

# Method for solving generalized convex nonsmooth mixed-integer nonlinear programming problems

Ville-Pekka Eronen · Jan Kronqvist ·  
Tapio Westerlund · Marko M. Mäkelä ·  
Napsu Karmitsa

Received: date / Accepted: date

**Abstract** In this paper, we generalize the extended supporting hyperplane algorithm for a convex continuously differentiable mixed-integer nonlinear programming problem to solve a wider class of nonsmooth problems. The generalization is made by using the subgradients of the Clarke subdifferential instead of gradients. Consequently, all the functions in the problems are assumed to be locally Lipschitz continuous. The algorithm is shown to converge to a global minimum of an MINLP problem if the objective function is convex and the constraint functions are  $f^\circ$ -pseudoconvex. With some additional assumptions, the constraint functions may be  $f^\circ$ -quasiconvex.

**Keywords** MINLP · Extended supporting hyperplane method · Convex optimization · Nonsmooth optimization · Clarke subdifferential · Generalized convexities

**Mathematics Subject Classification (2000)** 90C11 · 90C25

## 1 Introduction

The extended supporting hyperplane (ESH) algorithm to solve smooth (continuously differentiable) convex MINLP problems was presented in [10]. It is based on the classical supporting hyperplane method derived in [16]. Numerical comparisons in [10] suggest that the ESH algorithm is efficient and on par with current state-of-art MINLP solvers when solving MINLP problems with smooth convex objective and constraint functions.

Motivated by promising numerical results in [10], we will generalize the ESH algorithm to cover certain classes of nonsmooth convex MINLP problems. By a

---

Ville-Pekka Eronen, Marko M. Mäkelä and Napsu Karmitsa  
Department of Mathematics and Statistics  
University of Turku, Turku, Finland  
E-mail: vpoero@utu.fi

Tapio Westerlund and Jan Kronqvist  
Optimization and Systems Engineering  
Åbo Akademi, Turku, Finland

convex MINLP problem we mean that the objective function is convex and the feasible set is convex when discrete variables are relaxed to continuous ones. We require that the constraint functions are  $f^\circ$ -pseudoconvex. With an additional assumption that the subdifferentials of active constraint functions do not contain zero at points where supporting hyperplanes are computed, the constraint functions may be  $f^\circ$ -quasiconvex. These function classes are modifications of the classical pseudo- and quasiconvexity for a locally Lipschitz continuous function. With these constraint functions the feasible set is convex if the integers are relaxed to continuous variables. The generalization of the ESH algorithm follows similar steps to the generalization of the  $\alpha$ ECP algorithm in [7]. That is, instead of gradients we will use the subgradients of the Clarke subdifferential.

The generalization implies that the ESH algorithm presented in [10] is suitable for smooth pseudoconvex constraint functions as well. In fact, this was essentially noted in [16], but the term pseudoconvexity was not used.

The ESH and  $\alpha$ ECP methods share many similarities. In fact, the supporting hyperplanes were seen as an alternative to the cutting planes for the  $\alpha$ ECP algorithm in [14]. This will essentially lead to the ESH algorithm. Both methods solve a sequence of MILP problems and add linear constraints to the MILP problem. The linear constraints that the ESH method generates are supporting hyperplanes to the feasible set defined by the nonlinear constraints. Thus, a supporting hyperplane usually cuts off more of the infeasible region than a cutting plane. The downsides of ESH are that we have to know an inner point of the integer relaxed feasible set and we have to use a line search procedure.

The ESH algorithm is presented briefly in Section 3. In that section, we also prove that it can solve problems with  $f^\circ$ -pseudoconvex constraint functions. Some numerical examples are solved in Section 4. Concluding remarks are given in Section 5.

## 2 Preliminaries

In this section, we present some results on nonsmooth analysis and give the definitions of the generalized convexities that we use. First, we define the generalization of the gradient.

**Definition 1** [5] Let  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  be locally Lipschitz continuous at  $\mathbf{x} \in \mathbb{R}^n$ . The *Clarke generalized directional derivative* of  $f$  at  $\mathbf{x}$  in the direction  $\mathbf{d} \in \mathbb{R}^n$  is defined by

$$f^\circ(\mathbf{x}; \mathbf{d}) := \limsup_{\substack{\mathbf{y} \rightarrow \mathbf{x} \\ t \downarrow 0}} \frac{f(\mathbf{y} + t\mathbf{d}) - f(\mathbf{y})}{t}$$

and the *Clarke subdifferential* of  $f$  at  $\mathbf{x}$  by

$$\partial f(\mathbf{x}) := \{\boldsymbol{\xi} \in \mathbb{R}^n \mid f^\circ(\mathbf{x}; \mathbf{d}) \geq \boldsymbol{\xi}^T \mathbf{d} \text{ for all } \mathbf{d} \in \mathbb{R}^n\}.$$

Each element  $\boldsymbol{\xi} \in \partial f(\mathbf{x})$  is called a *subgradient* of  $f$  at  $\mathbf{x}$ .

Some basic properties of the subdifferential are given in the next theorem. The proofs can be found in [5].

**Theorem 1** Let  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  be locally Lipschitz continuous. Then

- (i)  $\partial f(\mathbf{x})$  is a nonempty, convex and compact set.
- (ii)  $f^\circ(\mathbf{x}; \mathbf{d}) = \max \{\boldsymbol{\xi}^T \mathbf{d} \mid \boldsymbol{\xi} \in \partial f(\mathbf{x})\}$  for all  $\mathbf{d} \in \mathbb{R}^n$ .

The following definitions present the main function classes we are dealing with.

**Definition 2** A function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is *pseudoconvex*, if it is smooth and for all  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$

$$f(\mathbf{y}) < f(\mathbf{x}) \quad \text{implies} \quad \nabla f(\mathbf{x})^T (\mathbf{y} - \mathbf{x}) < 0.$$

**Definition 3** A function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is  *$f^\circ$ -pseudoconvex* ( *$f^\circ$ -quasiconvex*), if it is locally Lipschitz continuous and for all  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$

$$f(\mathbf{y}) < (\leq) f(\mathbf{x}) \quad \text{implies} \quad f^\circ(\mathbf{x}; \mathbf{y} - \mathbf{x}) < (\leq) 0.$$

Some basic properties of these function classes can be found in e.g. [2]. The following results can be found on the pages 140-166.

**Theorem 2** Let  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  be locally Lipschitz continuous.

- (i) If  $f$  is convex or pseudoconvex, then it is  $f^\circ$ -pseudoconvex.
- (ii) If  $f$  is  $f^\circ$ -pseudoconvex, then it is  $f^\circ$ -quasiconvex.
- (iii) If  $f$  is  $f^\circ$ -pseudoconvex, then  $\mathbf{0} \in \partial f(\mathbf{x})$  implies that  $\mathbf{x}$  is a global minimizer of  $f$ .
- (iv) If  $f$  is  $f^\circ$ -quasiconvex, then it is quasiconvex.
- (v) If  $f_i$ ,  $i = 1, 2, \dots, m$  are  $f^\circ$ -pseudoconvex ( $f^\circ$ -quasiconvex), then  $\max_i f_i$  is  $f^\circ$ -pseudoconvex ( $f^\circ$ -quasiconvex).

As can be seen from Theorem 2 (i)  $f^\circ$ -pseudoconvexity is a generalization of the classical pseudoconvexity. By definition, the level sets of a quasiconvex function are convex. Thus, Theorem 2 (ii) and (iv) implies that the level sets of  $f^\circ$ -pseudoconvex or  $f^\circ$ -quasiconvex functions are also convex. The following result is a straightforward consequence of Theorem 2 (ii) and (iii).

**Corollary 1** Let  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  be an  $f^\circ$ -pseudoconvex function and  $\mathbf{x} \in \mathbb{R}^n$ . If there exists  $\mathbf{y} \in \mathbb{R}^n$  such that  $f(\mathbf{y}) < f(\mathbf{x})$ , then  $f$  is  $f^\circ$ -quasiconvex and  $\mathbf{0} \notin \partial f(\mathbf{x})$ .

With a certain assumption,  $f^\circ$ -quasiconvexity implies  $f^\circ$ -pseudoconvexity. For the proof, we need the following lemma, which is also useful in the next section.

**Lemma 1** Let  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  be an  $f^\circ$ -quasiconvex function,  $\mathbf{y} \in \mathbb{R}^n$ ,  $a > f(\mathbf{y})$  and  $A \subset \mathbb{R}^n$ . If  $a \leq f(\mathbf{x})$  and  $\mathbf{0} \notin \partial f(\mathbf{x})$  for all  $\mathbf{x} \in A$ , then there exists  $r > 0$  such that  $\boldsymbol{\xi}^T (\mathbf{y} - \mathbf{x}) \leq -r \|\boldsymbol{\xi}\| < 0$  for all  $\mathbf{x} \in A$  and  $\boldsymbol{\xi} \in \partial f(\mathbf{x})$ .

*Proof* Since  $f$  is continuous and  $f(\mathbf{y}) < a$ , there exists  $r > 0$  such that  $f(\mathbf{z}) < a$  for all  $\mathbf{z} \in \mathbb{R}^n$  such that  $\|\mathbf{z} - \mathbf{y}\| \leq r$ . Let  $\mathbf{x} \in A$  and  $\boldsymbol{\xi} \in \partial f(\mathbf{x})$  be arbitrary. Since  $\mathbf{0} \notin \partial f(\mathbf{x})$  we may define  $\hat{\mathbf{y}} = \mathbf{y} + \frac{\boldsymbol{\xi}}{\|\boldsymbol{\xi}\|} r$ . The  $f^\circ$ -quasiconvexity of  $f$  and the inequalities  $f(\hat{\mathbf{y}}) < a \leq f(\mathbf{x})$  imply

$$f^\circ(\mathbf{x}; \hat{\mathbf{y}} - \mathbf{x}) \leq 0. \tag{1}$$

By Theorem 1 (ii), inequality (1) implies  $\boldsymbol{\xi}^T (\hat{\mathbf{y}} - \mathbf{x}) \leq 0$ . Thus,

$$\boldsymbol{\xi}^T (\mathbf{y} - \mathbf{x}) = \boldsymbol{\xi}^T \left( \hat{\mathbf{y}} - \frac{\boldsymbol{\xi}}{\|\boldsymbol{\xi}\|} r - \mathbf{x} \right) = -r \|\boldsymbol{\xi}\| + \boldsymbol{\xi}^T (\hat{\mathbf{y}} - \mathbf{x}) \leq -r \|\boldsymbol{\xi}\|.$$

Since  $\mathbf{0} \notin \partial f(\mathbf{x})$  we have  $-r \|\boldsymbol{\xi}\| < 0$  proving the lemma.  $\square$

**Theorem 3** Let  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  be an  $f^\circ$ -quasiconvex function. If  $\mathbf{0} \in \partial f(\mathbf{x})$  implies that  $\mathbf{x}$  is a global minimum of  $f$ , then  $f$  is  $f^\circ$ -pseudoconvex.

*Proof* Let  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$  be points such that  $f(\mathbf{y}) < f(\mathbf{x})$ . By assumption  $\mathbf{0} \notin \partial f(\mathbf{x})$ . Choosing  $a = f(\mathbf{x})$  and  $A = \{\mathbf{x}\}$ , Lemma 1 implies

$$f^\circ(\mathbf{x}, \mathbf{y} - \mathbf{x}) = \max_{\boldsymbol{\xi} \in \partial f(\mathbf{x})} \boldsymbol{\xi}^T (\mathbf{y} - \mathbf{x}) < 0.$$

Thus,  $f$  is  $f^\circ$ -pseudoconvex.  $\square$

We denote the closed line segment  $[\mathbf{x}, \mathbf{y}] = \{\lambda \mathbf{x} + (1 - \lambda) \mathbf{y} \mid 0 \leq \lambda \leq 1\}$  and  $(\mathbf{x}, \mathbf{y})$  by the corresponding open line segment.

### 3 The ESH algorithm

In this section, we consider the ESH algorithm and its convergence properties. The global convergence is proved for a problem with a linear objective function and  $f^\circ$ -pseudoconvex constraint functions. With an additional assumption, the constraint functions may be  $f^\circ$ -quasiconvex. Note that any nonlinear convex objective function can be transformed to a linear objective function and a convex constraint. We begin by reformulating the ESH algorithm from [10] to deal with nonsmooth functions. After that, we prove the convergence results for an MINLP problem with  $f^\circ$ -pseudoconvex constraint functions.

#### 3.1 Convex constraint functions

The considered problem is

$$\begin{aligned} \min \quad & \mathbf{c}^T \mathbf{x} \\ \text{s. t.} \quad & g_m(\mathbf{x}) \leq 0 \quad \forall m = 1, \dots, M, \quad (\text{P}) \\ & \mathbf{x} \in L \cap Y, \end{aligned}$$

where each nonlinear function  $g_m : \mathbb{R}^n \rightarrow \mathbb{R}$  is convex but not necessarily smooth and the compact set  $L \subset \mathbb{R}^n$  defines linear constraints. Integer variables are defined by index set  $I_{\mathbb{Z}} \subseteq \{1, 2, \dots, n\}$  and set  $Y = \{\mathbf{x} \mid \mathbf{x} \in \mathbb{R}^n, \mathbf{x}_i \in \mathbb{Z} \text{ if } i \in I_{\mathbb{Z}}\}$ . Denote  $C_m = \{\mathbf{x} \mid g_m(\mathbf{x}) \leq 0\}$  and  $C = \bigcap_{m=1}^M C_m$ . Thus, the feasible set is  $C \cap L \cap Y$ . Denoting  $F(\mathbf{x}) = \max_m \{g_m(\mathbf{x})\}$  we can also write  $C = \{\mathbf{x} \mid F(\mathbf{x}) \leq 0\}$ . We denote indexes of active constraint functions on the set  $\{\mathbf{x} \mid F(\mathbf{x}) = 0\}$  by  $I_0(\mathbf{x}) = \{m \mid g_m(\mathbf{x}) = F(\mathbf{x}) = 0\}$ . For the moment, we assume that the *Slater constraint qualification* holds true. In other words,

$$\text{there exists a point } \mathbf{x} \in L \text{ such that } F(\mathbf{x}) < 0. \quad (2)$$

The algorithm will need a point  $\mathbf{x}_{\text{NLP}}$  such that  $F(\mathbf{x}_{\text{NLP}}) < 0$ . This can be obtained by solving the problem

$$\begin{aligned} \min \quad & F(\mathbf{x}) \quad (\text{NLP}) \\ \text{s. t.} \quad & \mathbf{x} \in L. \end{aligned}$$

Note that the problem (NLP) does not need to be solved to a global minimum. It suffices to find a point  $\mathbf{x}$  such that  $F(\mathbf{x}) < 0$ . The idea of the ESH algorithm is to solve a sequence of mixed-integer linear programming problems

$$\begin{aligned} \min \quad & \mathbf{c}^T \mathbf{x} \\ \text{s. t.} \quad & l_j(\mathbf{x}^j) \leq 0 \quad j = 1, 2, \dots, k-1 \quad (\text{MILP}_k) \\ & \mathbf{x} \in L \cap Y, \end{aligned}$$

where  $l_j(\mathbf{x}^j) \leq 0$  is a supporting hyperplane generated at a point  $\mathbf{x}^j$  such that  $F(\mathbf{x}^j) = 0$ . At the first iteration, no such planes exist. Solving (MILP<sub>1</sub>) then gives us a solution point  $\mathbf{x}_{\text{MILP}}^1$ . If  $F(\mathbf{x}_{\text{MILP}}^1) \leq 0$ , we have thus found a feasible point to the original problem (P). It will be a global minimum since it was found through minimization on a set containing the original feasible set.

Suppose then that  $F(\mathbf{x}_{\text{MILP}}^1) > 0$ . The point  $\mathbf{x}^1$  where the first supporting hyperplane is generated will be found through a line search between points  $\mathbf{x}_{\text{NLP}}$  and  $\mathbf{x}_{\text{MILP}}^1$ . Since  $F(\mathbf{x}_{\text{NLP}}) < 0 < F(\mathbf{x}_{\text{MILP}}^1)$  and the constraint functions are continuous, a point  $\mathbf{x}^1 \in [\mathbf{x}_{\text{NLP}}, \mathbf{x}_{\text{MILP}}^1]$  such that  $F(\mathbf{x}^1) = 0$  is guaranteed to exist. Since the constraint functions are convex so is  $F$  implying  $\mathbf{x}^1$  will be unique. A new problem (MILP<sub>2</sub>) will be formed by adding to (MILP<sub>1</sub>) the supporting hyperplane  $l_1(\mathbf{x}^1)$ . More accurately,  $l_1(\mathbf{x}^1) := \boldsymbol{\xi}^T(\mathbf{x} - \mathbf{x}^1) \leq 0$ , where  $\boldsymbol{\xi} \in \partial g_m(\mathbf{x}^1)$  for some  $m \in I_0(\mathbf{x}^1)$ . The ESH algorithm will continue solving problems (MILP<sub>k</sub>) accordingly until a stopping criterion is satisfied. The algorithm is as follows:

---

### Algorithm 3.1 The ESH algorithm

---

Give a tolerance parameter  $\varepsilon_g > 0$  and set  $k = 1$ .

1. If  $\mathbf{x}_{\text{NLP}}$  such that  $F(\mathbf{x}_{\text{NLP}}) < 0$  is not available, find that kind of point by solving (NLP).
  2. Solve the problem (MILP<sub>k</sub>). Denote the solution by  $\mathbf{x}_{\text{MILP}}^k$ .
  3. If  $F(\mathbf{x}_{\text{MILP}}^k) \leq \varepsilon_g$  then stop:  $\mathbf{x}_{\text{MILP}}^k$  is the final solution.  
Otherwise, find  $\mathbf{x}^k \in [\mathbf{x}_{\text{NLP}}, \mathbf{x}_{\text{MILP}}^k]$  such that  $F(\mathbf{x}^k) = \frac{\varepsilon_g}{2}$  with a line search.
  4. Generate (MILP<sub>k+1</sub>) by adding to (MILP<sub>k</sub>) the supporting hyperplane  $\boldsymbol{\xi}^T(\mathbf{x} - \mathbf{x}^k) \leq 0$ , where  $\boldsymbol{\xi} \in \partial g_m(\mathbf{x}^k)$  and  $m \in I_0(\mathbf{x}^k)$ .
  5. Set  $k = k + 1$  and go to step 2.
- 

Note that if  $\varepsilon_g > 0$  then we will make the supporting hyperplanes on points  $\mathbf{x}^k$  such that  $F(\mathbf{x}^k) = \frac{\varepsilon_g}{2} \neq 0$ . This ploy allows us to deal with problems that do not satisfy the Slater constraint qualification. In the interim, we will consider the theoretical case  $\varepsilon_g = 0$ .

In [10] it was shown that the ESH algorithm will converge to a global minimum if the constraint functions are smooth and convex (and  $\varepsilon_g = 0$ ). In the algorithm presented in [10] an LP step was used in order to speed up the algorithm. In this step the (MILP<sub>k</sub>) problem is solved with integer variables relaxed to continuous ones. This allows faster generation of supporting hyperplanes as an LP problem is easier to solve than an MILP problem. The LP step will stop after a certain amount of iteration. After the LP step the algorithm starts solving MILP problems as stated but now there are supporting hyperplanes already in (MILP<sub>1</sub>) giving an initial approximation of the feasible set. The LP step does not affect global convergence although it can result in faster convergence.

### 3.2 $f^\circ$ -pseudoconvex constraint functions

At this point, we are ready to prove that the ESH algorithm can be used successfully for a wider class of problems than those with a smooth and convex objective and constraint functions. We shall first consider the case  $\varepsilon_g = 0$  in which case we have to require that the Slater constraint qualification holds true. We shall assume that

- i)  $g_m$  is  $f^\circ$ -quasiconvex for all  $m = 1, 2, \dots, M$
- ii)  $\mathbf{0} \notin \partial g_m(\mathbf{x})$  if  $m \in I_0(\mathbf{x})$ .

These conditions are fulfilled for  $f^\circ$ -pseudoconvex constraint functions by Corollary 1 and the Slater constraint qualification (2). Since the level sets of  $f^\circ$ -quasi- and  $f^\circ$ -pseudoconvex functions are convex, we are dealing with a convex MINLP problem. That is, the objective function is convex and the feasible set is a convex set, if the integer variables are relaxed to continuous ones.

The convergence proof proceeds as follows. We will first show that supporting hyperplanes do not cut off any feasible points. However, they will cut off the previous solution points of  $(\text{MILP}_k)$  that were not within the set  $C$ . In the compact set  $L$ , this results in a solution sequence that has an accumulation point. This point will be shown to be feasible in  $C$ . Finally, this point proves to be a global minimum of  $(P)$ .

We begin by proving that supporting hyperplanes, generated for an  $f^\circ$ -pseudoconvex function, will not cut off any feasible point. For differentiable functions a corresponding proof has been given, for example, in [3].

**Theorem 4** *The supporting hyperplane*

$$\boldsymbol{\xi}^T(\mathbf{x} - \mathbf{x}^k) \leq 0, \quad \boldsymbol{\xi} \in \partial g_m(\mathbf{x}^k), \quad m \in I_0(\mathbf{x}^k) \quad (3)$$

*does not cut off feasible points.*

*Proof* It is sufficient to prove that the hyperplane (3) does not cut off any points in  $C_m \supset C$ . Let  $\mathbf{y} \in C_m$  be arbitrary. Then  $g_m(\mathbf{y}) \leq 0 = g_m(\mathbf{x}^k)$  and  $f^\circ$ -quasiconvexity implies  $g_m^\circ(\mathbf{x}^k; \mathbf{y} - \mathbf{x}^k) \leq 0$ . By Theorem 1 (ii),

$$\boldsymbol{\xi}^T(\mathbf{y} - \mathbf{x}^k) \leq \max_{\boldsymbol{\zeta} \in \partial g_m(\mathbf{x}^k)} \boldsymbol{\zeta}^T(\mathbf{y} - \mathbf{x}^k) = g_m^\circ(\mathbf{x}^k; \mathbf{y} - \mathbf{x}^k) \leq 0.$$

Thus, hyperplane (3) does not cut off  $\mathbf{y}$  proving the theorem.  $\square$

The next theorem shows that, if the current solution  $\mathbf{x}_{\text{MILP}}^k$  is infeasible, then it will be cut off by the supporting hyperplane generated at point  $\mathbf{x}^k$ .

**Theorem 5** *Let  $m \in I_0(\mathbf{x}^k)$  and  $\boldsymbol{\xi} \in \partial g_m(\mathbf{x}^k)$ . If  $F(\mathbf{x}_{\text{MILP}}^k) > 0$ , then  $\boldsymbol{\xi}^T(\mathbf{x}_{\text{MILP}}^k - \mathbf{x}^k) > 0$ .*

*Proof* We may write  $\mathbf{x}^k = \lambda \mathbf{x}_{\text{NLP}} + (1 - \lambda) \mathbf{x}_{\text{MILP}}^k$ , where  $\lambda \in [0, 1]$ . In point of fact, we have  $\lambda \in (0, 1)$  since  $F(\mathbf{x}_{\text{MILP}}^k) > 0$  and  $F(\mathbf{x}_{\text{NLP}}) < 0 = F(\mathbf{x}^k)$ . It is then straightforward to show that

$$-\frac{\lambda}{1 - \lambda}(\mathbf{x}_{\text{NLP}} - \mathbf{x}^k) = \mathbf{x}_{\text{MILP}}^k - \mathbf{x}^k.$$

Since  $g_m(\mathbf{x}_{\text{NLP}}) < g_m(\mathbf{x}^k)$  and  $\mathbf{0} \notin \partial g_m(\mathbf{x}^k)$  (assumption ii)) Lemma 1 implies (by choosing  $A = \{\mathbf{x}^k\}$  and  $a = g_m(\mathbf{x}^k)$ ) that there exists  $r > 0$  such that

$$\boldsymbol{\xi}^T(\mathbf{x}_{\text{MILP}}^k - \mathbf{x}^k) = -\frac{\lambda}{1-\lambda}\boldsymbol{\xi}^T(\mathbf{x}_{\text{NLP}} - \mathbf{x}^k) \geq -\frac{\lambda}{1-\lambda}(-r\|\boldsymbol{\xi}\|) > 0$$

proving the theorem.  $\square$

With Theorems 4 and 5 we can prove the uniqueness of  $\mathbf{x}^k$ .

**Corollary 2** *If  $F(\mathbf{x}_{\text{MILP}}^k) > 0$ , then  $\mathbf{x}^k$  is unique.*

*Proof* Suppose there exist  $\mathbf{y}, \mathbf{z} \in (\mathbf{x}_{\text{NLP}}, \mathbf{x}_{\text{MILP}}^k)$  such that  $F(\mathbf{y}) = F(\mathbf{z}) = 0$  and  $\mathbf{y} \neq \mathbf{z}$ . Without loss of generality we may assume that  $\mathbf{z} \in (\mathbf{y}, \mathbf{x}_{\text{MILP}}^k)$ . By Theorem 4,  $\mathbf{z}$  is not cut off by the hyperplane generated at  $\mathbf{y}$ . However, the hyperplane will then not cut off either  $\mathbf{x}_{\text{MILP}}^k$  contradicting Theorem 5.  $\square$

Suppose that at some iteration  $k$  we have  $F(\mathbf{x}_{\text{MILP}}^k) \leq 0$ . By Theorem 4, the feasible set of  $(\text{MILP}_k)$  includes  $C$ . Thus,  $\mathbf{x}_{\text{MILP}}^k$  is a global minimum of  $(P)$ . On the other hand, if  $F(\mathbf{x}_{\text{MILP}}^k) > 0$  for all  $k$ , Theorem 5 implies that the points in sequence  $(\mathbf{x}_{\text{MILP}}^k)$  are distinct. Since  $(\mathbf{x}_{\text{MILP}}^k) \subset L$  and  $L$  is a compact set, the sequence has an accumulation point by the Bolzano-Weierstrass Theorem. Hence, there exists a converging subsequence  $(\mathbf{x}_{\text{MILP}}^{k_j}) \subset (\mathbf{x}_{\text{MILP}}^k)$ .

Next, we will show that the subsequence  $(\mathbf{x}_{\text{MILP}}^{k_j})$  converges to a feasible point. To prove this, we need the following lemma.

**Lemma 2** *Let  $(\mathbf{x}_{\text{MILP}}^{k_j})$  be a converging sequence and  $\boldsymbol{\xi}_j \in \partial g_{m_j}(\mathbf{x}^{k_j})$ , where index  $m_j \in I_0(\mathbf{x}^{k_j})$ . Then*

$$\lim_{j \rightarrow \infty} \frac{\boldsymbol{\xi}_j^T}{\|\boldsymbol{\xi}_j\|}(\mathbf{x}_{\text{MILP}}^{k_j} - \mathbf{x}^{k_j}) = 0.$$

*Proof* Let  $\varepsilon > 0$  be arbitrary. Choose  $j$  such that  $\|\mathbf{x}_{\text{MILP}}^{k_{j+1}} - \mathbf{x}_{\text{MILP}}^{k_j}\| < \varepsilon$ . Then

$$\begin{aligned} & \left| \frac{\boldsymbol{\xi}_j^T}{\|\boldsymbol{\xi}_j\|}(\mathbf{x}_{\text{MILP}}^{k_{j+1}} - \mathbf{x}^{k_j}) - \frac{\boldsymbol{\xi}_j^T}{\|\boldsymbol{\xi}_j\|}(\mathbf{x}_{\text{MILP}}^{k_j} - \mathbf{x}^{k_j}) \right| \\ & \leq \frac{\|\boldsymbol{\xi}_j\|}{\|\boldsymbol{\xi}_j\|} \|\mathbf{x}_{\text{MILP}}^{k_{j+1}} - \mathbf{x}_{\text{MILP}}^{k_j}\| < \varepsilon. \end{aligned} \quad (4)$$

Since  $F(\mathbf{x}_{\text{MILP}}^{k_j}) > 0$  Theorem 5 implies  $0 < \frac{\boldsymbol{\xi}_j^T}{\|\boldsymbol{\xi}_j\|}(\mathbf{x}_{\text{MILP}}^{k_j} - \mathbf{x}^{k_j})$ . Then, by the feasibility of  $\mathbf{x}_{\text{MILP}}^{k_{j+1}}$  in problem  $(\text{MILP}_{k_{j+1}})$

$$\frac{\boldsymbol{\xi}_j^T}{\|\boldsymbol{\xi}_j\|}(\mathbf{x}_{\text{MILP}}^{k_{j+1}} - \mathbf{x}^{k_j}) \leq 0 < \frac{\boldsymbol{\xi}_j^T}{\|\boldsymbol{\xi}_j\|}(\mathbf{x}_{\text{MILP}}^{k_j} - \mathbf{x}^{k_j})$$

for all  $j$ . Hence, by inequality (4) we have  $\lim_{j \rightarrow \infty} \frac{\boldsymbol{\xi}_j^T}{\|\boldsymbol{\xi}_j\|}(\mathbf{x}_{\text{MILP}}^{k_j} - \mathbf{x}^{k_j}) = 0$ .  $\square$

Lemma 2 does not imply, thus far, that subsequence  $(\mathbf{x}_{\text{MILP}}^{k_j})$  would converge to a feasible point. It still leaves a possibility that vectors  $\boldsymbol{\xi}_j$  and  $\mathbf{x}_{\text{MILP}}^{k_j} - \mathbf{x}^{k_j}$  converge to vectors that are perpendicular to each other. With the help of  $f^\circ$ -quasiconvexity this case will be excluded in the next theorem.

**Theorem 6** *An accumulation point of the sequence  $(\mathbf{x}_{\text{MILP}}^k)$  is a feasible point.*

*Proof* By Lemma 2 we may re-index the convergent subsequence in a way that  $\frac{\boldsymbol{\xi}_j^T}{\|\boldsymbol{\xi}_j\|}(\mathbf{x}_{\text{MILP}}^{k_j} - \mathbf{x}^{k_j}) - \frac{1}{j} \leq 0$  for all  $j \in \mathbb{N}$ , where  $\boldsymbol{\xi}_j \in \partial g_{m_j}(\mathbf{x}^{k_j})$  and  $m_j \in I_0(\mathbf{x}^{k_j})$ . Furthermore,

$$\begin{aligned} \frac{\boldsymbol{\xi}_j^T}{\|\boldsymbol{\xi}_j\|}(\mathbf{x}_{\text{MILP}}^{k_j} - \mathbf{x}^{k_j}) - \frac{1}{j} &\leq 0 \\ &= \frac{\boldsymbol{\xi}_j^T}{\|\boldsymbol{\xi}_j\|}(\mathbf{x}^{k_j} - \mathbf{x}^{k_j}) \\ &= \frac{\boldsymbol{\xi}_j^T}{\|\boldsymbol{\xi}_j\|}(\lambda_{k_j} \mathbf{x}_{\text{NLP}} + (1 - \lambda_{k_j}) \mathbf{x}_{\text{MILP}}^{k_j} - \mathbf{x}^{k_j}). \end{aligned}$$

By rearranging the terms we have

$$\lambda_{k_j} \frac{\boldsymbol{\xi}_j^T}{\|\boldsymbol{\xi}_j\|}(\mathbf{x}_{\text{MILP}}^{k_j} - \mathbf{x}^{k_j}) - \frac{1}{j} \leq \lambda_{k_j} \frac{\boldsymbol{\xi}_j^T}{\|\boldsymbol{\xi}_j\|}(\mathbf{x}_{\text{NLP}} - \mathbf{x}^{k_j}).$$

By choosing  $a = 0$  and  $A_m = \{\mathbf{x} \in L \mid F(\mathbf{x}) = 0, g_m(\mathbf{x}) = 0\}$  we have  $\mathbf{x}^{k_j} \in A_{m_j}$  and  $g(\mathbf{x}_{\text{NLP}}) < a$ . Then Lemma 1 implies

$$\lambda_{k_j} \frac{\boldsymbol{\xi}_j^T}{\|\boldsymbol{\xi}_j\|}(\mathbf{x}_{\text{NLP}} - \mathbf{x}^{k_j}) \leq -\lambda_{k_j} \frac{\|\boldsymbol{\xi}_j\|}{\|\boldsymbol{\xi}_j\|} r_{m_j} = -\lambda_{k_j} r_{m_j},$$

for some  $r_{m_j} > 0$ . Since there are finitely many constraints,  $r = \min_{m=1,2,\dots,M} \{r_m\} > 0$  exists and

$$-\lambda_{k_j} r_{m_j} \leq -\lambda_{k_j} r \text{ for all } j \in \mathbb{N}.$$

Thus,

$$\lambda_{k_j} \frac{\boldsymbol{\xi}_j^T}{\|\boldsymbol{\xi}_j\|}(\mathbf{x}_{\text{MILP}}^{k_j} - \mathbf{x}^{k_j}) - \frac{1}{j} \leq -\lambda_{k_j} r.$$

By solving  $\lambda_{k_j}$  from this inequality we obtain

$$\lambda_{k_j} \leq \frac{1}{j \left( r + \frac{\boldsymbol{\xi}_j^T}{\|\boldsymbol{\xi}_j\|}(\mathbf{x}_{\text{MILP}}^{k_j} - \mathbf{x}^{k_j}) \right)} < \frac{1}{j \cdot r}.$$

Hence,  $\lim_{j \rightarrow \infty} \lambda_{k_j} = 0$  implying  $F(\hat{\mathbf{x}}) = 0$  where  $\hat{\mathbf{x}} = \lim_{j \rightarrow \infty} \mathbf{x}_{\text{MILP}}^{k_j}$ .  $\square$

Finally, the convergence result is given.

**Theorem 7** *An accumulation point of the sequence  $(\mathbf{x}_{\text{MILP}}^k)$  is a global minimum of the problem (P).*

*Proof* The proof is similar to Lemma 4 in [10].  $\square$

Consider the case where  $\varepsilon_g > 0$ . Then, supporting hyperplanes will be generated at points  $\mathbf{x}^k$  such that  $F(\mathbf{x}^k) = \frac{\varepsilon_g}{2}$ . This implies that we do not need the Slater constraint qualification to hold true. That is, it suffices that  $F(\mathbf{x}_{\text{NLP}}) \leq 0$  as was noticed in [10]. In the case of  $f^\circ$ -quasiconvex constraint functions, we have to require that  $\mathbf{0} \notin \partial g_m(\mathbf{x})$  if the supporting hyperplane is generated from  $g_m$  at  $\mathbf{x}$ . Again, an  $f^\circ$ -pseudoconvex constraint function meets this requirement unattended.

Theorem 4 is valid in the case  $\varepsilon_g > 0$  as well. In the other results, we have to replace 0 by  $\frac{\varepsilon_g}{2}$  or  $\varepsilon_g$  when appropriate. Then it follows that if we do not find a point  $\mathbf{x}$  such that  $F(\mathbf{x}) < \frac{\varepsilon_g}{2}$ , the MILP solutions sequence has an accumulation point  $\hat{\mathbf{x}}$  with  $F(\hat{\mathbf{x}}) = \frac{\varepsilon_g}{2}$ . Thus, we will find a point  $\mathbf{x}$  satisfying  $F(\mathbf{x}) \leq \varepsilon_g$  after a finite number of iterations. If the objective function is linear, the final solution point  $\mathbf{x}_{\text{MILP}}^k$  will be an  $\varepsilon_g$ -feasible global minimum. That is,  $F(\mathbf{x}_{\text{MILP}}^k) \leq \varepsilon_g$  and there exists no feasible point giving a lower objective function value. If a convex objective function is transformed to a linear objective function, there might be a feasible solution giving a lower objective function value than the one found. Nevertheless, this difference is at most  $\varepsilon_g$ .

### 3.3 On solving the NLP problem

To be able to solve a problem the ESH algorithm requires a point  $\mathbf{x}_{\text{NLP}}$  satisfying strictly all the nonlinear constraints. As previously mentioned, this can be found by solving (NLP). The compactness of  $L$  and the continuity of  $F$  guarantees that a solution exists. If functions  $g_m$  are  $f^\circ$ -pseudoconvex then so is  $F$  by Theorem 2 (v). In this case the nonsmooth problem (NLP) can be solved by e.g. the proximal bundle (PB) algorithm [15]. If the constraint functions are  $f^\circ$ -quasiconvex then  $F$  will be  $f^\circ$ -quasiconvex by Theorem 2 (v). The PB algorithm will find a stationary point, which is not guaranteed to solve the problem nor to be a feasible point.

Note that if  $\varepsilon_g > 0$  the point  $\mathbf{x}_{\text{NLP}}$  need only to satisfy  $F(\mathbf{x}_{\text{NLP}}) \leq 0$ . Then, by relaxing the integer variables of the original problem we can obtain a continuous problem, whose global minimum point can be set to  $\mathbf{x}_{\text{NLP}}$ . In the case of  $f^\circ$ -pseudoconvex constraint functions this problem can be solved by, for example, the  $\alpha$ ECP method. Again, if the constraint functions are  $f^\circ$ -quasiconvex, a stationary point is not guaranteed to be a feasible point. Consider a simple problem

$$\begin{aligned} \min & (x - 1)^2 \\ \text{s. t.} & (x - 1)^3 + 1 \leq 0 \\ & -2 \leq x \leq 2. \end{aligned}$$

Here,  $x = 1$  is a stationary point in the both proposed feasibility problems. However, it is not a feasible point. On our best knowledge there does not exist an algorithm that is guaranteed to find a feasible point if the constraint functions are  $f^\circ$ -quasiconvex. Nevertheless, if each of the  $f^\circ$ -quasiconvex constraints satisfies the condition

$$\mathbf{0} \notin \partial g_m(\mathbf{x}) \quad \text{or} \quad g_m(\mathbf{x}) < 0 \quad (5)$$

for all  $\mathbf{x} \in \mathbb{R}^n$ , then a stationary point is a feasible point. In addition, the function  $\hat{F}(\mathbf{x}) := \max\{0, F(\mathbf{x})\}$  will be  $f^\circ$ -pseudoconvex and the integer relaxed problem can be solved by  $\alpha$ ECP. Condition (5) is the well-known *Cottle constraint qualification*.

Recall that a convex objective function  $f$  may be transformed to a constraint function  $f - \mu$ , where the auxiliary variable  $\mu$  is the new linear objective function. If the objective function is convex and nonlinear, it sounds reasonable to use the relaxed problem to find  $\mathbf{x}_{\text{NLP}}$ . In this case, we can set  $\mu_{\text{NLP}} = f(\mathbf{x}_{\text{NLP}})$ . If (NLP) is used with constraint function  $f - \mu$ , the only other constraints that include  $\mu$  are user given box constraints. Hence,  $\mu_{\text{NLP}}$  would most probably be the given upper bound being somewhat arbitrary. The value of  $\mu_{\text{NLP}}$  affects the values of  $f - \mu$  in the line search and, thus, also the frequency of computing supporting hyperplanes from  $f - \mu$ .

#### 4 Numerical considerations & examples

In this section, we solve a few example problems with the ESH and the  $\alpha$ ECP [7, 17] algorithms. In addition, we will discuss some numerical details, especially, the line search. At first we will motivate why the line search is made to find a point  $\mathbf{x}^k$  that has been shifted  $\frac{\varepsilon_g}{2}$  from the set  $\{\mathbf{x} \mid F(\mathbf{x}) = 0\}$  in the function space. Another possibility is a shift in the variable space.

##### 4.1 On the line search

In the implementation and numerical comparison, we will have  $\varepsilon_g > 0$ . In the Algorithm 3.1, we apply a line search to find a point  $\mathbf{x}^k$  such that  $F(\mathbf{x}^k) = \frac{\varepsilon_g}{2}$ . This implies that we do not need the Slater constraint qualification to hold true, but the result will be an  $\varepsilon_g$ -feasible minimum. When comparing the point  $\mathbf{x}^k$  with the one obtained by the algorithm with  $\varepsilon_g = 0$  we will make a shift  $\frac{\varepsilon_g}{2}$  in the function value space towards  $\mathbf{x}_{\text{MILP}}^k$ , that is, away from  $\mathbf{x}_{\text{NLP}}$ .

Next, we will give an example problem which shows that if  $\varepsilon_g = 0$  or even if we do similar  $\frac{\varepsilon_g}{2}$  shift in the variable space we would need the Slater constraint qualification to hold. More accurately, we will create supporting hyperplanes on points  $\mathbf{x}^k + \boldsymbol{\delta}^k$ , where  $F(\mathbf{x}^k) = 0$  and

$$\boldsymbol{\delta}^k = \frac{\varepsilon_g}{2} \frac{\mathbf{x}_{\text{MILP}}^k - \mathbf{x}^k}{\|\mathbf{x}_{\text{MILP}}^k - \mathbf{x}^k\|}. \quad (6)$$

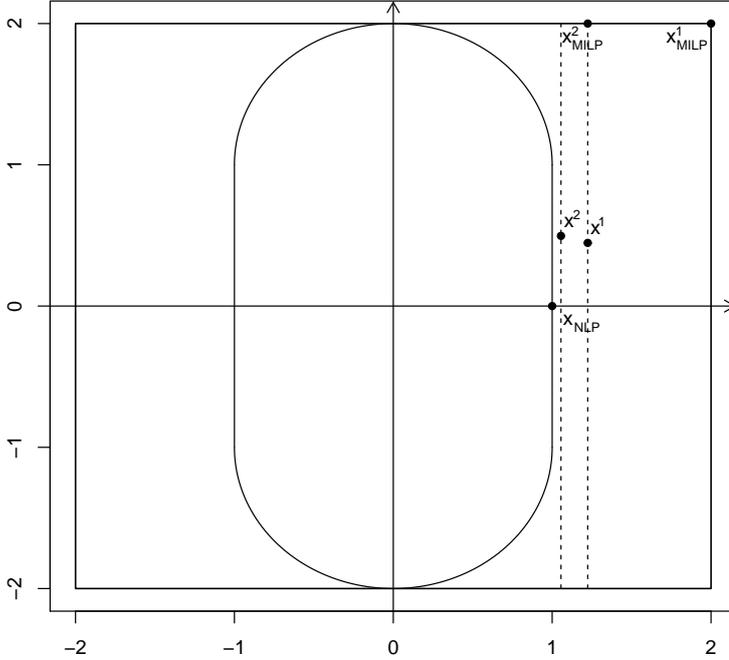
Consider problem (see Figure 1)

$$\begin{aligned} \min \quad & -x_1 - x_2 \\ \text{s. t.} \quad & g(\mathbf{x}) \leq 0 \\ & -2 \leq x_1 \leq 2, \quad -2 \leq x_2 \leq 2, \end{aligned} \quad (\text{P1})$$

where  $g(\mathbf{x}) = \max\{0, g_1(\mathbf{x})\}$  and

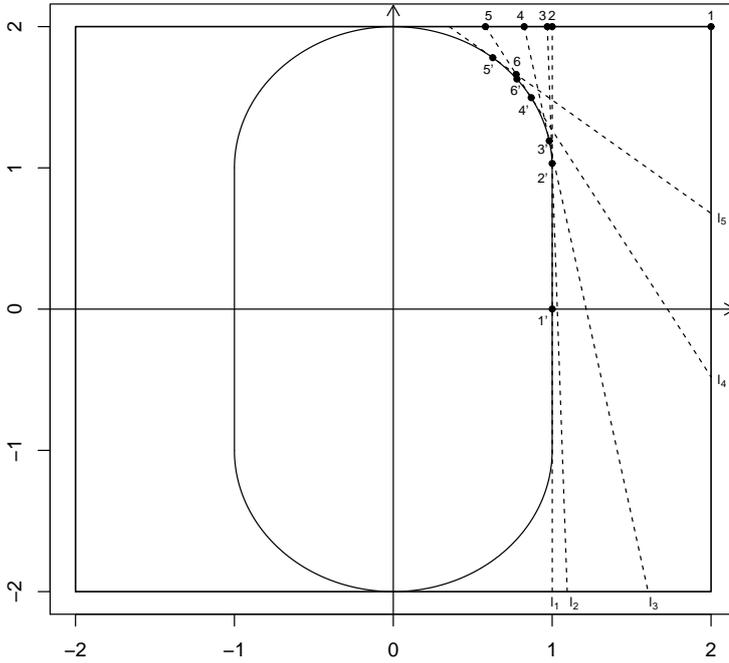
$$g_1(\mathbf{x}) = \begin{cases} (x_2 - 1)^2 + x_1^2 - 1, & x_2 > 1 \\ x_1^2 - 1, & -1 \leq x_2 \leq 1 \\ (x_2 + 1)^2 + x_1^2 - 1, & x_2 < -1 \end{cases}.$$

The constraint function  $g$  is convex and does not satisfy the Slater constraint qualification.



**Fig. 1** The feasible set and a part of the solving process of the given example when  $\delta = \frac{1}{2}$ . The dashed lines represent the first two hyperplanes and the bolded region is the feasible set.

Suppose that we have  $\mathbf{x}_{\text{NLP}} = (1,0)$ . Note that  $(1,0)$  is on the boundary. When trying to solve this problem with the ESH algorithm with shifts (6) and  $0 < \frac{\varepsilon_g}{2} < 1$ , the first solution will be  $\mathbf{x}_{\text{MILP}}^1 = (2,2)$ . The line search will find  $\mathbf{x}^k = \mathbf{x}_{\text{NLP}} = (1,0)$  and a supporting hyperplane  $\mathbf{x} \leq a$ , ( $a > 1$ ) will be added to the next MILP problem. More accurately  $a = 1 + \frac{\varepsilon_g}{2\sqrt{5}}$  according to (6), but  $a > 1$  is the property we are interested in. The next solution will be  $\mathbf{x}_{\text{MILP}}^2 = (a,2)$ . The solution process generates sequence  $(\mathbf{x}_{\text{MILP}}^k)$  that converges to the point  $(1,2)$ . However, this point is not even feasible. The ESH algorithm without the shift (6) and  $\varepsilon_g = 0$  will find  $\mathbf{x}_{\text{MILP}}^2 = (1,2)$  and become fixed there. This is due to the fact that the supporting hyperplane at  $\mathbf{x}_{\text{NLP}}$  does not cut off  $(1,2)$ . However, the Algorithm 3.1 can solve the problem when  $\varepsilon_g > 0$ . In Figure 2, the first six iterations are shown. The algorithm solves the problem after ten iterations.



**Fig. 2** Six iterations of the ESH algorithm when solving (P1) with tolerance  $\varepsilon_g = 0.001$ . Plain numbers indicate points  $x_{\text{MILP}}^k$  and dotted numbers indicate points  $x^k$ . The dashed lines represent supporting hyperplanes.

In the later examples, we use a line search based on the bisection method. This guarantees that at each iteration of the line search, we have found an interval  $[\mathbf{x}_L, \mathbf{x}_U]$  such that  $F(\mathbf{x}_L) \leq \frac{\varepsilon_g}{2}$  and  $\frac{\varepsilon_g}{2} < F(\mathbf{x}_U) \leq F(\mathbf{x}_{\text{MILP}}^k)$ . We will stop the line search when a point  $\mathbf{x}$  satisfying  $\frac{\varepsilon_g}{4} \leq F(\mathbf{x}) < \varepsilon_g$  has been found. The supporting hyperplanes will be generated at this point.

In the line search we do not need to calculate all of the constraint functions in each step. If  $g_m(\mathbf{x}_{\text{MILP}}^k) < \varepsilon_g$  then, by quasiconvexity (Theorem 2 (iv)),  $g_m(\mathbf{x}) < \varepsilon_g$  for all  $\mathbf{x} \in [\mathbf{x}_{\text{NLP}}, \mathbf{x}_{\text{MILP}}^k]$ . Hence, on the line segment  $[\mathbf{x}_{\text{NLP}}, \mathbf{x}_{\text{MILP}}^k]$ , it is enough to apply the line search on function

$$F_k(\mathbf{x}) = \max_{m \in M_k} g_m(\mathbf{x}), \text{ where } M_k = \left\{ m \mid g_m(\mathbf{x}_{\text{MILP}}^k) \geq \varepsilon_g \right\}.$$

#### 4.2 Theoretical comparison on the $\alpha$ ECP and ESH algorithms

Here we present some theoretical differences between the  $\alpha$ ECP and the ESH algorithms. For details on the  $\alpha$ ECP algorithm we refer to [7, 17].

The principal difference between the algorithms is the type of cutting plane that they use. The ESH algorithm creates supporting hyperplanes as explained in the previous section and the  $\alpha$ ECP algorithm creates  $\alpha$ -cutting planes. For a constraint function  $g_m$  the  $\alpha$ -cutting plane created at point  $\mathbf{x}_{\text{MILP}}^k$  reads

$$g_m(\mathbf{x}_{\text{MILP}}^k) + \alpha \cdot \boldsymbol{\xi}^T (\mathbf{x} - \mathbf{x}_{\text{MILP}}^k) \leq 0,$$

where  $\boldsymbol{\xi} \in \partial g_m(\mathbf{x}_{\text{MILP}}^k)$  and  $\alpha \geq 1$  is sufficiently large. For a convex function  $\alpha = 1$  is large enough but for an  $f^\circ$ -pseudoconvex function a sufficiently large  $\alpha$  value is usually not known. In practice,  $\alpha$  is first set to 1 but it is updated until the distance between the  $\alpha$ -cutting plane and  $\mathbf{x}_{\text{MILP}}^k$  is less than the given parameter  $\varepsilon_z > 0$ . Nevertheless, for a given  $\varepsilon_z$  it can not be guaranteed that an updated  $\alpha$ -cutting plane will not cut off feasible points. The supporting hyperplanes does not cut off any points from the feasible set even if the constraint functions are  $f^\circ$ -pseudoconvex. This makes ESH more appealing than  $\alpha$ ECP to solve problems involving  $f^\circ$ -pseudoconvex functions. With the ESH algorithm the nonlinear functions have to be evaluated also at points where integer variables attain non-integer values due to the line search. This restriction does not apply to  $\alpha$ ECP.

Presumably, a hyperplane that ESH creates usually cuts off more of the infeasible region than an updated  $\alpha$ -cutting plane. This could lead to a solving process with a fewer number of MILP problems. If there is more than one constraint function the approximation of the feasible set may be enhanced by adding more than one  $\alpha$ -cutting plane or supporting hyperplane per iteration. Naturally, creating more linear constraints leads to larger MILP problems. It is a straightforward process to create an  $\alpha$ -cutting plane for any violating constraint from a solution point  $\mathbf{x}_{\text{MILP}}^k$ . A supporting hyperplane may be created from a constraint that is active at the point  $\mathbf{x}^k$  at which the line search ends up. The quasiconvexity of the constraint functions implies that there are at least as many violating constraints at  $\mathbf{x}_{\text{MILP}}^k$  as there are active constraints at  $\mathbf{x}^k$ . In other words, if there are many constraint functions it is possible to make at least as many  $\alpha$ -cutting planes as supporting hyperplanes per iteration. Thus, the use of more than one cutting plane per iteration benefits  $\alpha$ ECP more than ESH.

For the ESH algorithm, the cost of solving less MILP problems is an increasing number of function evaluations per iteration due to the line search. This will most probably lead to a greater total number of nonlinear function evaluations. Nevertheless, MILP problems are usually difficult to solve and more time consuming than multiple nonlinear function evaluation. However, this is not always the case. In a chromatographic separation problem [6] some function evaluations require solving partial differential equations and are more time consuming than solving an MILP problem. Another type of time consuming constraint functions are probabilistic constraints which are considered, for example, in [1, 13].

### 4.3 Example problems

In addition to (P1) we apply the ESH algorithm to another simple example problem and to a facility layout problem [4]. The former of these problems is

$$\begin{aligned}
& \min \max \left\{ (x_1 - 2)^2, (x_2 - 4)^2 \right\} \\
& \text{s. t. } \frac{|x_1 - 3| - 10x_1}{3x_1 + x_2 + 1} + 2 \leq 0 \\
& \quad (x_1 - 7)^2 - 5x_2 \leq 0 \qquad \qquad \qquad (\text{P2}) \\
& \quad 1 \leq x_1, \quad x_2 \leq 8 \\
& \quad x_1 \geq 0, \quad x_2 \in \mathbb{Z}^+.
\end{aligned}$$

The objective function of (P2) is convex and nonsmooth. The first constraint function is  $f^\circ$ -pseudoconvex and the second constraint function is convex.

The third problem (P3) is the facility layout problem presented in [4] and [12]. The formulation of the problem is presented in the appendix. The problem (P3) was solved both by using only one linearization per iteration and by creating linearizations for all violating or active constraint functions.

The number and types of variables together with the types of constraints and objective functions are summarized in Table 1. Transformation of the convex objective function to a constraint function leads to one nonsmooth convex constraint in (P2) and 6 nonsmooth convex constraints in (P3). The use of only one objective function constraint in (P3) would result in a difficult constraint function. Since the objective function is a sum of 12  $l_1$ -norms of linear functions, the transformation to only one single constraint would have  $2^{12}$  different gradients on its domain of definition. Thus, to make a perfect approximation of the objective function,  $2^{12}$  linearizations would be needed. With 6 constraints, each will have only 4 different gradients.

**Table 1** Basic information on the example problems. Here cont=continuous, int=integers and bin=binary. In the problems (P2) and (P3) the objective function is nonsmooth. In addition, the  $f^\circ$ -pseudoconvex constraint in (P2) is nonsmooth. The other nonlinear constraints are smooth.

problem	objective	constraints			variables		
		linear	convex	$f^\circ$ -pseudoconvex	cont	int	bin
P1	linear	-	1	-	2	-	-
P2	convex	1	1	1	2	1	-
P3	convex	114	-	7	34	-	42

For the problems (P1) and (P2) the inner points are given as  $(x_1, x_2) = (1, 0)$  and  $(x_1, x_2) = (3, 4)$ , respectively. In the problem (P3) the feasibility problem is the relaxed problem discussed in Subsection 3.3. The feasibility problem was solved by the  $\alpha$ ECP algorithm.

In all of the problems, nonsmooth functions could be presented as a maximum of two functions. Consequently, a subgradient of a nonsmooth constraint function could be presented as a gradient of an active function. For example,  $h(x_1, x_2) = |x_1 - x_2|$  is a part of a constraint function in (P3) and its subdifferential is

$$\partial h(x_1, x_2) = \begin{cases} (1, -1) & , x_1 > x_2 \\ [(1, -1), (-1, 1)] & , x_1 = x_2 . \\ (-1, 1) & , x_1 < x_2 \end{cases}$$

When  $x_1 = x_2$  we use point  $(-1, 1)$  as the subgradient.

Parameters for the  $\alpha$ ECP algorithm were the same as those in problem 2 in [7]. In the ESH algorithm similar parameters were used, that is,  $\varepsilon_g = 0.001$ . Problems were solved using a 64-bit windows 7 computer with an Intel i3-2100 3.1GHz processor. The MILP problems were solved by CPLEX. Table 2 summarizes the results.

**Table 2** Numerical results. The number of function evaluations needed in the line search is in the column "f. eval. line s." The number of function and derivative evaluations, including the line search that ESH uses, is in the column "Function eval.". The number of MILP problems that was needed to solve the problem is in the column "# MILP". The number of linearizations at the last MILP problem is in the column "# lin.". Methods "ECP.all" and "ESH.all" creates linearizations for all violating/active constraints.

prob. name	method	obj. value	f. eval. line s.	function eval.	# MILP	# lin.	CPU time (s)
P1	ESH	-2.41	79	104	9	8	1.44
	$\alpha$ ECP	-2.41	-	28	10	9	1.40
P2	ESH	0.360	108	163	10	9	2.15
	$\alpha$ ECP	0.359	-	59	11	10	2.44
P3	ESH	20.73	5310	6991	108	107	240
	$\alpha$ ECP	20.73	-	1468	94	91	1180
P4	ESH.all	20.73	3742	5063	80	109	209
	$\alpha$ ECP.all	20.73	-	1173	24	239	265

The supposed strength of the ESH algorithm compared with the  $\alpha$ ECP algorithm is a fewer number of MILP problems due to the tighter underestimates of the nonlinear constraints. From Table 2, we can see that this was true only for problems (P1) and (P2) but not for problems (P3) and (P4). The  $\alpha$ ECP algorithm managed to solve all the problems in a fewer number of function evaluations, as expected. This holds true even without taking account of the effort needed to solve the feasibility problem to find the inner point. In problem (P3) solving the feasibility problem required 1468 nonlinear function evaluations and 3 seconds of CPU time. In the example problems, the nonlinear function evaluations were inexpensive. Consequently, the great difference in the number of function evaluations did not convert to a great difference in CPU time.

ESH solved problem (P3) in considerably shorter time. The reason for this is that the MILP problems that the ESH algorithm generated in (P3) were much easier than those generated by  $\alpha$ ECP. Both algorithms found the best-known solution in all of the problems. The difference between the objective function values in (P2) results from the feasibility tolerance:  $\alpha$ ECP found a solution that violates the constraints more but the violation is still within the tolerance.

From the last row of Table 2 we see that  $\alpha$ ECP benefited from using more linearizations per iteration. The number of function evaluations, the number of MILP problems and CPU time were all reduced. Even then  $\alpha$ ECP could not solve the problem faster than ESH. The size of the last MILP problem was increased as expected. In addition, while ESH also benefited from multiple hyperplanes it was not to the same extent as  $\alpha$ ECP.

Despite we can solve the nonsmooth formulation of the facility layout problem, a more efficient way to solve it is to use the same formulation as [4] and in MINLP

Library2 (<http://www.gamsworld.org/minlp/minlplib2/html/>) problem fo7. That is, the objective function is transformed to linear constraints and 7 pseudoconvex constraints are replaced by 14 convex constraints. As suggested by the results in Table 2, efficiency is further increased if all possible linearizations are created. In this case, the ESH algorithm solves the problem in 20 seconds while the  $\alpha$ ECP algorithm needed 15 seconds. If the objective function is handled as in (P3) but pseudoconvex constraints are replaced by convex constraints, the solving time for ESH is 170s and for  $\alpha$ ECP it is 11s. If the objective function is transformed to linear constraints and the pseudoconvex functions are kept as they are, the solving time for ESH is 190s and for  $\alpha$ ECP it is 370s. Based on these tests, it seems that the difficult MILP problems that  $\alpha$ ECP created in (P3) are related to the  $\alpha$ -cutting planes of the pseudoconvex functions.

## 5 Concluding remarks

In this paper, it was shown that the ESH method can be applied to a problem with nonsmooth locally Lipschitz continuous functions. The only change to the original method is to use Clarke subgradients instead of gradients. For a convex objective function and  $f^\circ$ -pseudoconvex constraint functions the algorithm was shown to converge to a global minimum. This result requires that the Slater constraint qualification holds true. If it does not, we can still solve the problem but the obtained solution might be only  $\varepsilon_g$ -feasible. If the subdifferentials of the constraint functions do not contain zero at the points where supporting hyperplanes are created, the convergence theorems are also valid for  $f^\circ$ -quasiconvex constraint functions.

Both the ESH and the  $\alpha$ ECP algorithms solved all the example problems considered to the best known solutions. For the ESH algorithm, the higher number of nonlinear function evaluation was compensated by fewer or easier MILP problems that led to faster solving times in problems (P2) and (P3). However, it is not guaranteed that the ESH algorithm solves a problem with a fewer number of MILP problems as was shown in problem (P3). The superiority of ESH to  $\alpha$ ECP when solving (P3) was related to the pseudoconvex constraints. If those constraints were replaced by appropriate convex constraints, then the  $\alpha$ ECP could solve the problem faster.

Generally, if the MILP problems of the original nonsmooth MINLP problem are hard to solve, ESH should do better than  $\alpha$ ECP. However, if the bottleneck of the solving process is expensive nonlinear function evaluations then  $\alpha$ ECP has an advantage over ESH.

There is still room for improvement for the ESH algorithm with respect to the problem classes considered in this paper. In addition to using the LP step, we may change the line search of the ESH algorithm and the algorithm that solves the feasibility problem to solve problems more efficiently. Furthermore, by using corresponding types of techniques as in [7,17] the ESH method to solve problems with an  $f^\circ$ -pseudoconvex objective function will be considered in a forthcoming paper.

**Acknowledgements** This research was supported by the grant no. 289500 of the Academy of Finland. Jan Kronqvist would like to thank the graduate school in chemical engineering.

## References

1. Arnold, T., Henrion, R., Moller, A., and Vigerske, S.: A mixed-integer stochastic nonlinear optimization problem with joint probabilistic constraints. *Pacific Journal of Optimization* 10, 5–20 (2014)
2. Bagirov, A., Mäkelä, M.M., and Karmitsa, N.: *Introduction to Nonsmooth Optimization: Theory, Practice and Software*. Springer International Publishing, Cham, Heidelberg (2014)
3. Bonami, P., Cournoéjols, G., Lodi, A., and Margot, F.: A feasibility pump for mixed integer nonlinear programs. *Mathematical Programming* 119, 331–352 (2009)
4. Castillo, I., Westerlund, J., Emet, S., and Westerlund T.: Optimization of block layout design problems with unequal areas: A comparison of MILP and MINLP optimization methods. *Computers & Chemical Engineering* 30, 54–69 (2005)
5. Clarke, F. H.: *Optimization and Nonsmooth Analysis*. Wiley-Interscience, New York (1983)
6. Emet S., and Westerlund, T.: Comparisons of solving a chromatographic separation problem using MINLP methods. *Computers & Chemical Engineering* 28, 673–682 (2004).
7. Eronen, V-P., Mäkelä, M.M., and Westerlund, T.: Extended cutting plane method for a class of nonsmooth nonconvex MINLP problems. *Optimization* 64:3, 641–661 (2015)
8. Haarala, N., Miettinen, K., and Mäkelä, M.M.: Globally convergent limited memory bundle method for large-scale nonsmooth optimization. *Mathematical Programming* 109:1, 181–205 (2007)
9. Jain, V., and Grossmann, I.: Cyclic scheduling of continuous parallel-process units with decaying performance. *AIChE Journal* 44, 1623–1636 (1999)
10. Kronqvist, J., Lundell, A., and Westerlund, T.: The ESH algorithm for convex mixed-integer nonlinear programming. *Journal of Global Optimization* (to appear). doi: 10.1007/s10898-015-0322-3
11. Mäkelä, M. M. and Neittaanmäki, P.: *Nonsmooth Optimization: Analysis and Algorithms with Applications to Optimal Control*. World Scientific Publishing Co., Singapore (1992)
12. Meller, R. D., Narayanan, V., and Vance, P. H.: Optimal facility layout design. *Operations Research Letters* 23, 117–127 (1999)
13. de Oliveira, W.: Regularized optimization methods for convex MINLP problems. *TOP* 24, 665–692 (2016)
14. Pörn, R.: *Mixed-Integer Non-Linear Programming: Convexification Techniques and Algorithm Development*. Ph.D. Thesis, bo Akademi University (2000)
15. Schramm, H. and Zowe, J.: A version of the bundle idea for minimizing a nonsmooth function: conceptual idea, convergence analysis, numerical results. *SIAM Journal on Optimization* 2:1, 121–152 (1992)
16. Veinott Jr, A.F.: The supporting hyperplane method for unimodal programming. *Operations Research* 15:1, 147–152 (1967)
17. Westerlund, T., and Pörn, R.: Solving pseudo-convex mixed integer optimization problems by cutting plane techniques. *Optimization and Engineering* 3, 253–280 (2002)

### A The facility layout problem: P3

In this problem, 7 departments should be placed in a facility. The width and the height of the facility are  $w_F = 8.54$  and  $h_F = 13$ , respectively. The decision variables and the indices of the problem are:

$i$  = indexes of the departments:  $1, 2, \dots, 7$ .

$x_i, y_i$  = coordinates of the center of department  $i$ .

$w_i$  = width of the department  $i$ .

$h_i$  = height of the department  $i$ .

$X_{ij}, Y_{ij}$  = auxiliary variables.

The problem reads

$$\min \sum_{i=1}^6 |x_i - x_{i+1}| + |y_i - y_{i+1}| \quad (7)$$

$$\text{s. t. } h_i w_i \geq a_i, i = 1, 2, \dots, 7 \quad (8)$$

$$x_i + \frac{1}{2} w_i \leq w_F, i = 1, 2, \dots, 7 \quad (9)$$

$$-x_i + \frac{1}{2} w_i \leq 0, i = 1, 2, \dots, 7 \quad (10)$$

$$y_i + \frac{1}{2} h_i \leq h_F, i = 1, 2, \dots, 7 \quad (11)$$

$$-y_i + \frac{1}{2} h_i \leq 0, i = 1, 2, \dots, 7 \quad (12)$$

$$\frac{1}{2}(w_i + w_j) - (x_i - x_j) \leq w_F(X_{ij} + Y_{ij}), 1 \leq i < j \leq 7 \quad (13)$$

$$\frac{1}{2}(w_i + w_j) - (x_j - x_i) \leq w_F(1 + X_{ij} - Y_{ij}), 1 \leq i < j \leq 7 \quad (14)$$

$$\frac{1}{2}(h_i + h_j) - (y_i - y_j) \leq h_F(1 - X_{ij} + Y_{ij}), 1 \leq i < j \leq 7 \quad (15)$$

$$\frac{1}{2}(h_i + h_j) - (y_j - y_i) \leq h_F(2 - X_{ij} - Y_{ij}), 1 \leq i < j \leq 7 \quad (16)$$

$$x_1 - x_2 \leq 0 \quad (17)$$

$$y_1 - y_2 \leq 0 \quad (18)$$

$$w_i^{low} \leq w_i \leq w_i^{up}, i = 1, 2, \dots, 7 \quad (19)$$

$$h_i^{low} \leq h_i \leq h_i^{up}, i = 1, 2, \dots, 7 \quad (20)$$

$$X_{ij} \in \{0, 1\}, 1 \leq i < j \leq 7 \quad (21)$$

$$Y_{ij} \in \{0, 1\}, 1 \leq i < j \leq 7 \quad (22)$$

The objective is to minimize the distance between consecutive departments. Constraints (7) define the minimum areas of the departments. Constraints (8)-(11) make sure the departments are located inside the facility. Constraints (12)-(15) prevent the overlapping of the departments. Constraints (16)-(17) erase symmetric solutions. The following parameters were used.

**Table 3** Parameters of the problem (P3)

i	1	2	3	4	5	6	7
$a_i$	16	16	16	36	9	9	9
$w_i^{low}$	2	2	2	3	1.5	1.5	1.5
$w_i^{up}$	8	8	8	8.54	6	6	6
$h_i^{low}$	2	2	2	4.2155	1.5	1.5	1.5
$h_i^{up}$	8	8	8	12	6	6	6