \tag{3.4e}$$

Take a real sequence $\{t_k\} \downarrow 0$ and define the sequence $\{x_k\} \subseteq \mathbb{R}^n$ by

$$x_k := x + t_k d.$$

Then, $x_k \to x$. We claim that, for k sufficiently large, $x_k \in \mathcal{D}_H$ and $H'(x_k)$ is a constant matrix J satisfying (3.4d), which will conclude the proof. Let $i \in [1:m]$.

- If $i \in \mathcal{A}(x)$, $(Ax_k + a)_i < (Bx_k + b)_i$ for k large, so that $x_k \in \mathcal{D}_H$ and $H_i'(x_k) = A_{i,:}$.
- If $i \in \mathcal{B}(x)$, $(Ax_k + a)_i > (Bx_k + b)_i$ for k large, so that $x_k \in \mathcal{D}_H$ and $H_i'(x_k) = B_{i,:}$.
- If $i \in \mathcal{E}^{=}(x)$, then $A_{i,:} = B_{i,:}$, so that $x_k \in \mathcal{D}_H$ and $H_i'(x_k) = A_{i,:} = B_{i,:}$.

- If $i \in \mathcal{E}^{\neq}(x)$, subtract side by side $(Ax_k + a)_i = (Ax + a)_i + t_k A_{i,:} d$ and $(Bx_k + b)_i = (Bx + b)_i + t_k B_{i,:} d$, use $(Ax + a)_i = (Bx + b)_i$ and next (3.4e) to get

$$(Bx_k + b)_i - (Ax_k + b)_i = t_k(B_{i,:} - A_{i,:})d = t_k V_{i,:}^{\mathsf{T}} d \begin{cases} > 0 \text{ if } s_i = +1, \\ < 0 \text{ if } s_i = -1. \end{cases}$$

Hence, $x_k \in \mathcal{D}_H$, ($H_i'(x_k) = A_{i,:}$ if $s_i = +1$) and ($H_i'(x_k) = B_{i,:}$ if $s_i = -1$). □

Equivalence 3.4 (B-differential ↔ signed feasibility of strict inequality systems) The equivalence between the original problem 3.1 and the signed feasibility of strict inequality system problem 3.2 is a consequence of the previous proposition with V given by (3.1), which shows the bijectivity of the map $\sigma : \partial_B H(x) \to \mathcal{S}$ defined by (3.3a). Therefore, knowing σ by its definition (3.3), determining $\partial_B H(x)$ or $\mathcal{S}$ are equivalent problems. □

3.2.2 Orthants encountered by the null space of a matrix

Recall the definition of $\mathcal{S}$ in (3.2), which is associated with some matrix $V \in \mathbb{R}^{n\times p}$, coming from (3.1), with nonzero columns. The equivalent form of problem 3.2 (hence of problem 3.1) introduced in this section is based on a bijection between the *complementary set* of $\mathcal{S}$ in $\{\pm1\}^p$, denoted $\mathcal{S}^c := \{\pm1\}^p \setminus \mathcal{S}$, and a collection $\mathcal{I}$ of subsets of $[1:p]$ (hence $\mathcal{I} \subseteq \mathfrak{P}([1:p])$), which refers to a collection of orthants of $\mathbb{R}^p$, those encountered by the null space of V. This equivalence will play a major part in the conception of the algorithms in section 5.2, in particular, but not only, in an algorithm describing the *complementary set* of $\partial_B H(x)$, which is interesting when $|\overline{\partial}_B H(x) \setminus \partial_B H(x)|$ is small. The concept of *stem vector*, defined in the second part of this section, has proven useful in this regard. The equivalence rests on a duality concept through Gordan's alternative.

Problem 3.5 (orthants encountered by the null space of a matrix) Let be given two positive integers n and $p \in \mathbb{N}^*$ and a matrix V in $\mathbb{R}^{n\times p}$ with nonzero columns. Associate with $I \subseteq [1:p]$ the following orthant of $\mathbb{R}^p$:

$$\mathcal{O}_I^p := \{y \in \mathbb{R}^p : y_I \geqslant 0,\ y_{I^c} \leqslant 0\},$$

where $I^c := [1:p] \setminus I$. It is requested to determine the set

$$\mathcal{I} := \{I \subseteq [1:p] : \mathcal{N}(V) \cap \mathcal{O}_I^p \neq \{0\}\}.$$ □

Note that, if $I \in \mathcal{I}$, then $I^c \in \mathcal{I}$ (because $y \in (\mathcal{N}(V) \cap \mathcal{O}_I^p) \setminus \{0\}$ implies that $-y \in (\mathcal{N}(V) \cap \mathcal{O}_{I^c}^p) \setminus \{0\}$), so that $|\mathcal{I}|$ is even (just like $|\mathcal{S}|$ and $|\mathcal{S}^c|$, see proposition 4.1).

The equivalence between problems 3.2 and 3.5 is obtained thanks to the following bijection

$$\imath : s \in \{\pm1\}^p \to \imath(s) := \{i \in [1:p] : s_i = +1\} \in \mathfrak{P}([1:p]), \tag{3.5}$$

whose reverse map is $\imath^{-1} : I \in \mathfrak{P}([1:p]) \to s \in \{\pm1\}^p$, where $s_i = +1$ if $i \in I$ and $s_i = -1$ if $i \notin I$. As announced above, this equivalence relies on Gordan's theorem of the alternative [36; 1873]: for a matrix $A \in \mathbb{R}^{m\times n}$,

$$\exists x \in \mathbb{R}^n : Ax > 0 \qquad \Longleftrightarrow \qquad \nexists \alpha \in \mathbb{R}^m_+ \setminus \{0\} : A^{\mathsf{T}}\alpha = 0. \tag{3.6}$$

Proposition 3.6 (bijection $\mathcal{S}^c \leftrightarrow \mathcal{I}$) *The map $\imath$ defined by (3.5) is a bijection from $\mathcal{S}^c$ onto $\mathcal{I}$.*

Proof Let $s \in \{\pm 1\}^p$ and set $I := \imath(s) = \{i \in [1\,{:}\,p] : s_i = +1\}$. Define $A := \operatorname{Diag}(s) V^{\mathsf{T}}$ to make the link with Gordan's alternative (3.6). One has the equivalences

$$\begin{aligned} s \in \mathcal{S}^c \quad &\iff \quad \nexists\, x \in \mathbb{R}^n : Ax > 0 \qquad \text{[definition of } \mathcal{S} \text{ in (3.2)]} \\ &\iff \quad \exists\, \alpha \in \mathbb{R}^m_+ \setminus \{0\} : A^{\mathsf{T}} \alpha = 0 \qquad \text{[Gordan's alternative (3.6)]} \\ &\iff \quad \exists\, \alpha \in \mathbb{R}^m_+ \setminus \{0\} : s \bullet \alpha \in \mathcal{N}(V) \\ &\iff \quad \mathcal{N}(V) \cap \mathcal{O}^p_I \neq \{0\} \qquad \text{[see below]} \\ &\iff \quad I \in \mathcal{I} \qquad \text{[definition of } \mathcal{I}\text{]}. \end{aligned} \tag{3.7}$$

The implication "$\Rightarrow$" in (3.7) is due to the fact that $s \bullet \alpha$ is nonzero and belongs to both $\mathcal{N}(V)$ and $\mathcal{O}^p_I$. The reverse implication "$\Leftarrow$" in (3.7) is due to the fact that there is a nonzero $y \in \mathcal{N}(V) \cap \mathcal{O}^p_I$, implying that $\alpha := s \bullet y$ is nonzero and $\geqslant 0$ and is such that $s \bullet \alpha = y \in \mathcal{N}(V)$.

Since $\imath : \{\pm 1\}^p \to \mathfrak{P}([1\,{:}\,p])$ is a bijection, the above equivalences show that $\imath$ is also a bijection from $\mathcal{S}^c$ onto $\mathcal{I}$. □

Equivalence 3.7 ($\mathcal{S}^c \leftrightarrow \mathcal{I}$) The equivalence between problems 3.2 and 3.5 is a consequence of the bijectivity of $\imath : \mathcal{S}^c \to \mathcal{I}$, established in proposition 3.6: to determine $\mathcal{S}$, it suffices to determine $\mathcal{S}^c = \imath^{-1}(\mathcal{I})$, hence to determine $\mathcal{I}$. □

Recall that the *nullity* of a matrix A, denoted by $\operatorname{null}(A)$, is the dimension of its null space. Let us introduce the following collection of index sets (from now on, J usually denotes a set of indices rather than a Jacobian matrix):

$$\mathcal{C} := \{J \subseteq [1\,{:}\,p] : J \neq \varnothing,\ \ \operatorname{null}(V_{:,J}) = 1, V_{:,J_0} \text{ is injective if } J_0 \subsetneq J\}, \tag{3.8}$$

where "$\subsetneq$" is used to denote strict inclusion. In the terminology of the *vector matroid* formed by the columns of V and its subsets made of linearly independent columns [48; proposition 1.1.1], the elements of $\mathcal{C}$ are called the *circuits* of the matroid [48; proposition 1.3.5(*iii*)]. The particular expression (3.8) of the circuit set is interesting in the present context, since it readily yields the following implication:

$$J \in \mathcal{C} \quad \Longrightarrow \quad \text{any nonzero } \alpha \in \mathcal{N}(V_{:,J}) \text{ has none zero component.} \tag{3.9}$$

From (3.8) and (3.9), one can associate with $J \in \mathcal{C}$ a pair of sign vectors $\pm\tilde{s} \in \{\pm 1\}^J$ by $\tilde{s} := \operatorname{sgn}(\alpha)$ for some nonzero $\alpha \in \mathcal{N}(V_{:,J})$; the sign vectors $\pm\tilde{s}$ do not depend on the chosen $\alpha \in \mathcal{N}(V_{:,J}) \setminus \{0\}$ since $\operatorname{null}(V_{:,J}) = 1$. We call such a sign vector a *stem vector*, because of proposition 3.9 below, which shows that any $s \in \mathcal{S}^c$ can be generated from such a stem vector.

Definition 3.8 (stem vector) A *stem vector* is a sign vector $\tilde{s} = \operatorname{sgn}(\alpha)$, where $\alpha \in \mathcal{N}(V_{:,J})$ for some $J \in \mathcal{C}$. □

Note that there are twice as many stem vectors as circuits and that the stem vectors do not have all the same size.

Proposition 3.9 (generating $\mathcal{S}^c$ from the stem vectors) *For $s \in \{\pm 1\}^p$,*

$$s \in \mathcal{S}^c \quad \iff \quad s_J = \tilde{s} \text{ for some } J \subseteq [1\,{:}\,p] \text{ and some stem vector } \tilde{s}. \tag{3.10}$$

Proof [⇒] The index set $J \subseteq [1:p]$ in the right-hand side of (3.10) can be determined as the one satisfying the following two properties:

$$\{d \in \mathbb{R}^n : s_j v_j^\mathsf{T} d > 0 \text{ for all } j \in J\} = \varnothing, \tag{3.11a}$$

$$\forall\, J_0 \subsetneq J,\ \{d \in \mathbb{R}^n : s_j v_j^\mathsf{T} d > 0 \text{ for all } j \in J_0\} \neq \varnothing. \tag{3.11b}$$

To determine such a J, start with $J = [1:p]$, which verifies (3.11a), since $s \in \mathcal{S}^c$. Next, remove an index j from $[1:p]$ if (3.11a) holds for $J = [1:p] \setminus \{j\}$. Pursuing the elimination of indices j in this way, one arrives to an index set J satisfying (3.11a) and $\{d \in \mathbb{R}^n : s_j v_j^\mathsf{T} d > 0$ for all $j \in J \setminus \{j_0\}\} \neq \varnothing$ for all $j_0 \in J$. Then, (3.11b) clearly holds. We claim that, for the J thus defined, s_J is a stem vector, which will conclude the proof of the implication.

We first show that J is a matroid circuit, sticking to definition 3.8. By (3.11a), $J \neq \varnothing$. By Gordan's alternative (3.6), (3.11a) and (3.11b) read

$$\exists\, \alpha \in \mathbb{R}^J_+ \setminus \{0\} \text{ such that } \textstyle\sum_{j\in J} s_j v_j \alpha_j = 0, \tag{3.11c}$$

$$\forall\, J_0 \subsetneq J,\ \nexists\, \alpha' \in \mathbb{R}^{J_0}_+ \setminus \{0\} \text{ such that } \textstyle\sum_{j\in J_0} s_j v_j \alpha'_j = 0. \tag{3.11d}$$

From these properties, one deduces that $\alpha > 0$ and that $\operatorname{null}(V_{:,J}) \geqslant 1$. To show that $\operatorname{null}(V_{:,J}) = 1$, we proceed by contradiction. Suppose that there is a nonzero $\alpha'' \in \mathbb{R}^J$ that is not colinear with α and that verifies $\sum_{j\in J} s_j v_j \alpha''_j = 0$. One can assume that $t := \max\{\alpha''_j/\alpha_j : j \in J\}$ is > 0 (take $-\alpha''$ otherwise). Set $J_0 := \{j \in J : \alpha''_j/\alpha_j < t\}$. By the non-colinearity of α and α'', on the one hand, and the definition of t, on the other hand, one has $\varnothing \subsetneq J_0 \subsetneq J$. Furthermore, $\alpha' := \alpha - \alpha''/t \geqslant 0$, $\alpha'_j > 0$ for $j \in J_0$ and $\alpha'_j = 0$ for $j \in J \setminus J_0$. Since $\sum_{j\in J_0} s_j v_j \alpha'_j = 0$, we have a contradiction with (3.11d).

To show that $J \in \mathcal{C}$, we still have to show that $V_{:,J_0}$ is injective when $J_0 \subsetneq J$. Let $J_0 \subsetneq J$ and suppose that $\sum_{j\in J_0} s_j v_j \beta_j = 0$ for some $\beta \in \mathbb{R}^{J_0}$. We only have to show that $\beta = 0$. The cases when $\beta \geqslant 0$ or $\beta \leqslant 0$ are easy since then, (3.11d) readily implies that $\beta = 0$. Suppose now that β has positive (> 0) and negative (< 0) components. Set $t := \min\{-\alpha_j/\beta_j : j \in J_0,\ \beta_j < 0\} > 0$. Then, $\alpha' := \alpha_{J_0} + t\beta \geqslant 0$ and $\sum_{j\in J_0} s_j v_j \alpha'_j = 0$. By (3.11d), $\alpha' = 0$, implying that $\beta \leqslant 0$. Like previously, (3.11d) implies that $\beta = 0$.

Now, since J is a matroid circuit of V, since $s_J \bullet \alpha \in \mathcal{N}(V_{:,J}) \setminus \{0\}$ by (3.11c) and since $s_J = \operatorname{sgn}(s_J \bullet \alpha)$, s_J is a stem vector.

[⇐] Since $\tilde{s}$ is a stem vector with indices in $J \subseteq [1:p]$, $\tilde{s} := \operatorname{sgn}(\alpha)$ for some $\alpha \in \mathbb{R}^J$ with nonzero components that satisfies $V_{:,J}\alpha = 0$. Let $J_+ := \{j \in J : \alpha_j > 0\} = \{j \in J : s_j = +1\}$, which may be empty. Since one has $\alpha \in \mathcal{N}(V_{:,J}) \cap \mathcal{O}^J_{J_+}$, the bijection $\imath$ in (3.5), restricted to $\{\pm 1\}^J$, tells us that $\imath^{-1}(J_+) = \operatorname{sgn}(\alpha) = \tilde{s}$ is such that there is no $d \in \mathbb{R}^n$ such that $\tilde{s}_i v_i^\mathsf{T} d > 0$ or $s_i v_i^\mathsf{T} d > 0$ for $i \in J$, hence certainly not for $i \in [1:p]$, meaning that $s \in \mathcal{S}^c$. □

To determine the stem vectors, which are based on the matroid circuits defined by (3.8), one has to select subsets of columns of V forming a rank one matrix, whose strict subsets form injective matrices. Actually, this last condition can be simplified by the following property.

Proposition 3.10 (matroid circuit detection) *Suppose that $I \subseteq [1:p]$ is such that $\operatorname{null}(V_{:,I}) = 1$ and that $\alpha \in \mathcal{N}(V_{:,I}) \setminus \{0\}$. Then, $J := \{i \in I : \alpha_i \neq 0\}$ is a matroid circuit of V and the unique one included in I.*

Proof 1) Let us show that J is a matroid circuit. Since $\alpha \neq 0$, $J \neq \varnothing$.

Let us show that $\operatorname{null}(V_{:,J}) = 1$. Since $J \subseteq I$, one has $\operatorname{null}(V_{:,J}) \leqslant \operatorname{null}(V_{:,I}) = 1$. Furthermore, $\alpha_J \in \mathcal{N}(V_{:,J}) \setminus \{0\}$ implies that $\operatorname{null}(V_{:,J}) \geqslant 1$.

Now, let $J_0 \subsetneq J$ and suppose that $V_{:,J_0}\beta = 0$. We have to show that $\beta = 0$. Since $V_{:,J}(\beta, 0_{J\setminus J_0}) = 0$, it follows that $(\beta, 0_{J\setminus J_0}) \in \mathcal{N}(V_{:,J})$, which is of dimension 1, so that $(\beta, 0_{J\setminus J_0})$ is colinear to α. Since the components of α are $\neq 0$, we get that $\beta = 0$.

2) Let us now show that J is the unique matroid circuit of V included in I.

Let J' be a matroid circuit of V included in I. Then $\operatorname{null}(V_{:,J'}) = 1$ and there is a nonzero $\alpha' \in \mathcal{N}(V_{:,J'})$. By (3.9), α' has nonzero components. Furthermore, $(\alpha', 0_{I\setminus J'}) \in \mathcal{N}(V_{:,I})$, which has unit dimension and contains α. Therefore, α and $(\alpha', 0_{I\setminus J'})$ are colinear. Since the components of α are $\neq 0$, we get that $J' = J$. $\square$

3.3 Convex analysis problems

The formulation of the original problem 3.1 in the form of the convex analysis problems 3.11 and 3.14 below may be useful to highlight some properties of $\partial_B H(x)$, thanks to the tools of that discipline.

3.3.1 Pointed cones by vector inversions

Recall that a *convex cone* K of $\mathbb{R}^n$ is a convex set verifying $\mathbb{R}_{++}K \subseteq K$ (or, more explicitly, $tx \in K$ when $t > 0$ and $x \in K$). A *closed* convex cone K is said to be *pointed* if $K \cap (-K) = \{0\}$ [15; p. 54], which amounts to saying that K does not contain a line (i.e., an affine subspace of dimension one) or that K has no nonzero direction z such that $-z \in K$. For $P \subseteq \mathbb{R}^n$, we also denote by "cone P" the smallest *convex* cone containing P.

Problem 3.11 (pointed cones by vector inversions) Let be given two positive integers n and $p \in \mathbb{N}^*$ and p vectors $v_1, \ldots, v_p \in \mathbb{R}^n \setminus \{0\}$. It is requested to determine all the sign vectors $s \in \{\pm 1\}^p$ such that $\operatorname{cone}\{s_i v_i : i \in [1:p]\}$ is pointed. $\square$

The equivalence between the original problem 3.1 and this problem 3.11 is obtained thanks to the next proposition, which gives another property ("cone pointedness") that is equivalent to those in (3.6) and that is adapted to the present concern. For a proof, see [37; theorem 2.3.29] [28].

Proposition 3.12 (pointed polyhedral cone) *For a finite collection of nonzero vectors* $\{v_i : i \in [1:p]\} \subseteq \mathbb{R}^n$*, the following properties are equivalent:*

(*i*) $\operatorname{cone}\{v_i : i \in [1:p]\}$ *is pointed,*
(*ii*) $\nexists\, \alpha \in \mathbb{R}^p_+ \setminus \{0\} : \sum_{i\in[1:p]} \alpha_i v_i = 0$,
(*iii*) $\exists\, d \in \mathbb{R}^n,\ \forall i \in [1:p] : v_i^\mathsf{T} d > 0$.

Equivalence 3.13 (signed linear system feasibility $\leftrightarrow$ pointed cone by vector inversion) The equivalence $(i) \Leftrightarrow (iii)$ of the previous proposition shows that the set $\mathcal{S}$ defined by (3.2) is also given by

$$\mathcal{S} = \{s \in \{\pm 1\}^p :\ \operatorname{cone}\{s_i v_i : i \in [1:p]\} \text{ is pointed}\}. \tag{3.12}$$

To put it in words, denoting by $v_1, \ldots, v_p$ the columns of the matrix V defined by (3.1), the original problem of section 3.1 is equivalent to problem 3.11. $\square$

3.3.2 Linearly separable bipartitions of a finite set

This section extends section 3.3.1 and adopts its concepts and notation. The point of view presented in this section was also shortly considered by Zaslavsky [67; 1975, § 6A].

Problem 3.14 (linearly separable bipartitioning) Let be given an affine space $\mathbb{A}$ and $p \in \mathbb{N}^*$ vectors $\bar{v}_1, \ldots, \bar{v}_p \in \mathbb{A}$. Let $\mathbb{A}_0 := \mathbb{A} - \mathbb{A}$ be the vector space parallel to $\mathbb{A}$, endowed with a scalar product $\langle \cdot, \cdot \rangle$. It is requested to find all the ordered bipartitions (i.e., the partitions made of two subsets) (I, J) of $[1:p]$ for which there exists a vector $\xi \in \mathbb{A}_0$ (also called *separating covector* below) such that

$$\forall\, i \in I,\ \forall\, j \in J: \quad \langle \xi, \bar{v}_i \rangle < \langle \xi, \bar{v}_j \rangle. \qquad \square$$

Of course, if (I, J) is an appropriate ordered bipartition to which a separating covector ξ corresponds, then (J, I) is also an appropriate ordered bipartition with separating covector $-\xi$. Therefore, only half of the appropriate ordered bipartitions (I, J) must be identified, a fact that is related to the symmetry of $\partial_B H(x)$ (proposition 4.1). Figure 3.1 shows the solution to this problem by draw-

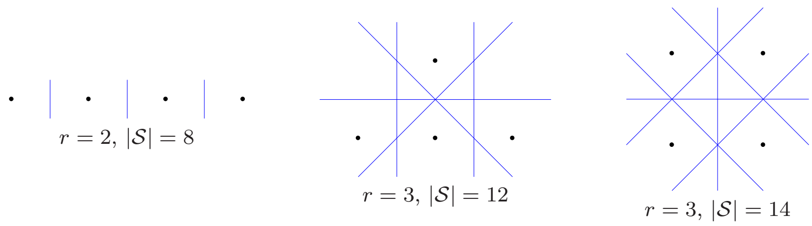

Fig. 3.1 Linearly separable bipartitions of a set of $p = 4$ points $\bar{v}_i$ in $\mathbb{R}^2$ (the dots in the figure). Possible separating hyperplanes are the drawn lines. We have not represented any separating line associated with the partition $(\varnothing, [1:p])$ or $([1:p], \varnothing)$, so that $|\mathcal{S}| = 2(n_s + 1)$, where n_s is the number of represented separating lines. We have set $r := \dim(\operatorname{vect}\{\bar{v}_1, \ldots, \bar{v}_p\}) + 1$.

ing the separating hyperplanes $\{\bar{v} \in \mathbb{A} : \xi^{\mathsf{T}} \bar{v} = t\}$ corresponding to some separating covector ξ and some $t \in \mathbb{R}$, for three examples with $n - 1 = 2$ and $p = 4$. Since it will be shown that $|\mathcal{S}|$ is the number of these searched linearly separable bipartitions, this one is denoted that way in the figure. Obviously, $|\mathcal{S}|$ not only depends on p and $r := \dim(\operatorname{vect}\{\bar{v}_1, \ldots, \bar{v}_p\}) + 1$, but it also depends on the arrangement of the $\bar{v}_i$'s in the affine space $\mathbb{A}$. We also see that $|\mathcal{S}|$ cannot take all the even values (proposition 4.1) between its lower bound $2p = 8$ and its upper bounds 8 (if $r = 2$) and 14 (if $r = 3$) given by propositions 4.7 and 4.10.

The equivalence between the linearly separable bipartitioning problem 3.14 of this section and the vector inversion problem 3.11 (hence, with the original problem 3.1) is grounded on the following construction and proposition.

Construction 3.15 1) Let be given two integers n and $p \in \mathbb{N}^*$ and p nonzero vectors $v_1, \ldots, v_p \in \mathbb{R}^n$ such that $K := \operatorname{cone}\{v_k : k \in [1:p]\}$ is a pointed cone. From proposition 3.12, there is a direction $d \in \mathbb{R}^n$ such that

$$\|d\| = 1 \qquad \text{and} \qquad \big(\forall\, k \in [1:p]: \quad v_k^{\mathsf{T}} d > 0\big).$$

Define

$$\mathbb{A} := \{\bar{v} \in \mathbb{R}^n : d^{\mathsf{T}} \bar{v} = 1\}, \qquad \mathbb{A}_0 := \mathbb{A} - \mathbb{A} = \{v \in \mathbb{R}^n : d^{\mathsf{T}} v = 0\},$$
$$\forall\, k \in [1:p]: \quad \bar{v}_k := v_k / (v_k^{\mathsf{T}} d) \in \mathbb{A}.$$

2) For a given bipartition (I, J) of $[1:p]$, define

$$K_I := \operatorname{cone}\{v_i : i \in I\} \qquad \text{and} \qquad K_J := \operatorname{cone}\{v_j : j \in J\}, \tag{3.13a}$$

$$C_I := K_I \cap \mathbb{A} \qquad \text{and} \qquad C_J := K_J \cap \mathbb{A}. \tag{3.13b}$$

□

Proposition 3.16 (pointed cone after vector inversions) *Let be given two integers n and $p \in \mathbb{N}^*$ and p nonzero vectors $v_1, \ldots, v_p \in \mathbb{R}^n$ such that $K := \operatorname{cone}\{v_k : k \in [1:p]\}$ is a pointed cone. Let (I, J) be a partition of $[1:p]$. Adopt the construction 3.15. Then, the following properties are equivalent:*

- (i) $\operatorname{cone}((-K_I) \cup K_J)$ *is pointed,*
- (ii) $K_I \cap K_J = \{0\}$,
- (iii) $C_I \cap C_J = \varnothing$,
- (iv) *there exists a vector* $\xi \in \mathbb{A}_0$ *such that* $\max_{i\in I} \xi^\mathsf{T} \bar{v}_i < \min_{j\in J} \xi^\mathsf{T} \bar{v}_j$.

Proof $[(i) \Rightarrow (ii)]$ We show the contrapositive. If there is $v \in (K_I \cap K_J) \setminus \{0\}$, then $-v \in (-K_I) \subseteq \operatorname{cone}((-K_I) \cup K_J)$ and $v \in K_J \subseteq \operatorname{cone}((-K_I) \cup K_J)$. Therefore, $\operatorname{cone}((-K_I) \cup K_J)$ is not pointed.

$[(ii) \Rightarrow (iii)]$ $\varnothing = \mathbb{A} \cap \{0\} = \mathbb{A} \cap K_I \cap K_J$ $[(ii)]$ $= (\mathbb{A} \cap K_I) \cap (\mathbb{A} \cap K_J) = C_I \cap C_J$.

$[(iii) \Rightarrow (iv)]$ We claim that

$$C_I \text{ is nonempty, convex and compact.}$$

Indeed, since C_I is nonempty (it contains the vectors $\bar{v}_i$ for $i \in I \neq \varnothing$), convex (because K_I and $\mathbb{A}$ are convex) and closed (because K_I and $\mathbb{A}$ are closed), it suffices to show that C_I is bounded or that its asymptotic cone (or recession cone in [57; p. 61]), namely $C_I^\infty = K_I \cap \mathbb{A}_0$, is reduced to $\{0\}$ [57; theorem 8.4]. This is indeed the case since $v^\mathsf{T} d > 0$ for all $v \in K_I \setminus \{0\} \subseteq K \setminus \{0\}$. For the same reason,

$$C_J \text{ is nonempty, convex and compact.}$$

Now, since $C_I \cap C_J = \varnothing$ by (iii), one can strictly separate the convex sets C_I and C_J in $\mathbb{A}$ [57; corollary 11.4.2]: there exists $\xi \in \mathbb{A}_0$ such that $\xi^\mathsf{T} v < \xi^\mathsf{T} w$, for all $v \in C_I$ and all $w \in C_J$. This shows that (iv) holds.

$[(iv) \Rightarrow (i)]$ Since $\operatorname{cone}((-K_I) \cup K_J) = \operatorname{cone}(\{-v_i : i \in I\} \cup \{v_j : j \in J\})$, by proposition 3.12, it suffices to find $d_{(I,J)} \in \mathbb{R}^n$ such that

$$\Big(-v_i^\mathsf{T} d_{(I,J)} > 0, \quad \forall\, i \in I\Big) \qquad \text{and} \qquad \Big(v_j^\mathsf{T} d_{(I,J)} > 0, \quad \forall\, j \in J\Big). \tag{3.14}$$

By (iv) and the fact that $\theta \in (0, \pi) \to \cot\theta \in \mathbb{R}$ is surjective, one can determine $\theta \in (0, \pi)$ such that

$$\max_{i\in I} \frac{\xi^\mathsf{T} v_i}{v_i^\mathsf{T} d} < -\cot\theta < \min_{j\in J} \frac{\xi^\mathsf{T} v_j}{v_j^\mathsf{T} d}. \tag{3.15}$$

Since $\sin\theta > 0$ for $\theta \in (0, \pi)$ and since $v_k^\mathsf{T} d > 0$ for all $k \in [1:p]$, this is equivalent to

$$\max_{i\in I} v_i^\mathsf{T} [(\cos\theta) d + (\sin\theta)\xi] < 0 < \min_{j\in J} v_j^\mathsf{T} [(\cos\theta) d + (\sin\theta)\xi].$$

Therefore, (3.14) is satisfied with $d_{(I,J)} := (\cos\theta) d + (\sin\theta)\xi$. □

One can now establish the link between the pointed cone problem of section 3.3.1 (problem 3.11) and the linearly separable bipartitioning problem (problem 3.14).

Equivalence 3.17 (pointed cone ↔ linearly separable bipartitioning) Let be given a matrix $V \in \mathbb{R}^{n\times p}$ with nonzero columns denoted by $v_1, \dots, v_p$ and take $s \in \mathcal{S}$, which is nonempty. By (3.12), $\operatorname{cone}\{s_i v_i : i \in [1:p]\}$ is pointed. Use the construction 3.15(1) with $v_i \curvearrowright s_i v_i$.

For $\tilde{s} \in \{\pm 1\}^p$, define a partition (I, J) of $[1:p]$ by

$$I := \{i \in [1:p] : \tilde{s}_i s_i = -1\} \quad \text{and} \quad J := \{i \in [1:p] : \tilde{s}_i s_i = +1\}.$$

Define also K_I and K_J by (3.13a) with $v_i \curvearrowright s_i v_i$. We claim that

$$\operatorname{cone}\{\tilde{s}_i v_i : i \in [1:p]\} \text{ is pointed} \quad \iff \quad \exists\, \xi \in \mathbb{A}_0 : \max_{i\in I} \xi^{\mathsf{T}} \bar{v}_i < \min_{j\in J} \xi^{\mathsf{T}} \bar{v}_j. \tag{3.16}$$

Indeed, one has

$$\begin{aligned}
&\operatorname{cone}\{\tilde{s}_i v_i : i \in [1:p]\} \text{ is pointed} \\
&\iff \operatorname{cone}\{\tilde{s}_i s_i (s_i v_i) : i \in [1:p]\} \text{ is pointed} \\
&\iff \operatorname{cone}((-K_I) \cup K_J) \text{ is pointed} \\
&\iff \exists\, \xi \in \mathbb{A}_0 : \max_{i\in I} \xi^{\mathsf{T}} \bar{v}_i < \min_{j\in J} \xi^{\mathsf{T}} \bar{v}_j,
\end{aligned}$$

where we have used the equivalence $(i) \Leftrightarrow (iv)$ of proposition 3.16 ($v_i \curvearrowright s_i v_i$).

The equivalence (3.16) establishes the expected equivalence between the pointed cone problem 3.11 (in which one looks for all the $\tilde{s} \in \{\pm 1\}^p$ such that $\operatorname{cone}\{\tilde{s}_i v_i : i \in [1:p]\}$ is pointed) and the linearly separable bipartitioning problem 3.14 of the vectors $\bar{v}_i = s_i v_i/(s_i v_i^{\mathsf{T}} d) = v_i/(v_i^{\mathsf{T}} d)$, $i \in [1:p]$, where d is associated with the pointed cone $\operatorname{cone}\{s_i v_i : i \in [1:p]\}$ by the equivalence $(i) \Leftrightarrow (iii)$ of proposition 3.12. □

3.4 Discrete geometry: hyperplane arrangements

The equivalent problem examined in this section has a long history, going back at least to the XIXth century [63,56]. More recently, it appears in *Computational Discrete Geometry* (the discipline has many other names), under the name of *hyperplane arrangements*. Contributions to this problem, or a more general version of it, with a discrete mathematics point of view, has been reviewed in [38, 32,62,2,40]. It has many applications [31,60,16].

Problem 3.18 (arrangement of hyperplanes containing the origin) Let be given two positive integers n and $p \in \mathbb{N}^*$ and p nonzero vectors $v_1, \dots, v_p \in \mathbb{R}^n$. Consider the hyperplanes containing the origin:

$$\mathcal{H}_i := \{d \in \mathbb{R}^n : v_i^{\mathsf{T}} d = 0\}. \tag{3.17}$$

It is requested to list the regions of $\mathbb{R}^n$ that are separated by these hyperplanes. Such a region is called a *cell* or a *chamber*, depending on the authors [5,59,2]. More specifically, let us define the half-spaces

$$\mathcal{H}_i^+ := \{d \in \mathbb{R}^n : v_i^{\mathsf{T}} d > 0\} \qquad \text{and} \qquad \mathcal{H}_i^- := \{d \in \mathbb{R}^n : v_i^{\mathsf{T}} d < 0\}.$$

The problem is to determine the following set of open sectors or cells of $\mathbb{R}^n$, indexed by the bipartitions (I_+, I_-) of $[1:p]$:

$$\mathfrak{C} := \left\{(I_+, I_-) \in \mathfrak{B}([1:p]) : (\cap_{i\in I_+} \mathcal{H}_i^+) \cap (\cap_{i\in I_-} \mathcal{H}_i^-) \neq \varnothing\right\}, \tag{3.18}$$

where $\mathfrak{B}([1:p])$ denotes the set of bipartitions of $[1:p]$. □

The link between problem 3.18 and the signed feasibility of strict linear inequality systems of section 3.2.1 is obtained from the bijection

$$\eta : (I_+, I_-) \in \mathfrak{B}([1:p]) \mapsto s \in \{\pm 1\}^p, \text{ where } s_i = \begin{cases} +1 \text{ if } i \in I_+, \\ -1 \text{ if } i \in I_- \end{cases} \tag{3.19}$$

and the setting $V = \begin{pmatrix} v_1 & \cdots & v_p \end{pmatrix}$, whose columns are nonzero by assumption here and in section 3.2.1. Recall the definition (3.2) of the set of sign vectors $\mathcal{S}$.

Proposition 3.19 (bijection $\mathfrak{C} \leftrightarrow \mathcal{S}$) *For the matrix $V \in \mathbb{R}^{n\times p}$, with nonzero columns v_i's, the map η given by (3.19) is a bijection from $\mathfrak{C}$ onto $\mathcal{S}$.*

Proof Let $(I_+, I_-) \in \mathfrak{B}([1:p])$ and $s := \eta((I_+, I_-))$. Then,

$$\begin{aligned}
(I_+, I_-) \in \mathfrak{C} \quad &\iff \quad \exists\, d \in (\cap_{i\in I_+} \mathcal{H}_i^+) \cap (\cap_{i\in I_-} \mathcal{H}_i^-) \\
&\iff \quad \exists\, d \in \mathbb{R}^n : (v_i^{\mathsf{T}} d > 0 \text{ for } i \in I_+) \text{ and } (v_i^{\mathsf{T}} d < 0 \text{ for } i \in I_-) \\
&\iff \quad \exists\, d \in \mathbb{R}^n : s \bullet V^{\mathsf{T}} d > 0 \\
&\iff \quad s \in \mathcal{S}.
\end{aligned}$$

These equivalences show the bijectivity of η from $\mathfrak{C}$ onto $\mathcal{S}$. □

Equivalence 3.20 (signed linear system feasibility ↔ hyperplane arrangement) The equivalence between problems 3.2 and 3.18 follows from the bijection of the map $\eta : \mathfrak{C} \to \mathcal{S}$ claimed in proposition 3.19. □

4 Description of the B-differential

This section gives some elements of description of the B-differential $\partial_B H(x)$, when H is the piecewise affine function given by (1.2) and $x \in \mathbb{R}^n$. This description is often carried out in terms of the matrix V defined by (3.1), whose p columns are denoted by $v_1, \ldots, v_p \in \mathbb{R}^n$ and are assumed to be nonzero. Some properties of $\partial_B H(x)$ are given in section 4.1, including those that are useful in [30]. Section 4.2 deals with the cardinal $|\partial_B H(x)|$ of the B-differential. Section 4.3 analyzes more precisely two particular configurations. Section 4.4 highlights two links between the B-differential and the C-differential of H.

4.1 Some properties of the B-differential

Let us start with a basic property of $\partial_B H(x)$, which is its symmetry in the sense of definitions 2.4. This property has been observed by many in other contexts [2; § 1.1.4], so that we leave its short proof, based on the equivalence 3.4, to [28].

Proposition 4.1 (symmetry of $\partial_B H(x)$) *Suppose that $p > 0$. Then, the B-differential $\partial_B H(x)$ is symmetric and $|\partial_B H(x)|$ is even.*

We now give a necessary and sufficient condition ensuring the completeness of $\partial_B H(x)$ in the sense of definition 2.3. The condition was shown to be sufficient in [17; corollary 2.1(i)] for the nonlinear case (1.5), using a different proof, but we shall see in [30] that it is an easy consequence of that property in the affine case (1.2). Thanks to the equivalence 3.4, the present proof is short.

Proposition 4.2 (completeness of the B-differential) *The B-differential of H at x, $\partial_B H(x)$, is complete if and only if the matrix $V \in \mathbb{R}^{n\times p}$ in (3.1) is injective. Hence, this property can hold only if $p \leqslant n$.*

Proof [$\Rightarrow$] We show the contrapositive. Assume that V is not injective, so that $V\alpha = 0$ for some nonzero $\alpha \in \mathbb{R}^p$. With $s \in \operatorname{sgn}(\alpha)$, one can write

$$\sum_{i\in[1:p]} |\alpha_i| s_i v_i = 0.$$

By Gordan's alternative (3.6), it follows that there is no $d \in \mathbb{R}^n$ such that $s \bullet V^{\mathsf{T}} d > 0$. By (3.2), this implies that $s \notin \mathcal{S}$. According to the equivalence 3.4, $\sigma^{-1}(s) \notin \partial_B H(x)$, showing that the B-differential is not complete.

[$\Leftarrow$] Assume the injectivity of V. Let $s \in \{\pm 1\}^p$. Since V^{T} is surjective, the system $V^{\mathsf{T}} d = s$ is feasible for $d \in \mathbb{R}^n$. For this d, $s \bullet V^{\mathsf{T}} d = e$, so that $s \bullet V^{\mathsf{T}} d > 0$ is feasible for $d \in \mathbb{R}^n$, so that the selected s is in $\mathcal{S}$. We have shown that $\mathcal{S} = \{\pm 1\}^p$ or that $\partial_B H(x) = \sigma^{-1}(\{\pm 1\}^p)$ (σ^{-1} is defined by (3.3b)) is complete. □

We focus now on the connectivity of $\partial_B H(x)$, a notion that is more easily presented in $\{\pm 1\}^p$ but that can be transferred straightforwardly to $\partial_B H(x)$ by the bijection σ defined in (3.3).

Definition 4.3 (adjacency in $\{\pm 1\}^p$) Two sign vectors s^1 and $s^2 \in \{\pm 1\}^p$ are said to be *adjacent* if they differ by a single component (i.e., the vertices s^1 and s^2 of the cube $\operatorname{co}\{\pm 1\}^p$ can be joined by a single edge). □

Definitions 4.4 (connectivity in $\{\pm 1\}^p$) A *path of length l in a subset S of* $\{\pm 1\}^p$ is a finite set of sign vectors $s^0, \ldots, s^l \in S$ such that s^i and s^{i+1} are adjacent for all $i \in [0 : l-1]$; in which case the path is said to be *joining* s^0 to s^l. One says that a subset S of $\{\pm 1\}^p$ is *connected* if any pair of points of S can be joined by a path in S. □

Proposition 4.5 (connectivity of the B-differential) *The set $\mathcal{S}$ defined by (3.2) is connected if and only if V has no colinear columns. In this case, any points s and $\tilde{s}$ of $\mathcal{S}$ can be joined by a path of length $l := \sum_{i\in[1:p]} |\tilde{s}_i - s_i|/2 \leqslant p$ in $\mathcal{S}$.*

Proof [$\Rightarrow$] We prove the contrapositive. Suppose that the columns v_i and v_j of V are colinear: $v_j = \alpha v_i$, for some $\alpha \in \mathbb{R}^*$. Assume that $\alpha > 0$ (resp. $\alpha < 0$). By (3.2), for any $s \in \mathcal{S}$, one can find $d \in \mathbb{R}^n$ such that $s \bullet V^{\mathsf{T}} d > 0$, implying that $s_i = s_j$ (resp. $s_i = -s_j$). Therefore, one cannot find a path in $\mathcal{S}$ joining $s \in \mathcal{S}$ and $-s \in \mathcal{S}$ (proposition 4.1), since one would have to change the two components with index in $\{i, j\}$ and that these components must be changed simultaneously for the sign vectors in $\mathcal{S}$, while the adjacency property along a path prevents from changing more than one sign at a time.

[$\Leftarrow$] We leave to [28] the proof of this implication and of the last claim of the proposition, since the conclusion of the implication is given in [2; section 1.10.4] as a simple observation with a very different point of view, related to graph theory. □

For $k \in [1 : p]$, we introduce

$$\mathcal{S}_k := \{s \in \{\pm 1\}^k : \ \exists\, d \in \mathbb{R}^n \text{ such that } s_i v_i^{\mathsf{T}} d > 0 \text{ for } i \in [1 : k]\}. \tag{4.3}$$

We also note $\mathcal{S}_k^c := \{\pm 1\}^k \setminus \mathcal{S}_k$. Hence $\mathcal{S} = \mathcal{S}_p$ and $\mathcal{S}^c = \mathcal{S}_p^c$. Point 1 of the next proposition will be used to motivate an improvement of algorithm 5.5 in section 5.2.4 and its points 2 and 3 will be used to get the equivalence in proposition 4.13, related to a fan arrangement.

Proposition 4.6 (incrementation)

1) *If* $s \in \mathcal{S}_k^c$*, then* $(s, \pm 1) \in \mathcal{S}_{k+1}^c$*. In particular,* $|\mathcal{S}_{k+1}^c| \geqslant 2|\mathcal{S}_k^c|$*.*
2) *If* $v_{k+1} \notin \operatorname{vect}\{v_1, \ldots, v_k\}$*, then,* $(s, \pm 1) \in \mathcal{S}_{k+1}$ *for all* $s \in \mathcal{S}_k$*. In particular,* $|\mathcal{S}_{k+1}| = 2|\mathcal{S}_k|$ *and* $|\mathcal{S}_{k+1}^c| = 2|\mathcal{S}_k^c|$*.*
3) *If* v_{k+1} *is not colinear to any of the vectors* $v_1, \ldots, v_k$*, then,* $(s, \pm 1)$ *and* $(-s, \pm 1) \in \mathcal{S}_{k+1}$ *for one* $s \in \mathcal{S}_k$ *and* $(s', +1)$ *or* $(s', -1) \in \mathcal{S}_{k+1}$ *for any* $s' \in \mathcal{S}_k$*. In particular,* $|\mathcal{S}_{k+1}| \geqslant |\mathcal{S}_k| + 2$*.*

Proof 1) If $s \in \mathcal{S}_k^c$, there is no $d \in \mathbb{R}^n$ such that $s_i v_i^{\mathsf{T}} d > 0$ for $i \in [1:k]$. Therefore, there is no $d \in \mathbb{R}^n$ such that ($s_i v_i^{\mathsf{T}} d > 0$ for $i \in [1:k]$) and $\pm v_{k+1}^{\mathsf{T}} d > 0$. Therefore, $(s, \pm 1) \in \mathcal{S}_{k+1}^c$. This implies that $|\mathcal{S}_{k+1}^c| \geqslant 2|\mathcal{S}_k^c|$.

2) Let P be the orthogonal projector on $\operatorname{vect}\{v_1, \ldots, v_k\}^{\perp}$ for the Euclidean scalar product. By assumption, $\mathrm{P}\, v_{k+1} \neq 0$. Let $s \in \mathcal{S}_k$, so that there is a direction $d \in \mathbb{R}^n$ such that $s_i v_i^{\mathsf{T}} d > 0$ for $i \in [1:k]$. For any $t \in \mathbb{R}$ and $i \in [1:k]$, the directions $d_{\pm} := d \pm t\, \mathrm{P}\, v_{k+1}$ verify $s_i v_i^{\mathsf{T}} d_{\pm} = s_i v_i^{\mathsf{T}} d > 0$ (because $v_i^{\mathsf{T}}\, \mathrm{P}\, v_{k+1} = 0$). In addition, for $t > 0$ sufficiently large, one has $\pm v_{k+1}^{\mathsf{T}} d_{\pm} = \pm v_{k+1}^{\mathsf{T}} d + t \|\mathrm{P}\, v_{k+1}\|^2 > 0$ (because $\mathrm{P}^2 = \mathrm{P}$ and $\mathrm{P}^{\mathsf{T}} = \mathrm{P}$). We have shown that both $(s, +1)$ and $(s, -1)$ are in $\mathcal{S}_{k+1}$. Therefore, $|\mathcal{S}_{k+1}| \geqslant 2|\mathcal{S}_k|$.

Now, $|\mathcal{S}_k| + |\mathcal{S}_k^c| = 2^k$, $|\mathcal{S}_{k+1}| + |\mathcal{S}_{k+1}^c| = 2^{k+1}$ and $|\mathcal{S}_{k+1}^c| \geqslant 2|\mathcal{S}_k^c|$ by point 1. Therefore, one must have $|\mathcal{S}_{k+1}| = 2|\mathcal{S}_k|$ and $|\mathcal{S}_{k+1}^c| = 2|\mathcal{S}_k^c|$.

3) We claim that one can find a direction $d \in \mathbb{R}^n$ such that

$$\Big(\forall\, i \in [1:k]: \ v_i^{\mathsf{T}} d \neq 0\Big) \qquad \text{and} \qquad v_{k+1}^{\mathsf{T}} d = 0. \tag{4.4}$$

Let us show this by induction. One can find a direction d_1 such that $v_1^{\mathsf{T}} d_1 \neq 0$ and $v_{k+1}^{\mathsf{T}} d_1 = 0$ (otherwise $\mathcal{N}(v_1^{\mathsf{T}}) \supseteq \mathcal{N}(v_{k+1}^{\mathsf{T}})$ or $\mathcal{R}(v_1) \subseteq \mathcal{R}(v_{k+1})$, which would imply that v_{k+1} and v_1 are colinear). Suppose now that, for some $j \in [1:k-1]$, one can find a direction $d_j \in \mathbb{R}^n$ such that $v_i^{\mathsf{T}} d_j \neq 0$ for $i \in [1:j]$ and $v_{k+1}^{\mathsf{T}} d_j = 0$. Like above, one can find a direction $p_j \in \mathbb{R}^n$ such that $v_{j+1}^{\mathsf{T}} p_j \neq 0$ and $v_{k+1}^{\mathsf{T}} p_j = 0$ (because v_{k+1} and v_{j+1} are not colinear). Then, for $\varepsilon > 0$ sufficiently small, $d_{j+1} := d_j + \varepsilon p_j$ satisfies $v_i^{\mathsf{T}} d_{j+1} \neq 0$ for $i \in [1:j+1]$ and $v_{k+1}^{\mathsf{T}} d_{j+1} = 0$.

Taking $s_i := \operatorname{sgn}(v_i^{\mathsf{T}} d)$ for $i \in [1:k]$, one deduces from (4.4) that there is a direction $d \in \mathbb{R}^n$ such that

$$\Big(\forall\, i \in [1:k]: \ s_i v_i^{\mathsf{T}} d > 0\Big) \qquad \text{and} \qquad v_{k+1}^{\mathsf{T}} d = 0.$$

It follows that, for $\varepsilon > 0$ sufficiently small, the directions $d_{\pm} := d \pm \varepsilon v_{k+1}$ satisfy

$$\Big(\forall\, i \in [1:k]: \ s_i v_i^{\mathsf{T}} d_{\pm} > 0\Big) \qquad \text{and} \qquad \pm\, v_{k+1}^{\mathsf{T}} d_{\pm} > 0.$$

This means that $(s, \pm 1) \in \mathcal{S}_{k+1}$. By symmetry (proposition 4.1), one also has $(-s, \pm 1) \in \mathcal{S}_{k+1}$, so that we have found 4 vectors in $\mathcal{S}_{k+1}$. Now, since, for any $s' \in \mathcal{S}_k \setminus \{\pm s\}$ (in number $|\mathcal{S}_k| - 2$), either $(s', +1) \in \mathcal{S}_{k+1}$ or $(s', -1) \in \mathcal{S}_{k+1}$, it follows that $|\mathcal{S}_{k+1}| \geqslant 4 + (|\mathcal{S}_k| - 2) = |\mathcal{S}_k| + 2$. □

4.2 Cardinal of the B-differential

4.2.1 Winder's formula

Giving the exact number of elements in $\partial_B H(x)$, that is $|\partial_B H(x)| = |\mathcal{S}| = |\mathfrak{C}| = 2^p - |\mathcal{S}^c| = 2^p - |\mathcal{I}|$, with the notation (3.2), (3.18) and (3.5), is a tricky task, even in the present affine case, since it subtly depends on the arrangement of the vectors v_i's in the space (see figure 3.1). Many contributions have been done on this subject; the earliest we cite dates from 1826 [63, 56, 38, 67, 65, 3, 4, 32, 20, 62, 2]. The formula (4.5) for $|\partial_B H(x)|$ is due to Winder [66; 1966] and reads for the matrix V with nonzero columns given by (3.1)

$$|\partial_B H(x)| = \sum_{I \subseteq [1:p]} (-1)^{\text{null}(V_{:,I})}, \tag{4.5}$$

where $\text{null}(V_{:,I})$ is the nullity of $V_{:,I}$ and the term in the right-hand side corresponding to $I = \varnothing$ is 1 (one takes the convention that $\text{null}(V_{:,\varnothing}) = 0$). Note that, in this formula, the columns of V can be colinear with each other. This amazing expression, with its only algebraic nature, potentially made of positive and negative terms, is explicit but, to our knowledge, has not been at the origin of a method to list the elements of $\partial_B H(x)$. We give in [28] a proof of (4.5) that follows the same line of reasoning as the one of Winder [66], but that is more analytic in that it uses the sign vectors introduced in section 3.2.1 rather than geometric arguments.

4.2.2 Bounds

When p is large, computing the cardinal $|\partial_B H(x)|$ from (4.5) by evaluating the 2^p ranks $\text{rank}(V_{:,I})$ for $I \subseteq [1:p]$ could be excessively expensive. Therefore, having simple-to-compute lower and upper bounds on $|\partial_B H(x)|$ may be useful in some circumstances, including theoretical ones. Proposition 4.7 gives elementary lower and upper bounds, while proposition 4.10 reinforces the upper bound, thanks to a lower semicontinuity argument (proposition 4.8). Necessary and sufficient conditions ensuring equality in the left-hand side or right-hand side inequalities in the next proposition are given in section 4.3.

Proposition 4.7 (lower and upper bounds on $|\partial_B H(x)|$) *For V given by* (3.1) *and* $r := \text{rank}(V)$, *one has* $\max(2p, 2^r) \leqslant 2^r + 2(p - r) \leqslant |\partial_B H(x)| \leqslant 2^p$.

Proof The first inequality is clear since $p \geqslant r \geqslant 1$ and $2r \leqslant 2^r$.

Consider the second inequality. One can assume that the first r columns of V are linearly independent, so that $|\mathcal{S}_r| = 2^r$ (notation (4.3) and proposition 4.6(2)). Next, by proposition 4.6(3), $|\mathcal{S}_{r+1}| \geqslant 2^r + 2$. By induction, the given lower bound holds for $|\mathcal{S}_p| = |\mathcal{S}| = |\partial_B H(x)|$.

The upper bound was already mentioned in proposition 2.2. □

Recall that a function $\varphi : x \in \mathbb{M} \to \varphi(x) \in \mathbb{R}$, defined on a metric space $\mathbb{M}$, is said to be *lower semicontinuous* if, for any $x \in \mathbb{M}$ and any sequence $\{x_k\}$ converging to x, one has $\varphi(x) \leqslant \liminf_{k\to\infty} \varphi(x_k)$. It is known that the rank of a matrix can only increase in the neighborhood of a given matrix, which implies its lower semicontinuity. The next lemma shows that the same property holds for $|\mathcal{S}| \in \mathbb{N}^*$, viewed as a function of V. Recall that the bijection σ is defined by (3.3).

Proposition 4.8 (lower semicontinuity of $|\partial_B H(x)|$) *Suppose that the set $\mathcal{S} = \sigma(\partial_B H(x))$ is viewed as a function of $V \in \mathbb{R}^{n\times p}$ given by* (3.1). *Then, $\mathcal{S}(V) \subseteq \mathcal{S}(\tilde{V})$ for $\tilde{V}$ near V in $\mathbb{R}^{n\times p}$. In particular, $|\partial_B H(x)| \in \mathbb{N}^*$ is a lower semicontinuous function of $V \in \mathbb{R}^{n\times p}$.*

Proof Suppose that $s \in \mathcal{S}(V)$. Then, by the definition (3.2) of $\mathcal{S}$, $s \bullet V^{\mathsf{T}} d > 0$ is feasible for $d \in \mathbb{R}^n$. Clearly, it follows that, for $\tilde{V}$ near V, $s \bullet \tilde{V}^{\mathsf{T}} d > 0$ is also feasible for $d \in \mathbb{R}^n$. Since $\mathcal{S}$ is finite, for any $\tilde{V}$ near V and any $s \in \mathcal{S}$, $s \bullet \tilde{V}^{\mathsf{T}} d > 0$ is also feasible for $d \in \mathbb{R}^n$. We have shown that $\mathcal{S}(V) \subseteq \mathcal{S}(\tilde{V})$ for $\tilde{V}$ near V.

As a direct consequence of this inclusion, we have that $|\mathcal{S}(V)| \leqslant |\mathcal{S}(\tilde{V})|$ for $\tilde{V}$ near V. The lower semicontinuity of $V \mapsto |\partial_B H(x)| = |\mathcal{S}|$ follows. □

Proposition 4.2 established a necessary and sufficient condition to have completeness of $\partial_B H(x)$. Here follows a less restrictive assumption, called *general position*, which is equivalent to have equality in (4.8) below. In connexion with this assumption, it is worth noting that, for a matrix $V \in \mathbb{R}^{n\times p}$ of rank r, one has

$$\forall\, I \subseteq [1:p]: \quad \operatorname{rank}(V_{:,I}) \leqslant \min(|I|, r). \tag{4.6}$$

Definition 4.9 (general position) The vectors $v_1, \ldots, v_p \in \mathbb{R}^n$ are said to be in *general position*, if the matrix $V := \begin{pmatrix} v_1 & \cdots & v_p \end{pmatrix}$ verifies

$$\forall\, I \subseteq [1:p]: \quad \operatorname{rank}(V_{:,I}) = \min(|I|, r), \tag{4.7}$$

where $r := \operatorname{rank}(V)$. □

This notion is used by Winder [66] when $r = n$. Example of vectors in general position are those in the left-hand side and right-hand side panes in figure 3.1 (the points are the normalized vectors $\bar{v}_i$'s so that the v_i's are actually in $\mathbb{R}^3$); note that in the first case $2 = r < n = 3$. Those in the middle pane are not in general position. This is due to the fact that $r := \operatorname{rank}(V) = 3$ while for the 3 bottom vectors, with indices in I say, one has $\min(|I|, r) - \operatorname{rank}(V_{:,I}) = 3 - 2 \neq 0$.

Equality in the upper estimate (4.8) of the next proposition was shown by Winder [66; 1966, corollary] when the columns of V are in general position and $r = n$, thanks to the identity (4.5). Long before him, the Swiss mathematician Ludwig Schläfli [58; p. 211] established the identity under the same assumptions, before 1852 [58; p. 174], without reference to (4.5), which was probably not known at that time. Note that equality does not hold in (4.8) for the middle configuration in figure 3.1 since $|\partial_B H(x)| = 12$, while the right-hand side of (4.8) reads $2[\binom{3}{0} + \binom{3}{1} + \binom{3}{2}] = 14$ (we have seen that the vectors in this pane are not in general position).

Proposition 4.10 (upper bound on $|\partial_B H(x)|$) *For V given by (3.1) and $r := \operatorname{rank}(V)$, one has*

$$|\partial_B H(x)| \leqslant 2 \sum_{i\in[0\,:\,r-1]} \binom{p-1}{i}, \tag{4.8}$$

with equality if and only if (4.7) holds.

Proof 1) The proof of the implication "(4.7) ⇒ (4.8) with equality" is established in [66; corollary], using the identity (4.5). See also [28].

2) Let us now show that (4.8) holds. Below, we systematically identify $\partial_B H(x)$ and $\mathcal{S}$, thanks to the equivalence 3.4. We also note $\mathcal{S} \equiv \mathcal{S}(V)$ to stress the dependence of $\mathcal{S}$ on V. Let β be the right-hand side of (4.8). We proceed by contradiction, assuming that there is a matrix $V \in \mathbb{R}^{n\times p}$ such that

$$|\mathcal{S}(V)| > \beta. \tag{4.9a}$$

It certainly suffices to show that one can find a sequence $\{V_k\} \subseteq \mathbb{R}^{n\times p}$ converging to V that satisfies

$$|\mathcal{S}(V_k)| = \beta, \tag{4.9b}$$

since then one would have the expected contradiction with the lower semicontinuity of $V \mapsto |\mathcal{S}(V)|$ ensured by proposition 4.8:

$$\liminf_{k\to\infty} |\mathcal{S}(V_k)| = \beta < |\mathcal{S}(V)|.$$

To find V_k arbitrarily close to V verifying (4.9b), we proceed as follows. Since (4.9a) holds, the first part of the proof implies that V does not satify (4.7). Our goal is to construct from V a matrix V_k

arbitrarily close to V with columns in general position and rank not exceeding $r = \operatorname{rank}(V)$, hence satisfying (4.9b) by the first part of the proof. To get $\operatorname{rank}(V_k) \leqslant r$, we arrange for $\mathcal{R}(V_k) \subseteq \mathcal{R}(V)$.

In view of (4.6), since V does not satify (4.7), there is some $I \subseteq [1:p]$ such that $\operatorname{rank}(V_{:,I}) < \min(|I|, r)$. We consider two complementary cases.

- If $|I| < r$, then, for an arbitrary small perturbation of the vectors $v_i \curvearrowright \tilde{v}_i$, with $i \in I$, one can get the $\tilde{v}_i$'s linearly independent in $\mathcal{R}(V)$. If one takes $\tilde{v}_i = v_i$ for $i \notin I$, the matrix $\tilde{V}$ formed of the vectors $\tilde{v}_i$'s verifies $\operatorname{rank}(\tilde{V}_{:,I}) = |I| = \min(|I|, r)$.
- If $|I| \geqslant r$, then, for an arbitrary small perturbation of the vectors $v_i \curvearrowright \tilde{v}_i$, with $i \in I$, one can get the $\tilde{v}_i$'s generate $\mathcal{R}(V)$, which is of dimension r. If one takes $\tilde{v}_i = v_i$ for $i \notin I$, the matrix $\tilde{V}$ formed of the vectors $\tilde{v}_i$'s verifies $\operatorname{rank}(\tilde{V}_{:,I}) = r = \min(|I|, r)$.

The perturbation of $V_{:,I}$ into $\tilde{V}_{:,I}$ also perturbs $V_{:,I'}$ for other index sets $I' \subseteq [1:p]$. However, one has $\operatorname{rank}(\tilde{V}_{:,I'}) \leqslant \min(|I'|, r)$ by (4.6) and $\mathcal{R}(\tilde{V}) \subseteq \mathcal{R}(V)$. Now, by the property of the rank, which can only increase in a neighborhood of a given matrix, if the perturbation taken above is sufficiently small, one has $\operatorname{rank}(V_{:,I'}) \leqslant \operatorname{rank}(\tilde{V}_{:,I'}) \leqslant \min(|I'|, r)$ for any $I' \subseteq [1:p]$. Therefore, $\operatorname{rank}(V_{:,I'}) = \min(|I'|, r)$ implies that $\operatorname{rank}(\tilde{V}_{:,I'}) = \min(|I'|, r)$. As a result, the modification of V to get $\tilde{V}$ described above increases by at least one the number of intervals $I' \subseteq [1:p]$ such that $\operatorname{rank}(\tilde{V}_{:,I'}) = \min(|I'|, r)$. Since the number of such intervals is finite, proceeding similarly with all the nonempty index sets $I'' \subseteq [1:p]$ such that $\operatorname{rank}(\tilde{V}_{:,I''}) < \min(|I''|, r)$, one finally obtains a matrix V_k, arbitrary close to V, such that (4.7) holds: $\operatorname{rank}((V_k)_{:,I}) = \min(|I|, r)$ for all $I \subseteq [1:p]$. By taking smaller and smaller perturbations of V, one also has $V_k \to V$.

3) One still has to show that "(4.8) with equality $\Rightarrow$ (4.7)". We proceed by contradiction, assuming that (4.8) holds with equality for $\partial_B H(x) = \sigma^{-1}(\mathcal{S})$ and V given by (3.1), but that (4.7) does not hold. By (4.6), there exists $I \subseteq [1:p]$ such that

$$\operatorname{rank}(V_{:,I}) < \min(|I|, r). \tag{4.9c}$$

Let β be the right-hand side of (4.8). It certainly suffices to show that, thanks to (4.9c), one can find a matrix $\tilde{V} \in \mathbb{R}^{n \times p}$ such that $\operatorname{rank}(\tilde{V}) \leqslant r$ and $|\mathcal{S}(\tilde{V})| > \beta$, since this would be in contradiction with what has been shown in part 2 of the proof. This matrix $\tilde{V}$ is obtained by perturbing V. By proposition 4.8, if the perturbation is sufficiently small, one has $\mathcal{S}(V) \subseteq \mathcal{S}(\tilde{V})$, so that it suffices to show that $\mathcal{S}(\tilde{V})$ contains a sign vector s that is not in $\mathcal{S}(V)$.

We claim that (4.9c) implies that one can find an index set $J \subseteq I$ such that

$$V_{:,J} \text{ is not injective} \qquad \text{and} \qquad |J| \leqslant r. \tag{4.9d}$$

Indeed, if $|I| \leqslant r$, one can take $J = I$ to satisfy (4.9d), since $\operatorname{rank}(V_{:,I}) < |I|$ by (4.9c), so that $V_{:,I}$ is not injective. If $|I| > r$, then $\operatorname{rank}(V_{:,I}) < r$ by (4.9c), which implies that any $J \subseteq I$ such that $|J| = r$ satisfies (4.9d).

Since $V_{:,J}$ is not injective, one can find $\alpha_J \in \mathbb{R}^J \setminus \{0\}$ such that

$$0 = \sum_{j \in J} \alpha_j v_j = \sum_{j \in J} \tilde{s}_j |\alpha_j| v_j,$$

for some $\tilde{s}_J \in \{\pm 1\}^J$ satisfying $\tilde{s}_j \in \operatorname{sgn}(\alpha_j)$ for all $j \in J$. Then, by Gordan's alternative (3.6),

$$\nexists\, d \in \mathbb{R}^n : \quad \tilde{s}_j v_j d > 0, \quad \text{for all } j \in J.$$

This implies that there is no $s \in \mathcal{S}(V)$ such that $s_J = \tilde{s}_J$. To conclude the proof, it suffices now to show that one can construct an arbitrary small perturbation $\tilde{V}$ of V, such that $\mathcal{R}(\tilde{V}) \subseteq \mathcal{R}(V)$ and with an $s \in \mathcal{S}(\tilde{V})$ satisfying $s_J = \tilde{s}_J$.

Let $J^c := [1:p] \setminus J$. By (4.9d), $|J| \leqslant r \leqslant n$ so that one can find vectors $\{\tilde{v}_j : j \in [1:p]\}$, such that $\tilde{v}_j = v_j$ for $j \in J^c$, the vectors $\{\tilde{v}_j : j \in J\}$ are linearly independent, $\tilde{v}_j - v_j$ is arbitrary small and $\{\tilde{v}_j : j \in [1:p]\} \subseteq \mathcal{R}(V)$. Since the vectors $\{\tilde{v}_j : j \in J\}$ are linearly independent, one can find a direction $d_0 \in \mathbb{R}^n$ such that $\tilde{v}_j^\mathsf{T} d_0 = \tilde{s}_j$ for $j \in J$, hence

$$\tilde{s}_j \tilde{v}_j^\mathsf{T} d_0 > 0, \quad \forall\, j \in J.$$

Let d be a discriminating vector given by lemma 2.5 (with an additional $v_0 = 0$, $v_i \curvearrowright \tilde{s}_i \tilde{v}_i$ and $\xi \curvearrowright d$) sufficiently close to d_0. It results that $\tilde{s}_j \tilde{v}_j^\mathsf{T} d > 0$ for $j \in J$ and $\tilde{s}_j \tilde{v}_j^\mathsf{T} d \neq 0$ for $j \in J^c$. Finally, we see that the sign vector $s \in \{\pm 1\}^p$ defined by $s_i = \operatorname{sgn}(\tilde{v}_i^\mathsf{T} d)$ for all $i \in [1:p]$ is in $\mathcal{S}(\tilde{V})$ and satisfies $s_J = \tilde{s}_J$, as desired. □

Corollary 4.11 (stability of the sign vector set) *The sign vector set $\mathcal{S} \subseteq \{\pm 1\}^p$ defined by (3.2) is unchanged by small variations of the matrix $V \in \mathbb{R}^{n \times p}$ preserving its rank, provided the columns $v_1, \ldots, v_p \in \mathbb{R}^n$ of V are in general position in the sense of definition 4.9.*

Proof If $\tilde{V}$ is near V, $\mathcal{S}(V) \subseteq \mathcal{S}(\tilde{V})$ by proposition 4.8. If the columns of V are in general position, proposition 4.10 tells us that $|\mathcal{S}(V)| = \beta$, where β is the right-hand side of Winder's bound (4.8) with $r = \operatorname{rank}(V)$. Now, by the fact that $\operatorname{rank}(\tilde{V}) = r$, proposition 4.10 ensures that $|\mathcal{S}(\tilde{V})| \leqslant \beta$. Therefore, one must have $\mathcal{S}(\tilde{V}) = \mathcal{S}(V)$. □

4.3 Particular configurations

We consider in this section some particular matrices V given by (3.1), which may be useful to get familiar with the B-differential of H. For these V's, $|\partial_B H(x)|$ can be computed easily. We consider two matrices V with the property that $r := \operatorname{rank}(V)$ takes the value 2 or p; they yield the lower and upper bounds on $|\partial_B H(x)|$ given by proposition 4.7. The lower bound $2p$ applies to the left-hand side pane of figure 3.1. As shown by the intermediate pane in figure 3.1, however, $|\partial_B H(x)|$ does not only depend on r.

Proposition 4.12 (injective matrix) *The matrix $V \in \mathbb{R}^{n \times p}$ given by (3.1) is injective if and only if $|\partial_B H(x)| = 2^p$.*

Proof Indeed, by proposition 4.2, the B-differential $\partial_B H(x)$ is complete (meaning that it is equal to $\overline{\partial}_B H(x)$, given by (2.2)) if and only if V is injective. Clearly, the completeness of $\partial_B H(x)$ is equivalent to $|\partial_B H(x)| = 2^p$. □

Proposition 4.13 (fan arrangement) *If $p \geqslant 2$ and the vectors v_i's are not two by two colinear, one has $\operatorname{rank}(V) = 2$ if and only if $|\partial_B H(x)| = 2p$.*

Proof [⇒] A short proof leverages Winder's bound (4.8) with equality. Since the v_i's are not two by two colinear, one has for any $I \subseteq [1:p]$:

$$\operatorname{rank}(V_{:,I}) = \begin{cases} |I| & \text{if } |I| \leqslant 2 \\ 2 & \text{if } |I| > 2. \end{cases}$$

Therefore (4.7) holds. By proposition 4.10, this implies that equality holds in (4.8), that is, with $r = 2$: $|\partial_B H(x)| = 2 \sum_{i \in [0:1]} \binom{p-1}{i} = 2p$.

[$\Leftarrow$] The $\operatorname{rank}(V) =: r$ cannot be 1, since $p \geqslant 2$ and the v_i's are not two by two colinear. We proceed by contradiction, assuming the $r > 2$. Then, one can find $k \in [2:p-1]$ such that $\dim \operatorname{vect}\{v_1, \ldots, v_k\} = 2$ and $\dim \operatorname{vect}\{v_1, \ldots, v_{k+1}\} = 3$. For any $k \in [1:p]$, denote by $\mathcal{S}_k$ the set defined by (4.3). By the first part of the proof, $|\mathcal{S}_k| = 2k$. Since $v_{k+1} \notin \{v_1, \ldots, v_k\}$, proposition 4.6(2) tells us that $|\mathcal{S}_{k+1}| = 4k$. Now, for $j \in [k+2:p]$, proposition 4.6(3) tells us that $|\mathcal{S}_j| \geqslant |\mathcal{S}_{j-1}| + 2$. As a result, we get $|\mathcal{S}_p| \geqslant 4k + 2(p-k-1) = 2p + 2(k-1) \geqslant 2p + 2$ (since $k \geqslant 2$), which contradicts the assumption $|\mathcal{S}_p| = 2p$. □

4.4 A glance at the C-differential

The section presents two links between the B-differential and the C-differential of the function H given by (1.2). The first proposition tells us that, whilst $\partial_C H(x)$ can be obtained from $\partial_B H(x)$ by taking its convex hull (it is its definition (1.1)), the latter can be obtained from the former by taking its extreme points. For a proof, see [28].

Proposition 4.14 (a link with the C-differential) $\partial_B H(x) = \operatorname{ext} \partial_C H(x)$.

The second proposition restates theorem 2.2 of Chen and Xiang [17; 2011], which applies to the more general nonlinear function (1.5). The interest of this restatement comes from its proof that is short, thanks to the use of the symmetry of the B-differential (proposition 4.1), and from the fact that proposition 4.15 can be used, straightforwardly, to recover Chen and Xiang's Jacobian, when H is given by (1.5); see [30]. Recall the notation (2.1) of the index sets.

Proposition 4.15 (a particular C-Jacobian) *One has $J \in \partial_C H(x)$ for the Jacobian whose ith row, $i \in [1:m]$, is defined by*

$$J_{i,:} = \begin{cases} A_{i,:} & \text{if } i \in \mathcal{A}(x), \\ \frac{1}{2}[A_{i,:} + B_{i,:}] & \text{if } i \in \mathcal{E}(x), \\ B_{i,:} & \text{if } i \in \mathcal{B}(x). \end{cases}$$

Proof Let $M \in \partial_B H(x)$, which is known to be nonempty. By proposition 2.2, $M_{i,:} = A_{i,:}$ for $i \in \mathcal{A}(x)$, $M_{i,:} = B_{i,:}$ for $i \in \mathcal{B}(x)$ and $M_{i,:} = A_{i,:} = B_{i,:}$ for $i \in \mathcal{E}^{=}(x)$. By the symmetry of $\partial_B H(x)$ (proposition 4.1), M' defined by $M'_{:,i} = M_{:,i}$ if $i \in \mathcal{A}(x) \cup \mathcal{E}^{=}(x) \cup \mathcal{B}(x)$ and by

$$M'_{i,:} = \begin{cases} B_{i,:} & \text{if } i \in \mathcal{E}^{\neq}(x) \text{ and } M_{i,:} = A_{i,:} \\ A_{i,:} & \text{if } i \in \mathcal{E}^{\neq}(x) \text{ and } M_{i,:} = B_{i,:} \end{cases}$$

is also in $\partial_B H(x)$. Therefore, $J = (M + M')/2$ is in $\operatorname{co} \partial_B H(x) = \partial_C H(x)$, by (1.1). This is the formula of J given in the statement of the proposition. □

Instead of taking $J_{1/2} := \frac{1}{2}(M + M')$ in the preceeding proof, one could also have taken $J_t := (1-t)M + tM'$, which is also in $\operatorname{co} \partial_B H(x) = \partial_C H(x)$ for any $t \in [0,1]$. The inconvenient of this latter choice, when $t \neq 1/2$, is that M is usually not known. In particular, it is not necessarily known whether $M_{i,:}$ may be $A_{i,:}$ or $B_{i,:}$, for a particular $i \in \mathcal{E}^{\neq}(x)$, while J_t depends on this value when $t \neq 1/2$. In contrast, $J_{1/2}$ has an explicit formula that does not require the knowledge of the value of $M_{i,:}$ for $i \in \mathcal{E}^{\neq}(x)$.

5 Computation of the B-differential

This section describes techniques to compute a single Jacobian (section 5.1) or all the Jacobians (section 5.2) of the B-differential $\partial_B H(x)$, in exact arithmetic, when H is the piecewise affine function given by (1.2). The piece of software ISF has been written to test the algorithms.

5.1 Computation of a single Jacobian

An interest of the problem equivalence highlighted in proposition 3.3(3) is to provide a method to find rapidly an element of $\partial_B H(x)$, which complements Qi's [52; 1993, final remarks (1)]. It is shown in [30], that this method extends to the computation of an element of the B-differential in the nonlinear case, i.e., when H is given by (1.5). The method is based on the following algorithm, which associates with p nonzero vectors $v_1, \ldots, v_p$, which may be identical or colinear, a direction d such that $v_i^\top d \neq 0$ for all $i \in [1:p]$; it is a variant of the technique used in the proof of [17; lemma 2.1]. When the v_i's are also distinct, the direction d can also be derived from lemma 2.5, by adding the vector $v_0 = 0$.

Algorithm 5.1 (computes $\boldsymbol{d} \in \mathbb{R}^{\boldsymbol{n}}$ such that $\boldsymbol{v_i^\top d \neq 0}$ for all $\boldsymbol{i}$)
Let be given p nonzero vectors $v_1, \ldots, v_p$ in $\mathbb{R}^n$ and take $d \in \mathbb{R}^n \setminus \{0\}$.
Repeat:

1. If $I := \{i \in [1:p] : v_i^\top d = 0\} = \varnothing$, exit.
2. Let $i \in I$.
3. Take $t > 0$ sufficiently small such that, for all $j \notin I$, $(v_j^\top d)(v_j^\top [d + tv_i]) > 0$.
4. Update $d := d + tv_i$.

Explanation. In step 3, any sufficiently small $t > 0$ is appropriate (the proof of [17; lemma 2.1] computes bounds explicitely), since $(v_j^\top d)(v_j^\top [d+tv_i])$ is positive for $t = 0$. The new direction d set in step 4 is such that $v_i^\top (d + tv_i) = t\|v_i\|^2 > 0$, so that this direction makes at least one more $v_j^\top d$ nonzero than the previous one. This implies that the algorithm finds an appropriate direction in at most p loops. □

The next procedure uses a direction d computed by algorithm 5.1 to obtain a single element of $\partial_B H(x)$. Recall that the map σ is defined by (3.3a) and is a bijection from $\partial_B H(x)$ onto $\mathcal{S}$, defined by (3.2) (proposition 3.3).

Algorithm 5.2 (computes a single Jacobian in $\boldsymbol{\partial_B H(x)}$)
Let H be given by (1.2), $x \in \mathbb{R}^n$ and suppose that $p \neq 0$.

1. Compute $V \in \mathbb{R}^{n \times p}$ by (3.1) and denote its columns by $v_1, \ldots, v_p \in \mathbb{R}^n$.
2. By algorithm 5.1, compute $d \in \mathbb{R}^n$ such that $v_i^\top d \neq 0$ for all $i \in [1:p]$.
3. Define $s \in \mathcal{S}$ by $s_i := \operatorname{sgn}(v_i^\top d)$, for $i \in [1:p]$.
4. Then, $\sigma^{-1}(s) \in \partial_B H(x)$.

Explanation. When $p = 0$, $\partial_B H(x) = \overline{\partial}_B H(x)$ contains a single Jacobian that is given by (2.2), which explains why algorithm 5.2 focuses on the case when $p > 0$. The sign vector s computed in step 3 is such that $s_i v_i^\top d > 0$ for all $i \in [1:p]$, so that it is indeed in $\mathcal{S}$ and, by proposition 3.3, $\sigma^{-1}(s)$ is a Jacobian in $\partial_B H(x)$. □

5.2 Computation of all the Jacobians

This section presents two algorithms, and some variants, for computing all the B-differential of H. They use the notion of $\mathcal{S}$-tree presented in section 5.2.1(A). The first algorithm is grounded on the notion of stem vector (section 3.2.2) and is described in section 5.2.2. The second algorithm is the culmination of a series of improvements brought to an algorithm by Rada and Černý [54; 2018] (section 5.2.1(B)) for computing the cells of a hyperplane arrangement, which is known to be an equivalent problem to the one of computing the B-differential of H when the hyperplanes contain zero (see section 3.4). The improvements are detailed in section 5.2.4 and the resulting algorithm is described in section 5.2.5. Finally, numerical experiments are presented in section 5.2.6 to compare the efficiency of the algorithms.

Algorithms for listing the elements of the finite set $\partial_B H(x)$ can be designed by looking at one of the various forms of the problem, those described in section 3 and others [5]; this is what we shall do. Nevertheless, the only algorithms we have found in the scientific literature take the point of view of hyperplane arrangements of section 3.4 and can usually be used for more general arrangements than those needed to describe $\partial_B H(x)$ (i.e., in which case the hyperplanes pass through zero). One can quote the contributions by Bieri and Nef [12; 1982], Edelsbrunner, O'Rourke and Seidel [31; 1986], Avis and Fukuda [5; 1996], improved Sleumer [59; 1998], and, more recently, Rada and Černý [54; 2018], which is described in section 5.2.1(B).

5.2.1 Incremental-recursive algorithms

The algorithms described in this section are incremental in the sense that the considered sign vectors have their length increased by one at each step. Furthermore, the algorithms explore the $\mathcal{S}$-tree described in subsection A below by recursive procedures, whose names are recognizable by their suffix "-REC". All the procedures end by returning to their calling program.

A. THE $\mathcal{S}$-TREE. A common feature of the algorithms considered in this paper is the construction of the $\mathcal{S}$-tree described below, incrementally and recursively. This idea was probably introduced by Rada and Černý [54; 2018].

The level k of the $\mathcal{S}$-tree is formed of a set of sign vectors denoted by

$$\mathcal{S}_k^1 := \{s \in \mathcal{S}_k : s_1 = +1\},$$

where $\mathcal{S}_k$ is the subset of $\{\pm 1\}^k$ defined by (4.3). In particular, the level 1 or root of the $\mathcal{S}$-tree contains the unique sign vector $+1 \in \{\pm 1\}^1$. There is indeed no reason to compute $\{s \in \mathcal{S} : s_1 = -1\}$ since this part of $\mathcal{S}$ is equal to $-\{s \in \mathcal{S} : s_1 = 1\}$ by the symmetry property of $\mathcal{S}$ (proposition 4.1). The $\mathcal{S}$-tree has p levels, where p is the number of vectors v_i, or columns of the given matrix $V \in \mathbb{R}^{n\times p}$. In order to avoid the memorization of the elements of $\mathcal{S}_k^1$, the $\mathcal{S}$-tree is constructed by a *depth-first search*, which can be schematized as follows.

Algorithm 5.3 (STREE) Let be given $V \in \mathbb{R}^{n\times p}$, with n and $p \in \mathbb{N}^*$, having nonzero columns.

1. Execute the recursive procedure STREE-REC$(V, +1)$.

Algorithm 5.4 (STREE-REC) Let be given $V \in \mathbb{R}^{n\times p}$, with n and $p \in \mathbb{N}^*$, having nonzero columns, and a sign vector $s \in \mathcal{S}_k^1$ for some $k \in [1:p]$.

1. If $k = p$, print s and return.

2. If $(s,+1) \in \mathcal{S}^1_{k+1}$, execute STREE-REC$(V,(s,+1))$.
3. If $(s,-1) \in \mathcal{S}^1_{k+1}$, execute STREE-REC$(V,(s,-1))$.

The method used to determine whether $(s,\pm 1)$ is in $\mathcal{S}^1_{k+1}$ depends on the specific algorithm and may or may not use a direction d intervening in (4.3). Note that, as emphasized in proposition 4.6(3), at least one of the sign vectors $(s,+1)$ and $(s,-1)$ belongs to $\mathcal{S}^1_{k+1}$ (maybe both). It is justified not to explore the $\mathcal{S}$-tree below an $(s,\pm 1)$ that is not in $\mathcal{S}^1_{k+1}$, since then $(s,\pm 1,s') \notin \mathcal{S}$ for any $s' \in \{\pm 1\}^{p-k-1}$. By construction, the algorithm STREE prints all the elements of $\mathcal{S}^1_p \equiv \mathcal{S}^1 := \{s \in \mathcal{S} : s_1 = +1\}$ in step 1 of the STREE-REC procedure.

B. Rada and Černý's algorithm. The algorithm proposed by Rada and Černý [54; 2018], which is referenced below as the RC algorithm, deals with the determination of the cells associated with a general hyperplane arrangement. We describe it below for an arrangement that results from the computation of the B-differential $\partial_B H(x)$, whose hyperplanes all contain zero (see section 3.4). We also use the linear algebra language of section 3.2.1, viewing the problem as the one of determining the set $\mathcal{S}$ of sign vectors $s \in \{\pm 1\}^p$ such that $s \bullet V^{\mathsf{T}} d > 0$ is feasible for $d \in \mathbb{R}^n$ (V is the matrix defined by (3.1)); in contrast, the language used in [54] is more geometric. The algorithm builds the $\mathcal{S}$-tree of the previous section A and, for each $s \in \mathcal{S}^1_k$, it solves a single LO problem to determine whether $(s,+1)$ or $(s,-1)$ is in $\mathcal{S}^1_{k+1}$.

The RC algorithm succeeds in solving only one LO problem to determine whether $(s,+1)$ and $(s,-1)$ are in $\mathcal{S}^1_{k+1}$, at the node $s \in \mathcal{S}^1_k$, thanks to the memorization of a direction d such that $s \bullet V_k^{\mathsf{T}} d > 0$ (we note $V_k := V_{:,[1:k]}$). Indeed, one has

$$\begin{array}{lll} v_{k+1}^{\mathsf{T}} d < 0 & \Longrightarrow & (s,-1) \in \mathcal{S}^1_{k+1}, \\ v_{k+1}^{\mathsf{T}} d > 0 & \Longrightarrow & (s,+1) \in \mathcal{S}^1_{k+1}, \end{array}$$

and one of these two cases takes place if we exclude the case where $v_{k+1}^{\mathsf{T}} d = 0$. In [54; Algorithm 1], the case where $v_{k+1}^{\mathsf{T}} d = 0$ is not dealt with completely since $(s,+1)$ is declared to belong to $\mathcal{S}^1_{k+1}$ in that case, while it is clear that $(s,-1)$ is also in $\mathcal{S}^1_{k+1}$. Indeed, in our implementation of the RC algorithm, we modify slightly d by adding a small positive or negative multiple of v_{k+1} to d when $v_{k+1}^{\mathsf{T}} d = 0$, so that both $(s,\pm 1)$ are accepted in $\mathcal{S}^1_{k+1}$ in that case. This choice may be at the origin of the differences that one observes in table 5.1 below between the statistics of the original RC algorithm in [54] and those of our implementation.

Next, when $(s,s_{k+1}) \in \{\pm 1\}^{k+1}$ is observed to belong to $\mathcal{S}^1_{k+1}$, the question of whether $(s,-s_{k+1})$ also belongs to $\mathcal{S}^1_{k+1}$ arises. In the RC algorithm, the answer to this question is obtained by solving a LO problem similar to

$$\begin{cases} \min_{(d,t)\in\mathbb{R}^n\times\mathbb{R}} \; t \\ s_i v_i^{\mathsf{T}} d \geqslant 1, \quad \forall\, i \in [1\!:\!k] \\ -s_{k+1} v_{k+1}^{\mathsf{T}} d \geqslant -t \\ t \geqslant -1. \end{cases} \tag{5.1}$$

When $s \in \mathcal{S}^1_k$, this problem is feasible (take d satisfying $s_i v_i^{\mathsf{T}} d \geqslant 1$, for all $i \in [1\!:\!k]$, and t sufficiently large) and bounded (its optimal value is $\geqslant -1$), so that it has a solution [18,13,14, 35]. Solving these LO problems is a time consuming part of the algorithms and in the numerical experiments of section 5.2.6, in particular in table 5.2, following [54], we measure the efficiency of the algorithms by the number of LO problems they solve.

One can now formally describe our version of the RC algorithm (the change is in step 2 of the RC-REC algorithm, which is not considered in the original RC algorithm).

Algorithm 5.5 (RC) Let be given $V \in \mathbb{R}^{n\times p}$, with n and $p \in \mathbb{N}^*$, having nonzero columns.

1. Execute the recursive procedure RC-REC$(V, v_1, +1)$.

Algorithm 5.6 (RC-REC) Let be given $V \in \mathbb{R}^{n\times p}$, with n and $p \in \mathbb{N}^*$, having nonzero columns, a direction $d \in \mathbb{R}^n$ and a sign vector $s \in \{\pm 1\}^k$ for some $k \in [1:p]$, such that $s_i v_i^\mathsf{T} d > 0$ for all $i \in [1:k]$.

1. If $k = p$, print s and return.
2. If $v_{k+1}^\mathsf{T} d \simeq 0$, then
   - 2.1. Execute RC-REC$(V, d_+, (s, +1))$, where $d_+ := d + t_+ v_{k+1}$ with $t_+ > 0$ chosen in the nonempty open interval
$$\left(0,\ \min_{\substack{i\in[1:k]\\ s_i v_i^\mathsf{T} v_{k+1}<0}} \frac{-v_i^\mathsf{T} d}{v_i^\mathsf{T} v_{k+1}}\right).$$
   - 2.2. Execute RC-REC$(V, d_-, (s, -1))$, where $d_- := d + t_- v_{k+1}$ with $t_- < 0$ chosen in the nonempty open interval
$$\left(\max_{\substack{i\in[1:k]\\ s_i v_i^\mathsf{T} v_{k+1}>0}} \frac{-v_i^\mathsf{T} d}{v_i^\mathsf{T} v_{k+1}},\ 0\right).$$
3. Else $s_{k+1} := \operatorname{sgn}(v_{k+1}^\mathsf{T} d)$.
   - 3.1. Execute RC-REC$(V, d, (s, s_{k+1}))$.
   - 3.2. Solve the LO problem (5.1) and denote by (d, t) a solution.
     If $t = -1$, execute RC-REC$(V, d, (s, -s_{k+1}))$.

The test $v_{k+1}^\mathsf{T} d \simeq 0$ done at the beginning of step 2 is supposed to take into account floating point arithmetic. In steps 2.1 and 2.2, the minimum and maximum are supposed to be infinite if their feasible set is empty. It is easy to see that the directions $d_\pm$ computed in steps 2.1 and 2.2 are such that $s_i v_i^\mathsf{T} d_\pm > 0$ for $i \in [1:k+1]$ and $s_{k+1} = \pm 1$, which justifies the recursive call to RC-REC with the given arguments. The most time-consuming part of the RC algorithm comes from the possible numerous LO problems to solve in step 3.2 of RC-REC.

5.2.2 An algorithm using stem vectors

When $s \in \mathcal{S}_k$, it is conceptually easy to check whether $(s, \pm 1)$ is in $\mathcal{S}_{k+1}$, provided a list of all the stem vectors associated with V is known. Indeed, by proposition 3.9, if no subvector of $(s, +1)$ (resp. $(s, -1)$) is a stem vector, then $(s, +1)$ (resp. $(s, -1)$) belongs to $\mathcal{S}_{k+1}$. Note also that, because any $s \in \mathcal{S}_k$ has at least one descendant in the $\mathcal{S}$-tree (proposition 4.6(3)), if it is observed that $(s, +1) \notin \mathcal{S}_{k+1}$, then, necessarily, $(s, -1) \in \mathcal{S}_{k+1}$. This observation prevents the algorithm from checking whether $(s, -1)$ contains a stem vector, which is a time consuming operation when the list of stem vectors is large. For future reference, we formalize this algorithm below.

Algorithm 5.7 (STEM) Let be given $V \in \mathbb{R}^{n\times p}$, with n and $p \in \mathbb{N}^*$, having nonzero columns.

1. Compute all the stem vectors associated with V.
2. Execute the recursive procedure STEM-REC$(V, +1)$.

Algorithm 5.8 (STEM-REC) Let be given $V \in \mathbb{R}^{n\times p}$, with n and $p \in \mathbb{N}^*$, having nonzero columns and a sign vector $s \in \{\pm 1\}^k$ for some $k \in [1:p]$.

1. If $k = p$, print s and return.
2. If no subvector of $(s, +1)$ is a stem vector, execute STEM-REC$(V, (s, +1))$.
3. If $(s, +1) \notin \mathcal{S}_{k+1}$ or no subvector of $(s, -1)$ is a stem vector, execute STEM-REC$(V, (s, -1))$.

This algorithm is improved below, as the option AD_4 of the ISF algorithm (see paragraphs A and D of section 5.2.4).

Note that, this algorithm need not generate directions d satisfying $s \cdot V_k^{\mathsf{T}} d > 0$, like the RC algorithm and need not solve any LO problem. Nevertheless, regarding the computation time, the algorithm has two bottlenecks that we now describe. Despite them, algorithm 5.7 is often the fastest in the numerical experiments of section 5.2.6.

The first bottleneck comes from the fact that the algorithm must compute all the stem vectors (or the set $\mathcal{C}$ of matroid circuits in (3.8)) associated with V. This is usually an expensive operation. For example, if V is randomly generated and of rank r, like in the test-cases `data_rand_*` in the experiments of section 5.2.6, any selection of r columns of V is likely to form an independent set of vectors, so that $\mathcal{C}$ is likely to be the sets of column indices of size $r+1$. Therefore, in this case, the number of circuits is likely to be the combination $\binom{p}{r+1}$ (and it is actually that number, see section 5.2.6(B.1)), which can be exponential in p (this number is bounded below by $2^{p/2}/(p+1)$ if p is even and $r+1 = p/2$ [23; (11.52)]). In the implemented ISF code, numerically tested in section 5.2.6, only the sets of columns whose cardinal is in $[3:r+1]$ are examined (since any group of two columns of V is supposed to be linearly independent and a group of $r+2$ columns or more is of nullity $\geqslant 2$, hence such group cannot form a matroid circuit; see (3.8)). In addition, the exploration is made using a tree structure for the column subsets, in order to discard the descendants of a circuit, which, by construction of the tree, contain this circuit and has more columns than this one. These two provisions are not sufficient, however, to prevent generating a lot of redundant circuits and, therefore, useless computation.

The second bottleneck is linked to the detection of a stem vector is the current sign vectors $(s, \pm 1)$. This operation requires to search the long list of stem vectors, which is a time consuming operation.

5.2.3 Linear optimization problem and stem vector

The property described in this section will be useful for the improvement D_2 of the ISF algorithm, described in section 5.2.4(D). It shows that a stem vector can be obtained easily from the dual solution of the LO problem (5.1), when $(s, -s_{k+1}) \notin \mathcal{S}_{k+1}$. Consider indeed the LO problem (5.1) and denote by (d, t) one of its solutions (these have been shown to exist). Then, either $t \geqslant 0$ (equivalently, $(s, -s_{k+1}) \notin \mathcal{S}_{k+1}$) or $t = -1$ (equivalently, $(s, -s_{k+1}) \in \mathcal{S}_{k+1}$).

Let σ_i, $i \in [1:k+1]$, be the multipliers associated with the first $k+1$ constraints of (5.1) and τ be the multiplier associated with its last constraint. Then, the Lagrangian dual of (5.1) reads [13, 11, 14, 34]

$$\begin{cases} \max_{(\sigma,\tau)\in\mathbb{R}^{k+1}\times\mathbb{R}} \ \sum_{i\in[1:k]} \sigma_i - \tau \\ \sigma \geqslant 0 \\ \tau \geqslant 0 \\ \sigma_{k+1} + \tau = 1 \\ \sigma_{k+1} s_{k+1} v_{k+1} = \sum_{i\in[1:k]} \sigma_i s_i v_i. \end{cases} \equiv \begin{cases} \max_{\sigma\in\mathbb{R}^{k+1}} \ \sum_{i\in[1:k+1]} \sigma_i - 1 \\ \sigma \geqslant 0 \\ \sigma_{k+1} \leqslant 1 \\ \sigma_{k+1} s_{k+1} v_{k+1} = \sum_{i\in[1:k]} \sigma_i s_i v_i, \end{cases} \tag{5.2}$$

where the second form of the dual is obtained by eliminating τ from the first form. By strong duality in linear optimization, the dual problems in (5.2) are feasible, have a solution and have the same optimal value as the primal problem. Let $(\sigma,\tau) \in \mathbb{R}^{k+1} \times \mathbb{R}$ be a dual solution. Then, $(s,-s_{k+1}) \in \mathcal{S}_{k+1}$ if and only if $t=-1$ if and only if $\sum_{i\in[1:k]} \sigma_i = 0$ and $\sigma_{k+1}=0$. We have shown that

$$(s,-s_{k+1}) \in \mathcal{S}_{k+1} \quad \iff \quad \sigma = 0.$$

Therefore, $(s,-s_{k+1}) \notin \mathcal{S}_{k+1}$ if and only if $\sigma \neq 0$ if and only if $\sigma_{k+1}=1$ (if $\sigma_{k+1}=0$, one can make the dual objective value as large as desired by multiplying σ by a factor going to $+\infty$; if $\sigma_{k+1} \in (0,1)$, the dual objective would by increased by replacing σ by σ/σ_{k+1}; in both cases the optimality of σ would be contradicted) if and only if $\tau = 0$. We have shown that

$$(s,-s_{k+1}) \notin \mathcal{S}_{k+1} \quad \iff \quad s_{k+1}v_{k+1} \in \operatorname{cone}\{s_i v_i : i \in [1:k]\}.$$

The next proposition shows how a matroid circuit can be detected from the dual solution σ when $(s,-s_{k+1}) \notin \mathcal{S}_{k+1}$.

Proposition 5.9 (matroid circuit detection) *Suppose that $(s,-s_{k+1}) \notin \mathcal{S}_{k+1}$ and that (σ,τ) is a solution to the dual problem in the left-hand side of (5.2) located at an extreme point of its feasible set. Then, $\{i \in [1:k+1] : \sigma_i > 0\}$ is a matroid circuit of V.*

Proof Necessarily, $\tau = 0$ and $\sigma_{k+1}=1$ when $(s,-s_{k+1}) \notin \mathcal{S}_{k+1}$. The fact that $(\sigma,0)$ is an extreme point of the feasible set of the problem in the left-hand side of (5.2) implies that the vectors [18,34]

$$\left\{ \begin{pmatrix} 0 \\ s_i v_i \end{pmatrix}_{i\in[1:k],\ \sigma_i>0}, \begin{pmatrix} 1 \\ -s_{k+1}v_{k+1} \end{pmatrix} \right\} \text{ are linearly independent.}$$

In particular, the vectors

$$\{s_i v_i : i \in [1:k],\ \sigma_i > 0\} \text{ are linearly independent.}$$

Since $s_{k+1}v_{k+1} = \sum_{i\in[1:k]} \sigma_i s_i v_i$, it follows that

$$\{s_i v_i : i \in [1:k+1],\ \sigma_i > 0\} \text{ has nullity one.}$$

The conclusion of the proposition follows from proposition 3.10. □

Recall that the dual-simplex algorithm finds a dual solution at an extreme point of the dual feasible set. For this reason, we use this approach in the ISF algorithm.

5.2.4 Improvements of the RC *and* STEM *algorithms*

This section presents several modifications of the RC algorithm and one modification of the STEM algorithm that significantly improve their performance. The modifications are indicated by the letters A, B, C and D, with reference to the sections where they are introduced. Additional numeric indices specify variants of the D option. The version AD_4 (modifications A and D_4) can be considered as an improvement of the new algorithm 5.7.

A. Taking the rank of V into account. Instead of starting with the vector $s = +1$, one can take into account the rank $r := \operatorname{rank}(V)$ to determine 2^r initial vectors s, hence avoiding to solve LO problems to determine these initial s's. This is especially useful when $p - r$ is small. In particular, when $p = r$, $\mathcal{S}$ is straightforwardly determined.

The algorithm selects $r := \operatorname{rank}(V)$ linearly independent vectors v_i, among the columns of $V \in \mathbb{R}^{n\times p}$. These vectors can be obtained by a QR factorization of

$$VP = QR,$$

where $P \in \{0,1\}^{p\times p}$ is a permutation matrix, $Q \in \mathbb{R}^{n\times n}$ is orthogonal (i.e., $Q^\mathsf{T} Q = I_n$) and $R \in \mathbb{R}^{n\times p}$ is upper triangular with $R_{[r+1\,:\,n],:} = 0$. To simplify the presentation, one can assume, without loss of generality, that $P = I$, in which case the vectors $v_1, \ldots, v_r$ are linearly independent (in practice, the vectors are symbolically reordered by using the permutation matrix P). By proposition 4.2,

$$\mathcal{S}_r = \{\pm 1\}^r. \tag{5.3}$$

Furthermore, for each $s \in \mathcal{S}_r$, we have, using $S := \operatorname{Diag}(s)$, $Q_r := Q_{:,[1\,:\,r]}$ and $R_r := R_{[1\,:\,r],[1\,:\,r]}$, that the vector

$$d_s = Q_r R_r^{-\mathsf{T}} s \tag{5.4}$$

is such that $s \bullet V_{:,[1\,:\,r]}^\mathsf{T} d_s = e > 0$, as desired.

For each $s \in \mathcal{S}_r$ and the associated d_s given by (5.4), the modified algorithm 5.5 will run the recursive function rc-rec(V, d_s, s) (see algorithm 5.11 below).

B. Special handling of the case where $v_{k+1}^\mathsf{T} d \simeq 0$. Directions $d_\pm := d + t_\pm v_{k+1}$ ensuring that $(s, \pm 1) \bullet V_{k+1}^\mathsf{T} d_\pm > 0$ can be computed not only when $v_{k+1}^\mathsf{T} d \simeq 0$ like in step 2 of the rc-rec algorithm 5.6, but also when $v_{k+1}^\mathsf{T} d$ is in the interval specified by (5.5) below. Note that the left-hand side in (5.5) is negative and the right-hand side is positive (this can be seen by multiplying numerators and denominators by s_i and by using $s_i v_i^\mathsf{T} d > 0$ for all $i \in [1\,:k]$), so that these inequalities are verified when $v_{k+1}^\mathsf{T} d = 0$. With the additional flexibility that (5.5) offers, the isf algorithm can sometimes avoid solving a significant number of LO problems of the form (5.1). For a proof of the next proposition, see [28].

Proposition 5.10 (two descendants without optimization) *Suppose that $s \in \{\pm 1\}^k$ verifies $s \bullet V_k^\mathsf{T} d > 0$, that $v_{k+1} \neq 0$ and that*

$$\max_{\substack{i\in[1\,:\,k]\\ s_i v_i^\mathsf{T} v_{k+1} > 0}} \frac{-v_i^\mathsf{T} d}{v_i^\mathsf{T} v_{k+1}} < \frac{-v_{k+1}^\mathsf{T} d}{\|v_{k+1}\|^2} < \min_{\substack{i\in[1\,:\,k]\\ s_i v_i^\mathsf{T} v_{k+1} < 0}} \frac{-v_i^\mathsf{T} d}{v_i^\mathsf{T} v_{k+1}}. \tag{5.5}$$

1) *The direction $d_+ := d + t_+ v_{k+1}$ verifies $s \bullet V_k^\mathsf{T} d_+ > 0$ and $v_{k+1}^\mathsf{T} d_+ > 0$ if and only if t_+ is in the nonempty open interval*

$$\left(\frac{-v_{k+1}^\mathsf{T} d}{\|v_{k+1}\|^2},\ \min_{\substack{i\in[1\,:\,k]\\ s_i v_i^\mathsf{T} v_{k+1} < 0}} \frac{-v_i^\mathsf{T} d}{v_i^\mathsf{T} v_{k+1}} \right). \tag{5.6a}$$

2) *The direction $d_- := d + t_- v_{k+1}$ verifies $s \bullet V_k^\mathsf{T} d_- > 0$ and $-v_{k+1}^\mathsf{T} d_- > 0$ if and only if t_- is in the nonempty open interval*

$$\left(\max_{\substack{i\in[1\,:\,k]\\ s_i v_i^\mathsf{T} v_{k+1} > 0}} \frac{-v_i^\mathsf{T} d}{v_i^\mathsf{T} v_{k+1}},\ \frac{-v_{k+1}^\mathsf{T} d}{\|v_{k+1}\|^2} \right). \tag{5.6b}$$

C. CHANGING THE ORDER OF THE VECTORS v_i'S. Each node s of the $\mathcal{S}$-tree described in section 5.2.1(A) has one or two descendants: $(s,+1)$ and/or $(s,-1)$. Since there is at most one LO problem solved per node of the $\mathcal{S}$-tree, decreasing the number of nodes should decrease the number of LO problems to solve, which significantly count in the computing time. To reach that goal, one can try to get as much as possible at the top of the tree the nodes having a single descendant. As shown below, this can be achieved by changing the order in which the vectors v_i's, the columns of V, are considered in the *depth-first search* of the tree; previously, the order was imposed by the modification A, taking into account the rank of V.

To implement this strategy, one associates with each node $s \in \mathcal{S}_k^1$ of the $\mathcal{S}$-tree, $k \in [1:p-1]$, the list of vectors considered so far at that node, denoted by $T_s := \{i_1,\dots,i_k\} \subseteq [1:p]$. Hence, we have to choose the next vector $v_{i_{k+1}}$ be selecting an index i_{k+1} in $T_s^c := [1:p] \setminus T_s$. Now, a natural idea is to restrict the set of possible indices to T_s^b, the set of indices j of T_s^c for which one of the intervals (5.6a) or (5.6b), with $v_{k+1} \equiv v_j$, is empty (implying that the technique used in the modification B will not give two descendants), if there is such an index, or T_s^c otherwise. To determine the index in T_s^b, we take

$$i_{k+1} = \operatorname*{arg\,max}_{i \in T_s^b} \frac{|v_i^{\mathsf{T}} d|}{\|v_i\|}, \tag{5.7}$$

which favors the vectors v_i for which $|v_i^{\mathsf{T}} d|/\|v_i\|$ is away from zero.

As table 5.2 indicates (section 5.2.6(C.3)), this modification has a significant impact on the decrease of LO to solve.

D. USING STEM VECTORS. We present in this section various modifications that use the concept of *stem vector*, introduced in the second part of section 3.2.2. These stem vectors are used to detect infeasible sign vectors, i.e., elements of $\mathcal{S}^c$, thanks to proposition 3.9. If $s \in \mathcal{S}_k^1$ and $(s, s_{k+1}) \in \mathcal{S}^c$ for $s_{k+1} \in \{\pm 1\}$, s has no descendant in $\mathcal{S}$ along (s, s_{k+1}), so that this part of the $\mathcal{S}$-tree does not need to be explored. From this point of view, computing all the stem vectors looks attractive, but, to our knowledge, this is a time consuming process, so that this option is not necessarily the most efficient one. The modifications presented below use more and more stem vectors, which require more and more computing time.

- D$_1$) Natural candidates as stem vectors are those obtained from the matroid circuits I made of $r+1$ columns of V ($r = \operatorname{rank}(V)$) formed of the r linear independent columns selected by the QR factorization of section 5.2.4(A) and one of the remaining $p-r$ columns of V. By proposition 3.10, such I contains exactly one circuit. Therefore, one detects in this way $p-r$ circuits and $2(p-r)$ stem vectors. This is not much compared to the total number of stem vectors, which may depend exponentially on p, so that the number of infeasible sign vectors detected by these stem vectors is usually relatively small (see table 5.2).
- D$_2$) With this option, when a LO problem (5.1) is solved at a certain node $s \in \mathcal{S}_k^1$ to see whether (s, s_{k+1}) belongs to $\mathcal{S}_{k+1}^1$, for $s_{k+1} \in \{\pm 1\}$, the dual solution is used to determine a stem vector, as shown by proposition 5.9. For this purpose, the ISF code solves the LO problems with the dual-simplex algorithm, so that the computed dual solution is at a vertex of the dual feasible set.
- D$_3$) With this option, all the stem vectors are computed, before running the recursive process that builds the $\mathcal{S}$-tree. At each note $s \in \mathcal{S}_k^1$, the algorithm still computes a direction $d \in \mathbb{R}^n$ such that $s_i v_i^{\mathsf{T}} d > 0$ for all $i \in T_s$ (the set of vector indices considered so far at s). The advantage of this direction is to allow the algorithm to use the beneficial modifications B and C and to easily determine one or two signs $s_{k+1} \in \{\pm 1\}$ such that $(s, s_{k+1}) \in \mathcal{S}_{k+1}^1$. If a single sign

$s_{k+1} \in \{\pm 1\}$ is selected, the stem vectors can decide whether $(s, -s_{k+1}) \in \mathcal{S}^1_{k+1}$. If this is the case, this option D_3 has the inconvenient of still requiring to solve a LO problem to get a direction associated with $(s, -s_{k+1})$. These LO problems (5.1) have an optimal value -1 and should not be solved exactly. Indeed, as soon as a feasible direction d for (5.1) gives a negative value to the objective of the problem, one could stop solving it, since this d verifies $s_i v_i^{\mathsf{T}} d > 0$ for all $i \in T_{(s,-s_{k+1})}$. We have not implemented that inexact solve of the LO problems, by lack of flexibility of the solver `Linprog` in Matlab.

D_4) Like with the option D_3, all the stem vectors are computed, before running the recursive process that builds the $\mathcal{S}$-tree. But now, unlike with option D_3, the algorithm computes no direction $d \in \mathbb{R}^n$. The approach can be viewed as an improvement of the algorithm 5.7 (STEM) presented in section 5.2.2, in the sense that option A is also activated.

Note that, knowing all the stem vectors, one could compute the complementary set $\mathcal{S}^c$ rather easily by completing with ± 1 the unspecified components of the stem vectors. Next, $\mathcal{S}$ could be obtained from $\mathcal{S}^c$ by taking its complementary set in $\{\pm 1\}^p$, but a straightforward implementation of this last operation looks rather expensive, so that we have not experimented it numerically.

5.2.5 ISF *algorithm*

We have named ISF (for Incremental Signed Feasibility) the algorithm that improves the RC algorithm 5.5 or the STEM algorithm 5.7 with the enhancements described in section 5.2.4. For the purpose of precision and reference, we formally state it in this section. It would be cumbersome and confusing, hence inappropriate, to mention all the options in its description, in particular because all of them have been specified separately in the previous section. As an example of algorithm, we provide a description with the options ABCD_2. It starts with a hat procedure ISF, similar to that of the RC algorithm but with the additional easy determination of $\mathcal{S}_r$ (modification A) and the computation of some stem vectors (modification D_1). Then, the hat procedure calls the recursive procedure ISF-REC.

Algorithm 5.11 (ISF, with options ABCD_2) Let be given $V \in \mathbb{R}^{n\times p}$, with n and $p \in \mathbb{N}^*$, having nonzero columns.

1. Compute the QR factorization of V. Let $r = \operatorname{rank}(V)$ and $T_r := \{i_1, \ldots, i_r\}$ be the indices of r selected linear independent columns of V.
2. Compute the $p - r$ matroid circuits containing T_r (see option D_1).
3. For each $s \in \mathcal{S}_r$, given by (5.3), and its associated d_s, given by (5.4), call the recursive procedure ISF-REC(V, T_r, d_s, s).

Algorithm 5.12 (ISF-REC, with options BCD_2) Let be given $V \in \mathbb{R}^{n\times p}$, with n and $p \in \mathbb{N}^*$ of rank r, having nonzero columns v_i, T_k a selection of k columns of V (with $k \in [r\!:\!p]$), a direction $d \in \mathbb{R}^n$ and a sign vector $s \in \{\pm 1\}^k$ for some $k \in [r\!:\!p]$. It is assumed that $s_i v_i^{\mathsf{T}} d > 0$ for all $i \in T_k$.

1. If $k = p$, print s and return.
2. Determine the index $i_{k+1} \in [1\!:\!p] \setminus T_k$ of the next vector to consider by option C and set $T_{k+1} := T_k \cup \{i_{k+1}\}$.
3. If (5.5) holds (with $[1\!:\!k]$ changed into T_k and $k+1$ into i_{k+1}), then
   3.1. Execute ISF-REC$(V, T_{k+1}, d_+, (s, +1))$, where $d_+ := d + t_+ v_{i_{k+1}}$ and t_+ is chosen in the nonempty open interval (5.6a).

3.2. Execute ISF-REC$(V, T_{k+1}, d_-, (s, -1))$, where $d_- := d + t_- v_{i_{k+1}}$ and $t_- < 0$ chosen in the nonempty open interval (5.6b).

4. Else $s_{k+1} := \operatorname{sgn}(v_{i_{k+1}}^\mathsf{T} d)$.

4.1. Execute ISF-REC$(V, T_{k+1}, d, (s, s_{k+1}))$.

4.2. If $(s, -s_{k+1})$ contains a stem vector, return.

4.3. Solve the LO problem (5.1) (with $[1\!:\!k]$ changed into T_k and $k+1$ into i_{k+1}) by the dual-simplex algorithm and denote by (d, t) a solution.

4.3.1. If $t = -1$, execute ISF-REC$(V, T_{k+1}, d, (s, -s_{k+1}))$.

4.3.2. Else, use the dual solution to store two more stem vectors by option D_2.

5.2.6 Numerical experiments

A. COMPUTER AND PROBLEM PRESENTATIONS. We present in tables 5.1, 5.2 and 5.3 the results obtained by running the ISF algorithm 5.11 (with several variants) on a small number of problems and compare it with our implementation of the RC algorithm 5.5, simulating the algorithm 1 (IE) in [54]. The implementations have been done in Matlab (version "`9.11.0.1837725 (R2021b) Update 2`") on a `MacBookPro18,2/10cores` with the system `macOS Monterey`, version `12.6.1`.

We have assessed the codes on randomly generated problems (function **`rand`** in Matlab, names prefixed by **`rand`** and **`srand`** in the first part of table 5.1) and problems adapted from [54] (names prefixed by `rc`). The **`rand-n-p-r`** problems have their data formed of a randomly generated matrix $V \in \mathbb{R}^{\mathtt{n}\times\mathtt{p}}$ with prescribed rank $\mathtt{r} := \operatorname{rank}(V)$. The matrix V of problem **`srand-n-p-q`** (`s` for structured) has its n first columns formed of the `n` basis vectors of $\mathbb{R}^{\mathtt{n}}$ and the last $\mathtt{p}-\mathtt{n} > 0$ columns have `q` nonzero random integer elements, randomly positioned, which induces many matroid circuits. The `rc` problems are adapted from [54] and given in the second part of table 5.1.

B. OBSERVATIONS ON TABLE 5.1. The dimensions n, p and r of the problems are given in columns 2-4 of table 5.1. Column 5 gives the number ς of matroid circuits of V. In column 6 and 7, one finds the cardinal $|\partial_B H(x)| = |\mathcal{S}|$ of the B-differential $\partial_B H(x)$ and the Winder upper bound (the right-hand side of (4.8)). The codes will be compared on the number of LO problems they solve, which is a good image of their computation effort, measured independently of the computer used to run the codes and the features of the LO solvers. A first example of comparison is given in columns 8 and 9 of table 5.1, where one finds the number of LO problems solved by the original RC algorithm and the simulated RC algorithm implemented in the ISF code respectively. The latter code will be used next, in the comparison with its improved versions, both regarding the LO problem counters (table 5.2) and the CPU times (table 5.3).

1) The randomly generated problems **`rand`** are likely to provide vectors v_i's (the columns of V) in general position, in the sense of definition 4.9. This can be seen indirectly on the numbers in table 5.1.
   - It is known from proposition 4.10 that (4.7) implies equality in (4.8). This equality indeed holds, as we can observe by comparing columns 6 and 7.
   - Incidentally, one can compute mentally Winder's bound β when p is even and $r = p/2$. In that case, the right-hand side of (4.8) reads

$$\beta = 2 \sum_{i\in[0\,:\,r-1]} \binom{2r-1}{i} = \sum_{i\in[0\,:\,2r-1]} \binom{2r-1}{i} = s^{2r-1} = 2^{p-1}.$$

This is what one observes in the table; for example when $p = 8$ and $r = 4$, one has $\beta = 128$, which is indeed 2^{8-1}.

- The number of circuits is also predictable for the random problems. Indeed, by the random generation of V, a subset of columns is likely to have nullity 1 (i.e., to form a circuit, in the matroid terminology) if and only if it contains $r + 1$ columns (r being the rank of V). Therefore, their number should be $\binom{p}{r+1}$ (see also [39; footnote 1]), which is indeed the number displayed in the 5th column of table 5.1 for the randomly generated problems.

2) One observes that when $r = 2$, one has $|\partial_B H(x)| = 2p$ (proposition 4.13).
3) The number of matroid circuits, given in the column labeled by ς, depends on the determination of the nonzero elements of the normalized vector $\alpha \in \mathcal{N}(V_{:,I}) \setminus \{0\}$ for the selected index set I (proposition 3.10). This operation is sensitive to a threshold value that is set to $10^5\varepsilon$, where $\varepsilon > 0$ is the machine epsilon; smaller values for this threshold have occasionally given larger numbers of matroid circuits. In other words, due to the floating point calculation, there is no certainty that the given number of circuits is the one that would be obtained in exact arithmetic.
4) A comparison between the "Original RC code" in Python and its "Simulated RC code" in Matlab shows that the latter is slightly more effective in terms of the number of LO problems solved. This is probably due to the special treatment in step 2 of the case where $v_{k+1}^\mathsf{T} d \simeq 0$ in algorithm 5.6, which is not considered in the original code.

C. Observations on table 5.2. Table 5.2 shows the effect of the modifications discussed in section 5.2.4 on the number of LO problems (LOP) solved, which significantly counts in the computing time. This will lead us to select three algorithms, those which bring the best profit on the LOP counter. The columns labeled "Ratio" show the acceleration ratio with respect to the simulated RC code in terms of LOP, that is the ratio of the LOP counter of the considered algorithm divided by the LOP counter of the simulated RC algorithm. On the last two lines of the table, one finds the mean and median values of these acceleration ratios, which may be viewed as a summary of the effect of the considered modification.

1) The modification A, proposed in section 5.2.4(A), which uses the QR factorization to get r linearly independent columns of V, does not bring a large benefit ("Ratio" is close to 1) and sometimes increases the number of LO problems to solve. The benefit is not important since it prevents "only" $\sum_{i \in [0:r]} 2^i = 2^{r+1} - 1$ nodes to run the LO solver, which is usually a small fraction of the total number of nodes of the $\mathcal{S}$-tree. One also observes that the number of solved LOP may increase (acceleration ratio < 1), which is sometimes due to the fact that the number 2^r of nodes at level r with modification A is larger than the one without modification A, which contributes to an increase in the total number of nodes of the constructed $\mathcal{S}$-tree and, therefore, tends to increase the number of LO to solve. Furthermore, the order in which the vectors are considered without/with modification A is not identical, which has also an impact on the number of solved LOP (see section 5.2.4(C)).
2) The modification B, proposed in section 5.2.4(B), which is able to detect two descendants of an $\mathcal{S}$-tree node, without solving any LO problem, has a significant impact on the total number of these problems. We see, indeed, that the (mean, median) acceleration ratio is raised to (1.24, 1.18).
3) Consider now the modification C, described in section 5.2.4(C), which changes the order in which the vectors v_i's are considered. We use the test-problem `rand-7-13-5` to show this effect in the next table.

	Number of nodes per level													Total
With modifications AB	1	2	4	8	16	31	57	99	163	256	386	562	794	2379
With modifications ABC	1	2	4	8	16	26	43	69	107	168	270	443	794	1951
$\mathcal{S}$-tree levels	1	2	3	4	5	6	7	8	9	10	11	12	13	

							LO problems solved in		
Problem	n	p	r	ς	$\lvert\partial_B H(x)\rvert$	Winder's bound	Original RC	Simulated RC	Difference
`rand-4-8-2`	4	8	2	56	16	16	29	28	1
`rand-7-8-4`	7	8	4	56	128	128	99	98	1
`rand-7-9-4`	7	9	4	126	186	186	163	162	1
`rand-7-10-5`	7	10	5	210	512	512	382	381	1
`rand-7-11-4`	7	11	4	462	352	352	386	385	1
`rand-7-12-6`	7	12	6	792	2048	2048	1486	1485	1
`rand-7-13-5`	7	13	5	1716	1588	1588	1586	1585	1
`rand-7-14-7`	7	14	7	3003	8192	8192	5812	5811	1
`rand-8-15-7`	8	15	7	6435	12952	12952	9908	9907	1
`rand-9-16-8`	9	16	8	11440	32768	32768	22821	22818	3
`rand-10-17-9`	10	17	9	19448	78406	78406	50643	50642	1
`srand-8-20-2`	8	20	8	990	24544	188368	28748	28620	128
`srand-8-20-4`	8	20	8	88752	157192	188368	136133	135566	567
`srand-8-20-6`	8	20	8	160074	186430	188368	167545	167262	283
`rc-2d-20-4`	4	19	4	926	136	1976	548	545	3
`rc-2d-20-5`	5	20	5	1317	272	10072	1096	1091	5
`rc-2d-20-6`	6	20	6	1120	512	33328	1936	1927	9
`rc-2d-20-7`	7	20	7	910	960	87592	3392	3343	49
`rc-2d-20-8`	8	20	8	728	1792	188368	5888	5855	33
`rc-perm-5`	5	15	5	268	720	2942	1211	1066	145
`rc-perm-6`	6	21	6	1649	5040	43400	10417	9346	1071
`rc-perm-7`	7	28	7	11874	40320	795188	99155	90169	8986
`rc-perm-8`	8	36	8	95097	362880	17463696	1036897	953009	83888
`rc-ratio-20-3-7`	3	19	3	3488	304	344	929	928	1
`rc-ratio-20-3-9`	3	19	3	1369	178	344	539	536	3
`rc-ratio-20-4-7`	4	20	4	15150	2278	2320	4954	4953	1
`rc-ratio-20-4-9`	4	20	4	14065	2016	2320	4393	4388	5
`rc-ratio-20-5-7`	5	20	5	34575	8470	10072	13798	13785	13
`rc-ratio-20-5-9`	5	20	5	31396	7826	10072	13798	13797	1
`rc-ratio-20-6-7`	6	20	6	56564	26194	33328	32993	32980	13
`rc-ratio-20-6-9`	6	20	6	64058	26758	33328	39823	39717	106
`rc-ratio-20-7-7`	7	20	7	112604	76790	87592	82751	82738	13
`rc-ratio-20-7-9`	7	20	7	75275	58468	87592	70214	70198	16

Table 5.1 Description of the test-problems and comparison of the "original RC algorithm in [54]", written in Python, and the "simulated RC algorithm 5.5", written in Matlab: "(n, p, r, ς)" are the dimensions of the problem ($V \in \mathbb{R}^{n\times p}$ is of rank r and has ς circuits), "Winder's bound" is the right-hand side of (4.8), "$|\partial_B H(x)|$" is the cardinal of the B-differential of H given by (1.2), "Original RC" gives the number of LO problems solved by the original piece of software in Python of Rada and Černý [54], "Simulated RC" gives the number of LO problems solved by the implementation in the Matlab code ISF of the Rada and Černý algorithm (see algorithm 5.5), "Difference" is the difference between the two previous columns.

The table gives the number of nodes for each level in the $\mathcal{S}$-tree, with the modifications AB and with the modifications ABC. Since $\operatorname{rank}(V) = 5$ for this problem and since the modification A is used in both cases, the number of nodes per level, only starts to differ from level 6 (before that it is equal to 2^{l-1}, where l is the $\mathcal{S}$-tree level). The final level is 13 (since there are $p = 13$ vectors) and its number of leaves is $|\mathcal{S}|/2 = 794$ (an observation from the table above or from table 5.2), necessary identical in both cases. The effect of the modification C can be seen on the smaller number of nodes per level and in all the $\mathcal{S}$-tree (rightmost column). This contributes to the decrease of the number of LO to solve: the (mean, median) acceleration ratio is raised to (2.35, 2.03).

4) The modifications D, described in section 5.2.4(D), deal with the contribution of the computed stem vectors, whose number increases from modification D_1 ($2(p-r)$ stem vectors after the QR factorization of V), D_2 (more stem vectors from the dual solution of the LO problem (5.1) when this one has a nonnegative optimal value), D_3 and D_4 (all the stem vectors).
   - We see that the option D_1 yields already some improvement (less LO to solve), but not much, raising the (mean, median) acceleration ratio from (2.35, 2.03) to (2.63, 2.15).
   - The use of the option D_2 is more beneficial since the (mean, median) acceleration ratio now goes up to (23.75, 3.29). We understand this fact to have its origin in the increase in the number of stem vectors detected from the dual solutions of some solved LO problems. Note that this last operation does not require much computation time.
   - With option D_3, only the LO problems (5.1) with the optimal value -1 are solved. This reduces even more significantly the number of LO to solve, with a (mean, median) acceleration ratio that now reaches (27.91, 4.63).
   - With option D_4, no LO problem is solved.

In conclusion of these observations, one could retain the following three solvers:

- ISF($ABCD_2$) is the most efficient solver that does not compute all the stem vectors,
- ISF($ABCD_3$) has room for improvement in a compiled language (compared to an interpreter, like Matlab) and therefore should not be discarded,
- ISF(AD_4) is the option combination without optimization problem to solve, which is an interesting feature (it is also the solver described in section 5.2.2 with the modification A of section 5.2.4 in addition). As we shall see in section 5.2.6(D), it is the solver that has usually the lowest (mean, median) CPU time on the considered test problems, but this good property is sometimes invalidated on problems with many stem vectors.

Problem	Simulated RC	ISF (A) LOP	ISF (A) Ratio	ISF (AB) LOP	ISF (AB) Ratio	ISF (ABC) LOP	ISF (ABC) Ratio	ISF (ABCD$_1$) LOP	ISF (ABCD$_1$) Ratio	ISF (ABCD$_2$) LOP	ISF (ABCD$_2$) Ratio	ISF (ABCD$_3$) LOP	ISF (ABCD$_3$) Ratio	ISF (AD$_4$) LOP	ISF (AD$_4$) Ratio
`rand-4-8-2`	28	27	1.04	21	1.33	16	1.75	10	2.80	9	3.11	0	—	0	—
`rand-7-8-4`	98	91	1.08	57	1.72	41	2.39	38	2.58	35	2.80	20	4.90	0	—
`rand-7-9-4`	162	155	1.05	108	1.50	80	2.02	74	2.19	61	2.66	35	4.63	0	—
`rand-7-10-5`	381	366	1.04	233	1.64	169	2.25	167	2.28	144	2.65	100	3.81	0	—
`rand-7-11-4`	385	378	1.02	287	1.34	167	2.31	168	2.29	132	2.92	75	5.13	0	—
`rand-7-12-6`	1485	1454	1.02	1012	1.47	748	1.99	735	2.02	628	2.36	495	3.00	0	—
`rand-7-13-5`	1585	1570	1.01	1234	1.28	763	2.08	759	2.09	589	2.69	401	3.95	0	—
`rand-7-14-7`	5811	5748	1.01	4222	1.38	3129	1.86	3100	1.87	2663	2.18	2233	2.60	0	—
`rand-8-15-7`	9907	9844	1.01	7642	1.30	5174	1.91	5199	1.91	4355	2.27	3638	2.72	0	—
`rand-9-16-8`	22818	22691	1.01	17586	1.30	13038	1.75	13023	1.75	11185	2.04	9943	2.29	0	—
`rand-10-17-9`	50642	50387	1.01	38167	1.33	28912	1.75	28839	1.76	25370	2.00	23266	2.18	0	—
`srand-8-20-2`	28620	28620	1.00	20207	1.42	6668	4.29	5535	5.17	2881	9.93	2851	10.04	0	—
`srand-8-20-4`	135566	136027	1.00	113493	1.19	60066	2.26	59267	2.29	45569	2.97	42445	3.19	0	—
`srand-8-20-6`	167262	167351	1.00	137450	1.22	77800	2.15	77752	2.15	62694	2.67	54980	3.04	0	—
`rc-2d-20-4`	545	540	1.01	480	1.14	256	2.13	196	2.78	43	12.67	0	—	0	—
`rc-2d-20-5`	1091	1080	1.01	960	1.14	528	2.07	408	2.67	44	24.80	0	—	0	—
`rc-2d-20-6`	1927	1904	1.01	1680	1.15	912	2.11	688	2.80	40	48.17	0	—	0	—
`rc-2d-20-7`	3343	3296	1.01	2912	1.15	2208	1.51	1792	1.87	52	64.29	0	—	0	—
`rc-2d-20-8`	5855	5760	1.02	4992	1.17	2752	2.13	1984	2.95	28	209.11	0	—	0	—
`rc-perm-5`	1066	1049	1.02	851	1.25	292	3.65	216	4.94	20	53.30	4	266.50	0	—
`rc-perm-6`	9346	9280	1.01	7898	1.18	2176	4.30	1836	5.09	92	101.59	61	153.21	0	—
`rc-perm-7`	90169	90094	1.00	79049	1.14	18794	4.80	16558	5.45	960	93.93	855	105.46	0	—
`rc-perm-8`	953009	952597	1.00	856597	1.11	168395	5.66	158989	5.99	9766	97.58	9393	101.46	0	—
`rc-ratio-20-3-7`	928	925	1.00	839	1.11	514	1.81	447	2.08	282	3.29	81	11.46	0	—
`rc-ratio-20-3-9`	536	564	0.95	541	0.99	456	1.18	432	1.24	152	3.53	23	23.30	0	—
`rc-ratio-20-4-7`	4953	4943	1.00	4522	1.10	2570	1.93	2500	1.98	1394	3.55	739	6.70	0	—
`rc-ratio-20-4-9`	4388	4498	0.98	4113	1.07	1998	2.20	1988	2.21	1054	4.16	562	7.81	0	—
`rc-ratio-20-5-7`	13785	15341	0.90	12979	1.06	7185	1.92	7064	1.95	3644	3.78	2467	5.59	0	—
`rc-ratio-20-5-9`	13797	13650	1.01	12220	1.13	6808	2.03	6719	2.05	3485	3.96	2454	5.62	0	—
`rc-ratio-20-6-7`	32980	35882	0.92	31967	1.03	17956	1.84	17505	1.88	10669	3.09	8765	3.76	0	—
`rc-ratio-20-6-9`	39717	40906	0.97	36485	1.09	21638	1.84	20142	1.97	11187	3.55	9061	4.38	0	—
`rc-ratio-20-7-7`	82738	81428	1.02	76158	1.09	47910	1.73	47748	1.73	30442	2.72	25841	3.20	0	—
`rc-ratio-20-7-9`	70198	(1)		51974	1.35	37448	1.87	35542	1.98	22295	3.15	19876	3.53	0	—
Mean			1.00		1.24		2.35		2.63		23.74		27.91		—
Median			1.01		1.18		2.03		2.15		3.29		4.63		—

Number of LO problems solved (LOP) and acceleration ratio (Ratio) with various options

Table 5.2 Improvement brought by the modifications described in section 5.2.4, in terms of the number of LO problems to solve: A (taking the rank of V into account), B (special handling of the case where $v_{k+1}^{\mathsf{T}} d \simeq 0$), C (changing the order of the vectors v_i's by taking i_{k+1} by (5.7)), D_1 (pre-computation of $2(p-r)$ stem vectors after the QR factorization), D_2 (D_1 and 2 additional stem vectors computed after solving a LO problem, whose optimal value is nonnegative), D_3 (all the stem vectors are first computed and, for $(s, \pm 1) \in \mathcal{S}_{k+1}$, a LO problem is solved to get a handle d), D_4 (all the stem vectors are first computed and no LO is solved). The "Ratio" (acceleration ratio) columns give for each considered problem the ratio (*LOP of the considered* ISF *version*)/(*LOP of the simulated* RC). Note: (1) failure of the LO solver `Linprog-'dual-simplex'`, which exceeds 5000 iterations.

D. Observations on table 5.3. Measuring the efficiency of the algorithms by the number of LO solved during execution, like in table 5.2, is sometimes misleading. If this is the main cost item for some algorithms, it is no longer the case when a large amount of stem vectors is computed. For two reasons. First, the time spent in the computation of these stem vectors is not negligible, far from it, at least in our implementation, in which each of them requires the computation of the nullity of a matrix and a null space vector. Next, verifying that a sign vector contains a stem vector (proposition 3.9) is also time consuming when there are many stem vectors. Therefore a comparison of the CPU time of the runs is welcome. This is done for a selection of versions of the isf codes in table 5.3, those selected at the end of section 5.2.6(C). Here are some observations on the statistics of this table.

1) A first observation is that the good behavior of the selected versions of the isf codes is confirmed, even though the acceleration ratios are not as large as the one based on the number of LO problems solved. This can be explained by the fact that the time spent in solving LO problems is counterbalanced by the handling of stem vectors for the version $ABCD_3$ and AD_4. Anyway, one observes that the CPU time acceleration ratios have (mean, median) values in the ranges (7..15, 3..14), which is significant.
2) The most effective combination of code options depends actually on the considered problems. It is difficult to state a rule that would predict which code behaves best because some solvers are better on some phases of the run, but worse on others (the three main phases are the detection of the stem vectors, the execution of LO problems and the search for stem vectors covered by a given sign vector). Actually, this multicriterion problem has no clear solution and we leave this question open for future numerical experiments.

6 Discussion

This paper deals with the description and computation of the B-differential of the componentwise minimum of two affine vector functions. The fact that this problem has many equivalent formulations, some of them being highlighted in section 3, implies that the present contribution has an impact on several domains, including on the description of the arrangement of hyperplanes in the space. To this respect, a singular aspect of this contribution is to propose a dual approach to solve the problem, using some or all the stem vectors, a concept made useful thanks to the convex analysis tool that is Gordan's alternative. Besides this contribution, the paper also brings various improvements of an algorithm of Rada and Černý [54], which was designed to determine the cells of an arrangement of hyperplanes in the space.

Even in the spirit of the methods proposed in this article, there is still room for improvement, in relation to three identified bottlenecks: (*i*) we have mentioned that with the option D_3, the LO problem (5.1) can be solved inexactly, since, in that case, the optimal value is -1, while any negative objective value for a feasible unknown would suffice, but this requires a better tuning of the linear optimization solver, (*ii*) computing more efficiently all the stem vectors (or matroid circuits) of the matrix V is certainly a source of improvement, (*iii*) a better storage of the stem vectors that would allow the algorithm to decide more rapidly that a sign vector contains a stem vector is also welcome. Some of these possible improvements are also linked to a better choice of programming language, probably one using a compilation phase.

This contribution has also various possible extensions. One would be to develop a dual approach to the problem of the arrangement in the space of hyperplanes *having no point in common* [29]. Another natural extension would be to see the implications of this work for computing the B-differential of the componentwise minimum of *nonlinear* vector functions [30].

Problem	Simulated RC	ISF (ABCD$_2$) Time	ISF (ABCD$_2$) Ratio	ISF (ABCD$_3$) Time	ISF (ABCD$_3$) Ratio	ISF (AD$_4$) Time	ISF (AD$_4$) Ratio
	CPU times (in sec)						
`rand-4-8-2`	1.13	0.95	1.19	0.15	7.53	0.09	12.56
`rand-7-8-4`	1.66	1.19	1.39	1.04	1.60	0.11	15.09
`rand-7-9-4`	2.18	1.41	1.55	1.17	1.86	0.12	18.17
`rand-7-10-5`	3.86	2.03	1.90	1.63	2.37	0.15	25.73
`rand-7-11-4`	3.90	1.94	2.01	1.50	2.60	0.16	24.38
`rand-7-12-6`	11.87	5.50	2.16	4.58	2.59	0.38	31.24
`rand-7-13-5`	12.80	5.31	2.41	4.08	3.14	0.48	26.67
`rand-7-14-7`	44.06	20.85	2.11	18.08	2.44	1.98	22.25
`rand-8-15-7`	73.31	33.09	2.22	30.78	2.38	5.22	14.04
`rand-9-16-8`	175.78	83.11	2.12	83.99	2.09	21.06	8.35
`rand-10-17-9`	410.86	185.85	2.21	217.53	1.89	70.34	5.84
`srand-8-20-2`	187.22	20.84	8.98	25.16	7.44	6.86	27.29
`srand-8-20-4`	985.69	351.11	2.81	639.82	1.54	686.00	1.44
`srand-8-20-6`	1079.92	516.67	2.09	1227.16	0.88	1557.59	0.69
`rc-2d-20-4`	4.79	1.35	3.55	0.34	14.09	0.25	19.16
`rc-2d-20-5`	8.67	1.42	6.11	0.44	19.70	0.34	25.50
`rc-2d-20-6`	14.71	1.44	10.22	0.58	25.36	0.53	27.75
`rc-2d-20-7`	26.58	2.09	12.72	1.04	25.56	0.83	32.02
`rc-2d-20-8`	44.32	2.02	21.94	1.43	30.99	1.29	34.36
`rc-perm-5`	8.07	1.23	6.56	1.02	7.91	0.34	23.74
`rc-perm-6`	64.31	2.62	24.55	3.92	16.41	3.31	19.43
`rc-perm-7`	675.87	13.77	49.08	52.43	12.89	85.43	7.91
`rc-perm-8`	6846.41	127.91	53.53	1614.33	4.24	5216.84	1.31
`rc-ratio-20-3-7`	6.91	3.07	2.25	1.82	3.80	0.38	18.18
`rc-ratio-20-3-9`	3.88	2.25	1.72	1.30	2.98	0.27	14.37
`rc-ratio-20-4-7`	32.43	11.70	2.77	8.76	3.70	3.51	9.24
`rc-ratio-20-4-9`	27.24	8.65	3.15	6.67	4.08	3.10	8.79
`rc-ratio-20-5-7`	86.35	27.34	3.16	30.58	2.82	22.61	3.82
`rc-ratio-20-5-9`	83.35	25.43	3.28	29.40	2.84	19.14	4.35
`rc-ratio-20-6-7`	203.89	75.41	2.70	115.68	1.76	95.12	2.14
`rc-ratio-20-6-9`	246.95	78.54	3.14	127.76	1.93	113.62	2.17
`rc-ratio-20-7-7`	533.16	221.18	2.41	481.12	1.11	525.48	1.01
`rc-ratio-20-7-9`	451.87	160.50	2.82	311.35	1.45	269.56	1.68
Mean			7.60		6.79		14.87
Median			2.77		2.84		14.37

Table 5.3 Comparison of the computing times.

Acknowledgments

We thank Rada and Černý for providing their code and test problems, those used in [54]; part of these were used in the numerical experiments.

References

1. M. Aganagić (1984). Newton's method for linear complementarity problems. *Mathematical Programming*, 28, 349–362. [doi]. 2
2. Marcelo Aguiar, Swapneel Mahajan (2017). *Topics in hyperplane Arrangements*. Mathematical Surveys and Monographs 226. American Mathematical Society, Providence, RI. [doi]. 14, 15, 16, 18
3. G.L. Alexanderson, John E. Wetzel (1981). Arrangements of planes in space. *Discrete Mathematics*, 34(3), 219–240. [doi]. 18
4. G.L. Alexanderson, John E. Wetzel (1983). Erratum: "arrangements of planes in space". *Discrete Mathematics*, 45(1), 140. [doi]. 18
5. David Avis, Komei Fukuda (1996). Reverse search for enumeration. *Discrete Applied Mathematics*, 65(1-3), 21–46. [doi]. 5, 14, 24
6. Pierre Baldi, Roman Vershynin (2019). Polynomial threshold functions, hyperplane arrangements, and random tensors. *SIAM Journal on Mathematics of Data Science*, 1(4), 699–729. [doi]. 5

7. I. Ben Gharbia, J.Ch. Gilbert (2012). Nonconvergence of the plain Newton-min algorithm for linear complementarity problems with a P-matrix. *Mathematical Programming*, 134, 349–364. [doi]. 2
8. I. Ben Gharbia, J.Ch. Gilbert (2013). An algorithmic characterization of P-matricity. *SIAM Journal on Matrix Analysis and Applications*, 34(3), 904–916. [doi]. 2
9. I. Ben Gharbia, J.Ch. Gilbert (2019). An algorithmic characterization of P-matricity II: adjustments, refinements, and validation. *SIAM Journal on Matrix Analysis and Applications*, 40(2), 800–813. [doi]. 2
10. I. Ben Gharbia, J. Jaffré (2014). Gas phase appearance and disappearance as a problem with complementarity constraints. *Mathematics and Computers in Simulation*, 99, 28–36. [doi]. 2
11. D.P. Bertsekas (1999). *Nonlinear Programming* (second edition). Athena Scientific. 27
12. H. Bieri, W. Nef (1982). A recursive sweep-plane algorithm, determining all cells of a finite division of $\mathbb{R}^d$. *Computing*, 28(3), 189–198. [doi]. 24
13. J.F. Bonnans, J.Ch. Gilbert, C. Lemaréchal, C. Sagastizábal (1997). *Optimisation Numérique – Aspects théoriques et pratiques*. Mathématiques et Applications 27. Springer Verlag, Berlin. [editor]. 25, 27
14. J.F. Bonnans, J.Ch. Gilbert, C. Lemaréchal, C. Sagastizábal (2006). *Numerical Optimization – Theoretical and Practical Aspects* (second edition). Universitext. Springer Verlag, Berlin. [authors] [editor] [doi]. 25, 27
15. J.M. Borwein, A.S. Lewis (2006). *Convex Analysis and Nonlinear Optimization – Theory and Examples* (second edition). CMS Books in Mathematics 3. Springer, New York. 2, 11
16. Michal Černý, Miroslav Rada, Jaromír Antoch, Milan Hladík (2022). A class of optimization problems motivated by rank estimators in robust regression. *Optimization*, 71(8), 2241–2271. [doi]. 14
17. X. Chen, S. Xiang (2011). Computation of generalized differentials in nonlinear complementarity problems. *Computational Optimization and Applications*, 50, 403–423. [doi]. 4, 5, 15, 22, 23
18. Vaček Chvátal (1983). *Linear Programming*. W.H. Freeman and Company, New York. 25, 28
19. F.H. Clarke (1983). *Optimization and Nonsmooth Analysis*. John Wiley & Sons, New York. Reprinted in 1990 by SIAM, Classics in Applied Mathematics 5 [doi]. 2, 3, 4
20. Kenneth L. Clarkson, Peter W. Shor (1995). Applications of random sampling in computational geometry, II. *Discrete & Computational Geometry*, 4, 387–421. [doi]. 18
21. R.W. Cottle, G.B. Dantzig (1970). A generalization of the linear complementarity problem. *Journal of Combinatorial Theory*, 8(1), 79–90. 2
22. R.W. Cottle, J.-S. Pang, R.E. Stone (2009). *The Linear Complementarity Problem*. Classics in Applied Mathematics 60. SIAM, Philadelphia, PA, USA. [doi]. 2
23. Thomas M. Cover, Joy A. Thomas (2006). *Elements of information theory* (second edition). Wiley-Interscience [John Wiley & Sons], Hoboken, NJ. 27
24. T. De Luca, F. Facchinei, C. Kanzow (2000). A theoretical and numerical comparison of some semismooth algorithms for complementarity problems. *Computational Optimization and Applications*, 16, 173–205. [doi]. 4
25. Jean-Pierre Dussault, M. Frappier, J.Ch. Gilbert (2019). A lower bound on the iterative complexity of the Harker and Pang globalization technique of the Newton-min algorithm for solving the linear complementarity problem. *EURO Journal on Computational Optimization*, 7(4), 359–380. [doi]. 2
26. Jean-Pierre Dussault, M. Frappier, J.Ch. Gilbert (2019). Polyhedral Newton-min algorithms for complementarity problems. Technical report. [hal-02306526, document]. 2
27. Jean-Pierre Dussault, J.Ch. Gilbert (2022). Exact computation of an error bound for the balanced linear complementarity problem with unique solution. *Mathematical Programming*. [doi]. 2
28. Jean-Pierre Dussault, J.Ch. Gilbert, B. Plaquevent-Jourdain (2023). On the B-differential of the componentwise minimum of two affine vector functions – The full report. Research report. [hal-03872711]. 3, 4, 11, 15, 16, 18, 19, 22, 29
29. Jean-Pierre Dussault, J.Ch. Gilbert, B. Plaquevent-Jourdain (2023). Arrangement of hyperplanes in the space by a dual approach. Research report (in preparation). 37
30. Jean-Pierre Dussault, J.Ch. Gilbert, B. Plaquevent-Jourdain (2023). Partial description of the B-differential of the componentwise minimum of two vector functions by linearization. Research report (in preparation). 15, 22, 23, 37
31. H. Edelsbrunner, J. O'Rourke, R. Seidel (1986). Constructing arrangements of lines and hyperplanes with applications. *SIAM Journal on Control*, 15(2), 341–363. [doi]. 14, 24
32. Herbert Edelsbrunner (1987). *Algorithms in Combinatorial Geometry*, volume 10 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, Berlin. [doi]. 14, 18
33. F. Facchinei, J.-S. Pang (2003). *Finite-Dimensional Variational Inequalities and Complementarity Problems* (two volumes). Springer Series in Operations Research. Springer. 2, 3
34. J.Ch. Gilbert (2021). *Fragments d'Optimisation Différentiable – Théorie et Algorithmes*. Lecture Notes (in French) of courses given at ENSTA and at Paris-Saclay University, Saclay, France. [hal-03347060, pdf]. 27, 28

35. J.Ch. Gilbert (2022). *Selected Topics on Continuous Optimization – Version* 2. Lecture notes of the Master-2 "Optmization" at the University Paris-Saclay. [hal]. 25
36. P. Gordan (1873). Über die Auflösung linearer Gleichungen mit reellen Coefficienten. *Mathematische Annalen*, 6, 23–28. 8
37. Rick Greer (1984). *Trees and Hills: Methodology for Maximizing Functions of Systems of Linear Relations*. Mathematical Studies 96, Annals of Discrete Mathematics 22. North-Holland. 11
38. Branko Grünbaum (1972). *Arrangements and Spreads*. Conference Board of the Mathematical Sciences Regional Conference Series in Mathematics 10. AMS, Providence, RI. 14, 18
39. Rohit Gurjar, Nisheeth K. Vishnoi (2019). On the number of circuits in regular matroids (with connections to lattices and codes). In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 861–880. SIAM, Philadelphia, PA. [doi]. 33
40. Dan Halperin, Micha Sharir (2018). Arrangements. In Jacob E. Goodman, Joseph O'Rourke, Csaba D. Tóth (editors), *Handbook of Discrete and Computational Geometry* (third edition), Discrete Mathematics its Applications, pages 723–762. CRC Press - Taylor & Francis Group. 14
41. J.-B. Hiriart-Urruty, C. Lemaréchal (2001). *Fundamentals of Convex Analysis*. Springer. 2
42. A.F. Izmailov, M.V. Solodov (2014). *Newton-Type Methods for Optimization and Variational Problems*. Springer Series in Operations Research and Financial Engineering. Springer. [doi]. 2
43. C. Kanzow, M. Fukushima (1998). Solving box constrained variational inequalities by using the natural residual with D-gap function globalization. *Operations Research Letters*, 23(1-2), 45–51. [doi]. 4
44. M. Kojima, S. Shindo (1986). Extension of Newton and quasi-Newton methods to systems of PC^1 equations. *Journal of Operations Research Society of Japan*, 29, 352–375. [doi]. 2
45. E. Marchand, T. Müller, P. Knabner (2012). Fully coupled generalised hybrid-mixed finite element approximation of two-phase two-component flow in porous media. Part II: numerical scheme and numerical results. *Computational Geosciences*, 16(3), 691–708. [doi]. 2
46. E. Marchand, T. Müller, P. Knabner (2013). Fully coupled generalised hybrid-mixed finite element approximation of two-phase two-component flow in porous media. Part I: formulation and properties of the mathematical model. *Computational Geosciences*, 17(2), 431–442. [doi]. 2
47. K.G. Murty (1988). *Linear Complementarity, Linear and Nonlinear Programming* (Internet edition, prepared by Feng-Tien Yu, 1997). Heldermann Verlag, Berlin. 2
48. James Oxley (2011). *Matroid theory* (second edition). Oxford Graduate Texts in Mathematics 21. Oxford University Press, Oxford. [doi]. 9
49. J.-S. Pang (1990). Newton's method for B-differentiable equations. *Mathematics of Operations Research*, 15, 311–341. [doi]. 2
50. J.-S. Pang (1991). A B-differentiable equation-based, globally and locally quadratically convergent algorithm for nonlinear programs, complementarity and variational inequality problems. *Mathematical Programming*, 51(1-3), 101–131. [doi]. 2
51. J.-S. Pang (1995). Complementarity problems. In R. Horst, P.M. Pardalos (editors), *Handbook of Global Optimization*, volume 2 of *Nonconvex Optimization and Its Applications*, pages 271–338. Kluwer, Dordrecht. [doi]. 2
52. Liqun Qi (1993). Convergence analysis of some algorithms for solving nonsmooth equations. *Mathematics of Operations Research*, 18, 227–244. [doi]. 2, 4, 23
53. Liqun Qi, Jie Sun (1993). A nonsmooth version of Newton's method. *Mathematical Programming*, 58, 353–367. [doi]. 2
54. Miroslav Rada, Michal Černý (2018). A new algorithm for enumeration of cells of hyperplane arrangements and a comparison with Avis and Fukuda's reverse search. *SIAM Journal on Discrete Mathematics*, 32(1), 455–473. [doi]. 3, 24, 25, 32, 34, 37, 38
55. H. Rademacher (1919). Über partielle und totale differenzierbarkeit. *I. Math. Ann.*, 89, 340–359. 2
56. Samuel Roberts (1887/88). 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. *Proc. London Math. Soc.*, 19, 405–422. [doi]. 14, 18
57. R.T. Rockafellar (1970). *Convex Analysis*. Princeton Mathematics Ser. 28. Princeton University Press, Princeton, New Jersey. 2, 13
58. L. Schläfli (1950). Theorie der vielfachen Kontinuität (in german). In *Gesammelte mathematische Abhandlugen,* Band 1, pages 168–387. Springer, Basel. [doi]. 19
59. Nora Helena Sleumer (1998). Output-sensitive cell enumeration in hyperplane arrangements. In *Algorithm theory-SWAT'98 (Stockholm)*, Lecture Notes in Comput. Sci. 1432, pages 300–309. Springer, Berlin. [doi]. 14, 24
60. Nora Helena Sleumer (2000). *Hyperplane arrangements – Construction, visualization and application*. PhD Thesis, Swiss Federal Institute of Technology, Zurich, Switzerland. [doi]. 14
61. Marek J. Śmietánski (2019). On a new exponential iterative method for solving nonsmooth equations. *Numerical Linear Algebra with Applications*, 25. [doi]. 2

62. R.P. Stanley (2007). An introduction to hyperplane arrangements. In Ezra Miller, Victor Reiner, Bernd Sturmfels (editors), *Geometric Combinatorics*, pages 389–496. [doi]. 14, 18
63. J. Steiner (1826). Einige Gesetze über die Theilung der Ebene und des Raumes. *Journal für die Reine und Angewandte Mathematik*, 1, 349–364. [doi]. 14, 18
64. R. Sznajder, M.S. Gowda (1994). The generalized order linear complementarity problem. *SIAM Journal on Matrix Analysis and Applications*, 15(3), 779–795. [doi]. 2
65. Michel Las Vergnas (1975). Matroïdes orientables. *C. R. Acad. Sci. Paris Série A-B*, 280, A61–A64. 18
66. Robert O. Winder (1966). Partitions of N-space by hyperplanes. *SIAM Journal on Applied Mathematics*, 14(4), 811–818. [doi]. 5, 18, 19
67. Thomas Zaslavsky (1975). Facing up to arrangements: face-count formulas for partitions of space by hyperplanes. *Memoirs of the American Mathematical Society*, Volume 1, Issue 1, Number 154. [doi]. 12, 18
68. C. Zhang, X. Chen, N. Xiu (2009). Global error bounds for the extended vertical LCP. *Computational Optimization and Applications*, 42(3), 335–352. [doi]. 2