Reflections on generating (disjunctive) cuts

Florent Cadoux · Claude Lemaréchal

Received: 24 April 2012 / Accepted: 21 December 2012 / Published online: 22 January 2013
© Springer-Verlag Berlin Heidelberg and EURO - The Association of European Operational Research Societies 2013

Abstract We consider the problem of constructing generic supporting hyperplanes to be used within a (branch and) bound-and-cut algorithm. We review some of the properties that these hyperplanes should enjoy. It is generally admitted that they should cut a parasitic point generated during the process. We argue that the advantages of restricting supports to cuts might be over-balanced by their drawbacks. Said otherwise, the cutting paradigm entails the reverse polar, which may not have definite advantages over the ordinary polar (associated with supports). In fact, we believe that moderate attention should be paid to the current relaxed solution, which plays only a temporary role in the process. We propose approaches to constructing supports, in such a way that the cutting property comes out as a by-product. In particular, squeezing the objective function is most desirable; but cuts merely based on the objective function, being parallel to each other all along the iterations, are generally recognized as inefficient. We propose a mechanism to revive the idea.

Keywords Cutting planes · Separation · Polarity

Mathematics Subject Classification (2010) 52A41 · 90C11

Framework

We are interested in the following general problem: a polyhedron $P \subset \mathbb{R}^n$ is given more or less explicitly and we want to generate valid inequalities for it. Just to simplify our development, we assume that P is bounded and full-dimensional.

F. Cadoux · C. Lemaréchal (✉)
Inria, 655 Avenue de l'Europe, Montbonnot, 38334 Saint Ismier, France
e-mail: claude.lemarechal@inria.fr

F. Cadoux
e-mail: florent.cadoux@inria.fr

Fig. 1 Constructing a disjunction

Example 1.1 (disjunction) Our motivating example is when P is the closed convex hull of two polyhedra: $P = \overline{\text{conv}}(P_0 \cup P_1)$. Even more specifically, we may have

$$P_i := \left\{ x \in \mathbb{R}^n : Ax \leq a, \pi_i^\top x \leq \beta_i \right\}, \quad i = 0, 1. \quad (1.1)$$

The notation in (1.1) just says that P_0 and P_1 differ by exactly one constraint. This occurs for example when solving an original problem of the form

$$\max b^\top x, \quad x \in H, \quad (1.2)$$

where $H \subset \mathbb{R}^n$ is a “horrible” set. During a process involving branching, bounding and cutting, we have relaxed H to some manageable $\bar{H}$, resulting in some $\bar{x} \in \bar{H} \setminus \overline{\text{conv}}(H)$; see Fig. 1a. Then we have defined the tighter relaxation P of (1.1), sandwiched between H and $\bar{H}$, with $\bar{x} \notin P$.

The $\overline{\text{conv}}(P_0 \cup P_1)$ of Fig. 1b might be obtained when H is some integral polyhedron involving constraints $Ax \leq a$; relaxing the integrality constraints gives $\bar{H}$. Some non-integral coordinate (say the i th) of $\bar{x}$ define a disjunction as in (1.1), with

$$\pi_0 = e_i, \quad \beta_0 = \lfloor \bar{x}_i \rfloor, \quad \pi_1 = -e_i, \quad \beta_1 = -\lceil \bar{x}_i \rceil$$

(e_i being the i th basis vector).

Disjunctive programming goes back to Balas (1979). Several disjunctions may also be considered. One may also conceive several points $\bar{x}$, coming from several branches in the B&B tree. The same idea will remain, things will just be more involved; see Perregaard and Balas (2001).

In a way, (1.1) defines “more or less explicitly” a polyhedron P . An explicit description of it is expensive, so one wishes to select an appropriate subset of inequalities defining it: “we want to generate valid inequalities” for P . $\square$

An affine inequality in $\mathbb{R}^n$ is characterized by $(d, r) \in \mathbb{R}^n \times \mathbb{R}$, which defines the hyperplane of equation $d^\top x = r$. Note that $(-d, -r)$ defines the same hyperplane; fixing the sign once and for all, we will say that (d, r) is a valid inequality for P when

$$\forall x \in P, \quad d^\top x \leq r \quad \text{or equivalently} \quad \max_{x \in P} d^\top x \leq r. \quad (1.3)$$

Fig. 2 Tight and loose valid inequalities

Our very first question is what are the “good” (d, r) ’s? Then, we will be faced with a second question: depending on our knowledge of P , how can one construct them?

Various arguments can be invoked to define good inequalities.

- (i) They will *touch* P (see Fig. 2): some $x \in P$ does satisfy $d^\top x = r$; this can also be expressed as $\max_{x \in P} d^\top x = r$.
- (ii) They will touch P on a *facet* (see Fig. 3): the argmax of $d^\top x$ has dimension $n - 1$ (knowing that P has dimension n).
- (iii) In the situation of Example 1.1, they will separate $\bar{x}$ from P : $d^\top \bar{x} > r$, we say that d is a *cut*.

These three requirements are commonly admitted for good inequalities. However, generating all such inequalities is not reasonable: they are too many if P has many facets and eventually many of them will be useless. Besides, we will see that (ii) and (iii) do not fully fit together.

In the framework of Example 1.1, several additional criteria may be required for good cuts:

- (iv) With respect to (iii), they will be *deep*: a cut will separate $\bar{x}$ and P well.
- (v) They will *squeeze down* the objective function: a cut will decrease the value $\max_P b^\top x$ well.

Of these five properties, (i) is the least disputable, and most specialists agree that (ii) is hardly less important. We note that (v) seems natural, but the way to let it play its role is not clear; results have been disappointing so far. As for (iii), it is blatantly necessary: what will be the use of the next bounding step if it does not tighten the current relaxation? This explains that the terminology *cutting* is universally used for

Fig. 3 Facet-exposing supports

the present problem, which is more generally to develop valid inequalities. Cutting $\bar{x}$ entails the concept of *reverse polar*, as opposed to the ordinary polar of convex analysis, associated with general supporting hyperplanes. Yet, we will see that the role of $\bar{x}$ should not be over-emphasized. This argument is also valid for (iv), which is a follower to (iii).

In the present notes, we will deal with criteria (i) and (ii) first: they appear as more intrinsic than the others. Then, we will consider (iv), (v) as additional criteria, which can be used to select the inequalities coming from (i), (ii)—and yield (iii) as a bonus. Our study will suggest that

- the reverse polar might not be a good object, just because it is inherently unbounded,
- the concept of deepest cut might not be too good either; among other things, it relies upon a normalization, which has to be arbitrary,
- “good” cuts should somehow take the objective function into account,
- after all, the cutting property is not fundamental; its importance is based on “myopic” arguments, giving too much importance to $\bar{x}$.

These notes are a follow-up to Cadoux (2008) and Cornuéjols and Lemaréchal (2005). Convex analysis is omnipresent; for an introduction of basic concepts and results, we suggest Hiriart-Urruty and Lemaréchal (2001), especially its Chap. C.

Valid and tight inequalities

For given $d \in \mathbb{R}^n$, criterion (i) fixes the righthand side r in (1.3) to its smallest possible value, namely $r = \max_{x \in P} d^\top x$. The special value thus defined is a function of d , a function which also depends on P .

Definition 2.1 (*Support function*) Given a set $P \subset \mathbb{R}^n$ and a vector $d \in \mathbb{R}^n$, the number

$$r_d := \sigma_P(d) := \sup_{x \in P} d^\top x \quad (2.1)$$

is called the (value at d of the) support function of P . $\square$

The support function is a fundamental object of convex analysis. An inequality $d^\top x \leq r$ is valid if $r \geq r_d$ and it is tight if $r = r_d$; see Fig. 2 (in which P is defined by very many inequalities!).

From now on, we will only consider tight inequalities, and we will call them supports.

Definition 2.2 (*Support*) A support (of P) is a nonzero vector $d \in \mathbb{R}^n$, which defines via (2.1) the hyperplane of equation $d^\top x = r_d$; it also defines the inequality $d^\top x \leq r_d$, which is valid and tight for P . Needless to say, d can be a support only if $\sigma_P(d)$ is a finite number; see Remark 2.3 below. Conversely, any d such that $\sigma_P(d) < +\infty$ is a (“potential”) support. $\square$

Thus, our valid inequalities have no longer the form (d, r) but merely d , knowing that $r = r_d$ is the corresponding righthand side.

Remark 2.3 Assuming boundedness of P in (2.1) guarantees that r_d is always a finite number: any $d \in \mathbb{R}^n$ is a support. If P were unbounded, this would no longer be true. Technically, a support d has to lie in the *polar* of the *recession cone* of P ; this means:

$$|x| \rightarrow +\infty \text{ and } x \in P \implies x^\top d \leq 0,$$

otherwise d cannot be a support.

Also, a support is defined up to multiplicative constant: if d is a support and $t > 0$, then $d' = td$ is again a support. A support is actually a *direction* in $\mathbb{R}^n$, which could for example be imposed to have norm 1. In fact, the support function is positively homogeneous: $\sigma_P(td) = t\sigma_P(d)$.

Incidentally, selecting the $\leq$ sign in (1.3) keeps us in the scope of most textbooks on the subject: the support function is also *convex* in d ; altogether, it is *sublinear*. $\square$

Admitting that a given $d \in \mathbb{R}^n$ is a support, to define the corresponding tight inequality amounts to computing r_d of (2.1), i.e., to solving an LP over P . This is easy in the disjunctive example (1.1): maximize $d^\top x$ over each P_i , then r_d is the higher of the two values.

Exposing facets

The argmax in (2.1) is called the face *exposed* by d . Criterion (ii) requires it to be actually an exposed facet: good supports are facet-exposing. Even though a facet is a subset of P , we will often use the same word to denote a nonzero *vector* in (the dual of) $\mathbb{R}^n$, namely the direction exposing that facet. In this sense, “facet” stands for the full wording “facet-exposing direction”, rather pedantic.

Figure 3 (where P now looks like an actual polyhedron!) illustrates the set of facets, seen from this viewpoint. It appears as a finite set (of directions) in $\mathbb{R}^n$. When the constraints defining P are available, say $Ax \leq a$, these directions are just given by these constraints—at least those that are non-redundant: each of them is a row of A . Otherwise, they may be more or less difficult to describe.

The following result is well-known.

Theorem 3.1 *Let Q be a full-dimensional polyhedron containing the origin as an interior point. Then the direction $d \in \mathbb{R}^n$ exposes a facet of Q if and only if it is (collinear to) an extreme point of the set*

$$Q^\circ := \left\{ d \in \mathbb{R}^n : \max_{x \in Q} d^\top x \leq 1 \right\}, \quad (3.1)$$

which is (closed and) bounded. $\square$

As already mentioned, the concept of support (2.1) is fundamental; it has appeared in Definition 2.2 and it appears again in the above definition of Q° (a polyhedron, called the *polar* of Q). It results directly from (2.1) that the property $0 \in Q$ implies that σ_Q is nonnegative, and even positive everywhere (remember $d \neq 0$) if 0 is actually interior.

The “facets” d defined through Q° are normalized through their support function rather than their length: the extreme points of Q° have $\sigma_Q(d) = 1$.

Remark 3.2 (Disjunction) In connection with Example 1.1, let $Q = Q_0 \cup Q_1$ be a disjunctive polyhedron. Then, it is clear enough that $Q^\circ = Q_0^\circ \cap Q_1^\circ$: the whole business to describe Q° is to describe the constraints defining each Q_i° . Stacking them together gives $(Q_0 \cup Q_1)^\circ$. This is mainly a theoretical suggestion, though. Not mentioning the complexity of describing each Q_i° , the total number of facets might be just too large. $\square$

Now, Fig. 3 reveals the obvious property that facets remain the same after a translation of P . Characterizing the facets of P can therefore be done with a change of origin: we take a point ω in the interior of P , so that 0 is an interior point of the translated polyhedron $Q := P - \omega$; and Q has the same facets as P . The support function of Q is easy to characterize via the change of variable $y = x - \omega$:

$$\sigma_Q(d) := \max_{y \in Q} d^\top y = \max_{x \in P} d^\top x - d^\top \omega = \sigma_P(d) - d^\top \omega,$$

a formula which can be condensed in $\sigma_{P-\omega}(d) = \sigma_P(d) - d^\top \omega$.

This translation has a simple impact on Theorem 3.1.

Corollary 3.3 *Let ω be an interior point of P . The facets of P are (collinear to) the extreme points of the polyhedron*

$$(P - \omega)^\circ = \left\{ d \in \mathbb{R}^n : d^\top (x - \omega) \leq 1 \text{ for all } x \in P \right\}, \quad (3.2)$$

which is bounded.¹ $\square$

Note again that, considered as *directions*, these extreme points do not depend on ω . However, their length does: when ω approaches the boundary of P , at least one of them goes to infinity.

Now, a natural idea to generate extreme points of a set is to minimize a linear function over it. So we can summarize this section: to generate facets of P ,

- take some ω in the interior of P (assumed nonempty),
- select $\ell \in \mathbb{R}^n$,
- minimize $\ell^\top d$ over $(P - \omega)^\circ$.

Remark 3.4 Because $(P - \omega)^\circ$ is a bounded polyhedron, the last step above is a linear program which always has an optimal solution. Whether this linear program is solvable depends on our knowledge of $(P - \omega)^\circ$ which, in view of (3.1), depends in turn on our knowledge of $\sigma_{P-\omega}$, i.e., of σ_P .

The next section develops this point.

¹ Polyhedrality goes of course with polyhedrality of P : in (3.2), the list of constraints can be restricted to the extreme points. Boundedness comes from Theorem 3.1 and can easily be seen: as already observed, $\sigma_P(d) - d^\top \omega$ is positive—hence > 1 for large d .

Decomposing a support into facets

Orientation The previous two sections have studied arguments (i) and (ii). Here, we postpone the study of (iii) and we address the following question: it will be most likely that a support satisfying (iv) or (v) does not expose a facet, so the present section aims at reconciling these last two properties with (ii).

Thus, we are given a “desired” support $\hat{d}$, say satisfying (iv) or (v), which is not a facet. The idea will be to express it as a positive combination of facets, i.e., of extreme points in $(P - \omega)^\circ$.

Recall that $\hat{d}$ is just a direction. It is a good idea to normalize it in such a way that it lies on the boundary of the bounded polyhedron $(P - \omega)^\circ$ (see Remark 4.1 and Theorem 6.2 below): it can be expressed as a convex combination of extreme points; say, for some k

$$\hat{d} = \sum_{i=1}^k \alpha_i d_i, \quad d_i \text{ extreme in } (P - \omega)^\circ, \quad \alpha_i \geq 0, \quad \sum_{i=1}^k \alpha_i = 1. \quad (4.1)$$

In what follows, we will use the notation

$$r_i := \sigma_P(d_i) = 1 + d_i^\top \omega, \quad \hat{\sigma} := \sigma_P(\hat{d}) = 1 + \hat{d}^\top \omega,$$

where the equalities come from the fact that the d_i ’s and $\hat{d}$ lie on the boundary of the polar.

From Theorem 3.1, the inequalities

$$d_i^\top x \leq 1 + d_i^\top \omega, \quad i = 1, \dots, k$$

are valid for P and expose k facets of P ; by (4.1), they imply the inequality $\hat{d}^\top x \leq \hat{\sigma}$. Figure 4 illustrates this trick with the P of Fig. 3; the position of ω in its first polyhedron (P) implies long d_1 and d_2 in its second polyhedron, while its third polyhedron displays the three supporting hyperplanes $d_1, d_2, \hat{d}$.

Fig. 4 Decomposing an arbitrary support

Remark 4.1 Normalizing $\hat{d}$ as above is possible when P (i.e., $P - \omega$) is bounded—as is assumed here: then 0 lies in the interior of $(P - \omega)^\circ$. This boundedness assumption is useless, though: because $\hat{d}$ is a support, the support function of P is finite at $\hat{d}$: $\hat{\sigma} := \sigma_P(\hat{d}) < +\infty$; and, once again, it is strictly bigger than $\hat{d}^\top \omega$ because ω is an interior point. Thus, we can always divide $\hat{d}$ by $\hat{\sigma} - \hat{d}^\top \omega > 0$. This, incidentally, places $\hat{d}$ on the boundary of $(P - \omega)^\circ$. $\square$

For an actual construction of the d_i 's, we propose an algorithm (Cadoux 2008, Alg.3.1), whose working horse is linear programming over $(P - \omega)^\circ$ (remember Remark 3.4). So we assume that, given $\ell \in \mathbb{R}^n$, an *oracle* is available to solve the problem

$$\min \ell^\top d, \quad d^\top (x - \omega) \leq 1 \text{ for all } x \in P. \quad (4.2)$$

At the current iteration of the algorithm, k extreme points have already been generated; call D_k their convex hull. Does $\hat{d}$ lie in D_k ? To answer this question, a possible idea is to solve the program with k variables

$$\min \left\| \sum_{i=1}^k \alpha_i d_i - \hat{d} \right\|, \quad \alpha_i \geq 0, \sum_{i=1}^k \alpha_i = 1, \quad (4.3)$$

where $\|\cdot\|$ is some norm. If the optimal value is 0 we are done. Otherwise, we need to generate a $(k+1)$ st extreme point and append it to the list. For this, we use a hyperplane $(\ell, r) \in \mathbb{R}^n \times \mathbb{R}$ separating the two closed convex set $\hat{d}$ and D_k . In fact, minimizing $\ell^\top d$ over $(P - \omega)^\circ$ will provide an extreme point d_{k+1} and we will have

$$\begin{aligned} \ell^\top d_{k+1} &\leq \ell^\top \hat{d} && \text{[definition of } d_{k+1}] \\ < r &\leq \ell^\top d_i, \quad i = 1, \dots, k. && \text{[separation of } \hat{d} \text{ and } D_k] \end{aligned}$$

The d_{k+1} thus exhibited “explains” why D_k is not big enough and can conveniently be used to enlarge D_k .

The separator ℓ is actually a by-product of (4.3) and can be obtained through the optimality conditions.² See Fig. 5, where B denotes the smallest possible ball associated to $\|\cdot\|$, centered at $\hat{d}$ and intersecting D_k . If the Euclidean norm is chosen in (4.3) (resulting in a rather simple quadratic program), and denoting by $d_k^* = \sum_i \alpha_i^* d_i$ the optimal solution, then $\ell = \hat{d} - d_k^*$.

In summary, the facet-generator works as follows.

Algorithm 4.2 Are given: $\hat{d} \in (P - \omega)^\circ$, an LP-solver for the oracle (4.2), and a norm for the master problem (4.3) (the ordinary Euclidean norm is suggested).

STEP 0 Compute some extreme point $d_1 \in (P - \omega)^\circ$ (for example, solve (4.2) with $\ell = \hat{d}$; cf. Remark 4.3). Set $k = 1$.

² This is well-known in nonlinear programming theory. Indeed, checking whether $\hat{d}$ lies in D_k amounts to showing that no separator exists between $\hat{d}$ and D_k .

The diagram shows an ellipse representing a set. Inside the ellipse is a point labeled $\hat{d}$. A dashed curve, labeled B , is drawn within the ellipse. A solid line segment, labeled D_k , connects three points on the ellipse: d_1 , d_2 , and d_k . A point d_{k+1} is marked on the ellipse. A dashed line segment connects d_{k+1} to $\hat{d}$. A line segment labeled ℓ is shown outside the ellipse, extending from the top left.

STEP 2 If $d_k^* = \hat{d}$ stop. Otherwise, obtain ℓ such that

$$\ell^\top \hat{d} < \ell^\top d_i, \quad i = 1, \dots, k.$$

Let us summarize: we concluded Sect. 3 by observing that the key ingredient to generate facets of P was linear programming over $(P - \omega)^\circ$. Here, we have given a *constructive* mechanism, using the following ingredients:

- The oracle (4.2), as expected.
- An interior point ω in P ; the question of its choice is left aside in these notes. Its role may not be fundamental, its main merit is to guarantee well-posedness of (4.2).
- A desired support $\hat{d}$, whose choice will be addressed in the following sections.
- The master (4.3); no matter which norm it uses, observe that its complexity is driven by the number k of iterations in Algorithm 4.2 and not by the dimension n of the original problem.

The stopping criterion in Step 2 is of course too crude: the computation of d_k^* is subject to roundoff errors. Some tolerance is needed, see Sect. 7.

The particular $\hat{d}$ -length mentioned in Remark 4.1 is indeed special; in the next result, we set again $r_i := r_{d_i} = \sigma_P(d_i)$ and $\hat{\sigma} := r_{\hat{d}}$.

Fig. 6 Decomposing a short $\hat{d}$

Proposition 4.4 Let $\hat{d}$ in (4.1) be such that $\sigma_{P-\omega}(\hat{d}) = 1$, hence $\hat{\sigma} = 1 + \hat{d}^\top \omega$, and call $I := \{i : \alpha_i > 0\}$ the set of active extreme points d_i . Then, $r_i = 1 + d_i^\top \omega$ for each $i \in I$.

Besides, any $x \in P$ exposed by $\hat{d}$ ($\hat{d}^\top x = \hat{\sigma}$) is also exposed by the d_i 's ($d_i^\top x = r_i$, for each $i \in I$).

Proof By definition of $(P - \omega)^\circ$, we have

$$r_i \leq 1 + d_i^\top \omega \quad \text{for each } i.$$

Suppose strict inequality held for some $i \in I$. Then, we would obtain by convex combination

$$\sum_{i=1}^k \alpha_i r_i < 1 + \hat{d}^\top \omega = \hat{\sigma};$$

but the support function $d \mapsto \sigma$ is convex, so this would be a contradiction.

Now take x as stated: by definition of the support function $d_i^\top x \leq r_i$. Again, strict inequality for some $i \in I$ would result in the same contradiction. $\square$

For given d , the above normalized $\hat{d} = td$ is obtained by solving for t the equation $\sigma_{P-\omega}(td) = 1$:

$$\hat{d} = \frac{d}{\sigma_{P-\omega}(d)} = \frac{d}{\sigma_P(d) - d^\top \omega}. \quad (4.4)$$

Proposition 4.4 is explained by Fig. 6, whose left part reproduces the second polyhedron of Fig. 4, while its right part illustrates the decomposition of a shorter $\hat{d}$ ($\sigma_{P-\omega}(\hat{d}) < 1$). A “parasitic” extreme point is then needed, for example d_3 . Thus, our special normalization places $\hat{d}$ and all of the d_i 's in the same hyperplane of equation $x^\top d = x^\top \hat{d}$, which has dimension $n - 1$: from Caratheodory's theorem, n extreme points suffice in (4.1). Besides, these extreme points have a nice interpretation in terms of the original problem (1.2), see Theorem 6.2 below.

Cuts and deepest cuts revisited

A support $d \in \mathbb{R}^n$ separates $\bar{x}$ from P if and only if $d^\top \bar{x} > r_d$ in (2.1). We will then say that d is a *cut*. On Fig. 2, d is a cut if $\bar{x}$ lies somewhere in the South-East part of the picture.

Fig. 7 Non-existence of facet-exposing cuts

A useful object characterizing cuts is the so-called *reverse polar*

$$(P - \bar{x})^- := \{d \in \mathbb{R}^n : d^\top (x - \bar{x}) \leq -1 \text{ for all } x \in P\};$$

to be compared with (3.2): indeed, the facets separating P from $\bar{x}$ are (collinear to) the extreme points of $(P - \bar{x})^-$ [which are in turn collinear to those of $(P - \omega)^\circ$]. Then, if $\hat{d}$ is a desired cut as in the previous section, Algorithm 4.2 can be used to decompose it into extreme points of $(P - \bar{x})^-$, instead of $(P - \omega)^\circ$. In practice, this means to decompose a cut into facet-exposing cuts. The only thing to change is the oracle (4.2), which becomes

$$\min \ell^\top d, \quad d^\top (x - \bar{x}) \leq -1 \text{ for all } x \in P.$$

However, a difficulty is that the reverse polar is inherently unbounded; as a result:

- during the course of the algorithm, some ℓ may be produced such that the oracle (4.2) has no finite solution,
- and in fact, a cut need not be a convex combination of facet-exposing cuts (in Fig. 7, d_1 is necessary to decompose $\hat{d}$ but is not a cut).

On the other hand, decomposing a desired cut into facet-exposing *supports* need not produce cuts: in Fig. 7, keep the same $\hat{d}$ but raise slightly $\bar{x}$; d_1 is still necessary but is not a cut at all. We conclude that combining (ii) and (iii) is not totally straightforward.

Unboundedness of the reverse polar presents another difficulty, perhaps more fundamental. To select a desired cut, one idea [which can be traced back to [Boyd \(1994\)](#)] is to take a “deepest” one, lying “as far as possible” from $\bar{x}$. Knowing that a cut is a d such that $d^\top \bar{x} - \sigma_P(d)$ is positive, one wishes d making this difference as large as possible. By positive homogeneity, however, this results in a d “at infinity”: a deepest cut should solve

$$\max_{d \in (P - \bar{x})^-} d^\top \bar{x} - \sigma_P(d), \quad (5.1)$$

which has no solution; some normalization is necessary. Choosing an adequate norm is an obscure issue, see [Balas et al. \(1993\)](#), [Bonami and Minoux \(2005\)](#), [Cadoux \(2008\)](#), [Cornuéjols and Lemaréchal \(2005\)](#), [Perregaard \(2003\)](#) and [Perregaard and Balas \(2001\)](#). Experience shows that this norm has a crucial impact on the quality of the cut produced; even worse: Definition 2.2 is usually ignored, a righthand side is also involved, which further changes the result.

When developing the cutting paradigm, one implicitly accepts that, to be useful, a support must be a cut. This idea is highly debatable, though: it puts much emphasis on the current relaxed solution $\bar{x}$, i.e., on one particular step of a process that may include many others. Most probably, future steps will construct cuts that do not separate the current $\bar{x}$ from the future P . In this paper, we put forward the suggestion that no support should be disregarded *a priori*, even if it is not a cut.

When this idea is accepted, calling for the polar to define a deepest cut becomes natural and several approaches come to mind. A first idea is to view the polar itself as a normalizing device and to replace (5.1) by

$$\max_{d \in (P-\omega)^\circ} d^\top \bar{x} - \sigma_P(d).$$

Any optimal solution $\hat{d}$ of this problem lies on the boundary of $(P-\omega)^\circ$ (positive homogeneity of the objective function), i.e., $\sigma_P(\hat{d}) = 1 + \hat{d}^\top \omega$. This suggests a simplification: linearize the nonlinear function σ_P , which results in solving

$$\max_{d \in (P-\omega)^\circ} d^\top (\bar{x} - \omega). \quad (5.2)$$

Proposition 5.1 *Any optimal solution of (5.2) is a cut.*

Proof Call $\hat{d}$ an optimal solution and suppose $\hat{d}^\top (\bar{x} - \omega) \leq 1$. Then

$$d^\top (\bar{x} - \omega) \leq \hat{d}^\top (\bar{x} - \omega) \leq 1, \quad \text{for any } d \in (P - \omega)^\circ.$$

This means that $\bar{x} - \omega$ lies in the polar of $(P - \omega)^\circ$, which is $P - \omega$ itself (a standard result in polarity theory): $\bar{x} - \omega \in P - \omega$, contradiction. We deduce

$$\hat{d}^\top (\bar{x} - \omega) > 1 \geq \hat{d}^\top (y - \omega) \quad \text{for any } y \in P,$$

where the second inequality is the definition of the polar. Thus $\hat{d}^\top \bar{x} > \sigma_P(\hat{d})$. $\square$

In addition to being simpler, the linear program (5.2) will most probably provide directly an extreme point of $(P - \omega)^\circ$, i.e., a facet of P : Algorithm 4.2 becomes useless.

Other proposals can be imagined. For example, let P have the form (1.1) and assume $\omega = 0$ for simplicity. As already mentioned, $(P_0 \cup P_1)^\circ = P_0^\circ \cap P_1^\circ$. Now, by LP duality, to say that $\sigma_{P_i}(d) \leq 1$ in (1.1) is to say that

$$d = A^\top \lambda_i + \mu_i \pi_i \quad \text{for some } \lambda_i \geq 0, \mu_i \geq 0 \text{ such that } a^\top \lambda_i + \beta_i \mu_i \leq 1;$$

besides, a d as above with $\mu_0 = \mu_1 = 0$ cuts nothing and is of no avail. An idea is therefore to solve

$$\begin{cases} \max_{\lambda, \mu} \min\{\mu_0, \mu_1\}, \\ A^\top \lambda_0 + \mu_0 \pi_0 = A^\top \lambda_1 + \mu_1 \pi_1, \\ a^\top \lambda_i + \beta_i \mu_i \leq 1, \quad i = 0, 1, \\ \lambda_i \geq 0, \mu_i \geq 0, \quad i = 0, 1, \end{cases}$$

which is indeed a linear program.

Fig. 8 Squeezing the objective function

In summary, our suggestion is to use the same techniques that have already been proposed, but to replace the reverse polar by the polar. Note, however, that an artificial aspect is still present: we have eliminated the need for a norm but we still need a translation ω .

Squeezing the objective function

Consider Example 1.1, more precisely (1.2). The horrible set H has been relaxed to $\bar{H}$, yielding $\bar{x}$ and the associated optimal value $b^\top \bar{x}$. Introducing an extra constraint defined by d will reduce this value to

$$v(d) := \max \left\{ b^\top x : x \in \bar{H}, d^\top x \leq \sigma_P(d) \right\} \quad (6.1)$$

(so that $v(0) = b^\top \bar{x}$) and it should be desirable to obtain the smallest possible value

$$v_{\min} := \max \left\{ b^\top x : x \in P \right\} = \sigma_P(b). \quad (6.2)$$

In other words, good supports are those that minimize $v(d)$. Figure 8 suggests, and the next result proves, that b itself answers the question.

Theorem 6.1 $v(b) = v_{\min}$.

Proof Observe that $v(d)$ is the value at b of the support function of the feasible set in (6.1). Because this set contains P if d is a support,

$$v(d) \geq \sigma_P(b) = v_{\min} \quad \text{for all support } d,$$

where the equality is (6.2).

On the other hand, take $d = b$ in (6.1):

$$v(b) = \max \left\{ b^\top x : x \in \bar{H}, b^\top x \leq \sigma_P(b) \right\}$$

so the inequality $v(b) \leq \sigma_P(b) = v_{\min}$ is trivial. $\square$

Once again, b need not be a facet of P and must be decomposed, either in supports or in cuts (if the latter is possible, remember Fig. 7). As mentioned in Sect. 5, it seems definitely advisable to accept supports that do not cut $\bar{x}$. This avoids lots of practical difficulties, but there is an even more profound reason to do so.

In fact, normalize b to $\hat{b}$ on the boundary of $(P - \omega)^\circ$ as in (4.4). The aim of Algorithm 4.2 is to write $\hat{b}$ as

$$[\hat{d}] \quad \hat{b} = \sum_{i=1}^k \alpha_i d_i$$

with $\alpha_i \geq 0$ and summing up to 1. This actually amounts to solving (6.2) and is the first step toward the optimality conditions of (1.2); the property that misses for the latter is that the d_i 's should support not only P but also (the convex hull of) H .

Theorem 6.2 *Let d_i be the extreme points produced by Algorithm 4.2 with $\hat{d} = \hat{b}$, together with their righthand sides $r_i := \sigma_P(d_i)$. If there exists $x^* \in P$ such that $d_i^\top x^* = r_i$, $i = 1, \dots, k$, this x^* solves (6.2). Besides, if $x^* \in H$, then it solves (1.2).*

Proof Because each inequality $d_i^\top x \leq r_i$ is a valid inequality for P and a fortiori for H , we have for each $x \in P$

$$d_i^\top x \leq r_i = d_i^\top x^*, \quad i = 1, \dots, k,$$

which implies $\hat{b}^\top x \leq \hat{b}^\top x^*$ by convex combination (and of course $b^\top x \leq b^\top x^*$). This means that x^* solves (6.2) [resp. (1.2)] if it lies in P [resp. H]. $\square$

This observation reveals the interest of placing $\hat{d}$ (i.e., $\hat{b}$) on the boundary of $(P - \omega)^\circ$, as in Remark 4.1. In fact, a point x described by Proposition 4.4 is a good candidate to being an x^* of Theorem 6.2: it belongs to P and activates the constraints $d_i^\top x = r_i$; if it also lies in $H \subset P$, we are done.

Note in passing that x^* is related to the multipliers coming from (4.2), which therefore should perhaps be replaced by its dual. These considerations might also shed some light on the initialization question (Remark 4.3). Such questions are worth meditating. Along the lines of our comments in Sect. 5, note also that the role of $\bar{x}$ may be deemed rather anecdotal, as compared with x^* ; especially as $\bar{x}$ has been found somehow “by chance”, via a particular relaxation of H . This is true for the concept of cut, a fortiori deepest!

The choice $\hat{d} = \hat{b}$, highly attractive, has already been known for quite long; but it does not have a good reputation. In fact, taking successive parallel cuts all along the B&C process, with tighter and tighter righthand sides, appears as very poor: what is needed is to delineate the convex hull of H (in the neighborhood of hopefully reasonable approximations $\bar{x}$). It appears as really crucial for efficiency to decompose $\hat{d} = \hat{b}$ into facets of the successive P 's.

An idea suggested by this section is therefore: to generate good supports, take b as the desired cut (no other cut can squeeze the objective better), normalize it to $\hat{b}$ via (4.4), and decompose $\hat{b}$ by Algorithm 4.2 into extreme points of $(P - \omega)^\circ$ —perhaps

not all those necessary to obtain (4.1) (if k reveals too large), but at least a “reasonable” subset of them. Unfortunately, this is not the panacea yet. In fact, b or $\hat{b}$ does define a cut when $v(b) < v(0)$; but despite Theorem 6.1, this latter property need not hold when (6.1) with $d = 0$ has several solutions, in addition to $\bar{x}$. This could happen for example with an instance as in Fig. 7 with $b = d_1$. Then, trying to describe P with the help of b is disastrous: no new facet will be generated and the objective function will stay the same. When using this b -technique, it is therefore strongly recommended to solve (6.2) first and to use something else (say, a deepest cut) if $v(b) = v(0)$. In this case, one *has to* accept a stagnation in the objective function (Theorem 6.1).

Remark 6.3 The optimal basis having produced $\bar{x}$ can fruitfully initialize Algorithm 4.2 with $k > 1$ facets. This will skip the re-computation of a bunch of supports that are useless anyway (they cut nothing from $\bar{H}$). Remembering Remark 4.3, we see that the overall bound-and-cut process can be made fully incremental.

Besides, one may wish to stop the algorithm as soon as a cut has been found, which should occur very early.

Another observation is that computing $v(b)$ is easy in case (1.1): optimize over each P_i and take the better value. $\square$

To conclude this section, let us mention that the supports with a 0-coefficient on π_i in the notation (1.1) are redundant. Any such a support generated by the algorithm can subsequently be dropped.

Final considerations

A simple illustration Let us first show on a simple two-dimensional example the essential content of this paper. Start from the split-polyhedron $P_0 \cup P_1$ in the left part of Fig. 9, where the splitting polyhedron P has introduced the additional facets F_3 and F_6 . The polar P° (on the right part of the picture, we assume that $\omega = 0$ is an interior point) has six extreme points d_j , including the two additional ones d_3 and d_6 . The interesting supports are the directions d between d_2 and d_4 , say d collinear to $\alpha d_2 + (1 - \alpha)d_4$ for $\alpha \in [0, 1]$. Those with $\alpha \in]0, 1[$ are cuts and the “ideal” one is d_3 .

Fig. 9 A simple two-dimensional disjunction

Fig. 10 Cutting with the “benefit-row” b

Fig. 11 Decomposing the benefit-row

Figure 10 extracts the interesting area of the picture. Its left part shows a possible objective function b , together with the corresponding (tight) cut F_b . Although imperfect, this cut yields the largest possible decrease of the value $b^\top \bar{x}$ (Theorem 6.1). In the right part of the picture, drawn in the dual space, b is normalized to the value $\hat{b}$ of (4.4).

Finally, Fig. 11 shows the decomposition $\hat{b} = \alpha d_3 + (1 - \alpha)d_4$. It corresponds in the primal (left part of the picture) to introducing F_3 and F_4 . Note that the support d_4 thus introduced is redundant (F_4 is already included in the description of $\bar{H}$); besides, d_4 is not a cut. The important feature is that, on this example, the decomposition has caught d_3 and describes accurately the intersecting portion of P .

Numerical considerations A serious and conclusive implementation of a cut generator based on our ideas would require important numerical work going beyond the scope of this paper. The bulk of the work is of course spent in Algorithm 4.2.

- This algorithm must be fast, to make the cost of one iteration roughly comparable with the cost of generating one cut in the lift-and-project algorithm, with which our method shares many similarities.
- It must also be robust, and this relies upon the algebraic calculations managing the sequential solutions of (4.3); they are well detailed in Wolfe (1976).

The stopping test in Step 2 is also crucial, let us explain how it should be designed. Abstractly, such a test will use some number ε_k tending to 0 and stop the algorithm when ε_k is deemed “small”. This requires a reference $z > 0$, to judge “smallness”

despite scale factors and roundoff errors; here, z will essentially be $v(0) - v(b) = b^\top \bar{x} - \sigma_P(b)$.

A first possible test uses the convex hull D_k of $d_1, \dots, d_k$. It is contained in $(P - \omega)^\circ$, its polar $Q_k := D_k^\circ$ therefore contains $P - \omega$, and it is aimed at describing the portion of P contributing the computation of $v(b)$ in (6.2). Then, call

$$v_k := \sup_{x - \omega \in Q_k} b^\top x = b^\top \omega + \sigma_{Q_k}(b)$$

the current (over-)approximation of $v(b)$, so that $\varepsilon_k := v_k - v(b) \downarrow 0$. Taking a tolerance $\kappa > 0$, the algorithm can be stopped when

$$v_k - v(b) \leq \kappa[v(0) - v(b)]; \quad (7.1)$$

if Remark 6.3 is exploited, v_k lies somewhere in the segment $[v_{\min} = v(b), v(0)]$ of Fig. 8. Note that computing v_k is relatively cheap: it is an LP with a reasonable number k of constraints, and remember Remark 6.3: $\bar{x}$ has been computed by a similar LP. Besides, this LP has an interest in its own, to produce an x_k^* as in Theorem 6.2. Actually, (7.1) is particularly appropriate in our context, as it assesses the quality of the supports defining our approximation of P .

Instead of connecting D_k and $(P - \omega)^\circ$, one may also use convergence to $\hat{b}$ of the projection d_k^* computed in Step 1. A test simpler than (7.1) can be designed as follows. Subtract $b^\top \omega$ from the relation $\bar{x}^\top b > \sigma_P(b)$:

$$(\bar{x} - \omega)^\top b > \sigma_P(b) - \omega^\top b = \sigma_{P-\omega}(p) > 0;$$

divide by $\sigma_{P-\omega}(p)$ and use (4.4):

$$z := (\bar{x} - \omega)^\top \hat{b} - 1 > 0$$

can serve as reference. A really cheap test is then

$$(\bar{x} - \omega)^\top d_k^* \geq (\bar{x} - \omega)^\top \hat{b} - \kappa \left[(\bar{x} - \omega)^\top \hat{b} - 1 \right], \quad (7.2)$$

where $\kappa > 0$ is a small tolerance as before; since $(\bar{x} - \omega)^\top d_k^* \rightarrow (\bar{x} - \omega)^\top \hat{b}$, (7.2) is eventually passed. In effect, we take $\kappa < 1$ (a natural assumption), so that

$$(\bar{x} - \omega)^\top d_k^* > (\bar{x} - \omega)^\top \hat{b} - \left[(\bar{x} - \omega)^\top \hat{b} - 1 \right] = 1.$$

On the other hand, $d_k^* \in D_k \subset (P - \omega)^\circ$; by definition (3.1) of a polar,

$$\sigma_P(d_k^*) - \omega^\top d_k^* \leq 1 < (\bar{x} - \omega)^\top d_k^*, \quad \text{i.e., } \sigma_P(d_k^*) < \bar{x}^\top d_k^*,$$

and this just means that d_k^* is a cut.

Thus, (7.2) guarantees a set of supports which do cut something from $\tilde{H}$. It does not guarantee a d_k^* close to $\hat{b}$; but (7.1) does not either, and this is of little importance anyway, since $\hat{b}$ in itself is a bad cut. Whether (7.1) or (7.2) is any good, as compared to full convergence of Algorithm 4.2, or even to a full description of P , is not clear and will remain so as long as overall convergence of the B&C process is not clarified. Along the same lines, we note that κ might not have to be a very small number, insofar as $\hat{b}$ might not have to be fully decomposed: a few facets might just be as efficient.

Conclusion This paper elaborates on the idea that, in the course of a branch-and-bound-and-cut algorithm, the role of the relaxed solution $\bar{x}$ is temporary; a valid inequality improving the current relaxation of the problem should not be deemed useless and discarded under the mere pretext that it does not cut $\bar{x}$. Said otherwise, considering only valid inequalities in the reverse polar (of the next relaxed polyhedron P) may be overly restrictive.

Accordingly, we suggest to rehabilitate the traditional direct polar as an interesting “support-generating set”, which can be used in various ways. For instance, the lift-and-project method can be mimicked by maximizing the violation of $\bar{x}$ over the direct polar. An advantage of doing so is that, contrary to the reverse polar, the direct polar is essentially a bounded set (although somewhat artificially since its shape depends on the choice of the origin) and no normalization constraint is necessary. This method, however, fails to achieve our initial goal of putting less emphasis on $\bar{x}$ and focusing on more intrinsic objects of the problem.

We therefore suggest another way to generate supports, completely *independent* of $\bar{x}$, as follows. We start from the (tight) cut directed by the objective function. It has the nice property of improving the value of the relaxed solution as much as possible, but it is usually considered as a poor choice because it just adds cuts that are parallel to each other instead of being parallel to facets of P —an equally desirable property. We suggest to decompose this appealing but powerless cut into extreme points of the direct polar and to use them as additional valid inequalities in the current relaxation of the optimization problem. This technique somehow yields the best of both worlds: the objective function is improved as much as possible and the valid inequalities are facet-exposing.

A final word: within a cutting-plane mechanism, an interior point can be used for not only cutting but also placing the relaxed point $\bar{x}$; the idea goes back to [Topkis \(1982\)](#) and was revived recently: see [Benameur and Neto \(2007\)](#) and [Fischetti and Salvagnin \(2010\)](#).

Acknowledgment We are indebted to anonymous referees, who helped us improving earlier versions of this paper.

References

- Balas E (1979) Disjunctive programming. *Ann Discret Math* 5:3–51
- Balas E, Ceria S, Cornuéjols G (1993) A lift-and-project cutting plane algorithm for mixed 0–1 programs. *Math Program* 58(3):295–324
- Benameur W, Neto J (2007) Acceleration of cutting plane and column generation algorithms: application to network design. *Networks* 49(1):3–17

- Bonami P, Minoux M (2005) Using rank-1 lift-and-project closures to generate cuts for 0G1 MIPs, a computational investigation. *Disc Opt* 2(4):288–307
- Boyd EA (1994) Fenchel cutting planes for integer programs. *Oper Res* 42(1):53–64
- Cadoux F (2008) Computing deep facet-defining disjunctive cuts for mixed-integer programming. *Math Program* 122(2):197–223. doi:[10.1007/s10107-008-0245-6](https://doi.org/10.1007/s10107-008-0245-6)
- Cornuéjols G, Lemaréchal C (2005) A convex-analysis perspective on disjunctive cuts. *Math Program* 106(3):567–586
- Fischetti M, Salvagnin D (2010) An in-out approach to disjunctive optimization. In: Lodi A, et al (eds) *Integration of AI and OR techniques in constraint programming for combinatorial optimization problems*, number 6140 in *Lecture Notes in Computer Sciences*. pp 136–140
- Hiriart-Urruty J-B, Lemaréchal C (2001) *Fundamentals of convex analysis*. Springer, Berlin
- Kiwiel KC (1986) A method for solving certain quadratic programming problems arising in nonsmooth optimization. *IMA J Numer Anal* 6:137–152
- Kiwiel KC (1994) A Cholesky dual method for proximal piecewise linear programming. *Numeri Math* 68:325–340
- Perregaard M (2003) *Generating disjunctive cuts for mixed integer programs*. Ph.D. thesis, Carnegie Mellon University
- Perregaard M, Balas E (2001) Generating cuts from multiple-term disjunctions. In: Aardal K et al (eds) *Integer programming and combinatorial optimization*, number 2081 in *Lecture Notes in Computer Science*, pp 348–360
- Topkis DM (1982) A cutting-plane algorithm with linear and geometric rates of convergence. *J Optim Theory Appl* 36(1):1–22
- Wolfe Ph (1976) Finding the nearest point in a polytope. *Math Program* 11(2):128–149