Journal of Global Optimization (2020) 78:161–179
https://doi.org/10.1007/s10898-020-00906-y

On the relation between the extended supporting hyperplane algorithm and Kelley's cutting plane algorithm

Felipe Serrano[1] · **Robert Schwarz**[1] · **Ambros Gleixner**[1]

Received: 20 May 2019 / Accepted: 13 April 2020 / Published online: 13 May 2020

Abstract
Recently, Kronqvist et al. (J Global Optim 64(2):249–272, 2016) rediscovered the supporting hyperplane algorithm of Veinott (Oper Res 15(1):147–152, 1967) and demonstrated its computational benefits for solving convex mixed integer nonlinear programs. In this paper we derive the algorithm from a geometric point of view. This enables us to show that the supporting hyperplane algorithm is equivalent to Kelley's cutting plane algorithm (J Soc Ind Appl Math 8(4):703–712, 1960) applied to a particular reformulation of the problem. As a result, we extend the applicability of the supporting hyperplane algorithm to convex problems represented by a class of general, not necessarily convex nor differentiable, functions.

Keywords Convex MINLP · Cutting plane algorithms · Supporting hyperplane algorithm · Nonsmooth Optimization

1 Introduction

A mixed integer convex program (MICP) is a problem of the form

$$\min\{c^{\mathsf{T}}x \ : \ x \in C \cap (\mathbb{Z}^p \times \mathbb{R}^{n-p})\}, \tag{1}$$

where C is a closed convex set, $c \in \mathbb{R}^n$, and p denotes the number of variables with integrality requirement. The use of a linear objective function is without loss of generality given that

The authors thank the Schloss Dagstuhl-Leibniz Center for Informatics for hosting the Seminar 18081 "Designing and Implementing Algorithms for Mixed-Integer Nonlinear Optimization" for providing the environment to develop the ideas in this paper. The described research activities are funded by the German Federal Ministry for Economic Affairs and Energy within the project EnBA-M (ID: 03ET1549D). The work for this article has been (partly) conducted within the Research Campus MODAL funded by the German Federal Ministry of Education and Research (BMBF grant numbers 05M14ZAM, 05M20ZBM).

✉ Felipe Serrano
serrano@zib.de

Robert Schwarz
schwarz@zib.de

Ambros Gleixner
gleixner@zib.de

[1] Department of Mathematical Optimization, Zuse Institute Berlin, Takustr. 7, 14195 Berlin, Germany

one can always transform a problem with a convex objective function into a problem of the form (1). We can represent the set C in different ways, one of the most common being as the intersection of sublevel sets of convex differentiable functions, that is,

$$C = \{x \in \mathbb{R}^n \,:\, g_j(x) \leq 0,\, j \in J\}. \tag{2}$$

Here, J is a finite index set and each g_j is convex and differentiable.

Several methods have been proposed for solving MICP. When the problem is continuous and represented as (2), one of the first proposed methods was Kelley's cutting plane algorithm [1]. This algorithm exploits the convexity of a constraint function g in the following way. The convexity and differentiability of g imply that $g(y) + \nabla g(y)(x - y) \leq g(x)$ for every $x, y \in \mathbb{R}^n$. Since every feasible point x must satisfy $g(x) \leq 0$, it follows that $g(y) + \nabla g(y)(x - y) \leq 0$, for a fixed y, is a valid linear inequality. If $\bar{x} \in \mathbb{R}^n$ does not satisfy the constraint $g(x) \leq 0$, that is, if $g(\bar{x}) > 0$, then

$$g(\bar{x}) + \nabla g(\bar{x})(x - \bar{x}) \leq 0 \tag{3}$$

separates $\bar{x}$ from the feasible solution. In the non-differentiable case

$$g(\bar{x}) + v^{\mathsf{T}}(x - \bar{x}) \leq 0, \quad \text{with } v \in \partial g(\bar{x}), \tag{4}$$

is also a separating valid inequality. Here $\partial g(\bar{x})$ denotes the *subdifferential* of g at $\bar{x}$ and we recall its definition later. We will call both inequalities (3) and (4) *gradient cut* of g at $\bar{x}$.

The idea of Kelley's cutting plane algorithm is to approximate the feasible region with a polytope, solve the resulting linear program (LP) and, if the LP solution is not feasible, separate it using gradient cuts to obtain a new polytope which is a better approximation of the feasible region and repeat, see Algorithm 1.

Algorithm 1: Kelley's cutting plane algorithm

```
LP = {x : x ∈ [l, u]}, x̄ ← arg min_{x∈LP} c^T x
while max_{j∈J} g_j(x̄) > ϵ do
    forall the j such that g_j(x̄) > 0 do
        LP ← LP ∩ {x : g_j(x̄) + ∇g_j(x̄)(x − x̄) ≤ 0}
    x̄ ← arg min_{x∈LP} c^T x
return x̄
```

Kelley shows that the algorithm converges to the optimum and it converges in finite time to a point close to the optimum. By solving integer programs (IP) using Gomory's cutting plane [2] instead of LP relaxations, Kelley shows that his cutting plane algorithm solves purely integer convex programs in finite time. The same algorithm works just as well for MICP. However, Kelley did not have access to a finite algorithm for solving mixed integer linear programs (MILP).

In an attempt to speed up Kelley's algorithm, Veinott [3] proposes the *supporting hyperplane algorithm* (SH). A possible issue with Kelley's algorithm is that, in general, gradient cuts do not support the feasible region, see Fig. 1. Therefore, it is expected that better relaxations can be achieved by using supporting cutting planes.

In order to construct supporting hyperplanes, Veinott suggests to build gradient cuts at boundary points of C. He uses an interior point of C to find the point on the boundary, $\hat{x}$, that intersects the segment joining the interior point and the solution of the current relaxation. These cuts are automatically supporting hyperplanes of C, at $\hat{x}$. However, since the cut is

computed at $\hat{x}$ which is in C, it might happen that the gradient of the constraints active at $\hat{x}$ vanishes. For this reason, Veinott also requires that the functions representing C have non-vanishing gradients at the boundary. This is immediately implied by, e.g., Slater's condition. Veinott also identifies that one can use his algorithm to solve (1) when representing C by quasi-convex functions, that is, functions whose sublevel sets are convex.

Recently, Kronqvist et al. [4] rediscovered and implemented Veinott's algorithm [3]. They call their algorithm the *extended supporting hyperplane algorithm* (ESH). They discuss the practical importance of choosing a good interior point and propose some improvements over the original method, such as solving LP relaxations during the first iterations instead of the more expensive MILP relaxation. As a result, they present a computationally competitive solver implementation for MICPs defined by convex differentiable constraint functions [5].

In this paper, we would like to understand when, given a convex differentiable function g, gradient cuts of g are supporting to the convex set $C = \{x \in \mathbb{R}^n : g(x) \leq 0\}$. This question is motivated by the fact that in this case Kelley's algorithm automatically becomes a supporting hyperplane algorithm. In Theorem 1 we give a necessary and sufficient condition for a gradient cut of g at a given point to be a supporting hyperplane of C. In particular, this condition suggests to look at *sublinear functions*, i.e., convex and positively homogeneous functions. As it turns out, this naturally leads to Veinott's algorithm.

Sublinear functions and convex sets are deeply related. When the origin is in the interior of a convex set C, then we can represent C via its *gauge* function φ_C, which is sublinear [6]. We give the formal definition of the gauge function in Sect. 4, but for now it suffices to know that we can represent C as $C = \{x \in \mathbb{R}^n : \varphi_C(x) \leq 1\}$ and that, in particular, for every $\bar{x} \neq 0$ a gradient cut of φ_C at $\bar{x}$ supports all of its sublevel sets. The following example illustrates this.

Example 1 Consider the convex feasible region given by

$$C = \{(x, y) \in \mathbb{R}^2 : g(x, y) \leq 0\},$$

where $g(x, y) = x^2 + y^2 - 1$. We show through an example that gradient cuts of g are not necessarily supporting to C, explain why this happens, and show that changing the representation of C to use its gauge function solves the issue.

Separating the infeasible point $\bar{x} = (\frac{3}{2}, \frac{3}{2})$ by a gradient cut of g at $\bar{x}$ gives

$$g(\bar{x}) + \nabla g(\bar{x})(x - \bar{x}) \leq 0$$
$$\Leftrightarrow x + y \leq \frac{11}{6}.$$

This cut does not support C, see Fig. 1. Alternatively, the gauge function of C is given by $\varphi_C(x, y) = \sqrt{x^2 + y^2}$ and $C = \{(x, y) : \sqrt{x^2 + y^2} \leq 1\}$. The gradient cut of φ_C at $\bar{x}$ is $x + y \leq \sqrt{2}$, which is supporting. □

From the previous discussion it is a natural idea to represent C via its gauge function, namely, $C = \{x \in \mathbb{R}^n : \varphi_C(x) \leq 1\}$. However, as mentioned before, C is usually given by (2). Our main contribution is to show that reformulating (2) to the gauge representation will naturally lead to the ESH algorithm, see Sect. 4.2. As a consequence, the convergence proofs of Veinott [3] and Kronqvist et al. [4] follow directly from the convergence proof of Kelley's cutting plane algorithm [1,7], see Sect. 5. In other words, we show that the ESH algorithm is Kelley's cutting plane algorithm applied to a different representation of the problem.[1]

[1] Strictly speaking, when the problem is mixed integer, the KCP algorithm only corresponds to the so-called LP-step [4] of the ESH algorithm.

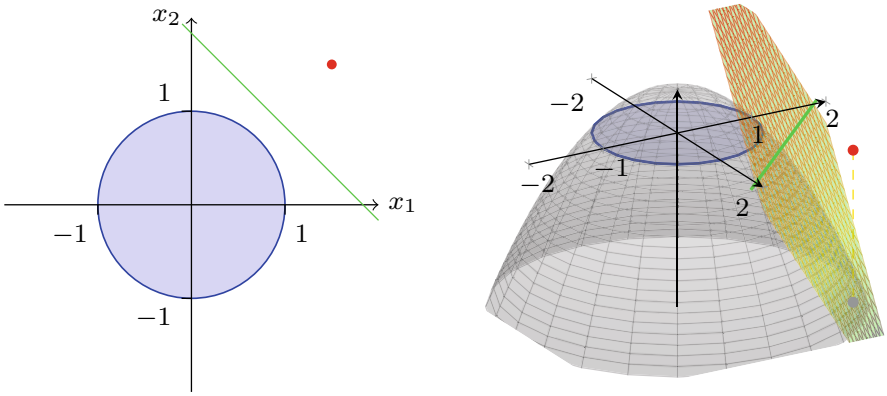

Fig. 1 The feasible region C and the infeasible point $\bar{x} = (\frac{3}{2}, \frac{3}{2})$ to separate. On the left we see that the separating hyperplane is not supporting to C. On the right we see why this happens: the linearization of g at $\bar{x}$ is tangent to the epigraph of g (shown upside-down for clarity) at $(\bar{x}, g(\bar{x}))$. However, when this hyperplane intersects the x-y-plane, it is already far away from the epigraph, and in consequence, from the sublevel set. The intersection of the hyperplane with the x–y-plane is the gradient cut

Motivated by this approach of representing C by its gauge function, we are able to show that the ESH algorithm applied to (1) converges even when C is not represented by convex functions. This is related to recent work of Lasserre [8] that tries to understand how different techniques behave when the convex set C is not represented via (2). Lasserre considers sets $C = \{x : g_j(x) \leq 0, j \in J\}$ where g_j are only differentiable, but not necessarily convex in the following setting:

Assumption 1 For all $x \in C$ and all $j \in J$, if $g_j(x) = 0$, then $\nabla g_j(x) \neq 0$.

Under this assumption, that is, if the gradients of active constraints do not vanish at the boundary of C, Lasserre shows that the KKT conditions are not only necessary but also sufficient for global optimality. In other words, every minimizer is a KKT point and every KKT point is a minimizer.

A series of generalizations follow the work of Lasserre. Dutta and Lalitha [9] generalize the previous result to the case where C is represented by locally Lipschitz functions, not necessarily differentiable nor convex, but regular in the sense of Clarke [10] (see also Definition 2). Martínez-Legaz [11] further generalize the result to the case where C is represented by *tangentially convex* functions [12,13]. Kabgani et al. [14] generalize the result to the case where C is represented by functions that admit an *upper regular convexificator* URC [15] (see also Definition 3). We note that regular functions in the sense of Clarke and tangentially convex functions admit a URC [14], thus the URC assumption is the most general among the ones considered in these works.

In terms of computations, Lasserre [16,17] proposes an algorithm to find the KKT point via log-barrier functions. He shows that the algorithm converges to the KKT point if Assumption 1 holds.

For all these concepts of generalized derivative, there is a notion of directional derivative and a notion of subdifferential. For example, for functions that admit a URC, the notion of

However, given that the KCP algorithm allows for an straightforward extension to the mixed integer case, we will continue to compare the KCP algorithm to the ESH algorithm with respect to their technique of generating cutting planes.

directional derivative is the upper Dini directional derivative and its subdifferential is the URC (see Definition 3). Let f be a function and let us denote by $f'(x;d)$ a generalized directional derivative. We say that the directional derivative is *well-behaved* if $f'(x;d) > 0$ implies that there exists $t_n \searrow 0$ such that $f(x + t_n d) > f(x)$.

In this sense we show that if C is represented by functions whose generalized directional derivatives are well-behaved, then the ESH converges to the global optimum, under the equivalent of Assumption 1 [see (10)] for the corresponding subdifferential. The upper Dini directional derivative is certainly well-behaved and, thus, our result shows that the ESH converges when C is represented by functions that admit a URC. We also show that for ∂°-pseudoconvex (see Definition 6) constraints, the Clarke directional derivative (see Definition 2) is well-behaved. Therefore, our result generalizes the result of [18] that the ESH converges when C is represented by ∂°-pseudoconvex functions.

We also show, via an example, that if we use Clarke's subdifferential [10], the ESH does not need to converge when the functions are only Lipschitz continuous but not regular in the sense of Clarke [10].

Finally, we provide a characterization of convex functions whose linearizations are supporting to their sublevel sets. Although elementary, the authors are not aware of its presence in the literature. In particular, this result allows us to identify some families of functions for which gradient cuts are never supporting (see Example 3) and some for which they are always supporting (see Corollary 2 and Example 2).

Overview of the paper. In the remainder of this section we introduce the notation that will be used throughout the paper. Section 2 provides a literature review on cutting plane approaches and efforts on obtaining supporting valid inequalities. In Sect. 3, we characterize functions whose linearizations are supporting hyperplanes to their 0-sublevel sets. Section 4 introduces the gauge function and shows how to use evaluation of the gauge function for building supporting hyperplanes. We note that evaluating the gauge function is equivalent to the line search step of the ESH algorithm [3,4]. This equivalence provides the link between the ESH and Kelley's cutting plane algorithm. In Sect. 5, we show that the cutting planes generated by the ESH algorithm can also be generated by Kelley's algorithm when applied to a reformulation of the problem. This implies that the convergence of the ESH algorithm follows from Kelley's. In Sect. 6, we show that we can apply the ESH algorithm to problem (1) when the convex set C is represented via functions whose generalized directional derivatives are well-behaved as long as 0 does not belong to the generalized subdifferential at points where the functions are zero. Finally, Sect. 7 presents our concluding remarks.

Notation and definitions. The boundary and the interior of a set C are denoted by ∂C and $\mathring{C}$, respectively. The epigraph of a function g is denoted by $\operatorname{epi} g$. The subdifferential of a convex function g at $\bar{x}$ is denoted by $\partial g(\bar{x})$. Recall that the subdifferential is the set of all subgradients of g at $\bar{x}$,

$$\partial g(\bar{x}) = \{v \in \mathbb{R}^n \,:\, g(\bar{x}) + v^\mathsf{T}(x - \bar{x}) \leq g(x), \forall x \in \mathbb{R}^n\}.$$

We say that an inequality $\alpha^\mathsf{T} x \leq \beta$ is *valid* for a set C if every $x \in C$ satisfies $\alpha^\mathsf{T} x \leq \beta$. Furthermore, we say that it is a *supporting hyperplane* of C, or that it *supports* C, if there is an $x \in \partial C$ such that $\alpha^\mathsf{T} x = \beta$.

A function g is *positively homogeneous* if $g(\lambda x) = \lambda g(x)$ for every $\lambda \geq 0$. A function is *sublinear* if it is positively homogeneous and convex.

2 Literature review

We can think of the algorithms of Kelley [1] and Veinott [3] as a mixture of two ingredients: which relaxation to solve and where to compute the cutting plane. Indeed, at each iteration we have a point x^k that we would like to separate with a linear inequality $\beta+\alpha^{\mathsf{T}}(x-x_0) \leq 0$. For Kelley's algorithm, $x_0 = x^k$, while for Veinott's algorithm, $x_0 \in \partial C$, and for both $\alpha \in \partial g(x_0)$ and $\beta = g(x_0)$. Choosing different relaxations and different points where to compute the cutting planes yields different algorithms. This framework is developed in Horst and Tuy [7].

Following the previous framework, Duran and Grossmann [19] propose the, so-called, *outer approximation algorithm* for MICP. The idea is to solve an MILP relaxation, but instead of computing a cutting plane at the MILP optimum, or at the boundary point on the segment between the MILP optimum and some interior point, they suggest to compute cutting planes at a solution of the nonlinear program (NLP) obtained after fixing the integer variables to the integer values given by the MILP optimal solution. This is a much more expensive algorithm but has the advantage of finite convergence. Of course, this does not work in complete generality and we need some assumptions, for example, requiring some constraint qualifications. Moreover, when obtaining an infeasible NLP after fixing the integer variables, care must be taken to prevent the same integer assignment in future iterations. To handle such cases, Duran and Grossmann propose the use of integer cuts. However, Fletcher and Leyffer [20] point out that this is not necessary. They show that the gradient cuts at the solution of a slack NLP separates the integer assignment. In [21] show that a naive generalization of the outer approximation algorithm to the non-differentiable case will not work. They provide a generalization for a particular class of function. Wei and Ali [22,23] provide further generalizations to the non-differentiable case.

A related algorithm to the outer approximation method is the so-called generalized Benders decomposition [24]. We refer to [19,20,25] for discussions about the relation between these two algorithms. A generalization of the generalized Benders decomposition to Banach spaces can be found in [26].

Westerlund and Pettersson [27] propose the so-called extended cutting plane algorithm. This algorithm is the extension of Kelley's cutting plane to MICP and they show that the algorithm converges. Further extensions and convergence proofs of cutting plane and outer approximation algorithms for non-smooth problems are given in [21]. An interesting generalization of the extended cutting plane algorithm to solve a class of non-convex problems is the so-called α extended cutting plane algorithm introduced by Westerlund et al. [28]. They consider problem (1) where C is represented by differentiable pseudoconvex constraints. The idea is that, even though a gradient cut might not be valid, one can tilt the cut in order to make it valid. The tilting is done by multiplying the gradient by some α, hence the name. We refer to [28] for more details.

As mentioned at the beginning, the assumption that the objective function is linear is without loss of generality, provided that the original objective function is convex. However, some classes of problems cannot be encompassed by (1), for example, when the objective function is quasi-convex. An extension of the KCP algorithm, the (α) extended cutting plane algorithm, and the ESH to convex problems with a class of quasi-convex objectives were developed by Plastria [29], Eronen et al. [30], and Westerlund et al. [31], respectively.

Yet another technique for producing tight cuts is to project the point to be separated onto C [7]. Using the projected point and the difference between the point and its projection, one can build a supporting hyperplane that separates the point. In the same reference, Horst and Tuy show that this algorithm converges.

There have been attempts at building tighter relaxations by ensuring that gradient cuts are supporting, in a more general context than convex mixed integer nonlinear programming. Belotti et al. [32] consider bivariate convex constraints of the form $f(x) - y \leq 0$, where f is a univariate convex function. They propose projecting the point to be separated onto the curve $y = f(x)$ and building a gradient cut at the projection. However, their motivation is not to find supporting hyperplanes, but to find the most violated cut. Indeed, as we will see, gradient cuts for these types of constraints are always supporting (Example 2). Other work along these lines includes [33], where the authors derive an efficient procedure to project onto a two dimensional constraint derived from a Gaussian linear chance constraint, thus building supporting valid inequalities.

Another algorithm for solving non-smooth convex optimization problems is the so-called bundle method [34]. This method has also been extended to consider the mixed integer case [35].

Finally, in terms of applications, we would like to point out that the supporting hyperplane algorithm is very popular in stochastic optimization [36–42].

3 Characterization of functions with supporting linearizations

We now give necessary and sufficient conditions for the linearization of a convex, not necessarily differentiable, function g at a point $\bar{x}$ to support the region $C = \{x \in \mathbb{R}^n : g(x) \leq 0\}$. In order for this to happen, the supporting hyperplane has to support the epigraph on the whole segment joining the point of C where it supports and $(\bar{x}, g(\bar{x}))$. In other words, the function must be affine on the segment joining the set C and $\bar{x}$. This is due to the convexity of g.

Theorem 1 *Let $g\colon \mathbb{R}^n \to \mathbb{R}$ be a convex function, $C = \{x \in \mathbb{R}^n : g(x) \leq 0\} \neq \emptyset$, and $\bar{x} \notin C$. There exists a subgradient $v \in \partial g(\bar{x})$ such that the valid inequality*

$$g(\bar{x}) + v^T(x - \bar{x}) \leq 0 \tag{5}$$

supports C, if and only if, there exists $x_0 \in C$ such that $\lambda \mapsto g(x_0 + \lambda(\bar{x} - x_0))$ is affine in $[0, 1]$.

Proof ($\Rightarrow$) Let $x_0 \in \partial C$ be the point where (5) supports C. The idea is to show that the affine function $x \mapsto g(\bar{x}) + v^{\mathsf{T}}(x - \bar{x})$ coincides g at two points, $\bar{x}$ and x_0. Then, by the convexity of g, it must coincide with g on the segment joining both points.

In more detail, by definition of x_0 we have,

$$g(\bar{x}) + v^{\mathsf{T}}(x_0 - \bar{x}) = 0. \tag{6}$$

For $\lambda \in [0, 1]$, let $l(\lambda) = x_0 + \lambda(\bar{x} - x_0)$ and $\rho(\lambda) = g(l(\lambda))$. Since g is convex and l affine, ρ is convex.

Since v is a subgradient,

$$g(\bar{x}) + v^{\mathsf{T}}(l(\lambda) - \bar{x}) \leq \rho(\lambda) \quad \text{for every } \lambda \in [0, 1].$$

After some algebraic manipulation and using that $\rho(1) = g(\bar{x}) = v^{\mathsf{T}}(\bar{x} - x_0)$, we obtain

$$\rho(1)\lambda \leq \rho(\lambda).$$

On the other hand, $\rho(0) = 0$ and $\rho(\lambda)$ is convex, thus we have $\rho(\lambda) \leq \lambda\rho(1) + (1 - \lambda)\rho(0) = \lambda\rho(1)$ for $\lambda \in [0, 1]$. Therefore, $\rho(\lambda) = \rho(1)\lambda$, hence $g(l(\lambda))$ is affine in $[0, 1]$.

($\Leftarrow$) The idea is to show that there is a supporting hyperplane H of epi $g \subseteq \mathbb{R}^n \times \mathbb{R}$ which contains the graph of g restricted to the segment joining x_0 and $\bar{x}$, that is, $A = \{(x_0 + \lambda(\bar{x} - x_0), g(x_0 + \lambda(\bar{x} - x_0))) : \lambda \in [0, 1]\}$. Then the intersection of such H with $\mathbb{R}^n \times \{0\}$ will give us (5).

The set A is a convex nonempty subset of epi g that does not intersect the relative interior of epi g. Hence, there exists a supporting hyperplane,

$$H = \{(x, z) \in \mathbb{R}^n \times \mathbb{R} : v^\mathsf{T} x + az = b\},$$

to epi g containing A ([6, Theorem 11.6]).

Since $g(x_0) \leq 0$ and $g(\bar{x}) > 0$, it follows that A is not parallel to the x-space. Therefore, H is also not parallel to the x-space and so $v \neq 0$. Since A is not parallel to the z-axis, it follows that $a \neq 0$. We assume, without loss of generality, that $a = -1$.

The point $(\bar{x}, g(\bar{x}))$ belongs to $A \subseteq H$, thus $v^\mathsf{T}\bar{x} - g(\bar{x}) = b$ and $H = \{(x, g(\bar{x}) + v^\mathsf{T}(x - \bar{x})) : x \in \mathbb{R}^n\}$. Given that H supports the epigraph, then v is a subgradient of g, in particular,

$$g(\bar{x}) + v^\mathsf{T}(x - \bar{x}) \leq g(x) \quad \text{for every } x \in \mathbb{R}^n.$$

Let $z(x)$ be the affine function whose graph is H, that is, $z(x) = g(\bar{x}) + v^\mathsf{T}(x - \bar{x})$. We now need to show that $g(\bar{x}) + v^\mathsf{T}(x - \bar{x}) \leq 0$ supports C by exhibiting an $\hat{x} \in C$ such that $g(\bar{x}) + v^\mathsf{T}(\hat{x} - \bar{x}) = 0$. By construction, $z(x_0 + \lambda(\bar{x} - x_0)) = g(x_0 + \lambda(\bar{x} - x_0))$. Since $z(x_0 + \lambda(\bar{x} - x_0))$ is non-positive for $\lambda = 0$ and positive for $\lambda = 1$, it has to be zero for some λ_0. Let $\hat{x} = x_0 + \lambda_0(\bar{x} - x_0)$. Then $g(\hat{x}) = z(\hat{x}) = 0$ and we conclude that $\hat{x} \in C$ and $g(\bar{x}) + v^\mathsf{T}(\hat{x} - \bar{x}) = 0$. □

Specializing the theorem to differentiable functions directly leads to the following:

Corollary 1 *Let $g : \mathbb{R}^n \to \mathbb{R}$ be a convex differentiable function, $C = \{x \in \mathbb{R}^n : g(x) \leq 0\}$, and $\bar{x} \notin C$. Then the valid inequality*

$$g(\bar{x}) + \nabla g(\bar{x})^T(x - \bar{x}) \leq 0,$$

supports C, if and only if, there exists $x_0 \in C$ such that $\lambda \mapsto g(x_0 + \lambda(\bar{x} - x_0))$ is affine in $[0, 1]$.

Proof Since g is differentiable, the subdifferential of g consists only of the gradient of g. □

A natural candidate for functions with supporting gradient cuts at every point are functions whose epigraph is a translation of a convex cone.

Corollary 2 (Sublinear functions) *Let $h(x)$ be a sublinear function. For this type of function, gradient cuts always support $C = \{x : h(x) \leq c\}$, for any $c \geq 0$.*

Proof This follows directly from Theorem 1, since $0 \in C$ and $\lambda \mapsto h(\lambda\bar{x})$ is affine in $\mathbb{R}_+$ for any $\bar{x}$. □

However, these are not the only functions that satisfy the conditions of Theorem 1 for every point. The previous theorem implies that linearizations always support the constraint set if a convex constraint $g(x) \leq 0$ is linear in one of its arguments.

Example 2 (Functions with linear variables) Let $f : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$ be a convex function of the form $f(x, y) = g(x) + a^\mathsf{T} y + c$, with $a \neq 0$ and $g : \mathbb{R}^m \to \mathbb{R}$ convex. Then gradient cuts support $C = \{(x, y) : f(x, y) \leq 0\}$. Indeed, assume without loss of generality that $a_1 > 0$ and let $(\bar{x}, \bar{y}) \notin C$. Then there exists a $\lambda > 0$ such that $f(\bar{x}, \bar{y} - \lambda e_1) = g(\bar{x}) + a^\mathsf{T}\bar{y} + c - a_1\lambda = 0$. The statement follows from Theorem 1.

Consider separating a point (x_0, z_0) from a constraint of the form $z = g(x)$ with $g : \mathbb{R} \to \mathbb{R}$ and convex, with $z_0 < g(x_0)$ (that is, separating on the convex constraint $g(x) \leq z$). As mentioned earlier, in [32] the authors suggest projecting (x_0, z_0) to the graph $z = g(x)$ and computing a gradient cut there. This example shows that this step is unnecessary when the sole purpose is to obtain a cut that is supporting to the graph. □

By contrast, if $g(x)$ is strictly convex, linearizations at points x such that $g(x) \neq 0$ are never supporting to $g(x) \leq 0$. This follows directly from Theorem 1 since $\lambda \mapsto g(x + \lambda v)$ is not affine for any v. We can also characterize convex quadratic functions with supporting linearizations.

Example 3 *(Convex quadratic functions)* Let $g(x) = x^\mathsf{T} Ax + b^\mathsf{T} x + c$ be a convex quadratic function, i.e., A is an n by n symmetric and positive semi-definite matrix. We show that gradient cuts support $C = \{x \in \mathbb{R}^n : g(x) \leq 0\}$, if and only if, b is not in the range of A, i.e., $b \notin R(A) = \{Ax : x \in \mathbb{R}^n\}$.

First notice that $l_v(\lambda) = g(x + \lambda v)$ is affine linear, if and only if, $v \in \ker(A)$.
Let $v \in \ker(A)$ and $\bar{x} \notin C$. Then there is a $\lambda \in \mathbb{R}$ such that $\bar{x} + \lambda v \in C$ if and only if l_v is not constant. Thus, gradient cuts are *not* supporting, if and only if, l_v is constant for every $v \in \ker(A)$. But l_v is constant for every $v \in \ker(A)$, if and only if, $b^\mathsf{T} v = 0$ for every $v \in \ker(A)$, which is equivalent to $b \in \ker(A)^\perp = R(A^\mathsf{T}) = R(A)$, since A is symmetric. Hence, gradient cuts support C, if and only if, $b \notin R(A)$.

In particular, if $b = 0$, i.e., there are no linear terms in the quadratic function, then gradient cuts are never supporting hyperplanes. Also, if A is invertible, $b \in R(A)$ and gradient cuts are not supporting. This is to be expected since in this case g is strictly convex. □

4 The gauge function

Any MICP of form (1) can be reformulated to an equivalent MICP with a single constraint for which every linearization supports the continuous relaxation of the feasible region. To this end, we can use any sublinear function whose 1-sublevel set is C. Each convex set C has at least one sublinear function that represents it, namely, the *gauge function* [6] of C.

Definition 1 Let $C \subseteq \mathbb{R}^n$ be a convex set such that $0 \in \mathring{C}$. The *gauge* of C is

$$\varphi_C(x) = \inf\{ t > 0 : x \in tC \}.$$

Proposition 1 ([43, Proposition 1.11]) *Let $C \subseteq \mathbb{R}^n$ be a convex set such that $0 \in \mathring{C}$, then $\varphi_C(x)$ is sublinear. If, in addition, C is closed, then it holds that*

$$C = \{x \in \mathbb{R}^n : \varphi_C(x) \leq 1\}$$

and

$$\partial C = \{x \in \mathbb{R}^n : \varphi_C(x) = 1\}.$$

Combining Proposition 1 with Corollary 2, we can see that the gauge function is appealing for separation, because it always generates supporting hyperplanes.

4.1 Using the gauge function for separation

Even though the gauge function is exactly what we need to ensure supporting gradient cuts, in general, there is no closed-form formula for it. Therefore, it is not always possible to explicitly reformulate C as $\varphi_C(x) \leq 1$.

Furthermore, if one is interested in solving mathematical programs with a numerical solver, performing such a reformulation might introduce some numerical issues one would have to take care of. Solvers usually solve up to a given tolerance, that is, they accept points that satisfy $g_j(x) \leq \varepsilon$ for some $\varepsilon > 0$. Then, even though $C = \{x : \varphi_C(x) \leq 1\}$, it might be that $\{x \in \mathbb{R}^n : \varphi_C(x) \leq 1 + \varepsilon\} \nsubseteq \{x \in \mathbb{R}^n : g_j(x) \leq \varepsilon\}$. In fact, even simple constraints show this behavior. Consider $C = \{x : x^2 - 1 \leq 0\}$. In this case, $\varphi_C(x) = |x|$ and for $x_0 = 1 + \varepsilon$, we have $\varphi_C(x_0) = 1 + \varepsilon$. Then x_0 would be ε-feasible for $\varphi_C(x) \leq 1$, although it would be infeasible for $x^2 - 1 \leq 0$, since $2\varepsilon + \varepsilon^2 > \varepsilon$.

Luckily, one does not need to reformulate in order to take advantage of the gauge function for tighter separation. The next propositions show how to use the gauge function and a point $\bar{x} \notin C$ to obtain a boundary point of C and that linearizing at that boundary point gives a supporting valid inequality that actually separates $\bar{x}$. For ensuring the existence of a supporting hyperplane we need Assumption 1. For example, Assumption 1 is satisfied whenever Slater's condition is satisfied for (1) with C represented by (2), that is, when there exists x_0 such that $g_j(x_0) < 0$ for every $j \in J$.

Before we state the propositions we start with a simple lemma.

Lemma 1 *Let $C \subseteq \mathbb{R}^n$ be a closed convex set such that $0 \in \mathring{C}$, let $\hat{x} \in \partial C$ and $\bar{x} \notin C$. Let $\alpha \in \mathbb{R}^n, \beta \in \mathbb{R}$ such that $\alpha \neq 0$ and $\alpha^T x \leq \beta$ is a valid inequality for C that supports C at $\hat{x}$. If the segment joining 0 and $\bar{x}$ contains $\hat{x}$, then the inequality separates $\bar{x}$ from C.*

Proof Consider $l(\lambda) = \alpha^T(\lambda\bar{x}) - \beta$ and let $\lambda_0 \in (0, 1)$ be such that $\lambda_0\bar{x} = \hat{x}$. The function l is a strictly increasing affine linear function. Indeed, $0 \in \mathring{C}$ implies that $l(0) < 0$, while $l(\lambda_0) = 0$. Thus, $l(1) > 0$, i.e., $\alpha^T\bar{x} > \beta$. □

Proposition 2 *Let $C \subseteq \mathbb{R}^n$ be a closed convex set such that $0 \in \mathring{C}$ and let $\bar{x} \notin C$. Then $\frac{\bar{x}}{\varphi_C(\bar{x})} \in \partial C$.*

Proof First, $\varphi_C(\bar{x}) \neq 0$ since $\bar{x} \notin C$. The positive homogeneity of φ_C implies that $\varphi_C\left(\frac{\bar{x}}{\varphi_C(\bar{x})}\right) = \frac{\varphi_C(\bar{x})}{\varphi_C(\bar{x})} = 1$. Proposition 1 implies $\frac{\bar{x}}{\varphi_C(\bar{x})} \in \partial C$. □

Let $J_0(x)$ be the set of indices of the active constraints at x, i.e., $J_0(x) = \{j \in J : g_j(x) = 0\}$.

Proposition 3 *Let $C = \{x : g_j(x) \leq 0, j \in J\}$ be such that $0 \in \mathring{C}$ and let φ_C be its gauge function. Assume that Assumption 1 holds. Given $\bar{x} \notin C$, define $\hat{x} = \frac{\bar{x}}{\varphi_C(\bar{x})}$. Then, for any $j \in J_0(\hat{x})$, the gradient cut of g_j at $\hat{x}$ yields a valid supporting inequality for C that separates $\bar{x}$.*

Proof By the previous proposition, we have that $\hat{x} \in \partial C$. Let $j \in J_0(\hat{x})$. Then the gradient cut of g_j at $\hat{x}$ yields a valid supporting inequality. The fact that it separates follows from Lemma 1. Note that Lemma 1 is applicable since Assumption 1 ensures that the normal of the gradient cut is nonzero. □

Hence, we can get supporting valid inequalities separating a given point $\bar{x} \notin C$ by using the gauge function to find the point $\hat{x} = \frac{\bar{x}}{\varphi_C(\bar{x})} \in \partial C$. Then Proposition 3 ensures that the gradient cut of any active constraint at $\hat{x}$ will separate $\bar{x}$ from C. But how do we compute $\varphi_C(\bar{x})$?

4.2 Evaluating the gauge function

Let $C = \{x : g_j(x) \leq 0,\ j \in J\}$ be a closed convex set such that $0 \in \mathring{C}$ and consider

$$f(x) = \max_{j \in J} g_j(x). \tag{7}$$

In general, evaluating the gauge function of C at $\bar{x} \notin C$ is equivalent to solving the following one dimensional equation

$$f(\lambda \bar{x}) = 0,\ \lambda \in (0, 1). \tag{8}$$

If λ^* is the solution, then $\varphi_C(\bar{x}) = \frac{1}{\lambda^*}$.

One can solve such an equation using a line search. Note that the line search is looking for a point $\hat{x} \in \partial C$ on the segment between 0 and $\bar{x}$. This is exactly what the (extended) supporting hyperplane algorithm performs when it uses 0 as its interior point.

We would also like to remark that a closed-form formula expression for the gauge function of C is equivalent to a closed-form formula for the solution of (8). It is possible to find such a formula for some functions, e.g., when f is a convex quadratic function.

Next, we briefly discuss what happens when 0 is not in the interior of C and when C has no interior. In the next section we discuss the implications of the fact that evaluating the gauge function is equivalent to the line search step of the supporting hyperplane algorithm.

4.3 Handling sets with empty interior

When $\mathring{C} = \emptyset$, we can still use the methods discussed above by applying a trick from [4]. Assuming $C = \{x \in \mathbb{R}^n : g_j(x) \leq 0,\ j \in J\} \neq \emptyset$, consider the set $C_\epsilon = \{x \in \mathbb{R}^n : g_j(x) \leq \epsilon,\ j \in J\}$. This set satisfies $\mathring{C}_\epsilon \neq \emptyset$ and optimizing over C_ϵ provides an ϵ-optimal solution.

4.4 Using a nonzero interior point

If $x_0 \in \mathring{C}$ and $x_0 \neq 0$, we can translate C so that 0 is in its interior. Equivalently, we can build a gauge function centered on x_0. This is given by

$$\varphi_{x_0, C}(x) = \varphi_{C - x_0}(x - x_0).$$

Then, given $\bar{x} \notin C$, the point

$$\hat{x} = \frac{\bar{x} - x_0}{\varphi_{C - x_0}(\bar{x} - x_0)} + x_0 \tag{9}$$

belongs to the boundary of C. Equivalently, $\hat{x} = x_0 + \lambda^*(\bar{x} - x_0)$, where λ^* solves

$$f(x_0 + \lambda(\bar{x} - x_0)) = 0,\ \lambda \in (0, 1),$$

with $f(x) = \max_{j \in J} g_j(x)$ as in (7).

5 Convergence proofs

Consider an MICP given by (1) with C represented as (2). Let f be defined as in (7). As mentioned above, the ESH algorithm [3,4] computes an interior point of C (which we will assume to be 0) and performs a line search between $\bar{x} \notin C$ and 0 in order to find a point on the

boundary. It computes a gradient cut at the boundary point, solves the relaxation again, and repeats the process. From our previous discussion, computing a gradient cut at the boundary point is equivalent to computing a gradient cut at $\frac{\bar{x}}{\varphi_C(\bar{x})}$. Therefore, the generated cuts are $f(\frac{\bar{x}}{\varphi_C(\bar{x})}) + v^{\mathsf{T}}(x - \frac{\bar{x}}{\varphi_C(\bar{x})}) \leq 0$, where $v \in \partial f(\frac{\bar{x}}{\varphi_C(\bar{x})})$.

To prove the convergence of the ESH algorithm, Veinott [3] and Kronqvist et al. [4] use tailored arguments. Here we show that the convergence of the algorithm follows from the convergence of Kelley's cutting plane algorithm (KCP) [1]. We note that the KCP algorithm still converges when C is represented by a convex non-differentiable function. One needs to replace gradients by subgradients and one can use any subgradient [7]. Therefore, given that $\varphi_C(x)$ is a convex function, we know that KCP converges when applied to $\min\{c^{\mathsf{T}}x : \varphi_C(x) \leq 1\}$. Thus, in order to prove that ESH converges, it is sufficient to show that the cutting planes generated by ESH can also be generated by KCP.

We first prove that the normals of (normalized) supporting valid inequalities are subgradients of the gauge function at the supporting point.

Lemma 2 *Let $\alpha^{\mathsf{T}}x \leq 1$ be a valid and supporting inequality for C. Let $\hat{x} \in \partial C$ be a point where it supports C, i.e., $\alpha^{\mathsf{T}}\hat{x} = 1$. Then $\alpha \in \partial\varphi_C(\hat{x})$.*

Proof We need to show that $\varphi_C(\hat{x}) + \alpha^{\mathsf{T}}(x - \hat{x}) \leq \varphi_C(x)$ for every x. Note that since $\hat{x} \in \partial C$, we have that $\varphi_C(\hat{x}) = 1$ and we just have to prove that $\alpha^{\mathsf{T}}x \leq \varphi_C(x)$.

When x is such that $\varphi_C(x) > 0$, we have $\frac{x}{\varphi_C(x)} \in C$. Due to the validity of $\alpha^{\mathsf{T}}x \leq 1$, it follows that $\alpha^{\mathsf{T}}\frac{x}{\varphi_C(x)} \leq 1$.

Now let x be such that $\varphi_C(x) = 0$. Then $\varphi_C(\lambda x) = 0$ for every $\lambda > 0$, i.e., $\lambda x \in C$ for every $\lambda > 0$. Hence, $\alpha^{\mathsf{T}}(\lambda x) \leq 1$ for every $\lambda > 0$ which implies that $\alpha^{\mathsf{T}}x \leq 0 = \varphi_C(x)$. □

Now we prove that the inequalities generated by the ESH algorithm can also be generated by the KCP algorithm. Given that the KCP algorithm converges even for non-smooth convex function [7], the next theorem implies the convergence of the ESH algorithm.

Theorem 2 *Consider an MICP given by (1) with C represented as (2) such that $0 \in \mathring{C}$ and Assumption 1 holds. Let f be defined as in (7) and let $\bar{x} \notin C$ be the current relaxation solution to separate. Let $f(\frac{\bar{x}}{\varphi_C(\bar{x})}) + v^{\mathsf{T}}(x - \frac{\bar{x}}{\varphi_C(\bar{x})}) \leq 0$, with $v \in \partial f(\frac{\bar{x}}{\varphi_C(\bar{x})})$, be the inequality generated by the ESH algorithm using 0 as the interior point. Then KCP applied to $\min\{c^{\mathsf{T}}x : \varphi_C(x) \leq 1\}$ can generate the same inequality.*

Proof Let $\hat{x} = \frac{\bar{x}}{\varphi_C(\bar{x})}$. First, let us show that Assumption 1 implies $v \neq 0$. Indeed, if $v = 0$, then $f(\hat{x}) + v^{\mathsf{T}}(x - \hat{x}) \leq f(x)$ and $0 \in C$ imply that $0 \geq f(0) \geq f(\hat{x}) + v^{\mathsf{T}}(0 - \hat{x}) = 0$. Let $j \in J$ be such that $g_j(0) = f(0) = 0$. Then $\lambda \mapsto g_j(\lambda\hat{x})$ is constant in $[0, 1]$. Thus, its derivative at 1 is 0, i.e., $\nabla g_j(\hat{x})^{\mathsf{T}}\hat{x} = 0$. This implies that $\nabla g_j(\hat{x})^{\mathsf{T}}\bar{x} = 0$. Furthermore, $\nabla g_j(\hat{x}) \neq 0$ by Assumption 1 and so Lemma 1 implies that $\nabla g_j(\hat{x})^{\mathsf{T}}(x - \hat{x}) \leq 0$ separates $\bar{x}$ from C. But this contradicts the equality $\nabla g_j(\hat{x})^{\mathsf{T}}\bar{x} = 0$.

Let us manipulate the inequality obtained by the ESH algorithm. Notice that $f(\hat{x}) = 0$ and so the inequality reads as $v^{\mathsf{T}}x \leq v^{\mathsf{T}}\hat{x}$. By Lemma 1, $\bar{x}$ is cut off by $v^{\mathsf{T}}x \leq v^{\mathsf{T}}\hat{x}$, i.e., $v^{\mathsf{T}}\bar{x} > v^{\mathsf{T}}\hat{x}$. This, together with $\varphi_C(\bar{x}) > 1$, implies that $v^{\mathsf{T}}\bar{x} > 0$. Summarizing, the inequality obtained by the ESH algorithm can be rewritten as

$$\left(\frac{\varphi_C(\bar{x})}{v^{\mathsf{T}}\bar{x}}v\right)^{\mathsf{T}} x \leq 1.$$

Lemma 2 implies that $\frac{\varphi_C(\bar{x})}{v^{\mathsf{T}}\bar{x}}v \in \partial\varphi_C(\hat{x})$. Since φ_C is positively homogeneous, $\partial\varphi_C(\hat{x}) = \partial\varphi_C(\bar{x})$. Hence, if the KCP algorithm applied to $\min\{c^{\mathsf{T}}x : \varphi_C(x) \leq 1\}$ separates $\bar{x}$ using

$\frac{\varphi_C(\bar{x})}{v^\mathsf{T}\bar{x}} v \in \partial\varphi_C(\bar{x})$, then it would generate the gradient cut

$$\varphi_C(\bar{x}) - 1 + \frac{\varphi_C(\bar{x})}{v^\mathsf{T}\bar{x}} v^\mathsf{T}(x - \bar{x}) \leq 0.$$

The left hand side of the above inequality is equivalent to $-1 + \frac{\varphi_C(\bar{x})}{v^\mathsf{T}\bar{x}} v^\mathsf{T} x$. This shows that the gradient cut constructed by the KCP algorithm is the same as the one construction by the ESH algorithm. □

6 Convex programs represented by non-convex non-smooth functions

In this section we consider problem (1) with C represented as

$$C = \{x \,:\, g_j(x) \leq 0,\, j \in J\},$$

where the functions g_j are not necessarily convex. As mentioned in the introduction, convex problems represented by non-convex functions have been considered in [8,9,11,14,16,17]. These different works have generalized each other by considering more general classes of non-smooth functions.

6.1 The ESH algorithm in the context of generalized differentiability

When a function is non-smooth there are many ways of extending the notion of differentiability. Informally, it is common to first define a notion of directional derivative and then a generalization of the gradient. As the directional derivative of g at x in the direction d is given by $\nabla g(x)^\mathsf{T} d$, the notion of generalized gradient tries to capture this relation.

A classic notion of generalized derivative is Clarke's subdifferential.

Definition 2 ([10,44]) The *Clarke directional derivative* of a function $g : \mathbb{R}^n \to \mathbb{R}$ at $\bar{x}$ in the direction $d \in \mathbb{R}^n$ is defined as

$$g^\circ(\bar{x}; d) = \limsup_{x \to \bar{x}, t \searrow 0} \frac{g(x + td) - g(x)}{t}.$$

The *Clarke subdifferential* of g at $\bar{x}$ is

$$\partial^\circ g(\bar{x}) = \{\eta \in \mathbb{R}^n : \eta^\mathsf{T} d \leq g^\circ(\bar{x}; d)\, \forall d \in \mathbb{R}^n\}.$$

We say that g is *directionally differentiable* at $\bar{x}$ if directional derivatives of g at $\bar{x}$ exist, that is,

$$g'(\bar{x}; d) = \lim_{t \searrow 0} \frac{g(\bar{x} + td) - g(\bar{x})}{t},$$

exists for every $d \in \mathbb{R}^n$. Finally, g is *regular in the sense of Clarke* at $\bar{x}$ if the g is directional differentiable at $\bar{x}$ and $g'(\bar{x}; d) = g^\circ(\bar{x}; d)$ for every $d \in \mathbb{R}$.

Another interesting class is the following.

Definition 3 ([15]) Let $g : \mathbb{R}^n \to \mathbb{R}$. The *upper Dini directional derivative* of g at $\bar{x}$ in the direction $d \in \mathbb{R}^n$ is

$$g^+(\bar{x}; d) = \limsup_{t \searrow 0} \frac{g(\bar{x} + td) - g(\bar{x})}{t}.$$

The function g has an *upper regular convexificator* (URC) at $\bar{x}$ if there exists a closed set $\partial^+ g(\bar{x}) \subseteq \mathbb{R}^n$ such that for each $d \in \mathbb{R}^n$,

$$g^+(\bar{x}; d) = \sup_{\alpha \in \partial^+ g(\bar{x})} \alpha^\mathsf{T} d.$$

We abstract the notion of directional derivative and subdifferential as follows.

Definition 4 Let $g : \mathbb{R}^n \to \mathbb{R}$ be a function. A *generalized directional derivative* of g is a function $h : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$, and the generalized directional derivative of g at x in the direction d is $h(x; d)$. We say that g *admits a generalized subdifferential* at x if there exists $A = A(x) \subseteq \mathbb{R}^n$ such that $h(x; d) = \sup_{v \in A(x)} v^\mathsf{T} d$ for all $d \in \mathbb{R}^n$.

For example, if g is locally Lipschitz, then Clarke's directional derivative is a generalized directional derivative and $\partial^\circ g(x)$ is a generalized subdifferential as $g^\circ(x; d) = \sup\{v^\mathsf{T} d : v \in \partial^\circ g(x)\}$ [44, Proposition 2.1.5]. Or, if g admits a URC, then Dini's directional derivative is a generalized directional derivative that admits a generalized subdifferential.

However, the above definition of generalized directional derivative and subdifferential is so general, that any support function of a set yields a generalized directional derivative that admits a generalized subdifferential. The following definition adds a further requirement in order to make this general notion useful.

Definition 5 Let h be a generalized directional derivative of g. We say that the generalized directional derivative is *well-behaved* if $h(x; d) > 0$ implies that there exists $t_n \searrow 0$ such that $g(x + t_n d) > g(x)$.

As we will see, this is the key property to show that the ESH algorithm converges.

Clearly, if g is differentiable, then the directional derivative is well-behaved. Also, Dini's directional derivative is well-behaved. As we will see in the next section, Clarke's directional derivative is not well-behaved in general. However, if the function is regular in the sense of Clarke, then it is well-behaved. Another important class of functions for which Clarke's directional derivative is well-behaved is the class of ∂°-pseudoconvex functions.

Definition 6 A function $g : \mathbb{R}^n \to \mathbb{R}$ is ∂°-*pseudoconvex* if

- it is locally Lipschitz and,
- for every $x, y \in \mathbb{R}^n$, if $g(y) < g(x)$, then $g^\circ(x; y - x) < 0$

To show that it is well-behaved, we need the following result.

Lemma 3 ([45, Lemma 5.3]) *If a function g is ∂°-pseudoconvex, then for every $x, y \in \mathbb{R}^n$, if $g(y) = g(x)$, then $g^\circ(x; y - x) \leq 0$. In particular, if $g(y) \leq g(x)$, then $g^\circ(x; y - x) \leq 0$.*

The contrapositive of the last statement is if $g^\circ(x; y - x) > 0$, then $g(y) > g(x)$. As $g^\circ(x; \cdot)$ is positively homogeneous [44, Proposition 2.1.1], we conclude that if g is ∂°-pseudoconvex, $g^\circ(x; d) > 0$ for some $d \in \mathbb{R}^n$, and $t > 0$, then $g(x + td) > g(x)$. Thus, if g is ∂°-pseudoconvex, then Clarke's directional derivative is well-behaved.

Now we are ready to prove the main result of this section. Recall that $J_0(x) = \{j \in J : g_j(x) = 0\}$.

Theorem 3 *Let $C = \{x : g_j(x) \leq 0, j \in J\}$ be such that C is convex, closed, and $0 \in \mathring{C}$. Assume that for each $x \in C$ and $j \in J_0(x)$, the function g_j has a well-behaved generalized directional derivative at x denoted by h_j, and that it admits a generalized subdifferential, $\partial^* g_j(x)$. Furthermore, assume that*

$$\partial^* g_j(x) \setminus \{0\} \neq \emptyset \text{ for all } x \in C \text{ and } j \in J_0(x). \tag{10}$$

Let φ_C be the gauge function of C. For $\bar{x} \notin C$, define $\hat{x} = \frac{\bar{x}}{\varphi_C(\bar{x})}$. Then, for every $j \in J_0(\hat{x})$ and every $v \in \partial^* g_j(\hat{x}) \setminus \{0\}$, the gradient cut, $g_j(\hat{x}) + v^T(x - \hat{x}) \leq 0$, is a valid supporting inequality for C that separates $\bar{x}$.

Proof By Proposition 2 we have that $\hat{x} \in \partial C$. Let $j \in J_0(\hat{x})$ and let us a consider an arbitrary $v \in \partial^* g_j(\hat{x}) \setminus \{0\}$. The gradient cut of g_j at $\hat{x}$ is $v^\mathsf{T}(x - \hat{x}) \leq 0$.

We first show that the gradient cut is valid, that is, $v^\mathsf{T}(y - \hat{x}) \leq 0$ for all $y \in C$. If this is not the case, then there exists $y_0 \in C$ for which $v^\mathsf{T}(y_0 - \hat{x}) > 0$.
Since g_j admits a generalized subdifferential at $\hat{x}$, we have that

$$h_j(\hat{x}; y_0 - \hat{x}) = \sup_{\eta \in \partial^* g_j(\hat{x})} \eta^\mathsf{T}(y_0 - \hat{x}).$$

As $v \in \partial^* g_j(\hat{x})$, it follows that $h_j(\hat{x}; y_0 - \hat{x}) > 0$. Since h_j is well-behaved, there is a sufficiently small $t \in (0, 1)$ such that $g_j(\hat{x} + t(y_0 - \hat{x})) > 0$. Thus, $\hat{x} + t(y_0 - \hat{x}) \notin C$. However, the convexity of C implies that $\hat{x} + \lambda(y_0 - \hat{x}) \in C$ for $\lambda \in [0, 1]$, which is a contradiction.

The fact that the gradient cut separates $\bar{x}$ follows from Lemma 1. Note that $v \neq 0$ by hypothesis. □

Theorem 3 extends the algorithm of Veinott [3] to further representations of the set C. In particular, it implies that the ESH converges (via an argument similar to Theorem 2's proof) when the constraints admit a URC or are ∂°-pseudoconvex. Thus, it generalizes the result of [18].

Remark 1 In [18], the authors assume that the constraint functions are ∂°-pseudoconvex. As we discussed above, for these functions the Clarke's directional derivative is well-behaved. However, being ∂°-pseudoconvex is a rather global property. In particular, if g is ∂°-pseudoconvex and $g^\circ(x; d) > 0$, then g is increasing in the direction d from x.

Theorem 3 states that the ESH will converge even if we only have this property locally. Indeed, a well-behaved Clarke differentiable function g satisfies the following property: If $g^\circ(x; d) > 0$, then for every $\varepsilon > 0$ there is a $t \in (0, \varepsilon)$ such that $g(x + td) > g(x)$. Thus, Theorem 3 includes functions that are not pseudoconvex. A simple example is $x \mapsto x^3 - x - 1$. □

Remark 2 Any representation of a convex set C as $\{x \in \mathbb{R}^n : g_j(x) \leq 0, j \in J\}$ yields a way to evaluate its gauge function, namely,

$$\varphi_C(x) = \inf \left\{ t > 0 : \max_j g_j\left(\frac{x}{t}\right) = 0 \right\}.$$

This infimum can be computed using a line search procedure.

However, what is more important is the ability to compute subgradients. Given any method to compute subgradients of the gauge function, we can apply the KCP algorithm using the implicitly defined gauge function. This allows us, for example, to drop (10). This algorithm is more general than the one proposed by Lasserre [16], but it will not necessarily converge to a KKT point of the original problem. □

6.2 Limits to the applicability of the ESH algorithm

The idea of the proof of Theorem 3 is that since C is convex, $\hat{x} + \lambda(y - \hat{x}) \in C$ for every $y \in C$ and $\lambda \in [0, 1]$. Hence, the functions g_j do not increase when moving in the direction

$y - \hat{x}$ from $\hat{x}$. Thus, a notion of subdifferential that characterizes a well-behaved directional derivative yields valid gradient cuts. The abstract definitions introduced above try to capture this line of reasoning.

Note that this is also how the proofs of the 'only if' parts of [8, Lemma 2.2], [14, Theorem 1], [9, Proposition 2.2], and the $\subseteq$ inclusion of [11, Proposition 6] work. For example, Lasserre [8] assumes that the g_j is differentiable, in which case the generalized subdifferential is just the singleton given by the gradient and the generalized directional derivative is the classic directional derivative. Dutta and Lalitha [9] assume that the functions are locally Lipschitz and regular in the sense of Clarke.

It is a natural question to wonder how important the regularity assumption is. As the following example shows, the ESH algorithm can produce invalid cutting planes when using Clarke's subdifferential and the constraints are not regular in the sense of Clarke. In particular, this shows that, without the assumption of regularity, Clarke's directional derivative is not well-behaved, in general.

Example 4 Consider the non-convex function $g(x_1, x_2) = \max\{\min\{3x_1 + x_2, 2x_1 + 3x_2\}, x_1\}$. The set $C = \{(x_1, x_2) : g(x_1, x_2) \leq 0\}$ is convex, closed and its interior is nonempty as shown in Fig. 2. Note that as g is piecewise linear, it is globally Lipschitz continuous [46, Proposition 2.2.7]. Using [44, Theorem 2.8.1], it follows that $\partial^\circ g(0) = \text{conv}\{(3, 1), (2, 3), (1, 0)\}$. Then $2x_1 + 3x_2 \leq 0$ is a gradient cut of g at 0. However, it is not valid as $(-1, 3)$ is feasible but $-2 + 9 > 0$.

In particular, it must be that g is not regular in the sense of Clarke and that g° is not well-behaved. To see that g is not well-behaved, consider the direction $d = (-1, 1)$. Notice that $g((0, 0) + td) = tg(-1, 1) = -t$, and so g is strictly decreasing in the direction d. However, $g^\circ(0; d) = \max_{v \in \partial^\circ g(0)} -v_1 + v_2 = 1$. This also shows that g is not regular. The directional derivative of g at 0 in the direction d is $-1 \neq 1$. □

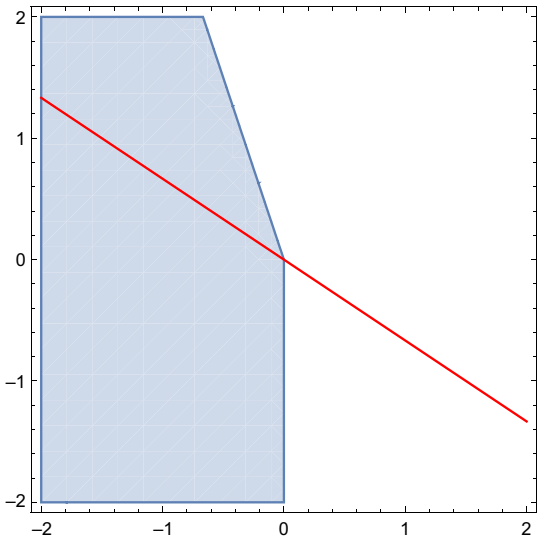

Fig. 2 Counterexample showing that, in general, the ESH algorithm can generate invalid cutting planes if the constraints are just Lipschitz continuous. The convex feasible region $\max\{\min\{3x_1 + x_2, 2x_1 + 3x_2\}, x_1\} \leq 0$ in blue and the boundary of the invalid gradient cut $2x_1 + 3x_2 \leq 0$ in red. (Color figure online)

7 Concluding remarks

In this paper, we have shown that the extended supporting hyperplane algorithm studied by Veinott [3] and Kronqvist et al. [4] is identical to Kelley's classic cutting plane algorithm applied to a suitable reformulation of the problem. We used this new perspective in order to prove the convergence of the method for the larger class of problems with convex feasible regions represented by non-convex non-smooth constraints which admit a generalized subdifferential and whose generalized directional derivative is well-behaved. This class includes ∂°-pseudoconvex functions and functions that admit a URC. Functions that admit a URC include differentiable functions and locally Lipschitz functions that are regular in the sense of Clarke. More generally, the algorithm extends to any representation of a convex set that allows to compute subgradients of its gauge function. These theoretical results bear relevance in practice, as the experimental results in [4,5] have already demonstrated the computational benefits of the supporting hyperplane algorithm in comparison to alternative state-of-the-art solving methods.

Acknowledgements Open Access funding provided by Projekt DEAL. The authors would like to thank Benjamin Müller, Stefan Vigerske, and Emilio Vilches for helpful discussions. They would also like to thank three anonymous reviewers for their comments that improved this manuscript.

References

1. Kelley Jr., J.E.: The cutting-plane method for solving convex programs. J. Soc. Ind. Appl. Math. **8**(4), 703–712 (1960). https://doi.org/10.1137/0108053
2. Gomory, R.E.: Outline of an algorithm for integer solutions to linear programs. Bull. Am. Math. Soc. **64**(5), 275–279 (1958). https://doi.org/10.1090/s0002-9904-1958-10224-4
3. Veinott, A.F.: The supporting hyperplane method for unimodal programming. Oper. Res. **15**(1), 147–152 (1967). https://doi.org/10.1287/opre.15.1.147
4. Kronqvist, J., Lundell, A., Westerlund, T.: The extended supporting hyperplane algorithm for convex mixed-integer nonlinear programming. J. Global Optim. **64**(2), 249–272 (2016). https://doi.org/10.1007/s10898-015-0322-3
5. Kronqvist, J., Bernal, D.E., Lundell, A., Grossmann, I.E.: A review and comparison of solvers for convex MINLP. Optim. Eng. **20**(2), 397–455 (2018). https://doi.org/10.1007/s11081-018-9411-8
6. Rockafellar, R.T.: Convex Analysis. Princeton University Press, Princeton (1970)
7. Horst, R., Tuy, H.: Global Optimization. Springer, Berlin (1990). https://doi.org/10.1007/978-3-662-02598-7
8. Lasserre, J.B.: On representations of the feasible set in convex optimization. Optimization Letters **4**(1), 1–5 (2009). https://doi.org/10.1007/s11590-009-0153-6
9. Dutta, J., Lalitha, C.S.: Optimality conditions in convex optimization revisited. Optim. Lett. **7**(2), 221–229 (2011). https://doi.org/10.1007/s11590-011-0410-3
10. Clarke, F.H.: Optimization and nonsmooth analysis. Soc. Ind. Appl. Math. (1990). https://doi.org/10.1137/1.9781611971309
11. Martínez-Legaz, J.E.: Optimality conditions for pseudoconvex minimization over convex sets defined by tangentially convex constraints. Optim. Lett. **9**(5), 1017–1023 (2014). https://doi.org/10.1007/s11590-014-0822-y

12. Lemaréchal, C.: An introduction to the theory of nonsmooth optimization. Optimization **17**(6), 827–858 (1986). https://doi.org/10.1080/02331938608843204
13. Pshenichnyi, B.N.: Necessary Conditions for an Extremum. Marcel Dekker Inc, New York (1971)
14. Kabgani, A., Soleimani-damaneh, M., Zamani, M.: Optimality conditions in optimization problems with convex feasible set using convexificators. Math. Methods Oper. Res. **86**(1), 103–121 (2017). https://doi.org/10.1007/s00186-017-0584-2
15. Jeyakumar, V., Luc, D.T.: Nonsmooth calculus, minimality, and monotonicity of convexificators. J. Optim. Theory Appl. **101**(3), 599–621 (1999). https://doi.org/10.1023/a:1021790120780
16. Lasserre, J.B.: On convex optimization without convex representation. Optim. Lett. **5**(4), 549–556 (2011). https://doi.org/10.1007/s11590-011-0323-1
17. Lasserre, J.B.: Erratum to: on convex optimization without convex representation. Optim. Lett. **8**(5), 1795–1796 (2014). https://doi.org/10.1007/s11590-014-0735-9
18. Eronen, V.P., Kronqvist, J., Westerlund, T., Mäkelä, M.M., Karmitsa, N.: Method for solving generalized convex nonsmooth mixed-integer nonlinear programming problems. J. Global Optim. **69**(2), 443–459 (2017). https://doi.org/10.1007/s10898-017-0528-7
19. Duran, M.A., Grossmann, I.E.: An outer-approximation algorithm for a class of mixed-integer nonlinear programs. Math. Program. **36**(3), 307–339 (1986). https://doi.org/10.1007/bf02592064
20. Fletcher, R., Leyffer, S.: Solving mixed integer nonlinear programs by outer approximation. Math. Program. **66**(1), 327–349 (1994). https://doi.org/10.1007/BF01581153
21. Eronen, V.P., Mäkelä, M.M., Westerlund, T.: On the generalization of ECP and OA methods to nonsmooth convex MINLP problems. Optimization **63**(7), 1057–1073 (2012). https://doi.org/10.1080/02331934.2012.712118
22. Wei, Z., Ali, M.M.: Outer approximation algorithm for one class of convex mixed-integer nonlinear programming problems with partial differentiability. J. Optim. Theory Appl. **167**(2), 644–652 (2015). https://doi.org/10.1007/s10957-015-0715-y
23. Wei, Z., Ali, M.M.: Convex mixed integer nonlinear programming problems and an outer approximation algorithm. J. Global Optim. **63**(2), 213–227 (2015). https://doi.org/10.1007/s10898-015-0284-5
24. Geoffrion, A.M.: Generalized benders decomposition. J. Optim. Theory Appl. **10**(4), 237–260 (1972). https://doi.org/10.1007/bf00934810
25. Quesada, I., Grossmann, I.E.: An LP/NLP based branch and bound algorithm for convex minlp optimization problems. Comput. Chem. Eng. **16**(10–11), 937–947 (1992)
26. Wei, Z., Ali, M.M.: Generalized benders decomposition for one class of MINLPs with vector conic constraint. SIAM J. Optim. **25**(3), 1809–1825 (2015). https://doi.org/10.1137/140967519
27. Westerlund, T., Pettersson, F.: An extended cutting plane method for solving convex MINLP problems. Comput. Chem. Eng. **19**, 131–136 (1995). https://doi.org/10.1016/0098-1354(95)87027-x
28. Westerlund, T., Skrifvars, H., Harjunkoski, I., Pörn, R.: An extended cutting plane method for a class of non-convex MINLP problems. Comput. Chem. Eng. **22**(3), 357–365 (1998). https://doi.org/10.1016/s0098-1354(97)00000-8
29. Plastria, F.: Lower subdifferentiable functions and their minimization by cutting planes. J. Optim. Theory Appl. **46**(1), 37–53 (1985). https://doi.org/10.1007/bf00938758
30. Eronen, V.P., Mäkelä, M.M., Westerlund, T.: Extended cutting plane method for a class of nonsmooth nonconvex MINLP problems. Optimization (2013). https://doi.org/10.1080/02331934.2013.796473
31. Westerlund, T., Eronen, V.P., Mäkelä, M.M.: On solving generalized convex MINLP problems using supporting hyperplane techniques. J. Global Optim. **71**(4), 987–1011 (2018). https://doi.org/10.1007/s10898-018-0644-z
32. Belotti, P., Lee, J., Liberti, L., Margot, F., Wächter, A.: Branching and bounds tightening techniques for non-convex MINLP. Optim. Methods Softw. **24**(4–5), 597–634 (2009)
33. Prékopa, A., Szántai, T.: Flood control reservoir system design using stochastic programming. In: Mathematical Programming in Use, pp. 138–151. Springer, Berlin (1978). https://doi.org/10.1007/bfb0120831
34. Hiriart-Urruty, J.B., Lemaréchal, C.: Convex Analysis and Minimization Algorithms II. Springer, Berlin (1993). https://doi.org/10.1007/978-3-662-06409-2
35. de Oliveira, W.: Regularized optimization methods for convex MINLP problems. TOP **24**(3), 665–692 (2016). https://doi.org/10.1007/s11750-016-0413-4
36. van Ackooij, W., Finardi, E.C., Ramalho, G.M.: An exact solution method for the hydrothermal unit commitment under wind power uncertainty with joint probability constraints. IEEE Trans. Power Syst. **33**(6), 6487–6500 (2018). https://doi.org/10.1109/tpwrs.2018.2848594
37. van Ackooij, W., Henrion, R., Möller, A., Zorgati, R.: Joint chance constrained programming for hydro reservoir management. Optim. Eng. (2013). https://doi.org/10.1007/s11081-013-9236-4
38. van Ackooij, W., de Oliveira, W.: Convexity and optimization with copulæ structured probabilistic constraints. Optimization **65**(7), 1349–1376 (2016). https://doi.org/10.1080/02331934.2016.1179302

39. Arnold, T., Henrion, R., Möller, A., Vigerske, S.: A mixed-integer stochastic nonlinear optimization problem with joint probabilistic constraints. Stoch. Program. E-print Ser. (2013). https://doi.org/10.18452/8435
40. Prékopa, A.: Stochastic Programming. Springer Netherlands (1995). https://doi.org/10.1007/978-94-017-3087-7. 10.1007%2F978-94-017-3087-7
41. Prékopa, A., Szántai, T.: Flood control reservoir system design using stochastic programming. In: Balinski, M.L., Lemarechal, C. (eds.) Mathematical Programming in Use, pp. 138–151. Springer, Berlin (1978). https://doi.org/10.1007/bfb0120831
42. Szántai: Numerical Techniques for Stochastic Optimization, chap. A computer code for solution of probabilistic-constrained stochastic programming problems, pp. 229–235. Springer, (1988)
43. Tuy, H.: Convex Analysis and Global Optimization. Springer, Berlin (2016). https://doi.org/10.1007/978-3-319-31484-6
44. Clarke, F.H., Ledyaev, Y.S., Stern, R.J., Wolenski, P.R.: Nonsmooth Analysis and Control Theory. Springer, New York (1998). https://doi.org/10.1007/b9765010.1007/b97650
45. Bagirov, A., Karmitsa, N., Mäkelä, M.M.: Introduction to Nonsmooth Optimization. Springer, Berlin (2014). https://doi.org/10.1007/978-3-319-08114-4
46. Scholtes, S.: Introduction to Piecewise Differentiable Equations. Springer, New York (2012). https://doi.org/10.1007/978-1-4614-4340-7