

# An algorithm for nonsmooth optimization by successive piecewise linearization

Sabrina Fiege<sup>1</sup>  · Andrea Walther<sup>1</sup> ·  
Andreas Griewank<sup>2</sup>

Received: 8 December 2016 / Accepted: 30 March 2018 / Published online: 6 April 2018  
© Springer-Verlag GmbH Germany, part of Springer Nature and Mathematical Optimization Society 2018

**Abstract** We present an optimization method for Lipschitz continuous, piecewise smooth (PS) objective functions based on successive piecewise linearization. Since, in many realistic cases, nondifferentiabilities are caused by the occurrence of **abs()**, **max()**, and **min()**, we concentrate on these nonsmooth elemental functions. The method's idea is to locate an optimum of a PS objective function by explicitly handling the kink structure at the level of piecewise linear models. This piecewise linearization can be generated in its *abs-normal-form* by minor extension of standard algorithmic, or automatic, differentiation tools. In this paper it is shown that the new method when started from within a compact level set generates a sequence of iterates whose cluster points are all Clarke stationary. Numerical results including comparisons with other nonsmooth optimization methods then illustrate the capabilities of the proposed approach.

**Keywords** Piecewise smoothness · Nonsmooth optimization · Algorithmic differentiation · Abs-normal form · Clarke stationary

**Mathematics Subject Classification** 49J52 · 90C56

## 1 Introduction

Even today, only few practical methods for the minimization of Lipschitzian piecewise smooth functions  $f : \mathbb{R}^n \mapsto \mathbb{R}$  are available. On convex objectives, the use of subgradients in combination with merely square summable step lengths yields only

---

✉ Sabrina Fiege  
sfiege@math.uni-paderborn.de

<sup>1</sup> Department of Mathematics, Paderborn University, Paderborn, Germany

<sup>2</sup> School of Mathematical Science and Information Technology, Yachaytech, Urcuqui, Ecuador

a sublinear rate of convergence, see, e.g., [20, Chapter 2]. A more reasonable rate of convergence can be expected from bundle methods and gradient sampling, see, e.g., [1, 3, 12, 13, 16], but their performance is somewhat erratic. Another option is to adapt quasi-Newton methods for the nonsmooth case, as proposed in [14].

In [8], we proposed a new algorithm for computing a stationary point of a piecewise linear (PL) function. Key concepts of this algorithm, in particular the piecewise linearization, were initially proposed in [4]. The local piecewise linear models can be efficiently and stably represented in the so-called abs-normal form. Just like local linearizations in the smooth case piecewise linear models can be used for various numerical purposes in the nonsmooth case. The solution of square nonsmooth equation systems in a generalized Newton type fashion can be based on successively solving the local models. The local subproblem, namely solving a piecewise linear equation system in abs-normal form, was considered in [5], which surveys a wide range of finite and iterative methods.

In this paper we will consider only scalar valued functions in several variables that serve as objectives in unconstrained optimization. That was also the setting in the predecessor paper [8]. Following the seminal work of Hiriart-Urruty and Lemaréchal [11], we could demonstrate finite convergence in the convex case and verify it numerically on a few selected test problems. Moreover, the basic method was formulated for general nonconvex piecewise linear problems with an additional proximal term, and it was found to always reach a stationary point in preliminary numerical experiments. The performance of this new algorithm compared favorably to an adapted BFGS method with gradient sampling [1, 14] and a bundle method implementation [16].

As our efforts to theoretically exclude the possibility of Zenon-like zigzagging on nonconvex functions failed, we have modified the original method [8] for piecewise linear functions by replacing a line-search with a positive definite QP solve as sketched in Sect. 3. That modification reduces the number of iterations and immediately implies finite convergence to a stationary point for general piecewise linear functions with a proximal term. The resulting numerical performance of our code module PLMin on piecewise linear functions is reported in Sect. 4 of [8]. Both the method introduced in [8] and the modified method proposed in this paper determine a descent trajectory along a path of essential polyhedra towards a stationary point. In this process, the choice of the succeeding essential polyhedral is crucial. Currently, this choice is realized by a bundle approach which turns out to be very robust. Another possible approach is a stationary test based on new optimality conditions defined in [7], which have as of now no algorithmic realization.

As already foreshadowed in [4], the main thrust of our algorithmic development is the minimization of piecewise smooth (PS) and other semi-smooth functions by successive piecewise linearization using PLMin as an inner solver. For a conservative updating strategy of the proximal term it had already been shown that successive piecewise linearization generates a subsequence that converges to a stationary point from within a compact level set. After some numerical experiments we developed the more aggressive updating strategy described in Sect. 3, which maintains global convergence in the above sense.

The final Sect. 4 contains numerical results for a wide range of test problems. A direct comparison with other nonsmooth solvers is difficult because they utilize

much less information about the objective than our approach. However, this additional structural information given by the abs-normal form of PL functions [4] is easy to come by, not only on all the usual test problems but also on large scale applications from scientific computing. It can be obtained by an extension of algorithmic differentiation (AD) to evaluation codes involving smooth elemental functions and the absolute value function as well as the min and max operator as shown in [4].

## 2 Notation and background

Throughout the paper, we will consider only objective functions  $f : \mathbb{R}^n \mapsto \mathbb{R}$  that can be described as a composition of elementary functions  $\varphi \in \Phi$  such that  $f$  can be given by an evaluation procedure. If all elemental functions  $\varphi$  are  $d$  times continuously differentiable with  $1 \leq d \leq \infty$  on an open domain, this evaluation procedure can be illustrated as in Table 1. The evaluation procedure consists of three parts. First, the independent variables  $v_{i-n}, i = 1, \dots, n$ , are declared. Then each intermediate variable corresponds to one elemental function, i.e.,  $v_i = \varphi_i(v_j)_{j < i}$  where the precedence relation  $j < i$  denotes that  $v_i$  depends directly on  $v_j$  for  $j < i$ . Finally, the dependent variable is assigned. This notation is based on standard algorithmic differentiation theory. For further information see [6].

In the following, we assume that these elemental functions  $\varphi \in \Phi$  are either the absolute value function or Lipschitz continuously differentiable in the domain  $D \subset \mathbb{R}^n$  of interest. Using the reformulations

$$\begin{aligned} \min(v, u) &= (v + u - \text{abs}(v - u))/2 \quad \text{and} & (1) \\ \max(v, u) &= (v + u + \text{abs}(v - u))/2, & (2) \end{aligned}$$

a quite large range of piecewise differentiable and locally Lipschitz continuous functions are covered. It follows from this assumption that the resulting objective function  $f(x)$  is piecewise smooth in the sense of Scholtes [19, Chapter 4]. Subsequently, these functions will be denoted as composite piecewise differentiable functions.

Conceptually combining consecutive smooth elemental functions  $\varphi_i$  into larger smooth elemental functions  $\psi_i$ , one obtains the reduced evaluation procedure shown in Table 2, where all evaluations of the absolute value function can be clearly identified and exploited.

**Table 1** Evaluation procedure, if all  $\varphi \in \Phi$  are  $d$  times continuously differentiable

$v_{i-n} = x_i$	$i = 1 \dots n$
$v_i = \varphi_i(v_j)_{j < i}$	$i = 1 \dots l$
$y = v_l$	

**Table 2** Reduced evaluation procedure for composite piecewise differentiable functions

$v_{i-n} = x_i$	$i = 1 \dots n$
$\left. \begin{aligned} z_i &= \psi_i(v_j)_{j < i} \\ \sigma_i &= \text{sign}(z_i) \\ v_i &= \sigma_i z_i = \text{abs}(z_i) \end{aligned} \right\}$	$i = 1 \dots s$
$y \equiv v_{s+1} = \psi_{s+1}(v_j)_{j < s+1}$	

Notice that in Table 2 the intermediate variables  $v_i$  and the precedence relation  $j < i$  have been redefined. This will not cause any confusion since in the remainder of the paper we will never refer to the original, fine-grained decomposition given in Table 1, which is the basis of AD tools and their PS extensions. They then yield all derivatives of the  $\psi_i$  by partial pre-accumulation of the smooth elemental functions. By inspection of Table 2 we note that  $s \in \mathbb{N}$  denotes the actual number of evaluations of the absolute value function. Since the intermediate value  $z_i$  is used as the argument of the absolute value and hence causes also the switches in the corresponding derivative values, the vector  $z = (z_i) \in \mathbb{R}^s$  is called the *switching vector* defining also the *signature vector*  $\sigma = (\sigma_i(x))_{i=1,\dots,s} \equiv (\text{sign}(z_i(x)))_{i=1,\dots,s} \in \mathbb{R}^s$ .

Throughout the rest of the paper, we will use the following example to illustrate our approach.

*Example 1* We consider the nonconvex, composite piecewise differentiable function

$$f : \mathbb{R}^2 \rightarrow \mathbb{R}, \quad f(x_1, x_2) = (x_2^2 - (x_1)_+)_+ \quad \text{with } y_+ \equiv \max(0, y), \quad (3)$$

which can be rewritten in terms of the switching vector  $z$  as

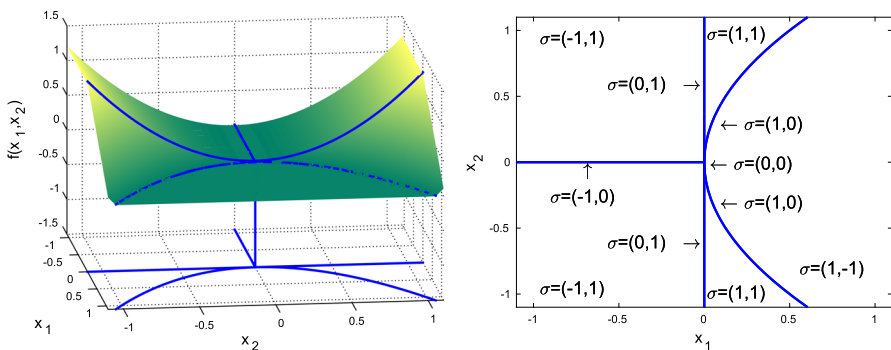
$$f(x_1, x_2) = \frac{1}{2} (z_2 + |z_2|) \quad \text{with } z_1 = x_1 \quad \text{and} \quad z_2 = x_2^2 - \frac{1}{2} (z_1 + |z_1|).$$

Since the absolute value function is evaluated twice we have  $s = 2$ . The plot of the function is shown in Fig. 1. All points of nondifferentiability are marked both in the domain of the function and in its value set. Additionally, the decomposition of the argument space is marked by the corresponding vectors  $\sigma$ . The reduced evaluation procedure of the function is illustrated in Table 3

Given the class of piecewise smooth functions considered in this paper, it follows that they can be represented as

$$f(x) \in \{f_\sigma(x) : \sigma \in \mathcal{E} \subset \{-1, 0, 1\}^s\} \quad \text{at } x \in \mathbb{R}^n,$$

where the selection functions  $f_\sigma$  are continuously differentiable on neighborhoods of points where they are active, i.e., coincide with  $f$ , as described in [19]. We will assume



**Fig. 1** PS function (3) with all points of nondifferentiability and corresponding signatures

**Table 3** Reduced adapted evaluation procedure of Example 1

$v_{-1} = x_1$
$v_0 = x_2$
$z_1 = \Psi_1(v_j)_{j < 1} = v_{-1}$
$\sigma_1 = \text{sign}(z_1)$
$v_1 = \sigma_1 z_1$
$z_2 = \Psi_2(v_j)_{j < 2} = v_0^2 - \frac{1}{2}(v_{-1} + v_1)$
$\sigma_2 = \text{sign}(z_2)$
$v_2 = \sigma_2 z_2$
$y = \Psi_3(v_j)_{j < 3} = \frac{1}{2} \left( v_0^2 - \frac{1}{2}(v_{-1} + v_1) + v_2 \right)$

that all  $f_\sigma$  with  $\sigma \in \mathcal{E}$  are essential in that their coincidence sets  $\{f(x) = f_\sigma(x)\}$  are the closures of their interiors, i.e.,  $\sigma \in \mathcal{E}$  if and only if

$$\mathcal{E} \equiv \{\sigma(x) \in \{-1, 0, 1\}^s \mid x \in \text{cl}(\text{int}\{x \in \mathbb{R}^n \mid f(x) = f_\sigma(x)\})\}.$$

The particular form of the index set  $\mathcal{E} \subset \{-1, 0, 1\}^s$  stems from our function evaluation model described in Table 2. According to [2], one obtains the *Clarke subdifferential* of  $f$  as

$$\partial f(x) \equiv \text{conv}(\partial^L f(x))$$

with the *limiting subdifferential*

$$\partial^L f(x) \equiv \{g \in \mathbb{R}^n : \exists \{x_i\}_{i \in \mathbb{N}} \text{ with } x_i \notin \Omega_f \text{ s.t. } x_i \rightarrow x \text{ and } \nabla f(x_i) \rightarrow g\},$$

where  $\Omega_f$  is the set where  $f$  is not differentiable. The elements of  $\partial^L f(x)$  are called the *limiting subgradients* of  $f$  at  $x$ . Moreover, it was proven in [19] that for piecewise smooth functions the limiting subdifferential can be given as

$$\partial^L f(x) \equiv \{\nabla f_\sigma(x) : f_\sigma(x) = f(x) : \sigma \in \mathcal{E}\}.$$

A *directionally active gradient*  $g$  is given by

$$g \equiv g(x; d) \in \partial^L f(x) \quad \text{such that} \quad f'(x; d) = g^\top d. \tag{4}$$

Here  $f'(x; d)$  is the directional derivative of  $f$  at  $x$  in direction  $d$  and  $g(x; d)$  equals the gradient  $\nabla f_\sigma(x)$  of a locally differentiable selection function  $f_\sigma$  that coincides with  $f$  on a set, whose tangent cone at  $x$  contains  $d$  and has a nonempty interior.

To obtain a piecewise linearization of the objective function  $f$ , one has to construct for each elemental function a tangent approximation. For a given argument  $x$  and a direction  $\Delta x$ , we will use the elemental linearizations

$$\Delta v_i = \Delta v_j \pm \Delta v_k \quad \text{for } v_i = v_j \pm v_k, \tag{5}$$

$$\Delta v_i = v_j * \Delta v_k + v_k * \Delta v_j \quad \text{for } v_i = v_j * v_k, \tag{6}$$

$$\Delta v_i = \varphi'(v_j)_{j < i} * \Delta(v_j)_{j < i} \quad \text{for } v_i = \varphi_i(v_j)_{j < i} \neq \text{abs}(v_j), \tag{7}$$

$$\Delta v_i = \text{abs}(v_j + \Delta v_j) - v_i \quad \text{for } v_i = \text{abs}(v_j). \tag{8}$$

The linearizations (5)–(7) are well-known from standard AD theory, see [6], whereas the linearization (8) was proposed in [4]. These linearizations can be used to compute the increment  $\Delta f(x; \Delta x)$  and therefore also the piecewise linearization

$$f_x^{PL}(\Delta x) \equiv f(x) + \Delta f(x; \Delta x) \tag{9}$$

of the original PS function  $f$  at a given point  $x$  with the argument  $\Delta x$ .

As shown in [5], any piecewise linear function  $y^{PL} = f^{PL}(\Delta x)$  with

$$f^{PL} : \mathbb{R}^n \rightarrow \mathbb{R}^m$$

can be expressed using the argument  $\Delta x$  and the resulting switching vector  $z \in \mathbb{R}^s$  in the *abs-normal* form given by

$$\begin{bmatrix} z \\ y^{PL} \end{bmatrix} = \begin{bmatrix} c_z \\ c_y \end{bmatrix} + \begin{bmatrix} Z & L \\ Y & J \end{bmatrix} \begin{bmatrix} \Delta x \\ |z| \end{bmatrix}, \tag{10}$$

where  $c_z \in \mathbb{R}^s$ ,  $c_y \in \mathbb{R}^m$ ,  $Z \in \mathbb{R}^{s \times n}$ ,  $L \in \mathbb{R}^{s \times s}$ ,  $Y \in \mathbb{R}^{m \times n}$  and  $J \in \mathbb{R}^{m \times s}$ . The matrix  $L$  is strictly lower triangular, i.e., each  $z_i$  is an affine function of absolute values  $|z_j|$  with  $j < i$  and the input values  $\Delta x_k$  for  $1 \leq k \leq n$ . The matrices  $Y$  and  $J$  are row vectors in this optimization context, since we consider functions with  $m = 1$ . Correspondingly,  $c_y$  is a real number instead of a vector in the more general setting. Note that the signature vector  $\sigma(\Delta x)$  of the piecewise linearization defined in Eq. (10) does not coincide with the signature vector  $\sigma(x)$  of the piecewise smooth function defined in Table 2, since the underlying functions and therewith, the switching vectors differ. The switching vector of the linearization is given by  $\sigma(\Delta x) = \text{sign}(z(\Delta x))$ . Freezing  $x$  and  $\sigma \in \{-1, 0, 1\}^s$  we obtain the polytope of corresponding  $\Delta x$  as

$$P_\sigma \equiv \{\Delta x \in \mathbb{R}^n \mid \sigma = \text{sign}(z(\Delta x))\}. \tag{11}$$

Using the diagonal signature matrix

$$\Sigma \equiv \Sigma(\Delta x) \equiv \text{diag}(\sigma(\Delta x)) \in \{-1, 0, 1\}^{s \times s}$$

one derives from the first equation of Eq. (10) with  $|z| \equiv \Sigma z$  for  $\Delta x \in P_\sigma$  the linear relations

$$(I - L\Sigma)z = c_z + Z\Delta x \quad \text{and} \quad z = (I - L\Sigma)^{-1}(c_z + Z\Delta x). \tag{12}$$

Notice that due to the strict triangularity of  $L \Sigma$  the inverse  $(I - L \Sigma)^{-1}$  is well defined and polynomial in the entries of  $L \Sigma$ . Substituting this expression into the last equation of Eq. (10), it follows for the function value that

$$f_x^{PL}(\Delta x) \equiv \gamma_\sigma + g_\sigma^\top \Delta x \quad \text{for } \Delta x \in P_\sigma \tag{13}$$

with

$$\gamma_\sigma = c_y + J \Sigma (I - L \Sigma)^{-1} c_z \quad \text{and} \quad g_\sigma^\top = Y + J \Sigma (I - L \Sigma)^{-1} Z. \tag{14}$$

That is, the gradient evaluation for one of the linear pieces reduces to the solution of a linear system with a triangular matrix. This will be exploited for a cheap gradient calculation in the inner loop of the optimization algorithm presented in the next section.

*Example 2* The piecewise linearization  $f_x^{PL}$  of the function  $f$  introduced in Example 1 evaluated at the base point  $\bar{x}$  with the argument  $\Delta x = x - \bar{x}$  is given by

$$f_{\bar{x}}^{PL}(\Delta x) = \frac{1}{2}(z_2 + |z_2|) \tag{15}$$

where the switching vector  $z$  of  $f_x^{PL}$  is given by

$$z_1 = \hat{x}_1 + \Delta x_1 \quad \text{and} \quad z_2 = \hat{x}_2^2 + 2\hat{x}_2 \Delta x_2 - \frac{1}{2}(z_1 + |z_1|).$$

The piecewise linearization can be written in its abs-normal form given by

$$\begin{bmatrix} z_1 \\ z_2 \\ y \end{bmatrix} = \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2^2 - \frac{1}{2}\hat{x}_1 \\ \frac{1}{2}(\hat{x}_2^2 - \frac{1}{2}\hat{x}_1) \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{2} & 2\hat{x}_2 & -\frac{1}{2} & 0 \\ -\frac{1}{4} & \hat{x}_2 & -\frac{1}{4} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ |z_1| \\ |z_2| \end{bmatrix}.$$

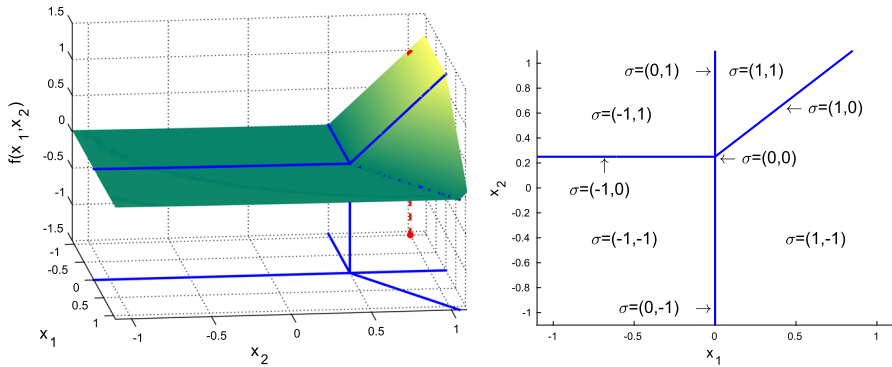
Applying Eqs. (13) and (14), the function  $f_{\hat{x}}^{PL}$  with  $\hat{x} = (-1, 0.5)$  and the gradient  $g_\sigma$  can be computed explicitly for each signature vector  $\sigma$ . For  $\sigma = (-1, 1)$ , one obtains

$$g_\sigma(\Delta x) = (0, 1)^\top \quad \text{and} \quad f_{\hat{x}}^{PL}(\Delta x) = 0.25 + \Delta x_2 \quad \forall \Delta x \in P_\sigma$$

and for  $\sigma = (1, 1)$ , one obtains

$$g_\sigma(\Delta x) = (-1, 1)^\top \quad \text{and} \quad f_{\hat{x}}^{PL}(\Delta x) = 2.375 - \Delta x_1 + \Delta x_2 \quad \forall \Delta x \in P_\sigma.$$

The domain of the piecewise linearization is decomposed by two absolute value functions into polyhedra with corresponding signatures  $\sigma \in \{-1, 0, 1\}^2$  as can be seen in Fig. 2. Note that the decomposition of the underlying piecewise smooth function presented in Example 1 differs clearly from the decomposition of the piecewise linearization, e.g., the number of polyhedra and the occurring signatures.



**Fig. 2** Left: piecewise linearization of Eq. (3) evaluated at  $\hat{x} = (-1, 0.5)$ . Right: the decomposition of its domain and its signatures

### 3 Successive piecewise linearization

#### 3.1 Minimization of the piecewise smooth function

As already stated in [4], we propose to minimize piecewise smooth functions using an algorithm called **LiPsMin** (Algorithm 1) that relies on an algorithm **PLMin** (Algorithm 3) to solve the local subproblems obtained by piecewise linearization. **PLMin** in turn relies on an algorithm **ComputeDesDir** (Algorithm 2) to generate descent directions at vertices of the PL model. Their nesting without an outer stopping criterion generates conceptually an infinite sequence of iterates  $\{x^k\}_{k \in \mathbb{N}}$ .

**Algorithm 1 (LiPsMin)**

**LiPsMin**( $x, q^{lb}, \kappa$ )

// Precondition:  $x \in \mathbb{R}^n, \kappa > 1$  and  $q^{lb} > 0$

Set  $x^0 = x$  and  $q^0 = q^{lb}$ .

for  $k = 0, 1, 2, \dots$

1. Generate the PL model  $f_{x^k}^{PL}(\cdot)$  at the current iterate  $x^k$ .
2. Use  $PLMin(x^k, q^k)$ , see Algorithm 3, to solve

$$\Delta x^k = \arg \min_{\Delta x \in \mathbb{R}^n} f_{x^k}^{PL}(\Delta x) + \frac{\kappa}{2} q^k \|\Delta x\|^2.$$

3. Set  $x^{k+1} = x^k + \Delta x^k$  if  $f(x^k + \Delta x^k) < f(x^k)$  and  $x^{k+1} = x^k$  otherwise.
4. Compute

$$\hat{q}^{k+1} \equiv \hat{q}^k(x^{k+1}, \Delta x^k) \equiv \frac{2|f(x^{k+1}) - f_{x^k}^{PL}(\Delta x^k)|}{\|\Delta x^k\|^2}$$

and set  $q^{k+1} = \max\{\hat{q}^{k+1}, \mu q^k + (1 - \mu)\hat{q}^{k+1}, q^{lb}\}$  with  $\mu \in [0, 1]$ .



As can be seen, the main ingredient of the approach is the successive piecewise linearization which was introduced in the previous section. The local model will always be generated in step 1 of LiPsMin. In step 2 an overestimation of a local subproblem

$$\hat{f}_{x^k}(\Delta x) \equiv f_{x^k}^{PL}(\Delta x) + \frac{\kappa}{2} q^k \|\Delta x\|^2, \tag{16}$$

with  $\kappa > 1$  is solved by an inner loop which is discussed explicitly in Sect. 3.2. Actually, while we aim to compute the global minimum of  $\hat{f}_{x^k}(\Delta x)$  all we can guarantee is to reach a Clarke stationary point of the PL model.

The overestimation is necessary to ensure lower boundedness of the model and to obtain the required convergence behavior. In this algorithmic specification we have not yet given a termination criterion so that the conceptual algorithm generates an infinite sequence of iterates  $\{x^k\}_{k \in \mathbb{N}}$  that can be examined in the convergence analysis. Naturally, we would like Algorithm 1 to generate iterates with cluster points  $x_*$  that are minimizers or at least Clarke stationary, i.e.,  $0 \in \partial f(x_*)$ , not only for the PL models but for the underlying PS objective. Therefore, we establish the following assertion.

- Lemma 1** *i) If the piecewise smooth function  $f$  is locally minimal at  $x$ , then the piecewise quadratic model  $\hat{f}_x$  is locally minimal at  $\Delta x = 0$  for all  $q \geq 0$ .  
 ii) If the piecewise quadratic model  $\hat{f}_x$  is Clarke stationary at  $\Delta x = 0$  for some  $q \geq 0$ , then the piecewise smooth function  $f$  is Clarke stationary at  $x$ .*

*Proof* Note that according to Proposition 9 in [4] the subdifferential of the piecewise smooth function  $f$  at  $x$  contains that of the piecewise linearization (9) based on  $x$  at  $\Delta x = 0$ , i.e.,

$$\partial f(x) \supset \partial f_x^{PL}(0).$$

We define  $h : \mathbb{R}^n \rightarrow \mathbb{R}$  as  $h(\Delta x) := \frac{q}{2} \|\Delta x\|^2$  which is a twice continuously differentiable function with a minimizer at  $\Delta x = 0$ . The minimizer is unique, if  $q > 0$ . The subdifferential of  $h$  is given by  $\partial h(\Delta x) = \{q \Delta x\}$ . Then, the quadratic model can be written as  $\hat{f}_x(\Delta x) = f_x^{PL}(\Delta x) + h(\Delta x)$ .

i) Let us assume for simplicity that  $f$  is locally minimal at  $x$  with  $f(x) = 0$  and hence  $\hat{f}_x(0) = 0$ . Suppose that  $\hat{f}_x(\cdot)$  is not minimal at 0 for some  $q \geq 0$ . Then we have for some  $\Delta x$  and  $t > 0$

$$\hat{f}_x(t \Delta x) = t g_\sigma^\top \Delta x + o(t) < 0,$$

where we have used the directional differentiability of the piecewise linear model and  $g_\sigma$  is a suitable subgradient. Then it follows by the generalized Taylor expansion [4] that for sufficiently small  $t$  also

$$f(x + t \Delta x) - f(x) = t g_\sigma^\top \Delta x + o(t) < 0,$$

yielding a contradiction to the minimality of  $f$  at  $x$ .

ii) If  $\hat{f}_x$  is Clarke stationary in  $\Delta x = 0$ , it implies that

$$0 \in \partial \hat{f}_x(0) = \partial(f_x^{PL} + h)(0) \subseteq \partial f_x^{PL}(0) + \partial h(0).$$

Since  $\partial h(0) = \{0\}$  one obtains that  $0 \in \partial f_x^{PL}(0)$ . By using the inclusion relation of the subdifferentials noted above this implies that also  $0 \in \partial f(x)$  and hence,  $f$  is Clarke stationary in  $x$ . □

### 3.2 Minimization of the piecewise linear subproblem as sequence of QPs

For the general scenario we have Algorithm 3 to compute a stationary point of the local model

$$\arg \min_{\Delta x \in \mathbb{R}^n} f_x^{PL}(\Delta x) + \frac{\kappa}{2} q \|\Delta x\|^2 \tag{17}$$

for step 2 of Algorithm 1. Since we consider the  $k$ th iteration of Algorithm 1 throughout this subsection, we use  $x$ ,  $\Delta x$ , and  $q$  instead of  $x^k$ ,  $\Delta x^k$  and  $q^k$  for simplicity. The essential difference to the *true descent algorithm* introduced in [8, Algorithm 4] is the solution of a quadratic subproblem on polyhedral subdomains instead of computing just a critical step multiplier to traverse them. In both approaches the objective (16) is consistently reduced but since we now make sure that in each polyhedral subdomain the minimal value is reached it can be visited only once. Hence there can be no cycling, which previously we could only exclude for the descent trajectory under the assumption of convexity. In this way we get a finite sequence of iterates  $\{\Delta x_j\}_{j < \ell}$  for Algorithm 3, where the upper bound  $\ell$  may of course be quite large depending on the particular problem.

To solve the piecewise quadratic model (17) Algorithm 3 builds a sequence of constrained QPs each defined on an essential polyhedron  $P_\sigma$ . Therefore, we face two tasks. First, the individual QPs have to be formulated. Second, having computed the minimizer of the current QP, this minimizer has to be tested for stationarity. If the minimizer of the current QP is in the interior of its domain, we have found at least a local minimizer of (17) so that Algorithm 3 can terminate; otherwise, Algorithm 2 can be applied to check stationarity. In case the minimizer is not a stationary point Algorithm 2 returns a descent direction  $d$  which then can be used to identify a new essential polyhedron and therewith, the subsequent QP. Algorithm 2 gets the current iterate of the inner loop  $\Delta x$  and the overestimated quadratic coefficient  $q\kappa$  as input parameters together with the initial bundle  $G = \{g_\sigma\}$  containing the gradient of the current polyhedron  $P_\sigma$ .

#### Algorithm 2 (Computation of Descent Direction)

**ComputeDesDir**( $\Delta x, q\kappa, G$ )

// Precondition:  $\Delta x \in \mathbb{R}^n, q \geq 0, \kappa > 1, \emptyset \neq G \subset \partial^L f_x^{PL}(\Delta x)$

*repeat*

Compute  $d = -\mathbf{short}(q\kappa \Delta x, G)$  as defined in Equation (18).

Evaluate  $g = g(\Delta x; d)$ .

Augment the bundle  $G = G \cup \{g\}$ .  
**until**  $(g + q\kappa \Delta x)^\top d \leq -\beta \|d\|^2$   
 Set  $G = \emptyset$ .  
 return  $d$

In this algorithm,  $G$  is a subset of the limiting subdifferential of the PL function  $f_x^{PL}$  at the current iterate  $\Delta x$ , i.e.,  $G \subseteq \partial^L f_x^{PL}(\Delta x)$ . For an arbitrary quadratic coefficient  $q$  and increment  $\Delta x$  the direction  $d$  is defined as

$$d \equiv -\text{short}(q\Delta x, G)$$

$$\equiv -\arg \min \left\{ \|g\| \mid g = \sum_{j=1}^{|G|} \lambda_j g_j + q\Delta x, g_j \in G, \lambda_j \geq 0, \sum_{j=1}^{|G|} \lambda_j = 1 \right\}. \quad (18)$$

Hence, the evaluation of each  $d$  requires the solution of a QP similar to the one occurring in bundle methods. Subsequently, the bundle  $G$  gets augmented by further directionally active gradients  $g(x; d)$  as defined in Eq. (4) corresponding to neighboring polyhedra. This can only happen finitely often so that the algorithm must terminate as proven in Proposition 3.2 of [8]. We introduce here the additional multiplier  $\beta \in (0, 1)$  to relax the descent condition compared to [8, Algorithm 2] where  $\beta$  was set to 1. However, for the more general case considered here, we only want to identify a polyhedron  $P_\sigma$  that provides descent compared to the current polyhedron.

Now, given a point  $\Delta x$  and a descent direction  $d \neq 0$  we must look for a signature  $\sigma$  such that the closure  $\bar{P}_\sigma$  of  $P_\sigma$  as defined in Eq. (11) contains  $\Delta x$  as an element and  $d$  is an element of its tangent cone at  $\Delta x$ . This means we must identify an essential polyhedron  $P_\sigma$  with  $\sigma \in \mathcal{E}$  at each point  $\Delta x \in \mathbb{R}^n$  whether  $\Delta x$  is differentiable or not. Therefore, we define the *directionally active signature vector*  $\sigma \equiv \sigma(\Delta x; d)$  which corresponds to an essentially active selection function  $f_\sigma$  that coincides with  $f_x^{PL}$  on a nonempty and open  $P_\sigma$  such that  $x, x + \tau d \in \bar{P}_\sigma$  with  $\tau > 0$  arbitrary small. These signature vectors are, unlike those signature vectors defined in Table 2, necessarily elements of the set  $\mathcal{E}$  of essentially active signature vectors. Hence, it follows by continuity that  $P_\sigma$  is open and as shown in Proposition 2 of [8]

$$\bar{P}_\sigma \equiv \{\Delta x \in \mathbb{R}^n : \sigma(\Delta x) \leq \sigma\}. \quad (19)$$

Such directionally active signature vectors  $\sigma(\Delta x; d)$  are defined by

$$\sigma(\Delta x; d) = (\sigma_i(\Delta x; d))_{i=1, \dots, s} \equiv (\text{firstsign}(z_i(\Delta x); \nabla z_i^\top(\Delta x)E))_{i=1, \dots, s}, \quad (20)$$

where  $d \in \mathbb{R}^n$  is the direction obtained from Algorithm 2,  $E \in \mathbb{R}^{n \times n}$  is a nonsingular matrix and  $\text{firstsign}(z; \nabla z^\top(\Delta x)E)$  returns for each component  $\sigma_i, i = 1, \dots, s$ , the sign of the first nonvanishing entry of the vector  $(z_i(\Delta x); \nabla z_i^\top(\Delta x)E) \in \mathbb{R}^{n+1}$ . For our application we choose

$$E = [d, e_1, \dots, e_{j^*-1}, e_{j^*+1}, e_n] \quad \text{with} \quad j^* = \operatorname{argmax}_{j=1, \dots, n} |d_j|,$$

where  $e_i, i = 1, \dots, n$ , are the unit vectors. This computation of the directionally active signature yields lexicographic differentiation as defined in [18] and applies the idea of polynomial escape introduced in [4]. There, it is shown that the polynomial arc

$$a(\tau) = \Delta x + \tau d + \sum_{j=2}^n t^j d_j \quad \text{with} \quad \det(d, d_2, \dots, d_n) \neq 0,$$

where  $d_j \in \mathbb{R}^n$  must be  $n$  linearly independent vectors with  $d_1 \equiv d$ , cannot be trapped in one of the kink surfaces of  $f_x^{PL}(\Delta x)$ .

Due to the previous considerations we can now build a sequence of constrained QPs. An initial direction  $d$  can be computed by Algorithm 2 and an initial signature vector  $\sigma \equiv \sigma(\Delta x; d)$  corresponding to an essential polyhedron can be identified according to Eq. (20).

For all  $\Delta x + \delta x \in P_\sigma$  the switching vector  $z = z(\Delta x + \delta x)$  is given by the affine function

$$z = \underbrace{(I - L\Sigma)^{-1}(c_z + Z\Delta x)}_{z(\Delta x)} + \underbrace{(I - L\Sigma)^{-1}Z\delta x}_{\nabla z(\Delta x)\delta x} \tag{21}$$

according to Eq. (12) where  $\bar{j}$  is the number of the previously solved QPs and  $\Delta x = \sum_{i=0}^{\bar{j}} \delta x_j$  sums up all the obtained minimizers  $\delta x_j, j = 0, \dots, \bar{j}$ . This is necessary to maintain the relationship between the current essential polyhedron  $P_\sigma$  and the base point  $x$  of the piecewise linearization. Hence, one obtains the following subproblem

$$\begin{aligned} \delta x_j &= \arg \min_{\delta x \in \mathbb{R}^n} f_\sigma(\Delta x + \delta x) + \frac{\kappa}{2} q \|\Delta x + \delta x\|^2, \\ \text{s.t. } z_i(\Delta x) + \nabla z_i(\Delta x)^\top (\Delta x + \delta x) &\begin{cases} \leq 0 & \text{if } \sigma_i^j < 0 \\ \geq 0 & \text{if } \sigma_i^j > 0 \\ = 0 & \text{if } \sigma_i^j = 0 \end{cases} \quad \text{for } i = 1, \dots, s. \end{aligned} \tag{22}$$

By construction any such QP is convex, feasible and bounded so that theoretically a solution always exists. However, the minimizer may be degenerate and there may be a large number of redundant constraints slowing down the computation.

The above approach can be summarized as follows, where we use the base point  $x$  and the step  $\Delta x$  as in Algorithm 1 for clarity. The base point  $x$  and the quadratic coefficient  $q$  serve as input variables. The increment  $\Delta x$  is the parameter returned to Algorithm 1.

**Algorithm 3 (PLMin)**

**PLMin**( $x, q$ ) // *Precondition:*  $x \in \mathbb{R}^n, q > 0$

Set  $\Delta x^0 = 0$ .

For  $j = 0, 1, 2, \dots, \ell$

1. Compute direction  $d$  by Algorithm 2.
2. If  $\|d\| = 0$ : STOP.
3. Identify new polyhedron  $P_{\sigma^j}$  using direction  $d$ .
4. Determine solution  $\delta x^j$  of local QP (22) on current polyhedron  $P_{\sigma^j}$ .
5. Update  $\Delta x^{j+1} = \Delta x^j + \delta x^j$ .

return  $\Delta x$

Algorithm 3 converges finitely to a stationary point  $\Delta x$  since any polyhedron  $P_\sigma$  can only be visited at most once as argued at the beginning of this section.

### 3.3 Convergence results for LiPsMin

To prove that cluster points of Algorithm 1 are Clarke stationary, we suppose that our piecewise smooth objective function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is bounded below and has a bounded level set  $\mathcal{N}_0 \equiv \{x \in \mathbb{R}^n : f(x) \leq f(x^0)\}$  with  $x^0$  the starting point of the generated sequence of iterates. Hence, the level set is compact. Furthermore, we assume that  $f$  satisfies all the assumptions of Sect. 2 on an open neighborhood  $\tilde{\mathcal{N}}_0$  of  $\mathcal{N}_0$ . In [4] it was proven that the piecewise linearization  $f_x^{PL}$  yields a second order approximation of the underlying function  $f$ . Therewith, it holds that

$$\begin{aligned} f(x + \Delta x) &= f(x) + \Delta f(x; \Delta x) + \mathcal{O}(\|\Delta x\|^2) \\ &\leq f(x) + \Delta f(x; \Delta x) + c\|\Delta x\|^2 \end{aligned} \tag{23}$$

with the coefficient  $c \in \mathbb{R}$ . Subsequently, this coefficient is set as  $c := \frac{1}{2}\tilde{q}$ . The coefficient  $\tilde{q}(x; \Delta x)$  can be computed for certain  $x$  and  $\Delta x$ . However, it is possible that  $\tilde{q}(x; \Delta x)$  is negative and thus, the local quadratic model is not bounded below. Therefore, the coefficient  $\hat{q}(x; \Delta x)$  is chosen as

$$\hat{q}(x; \Delta x) \equiv |\tilde{q}(x; \Delta x)| = \frac{2|f(x + \Delta x) - f(x) - \Delta f(x; \Delta x)|}{\|\Delta x\|^2}. \tag{24}$$

By doing so, one obtains from Eq. (23) for all descent directions  $\Delta x$  the estimate

$$f(x + \Delta x) - f(x) \leq \Delta f(x; \Delta x) + \frac{1}{2}\hat{q}(x; \Delta x)\|\Delta x\|^2 \leq 0. \tag{25}$$

In [4, Proposition 1] it was proven as well that there exists a monotonic mapping  $\bar{q}(\delta) : [0, \infty) \rightarrow [0, \infty)$  such that for all  $x \in \mathcal{N}_0$  and  $\Delta x \in \mathbb{R}^n$

$$\frac{2|f(x + \Delta x) - f(x) - \Delta f(x; \Delta x)|}{\|\Delta x\|^2} \leq \bar{q}(\|\Delta x\|) \tag{26}$$

under the assumptions of this section. This holds because if the line segment  $[x, x + \Delta x]$  is fully contained in  $\tilde{\mathcal{N}}_0$ , then the scalar  $\bar{q}(\|\Delta x\|)$  denotes the constant of [4, Proposition 1]. Otherwise those steps  $\Delta x$  for which the line segment  $[x, x + \Delta x]$  is not fully contained in  $\tilde{\mathcal{N}}_0$  must have a certain minimal size, since the base points  $x$  are restricted

to  $\mathcal{N}_0$ . Then the denominators in Eq. (26) are bounded away from zero so that  $\bar{q}(\|\Delta x\|)$  exists.

Since  $\bar{q}$  is a monotonic descending mapping which is bounded below, it converges to some limit  $\bar{q}^* \in (0, \infty)$ . Nevertheless  $\bar{q}$  will generally not be known, so that we approximate it by estimates, referred to as quadratic coefficients throughout. We generate the sequences of iterates  $\{x^k\}_{k \in \mathbb{N}}$  with  $x^k \in \mathcal{N}_0$  and corresponding steps  $\{\Delta x^k\}_{k \in \mathbb{N}}$  with  $\Delta x^k \in \mathbb{R}^n$  by Algorithm 1 and consistently update the quadratic coefficient starting from some  $q^0 > 0$  according to

$$q^{k+1} = \max\{\hat{q}^{k+1}, \mu q^k + (1 - \mu)\hat{q}^{k+1}, q^{lb}\} \tag{27}$$

with  $\hat{q}^{k+1} := \hat{q}(x^k; \Delta x^k)$ ,  $\mu \in [0, 1]$  and where  $q^{lb} > 0$  is a lower bound. Then the following lemma holds.

**Lemma 2** *Under the general assumptions of this section, one has:*

- a) *The sequence of steps  $\{\Delta x^k\}_{k \in \mathbb{N}}$  exists.*
- b) *The sequences  $\{\Delta x^k\}_{k \in \mathbb{N}}$  and  $\{\hat{q}^k\}_{k \in \mathbb{N}}$  are uniformly bounded.*
- c) *The sequence  $\{q^k\}_{k \in \mathbb{N}}$  is bounded.*

*Proof* a) By minimizing the supposed upper bound  $\Delta f(x^k; \Delta x) + \frac{1}{2}q^k \kappa \|\Delta x\|^2$  on  $f(x^k + \Delta x) - f(x^k)$  at least locally we always obtain a step

$$\Delta x^k \equiv \arg \min_s (\Delta f(x^k; s) + \frac{\kappa}{2}q^k \|s\|^2).$$

A globally minimizing step  $\Delta x^k$  must exist since  $\Delta f(x^k; s)$  can only decrease linearly so that the positive quadratic term always dominates for large  $\|s\|$ . Moreover,  $\Delta x^k$  vanishes only at first order minimal points  $x^k$  where  $\Delta f(x^k; s)$  and  $f'(x^k; s)$  have the local minimizer  $s = 0$ .

b) It follows from  $q^k \geq q^{lb} > 0$  and the continuity of all quantities on the compact set  $\mathcal{N}_0$  that the step size  $\delta \equiv \|\Delta x\|$  must be uniformly bounded by some  $\bar{\delta}$ . This means that the  $\hat{q}$  are uniformly bounded by  $\bar{q} \equiv \bar{q}(\bar{\delta})$ .

c) Obviously, the sequence  $\{q^k\}_{k \in \mathbb{N}}$  is bounded below by  $q^{lb}$ . Considering the first two arguments of Eq. (27), one obtains that  $q^{k+1} = \hat{q}^{k+1}$  and  $q^{k+1} > q^k$  if  $\hat{q}^{k+1} > \mu q^k + (1 - \mu)\hat{q}^{k+1}$ . Respectively, if  $\hat{q}^{k+1} \leq \mu q^k + (1 - \mu)\hat{q}^{k+1}$ , one obtains  $q^{k+1} \geq \hat{q}^{k+1}$  and  $q^{k+1} \leq q^k$ . This means that the maximal element of the sequence is given by a  $\hat{q}^j$  with  $j \in \{1, \dots, k + 1\}$  and thus bounded by  $\bar{q}(\|\Delta x^j\|)$ . Therefore, the sequence  $\{q^k\}_{k \in \mathbb{N}}$  is bounded above.  $\square$

The proof of Lemma 2 c) gives us the important insight that  $q^{k+1} \geq \hat{q}^{k+1}$  holds. With these results we can now prove the main convergence result of this paper.

**Theorem 4** *Let  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  be a piecewise smooth function as described at the beginning of Sect. 2 which is bounded below and has a bounded level set  $\mathcal{N}_0 = \{x \in \mathbb{R}^n \mid f(x) \leq f(x^0)\}$  with  $x^0$  the starting point of the generated sequence of iterates  $\{x^k\}_{k \in \mathbb{N}}$ .*

Then a cluster point  $x^*$  of the infinite sequence  $\{x^k\}_{k \in \mathbb{N}}$  generated by Algorithm 1 exists. All cluster points of the infinite sequence  $\{x^k\}_{k \in \mathbb{N}}$  are Clarke stationary points of  $f$ .

*Proof* The sequence of steps  $\{\Delta x^k\}_{k \in \mathbb{N}}$  is generated by solving the overestimated quadratic problem in step 2 of Algorithm 1 of the form

$$\Delta x^k = \arg \min_s (\Delta f(x; s) + \frac{1}{2} \kappa q^k \|s\|^2).$$

Unless  $x^k$  satisfies first order optimality conditions the step  $\Delta x^k$  satisfies

$$\Delta f(x^k; \Delta x^k) + \frac{1}{2} \kappa q^k \|\Delta x^k\|^2 < 0. \tag{28}$$

Therewith, one obtains from Eq. (25)

$$f(x^k + \Delta x^k) - f(x^k) \leq \frac{1}{2} [q^{k+1} - \kappa q^k] \|\Delta x^k\|^2 \tag{29}$$

where  $\hat{q}^{k+1} \leq q^{k+1}$  holds as a result of Eq. (27) and due to Eq. (28) one has  $\Delta f(x^k; \Delta x^k) \leq -\frac{1}{2} q^k \kappa \|\Delta x^k\|^2$ . The latter inequality can be overestimated by applying the limit superior  $\bar{q} = \limsup_{k \rightarrow \infty} q^{k+1}$  as follows:

$$f(x^k + \Delta x^k) - f(x^k) \leq \frac{1}{2} [\bar{q} - \kappa q^k] \|\Delta x^k\|^2.$$

Considering a subsequence of  $\{q^{k_j}\}_{j \in \mathbb{N}}$  converging to the limit superior, it follows that for each  $\epsilon > 0$  there exists  $\bar{j} \in \mathbb{N}$  such that for all  $j \geq \bar{j}$  one obtains  $\|\bar{q} - q^{k_j}\| < \epsilon$ . Therewith the overestimated local problem provides that the term  $\bar{q} - \kappa q^{k_j} < 0$ . Since the objective function  $f$  is bounded below on  $\mathcal{N}_0$ , infinitely many significant descent steps can not be performed and thus  $f(x^{k_j} + \Delta x^{k_j}) - f(x^{k_j})$  has to converge to 0 as  $j$  tends towards infinity. As a consequence, the right hand side of Eq. (29) has to tend towards 0 as well. Therefore, the subsequence  $\{\Delta x^{k_j}\}_{j \in \mathbb{N}}$  is a null sequence. Since the level set  $\mathcal{N}_0$  is compact, the sequence  $\{x^{k_j}\}_{j \in \mathbb{N}}$  has a subsequence that tends to a cluster point  $x^*$ . Hence, a cluster point  $x^*$  of the sequence  $\{x^k\}_{k \in \mathbb{N}}$  exists.

Assume that the subsequence  $\{x^{k_j}\}$  of  $\{x^k\}$  converges to a cluster point. As shown above the corresponding sequence of penalty coefficients  $\{\Delta x^{k_j}\}_{j \in \mathbb{N}}$  converges to zero if  $j$  tends to infinity. Therewith, one can apply Lemma 1 at the cluster point  $x^*$ , where it was proven that if  $\hat{f}_x$  is Clarke stationary at  $\Delta x = 0$  for one  $q \geq 0$ , then the piecewise smooth function  $f$  is Clarke stationary in  $x$  yielding the assertion.  $\square$

### 4 Numerical results

The nonsmooth optimization method LiPsMin introduced in this paper will be tested in the following section. Therefore, we introduce piecewise linear and piecewise smooth test problems in the Sects. 4.1 and 4.2. In both cases the test set contains convex and

nonconvex test problems. In Sect. 4.3 results of numerous optimization runs will be given and compared to other nonsmooth optimization software.

### 4.1 Piecewise linear test problems

The test set of piecewise linear problems comprises:

#### 1. Counterexample of HUL

$$f(x) = \max \{-100, 3x_1 \pm 2x_2, 2x_1 \pm 5x_2\}, \quad (x_1, x_2)^0 = (9, -2).$$

#### 2. MXHILB

$$f(x) = \max_{1 \leq i \leq n} \left| \sum_{j=1}^n \frac{x_j}{i+j-1} \right|, \quad x_i^0 = 1, \quad \text{for all } i = 1, \dots, n.$$

#### 3. MAXL

$$f(x) = \max_{1 \leq i \leq n} |x_i|, \quad x_i^0 = i, \quad \text{for all } i = 1, \dots, n.$$

#### 4. Second Chebyshev–Rosenbrock

$$f(x) = \frac{1}{4} |x_1 - 1| + \sum_{i=1}^{n-1} |x_{i+1} - 2|x_i| + 1|$$

$$x_i^0 = -0.5, \text{ when } \text{mod}(i, 2) = 1, i = 1, \dots, n \text{ and}$$

$$x_i^0 = 0.5, \text{ when } \text{mod}(i, 2) = 0, i = 1, \dots, n.$$

Table 4 provides further information about the test problems such as the optimal value  $f^*$  of the function, the dimension  $n$  and the number of absolute value functions  $s$  occurring during the function evaluation depending on the dimension  $n$ . With  $s$  given in this way the relation of  $n$  and  $s$  can be given as well. Additionally, a reference is given for each test problem.

**Table 4** Information about piecewise linear test problems

	$n$	$s$	$s \sim n$	$f^*$	Properties	References
1	2	$2n$	$n < s$	-100	PL, convex	[11]
2	Any	$2n - 1$	$n \leq s$	0	PL, convex	[10]
3	Any	$2n - 1$	$n \leq s$	0	PL, convex	[16]
4	Any	$2n - 1$	$n \leq s$	0	PL, nonconvex	[9]



### 4.2 Piecewise smooth test problems

The test set of piecewise smooth problems is listed below.

#### 5. MAXQ

$$f(x) = \max_{1 \leq i \leq n} x_i^2,$$

$$x_i^0 = i, \text{ for } i = 1, \dots, n/2 \quad \text{and}$$

$$x_i^0 = -i, \text{ for } i = n/2 + 1, \dots, n.$$

#### 6. Chained LQ

$$f(x) = \sum_{i=1}^{n-1} \max \left\{ -x_i - x_{i+1}, -x_i - x_{i+1} + (x_i^2 + x_{i+1}^2 - 1) \right\}$$

$$x_i^0 = -0.5, \text{ for all } i = 1, \dots, n.$$

#### 7. Chained CB3 II

$$f(x) = \max \{ f_1(x), f_2(x), f_3(x) \},$$

with  $f_1(x) = \sum_{i=1}^{n-1} (x_i^4 + x_{i+1}^2), \quad f_2(x) = \sum_{i=1}^{n-1} ((2 - x_i)^2 + (2 - x_{i+1})^2)$

and  $f_3(x) = \sum_{i=1}^{n-1} (2e^{-x_i + x_{i+1}}),$

$$x_i^0 = 2, \text{ for all } i = 1, \dots, n.$$

#### 8. MAXQUAD

$$f(x) = \max_{1 \leq i \leq 5} (x^\top A^i x - x^\top b^i)$$

$$A_{kj}^i = A_{jk}^i = e^{j/k} \cos(jk) \sin(i), \text{ for } j < k, \quad j, k = 1, \dots, 10$$

$$A_{jj}^i = \frac{j}{10} |\sin(i)| + \sum_{k \neq j} |A_{jk}^i|,$$

$$b_j^i = e^{j/i} \sin(ij),$$

$$x_i^0 = 0, \text{ for all } i = 1, \dots, 10.$$

#### 9. Chained Crescent I

$$f(x) = \max \{ f_1(x), f_2(x) \},$$

with  $f_1(x) = \sum_{i=1}^{n-1} (x_i^2 + (x_{i+1} - 1)^2 + x_{i+1} - 1)$

**Table 5** Information about piecewise smooth test problems

$n$	$s$	$s \sim n$	$f^*$	Properties	References
5	Any	$n - 1$	$s < n$	0	PS, convex [10]
6	Any	$n - 1$	$s < n$	$-(n - 1)2^{1/2}$	PS, convex [10]
7	Any	2	$s \leq n$	$2(n - 1)$	PS, convex [10]
8	10	4	$s < n$	$-0.8414083$	PS, convex [15]
9	Any	2	$s \leq n$	0	PS, nonconvex [10]
10	Any	$n - 1$	$s \leq n$	0	PS, nonconvex [10]
11	Any	$n - 1$	$s < n$	0	PS, nonconvex [9]
12	Any	$n + 1$	$n < s$	0	PS, nonconvex [10]

and  $f_2(x) = \sum_{i=1}^{n-1} \left( -x_i^2 - (x_{i+1} - 1)^2 + x_{i+1} + 1 \right)$ ,  
 $x_i^0 = -1.5$ , when  $\text{mod}(i, 2) = 1, i = 1, \dots, n$  and  
 $x_i^0 = 2$ , when  $\text{mod}(i, 2) = 0, i = 1, \dots, n$ .

**10. Chained Crescent II**

$f(x) = \sum_{i=1}^{n-1} \max \{ f_{1,i}(x), f_{2,i}(x) \}$ ,  
 with  $f_{1,i}(x) = x_i^2 + (x_{i+1} - 1)^2 + x_{i+1} - 1$ ,  
 and  $f_{2,i}(x) = -x_i^2 - (x_{i+1} - 1)^2 + x_{i+1} + 1$ ,  
 $x_i^0 = -1.5$ , when  $\text{mod}(i, 2) = 1, i = 1, \dots, n$  and  
 $x_i^0 = 2$ , when  $\text{mod}(i, 2) = 0, i = 1, \dots, n$ .

**11. First Chebyshev–Rosenbrock**

$f(x) = \frac{1}{4}(x_1 - 1)^2 + \sum_{i=1}^{n-1} \left| x_{i+1} - 2x_i^2 + 1 \right|$   
 $x_i^0 = -0.5$ , when  $\text{mod}(i, 2) = 1, i = 1, \dots, n$  and  
 $x_i^0 = 0.5$ , when  $\text{mod}(i, 2) = 0, i = 1, \dots, n$ .

**12. Number of active faces**

$f(x) = \max_{1 \leq i \leq n} \left\{ g \left( -\sum_{j=1}^n x_j \right), g(x_i) \right\}$ , where  $g(y) = \ln(|y| + 1)$ .  
 $x_i^0 = 1$ , for all  $i = 1, \dots, n$ .

In Table 5 further information about the test problems are given, compare Table 4.

### 4.3 Performance results of LiPsMin and comparison with other nonsmooth optimization methods

In the following, the introduced routine LiPsMin is compared with the nonsmooth optimization routine MPBNGC, which is a proximal bundle method described in [16], and the quasi-Newton type method HANSO described in [14].

The idea of the bundle method MPBNGC is to approximate the subdifferential of the objective function at the current iterate by collecting subgradients of previous iterates and storing them into a bundle. Thus, more information about the local behavior of the function is available. To reduce the required storage the amount of stored subgradients has to be restricted. Therefore an aggregated subgradient is computed from several previous subgradients so that these subgradients can be removed without losing their information. For more details see [16, 17].

HANSO 2.2 combines the BFGS method [14] with an inexact line search and the gradient sampling approach described in [1]. The gradient sampling approach is a stabilized steepest descent method. At each iterate the corresponding gradient and additional gradients of nearby points are evaluated. The descent direction is chosen as the vector with the smallest norm in the convex hull of these gradients. However, by default the gradient sampling method is not used in HANSO 2.2, since it is mainly required for coherent convergence results.

As stopping criteria of routines LiPsMin and PLMin we used  $\epsilon = 1e-8$  and the maximal iteration number  $maxIter = 1000$ . In the implementation of Algorithm 1 we chose the parameter  $\mu = 0.9$ . Under certain conditions, e.g., a large number of active constraints at the optimal point, it is reasonable to add a termination criterion that considers the reduction of the function value in two consecutive iterations, i.e.,  $|f(x_k) - f(x_{k+1})| < \epsilon$ . For the bundle method MPBNGC we choose the following parameter settings. The maximal bundle size equals the dimension  $n$ . If the considered test function is convex then the parameter  $\gamma_{\text{am}}$  equals 0 otherwise  $\gamma_{\text{am}}$  is set to 0.5. Further stopping criteria are the number of iterations  $nIter = 10,000$  (Iter), the number of function and gradient evaluations  $NFASG = 10,000$  ( $\#f = \#\nabla f$ ) and the final accuracy  $EPS = 1e-8$ . For HANSO we choose  $normtol = 1e-8$  and  $maxit = 10,000$  (Iter).

#### *Performance results of piecewise linear test problems*

First, we consider the problems of the piecewise linear test set, see Sect. 4.1. The results of the piecewise linear and convex problems are presented in Table 6 and Table 7. Each table contains all results of a single test problem generated by the three different optimization routines mentioned. The columns of the tables give the dimension  $n$  of the problem, the final function value  $f^*$ , the number of function evaluations  $\#f$ , the number of gradient evaluations  $\#\nabla f$  and number of iterations. Additionally, the initial penalty coefficient  $q^0$  of LiPsMin is given. Since we assume that all our test problems are bounded below and we consider piecewise linear problems, the initial penalty coefficient is chosen as  $q^0 = 0$ . For the test problem MXHILB the additional stopping criterion considering the function value reduction was added.

**Table 6** Results for Counterexample of HUL

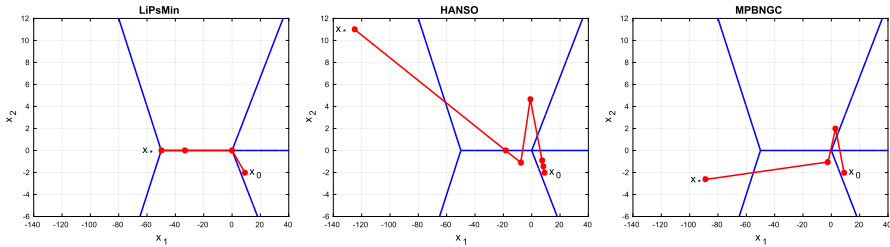
	$n$	$f^*$	$\#f$	$\#\nabla f$	Iter
LiPsMin	2	-100	3	8	2
HANSO	2	-100	9	9	3
MPBNGC	2	-100	7	7	6

**Table 7** Results for MXHILB and MAXL

	$n$	MXHILB				MAXL			
		$f^*$	$\#f$	$\#\nabla f$	Iter	$f^*$	$\#f$	$\#\nabla f$	Iter
LiPsMin	2	5.6e-17	3	7	2	0	3	7	2
	5	2.7e-10	3	35	2	0	3	10	2
	10	5.6e-10	3	26	2	0	3	15	2
	20	4.7e-9	3	34	2	0	3	25	2
	50	3.0e-9	3	20	2	0	3	203	2
	100	2.1e-12	3	8	2	0	3	404	2
HANSO	2	5.5e-7	47	47	23	1.5e-5	21	21	17
	5	1.4e-29	636	636	255	7.6e-6	28	28	21
	10	1.5e-13	643	643	259	7.6e-6	38	38	26
	20	1.4e-12	782	782	328	3.8e-6	59	59	37
	50	3.0e-11	879	879	382	3.8e-6	119	119	67
	100	2.8e-11	1233	1233	494	1.9e-6	220	220	118
MPBNGC	2	4.2e-17	9	9	8	3.4e-9	67	67	42
	5	1.8e-8	10,000	10,000	1681	1.7e-16	17	17	15
	10	2.9e-10	11	11	10	1.2e-13	29	29	27
	20	1.6e-9	13	13	11	3.8e-11	56	56	54
	50	5.5e-12	15	15	13	1.0e-12	123	123	121
	100	4.0e-10	18	18	15	1.7e-9	176	176	165

In Fig. 3 a comparison of the behavior of optimization runs generated by LiPsMin, HANSO, and MPBNGC is illustrated for the Counterexample of HUL. As intended LiPsMin uses the additional information of the polyhedral decomposed domain efficiently in order to minimize the number of iterations. As a consequence the optimization run computed by LiPsMin is more predictable and purposeful than the runs computed by HANSO and MPBNGC. This behavior is characteristic for all piecewise linear problems solved by LiPsMin. In contrast to the first three test problems the 2nd Chebyshev–Rosenbrock function is nonconvex. The corresponding results are given in Table 8. For  $n > 2$ , all optimization routines failed to detect the unique minimizer. The detected points are Clarke stationary points.

To distinguish minimizers and Clarke stationary points that are not optimal, new optimality conditions were established in [7]. These optimality conditions are based on the *linear independent kink qualification* (LIKQ) which is a generalization of LICQ familiar from nonlinear optimization. It is shown in the mentioned article that the 2nd Chebyshev–Rosenbrock function satisfies LIKQ globally, i.e., throughout  $\mathbb{R}^n$ . The following special purpose algorithm for piecewise linear functions was designed such that it can only stop at stationary points satisfying first order minimality conditions. The call of the routine ComputeDesDir() in Algorithm 3 is replaced by a reflection



**Fig. 3** Optimization runs of test problem HUL performed by LiPsMin, HANSO and MPBNGC

**Table 8** Results for 2nd Chebyshev–Rosenbrock

	$n$	$f^*$	$\#f$	$\#\nabla f$	Iter
LiPsMin	2	1.29e-11	4	11	3
	5	1.9e-1	5	53	4
	10	4.0e-1	4	42	3
	20	4.0e-1	3	45	2
	50	4.0e-1	3	57	2
HANSO	100	4.0e-1	3	120	2
	2	3.8e-7	211	211	61
	5	3.1e-1	8742	8742	2521
	10	4.0e-1	1383	1383	514
	20	4.0e-1	5268	5268	2177
MPBNGC	50	4.0e-1	2607	2607	463
	100	4.0e-1	2278	2278	290
	2	1.7e-16	80	80	52
	5	2.5e-1	10,000	10,000	3561
	10	4.0e-1	10,000	10,000	9807
	20	4.0e-1	66	66	65
	50	4.0e-1	188	188	187
	100	3.5e-1	251	251	249

of the signature vector  $\sigma$  of the current polyhedron into the opposing polyhedron by switching all active signs from 1 to -1 or vice versa.

**Algorithm 5 (PLMin\_Reflection)**

**PLMin\_Reflection**( $x$ ) // Precondition:  $x, \Delta x \in \mathbb{R}^n, \Delta x = 0$

1. Determine solution  $\delta x$  of local QP (22) on current  $P_\sigma$ .
2. If  $\|\delta x\| > \epsilon$   
 Compute new  $\sigma$  by switching all active signs.  
 Update  $\Delta x = \Delta x + \delta x$ .  
 Go to 1.

else

STOP.

return  $\Delta x$

The results of Algorithm 5 applied on the 2nd Chebyshev–Rosenbrock function are given in Table 9. As stopping criterion we used the tolerance  $\epsilon = 1e-8$ . Each evaluated gradient  $\nabla f$  represents an open polyhedron. The algorithm takes  $2^{n-1}$  iterations, which indicates that from the given starting point it visits almost all the other  $2^{n-1}$

**Table 9** Results for 2nd Chebyshev–Rosenbrock with Algorithm 5 incorporated into LiPsMin

$n$	$f^*$	# f	# $\nabla f$	Iter
2	1.3e–11	3	6	2
5	1.1e–16	3	17	2
10	1.6e–14	3	414	2
20	1.1e–16	4	419,438	3

**Table 10** Results for MAXQ,  $q^0 = 0.1$  and Chained LQ,  $q^0 = 0.1$

	$n$	MAXQ				Chained LQ			
		$f^*$	#f	# $\nabla f$	Iter	$f^*$	#f	# $\nabla f$	Iter
LiPsMin	2	2.3e–9	27	156	26	– 1.41421	10	26	9
	5	1.8e–9	36	486	35	– 5.65685	47	471	46
	10	2.7e–9	34	939	33	– 12.7278	15	128	18
	20	1.9e–9	36	1980	35	– 26.8701	15	258	14
	50	1.4e–8	58	8011	57	– 69.2965	15	646	14
HANSO	100	3.5e–8	117	32,359	116	– 140.007	15	1341	14
	2	3.2e–19	18	18	16	– 1.41421	156	156	51
	5	3.0e–17	242	242	116	– 5.65685	190	190	52
	10	6.2e–16	787	787	352	– 12.7279	315	315	100
	20	1.1e–16	1362	1362	637	– 26.8701	578	578	165
MPBNGC	50	2.1e–16	4409	4409	1906	– 69.2965	1238	1238	274
	100	3.0e–16	8922	8922	3991	– 140.007	2353	2353	416
	2	7.6e–9	15	15	14	– 1.41421	9	9	8
	5	3.1e–9	60	60	49	– 5.65685	34	34	30
	10	3.4e–9	126	126	34	– 12.7279	40	40	33
	20	2.6e–9	244	244	222	– 26.8701	63	63	61
	50	3.8e–9	577	577	549	– 69.2965	143	143	108
	100	4.5e–9	1118	1118	1083	– 140.007	468	468	273

stationary points before reaching the unique minimizer. That enormous effort may be unavoidable for a constant descent method. We are currently in the process of incorporating a variant of the reflection idea into the general purpose PLMin algorithm.

In summary it can be said, that we use many fewer iterations on PL problems than the other software packages HANSO and MPBNGC, but searching for a descent direction still requires a significant number of active gradients that overall corresponds to the number of gradient (and function) evaluations of the bundle method.

*Performance results of piecewise smooth test problems*

In the following, we consider the problems of the piecewise smooth test set, as introduced in Sect. 4.2. The results of the piecewise smooth problems are presented in Tables 10, 11, 12 and 13. Each table contains all results of a single test problem generated by the three different optimization routines mentioned above. The initial penalty coefficient is chosen as  $q_0 = 0.1$  in most cases. In the other cases it is chosen as  $q_0 = 1$ .

In Tables 10, 11 and 12 the results of the piecewise smooth and convex test problems are presented. The additional stopping criterion of LiPsMin considering the function

**Table 11** Results for Chained CB3 II,  $q^0 = 1$

	$n$	$f^*$	$\#f$	$\#\nabla f$	Iter
LiPsMin	2	2.00000	12	72	11
	5	8.00000	69	530	68
	10	18.0000	67	515	66
	20	38.0000	63	482	62
	50	98.0000	61	465	60
	100	198.000	59	449	58
HANSO	2	2.00000	193	193	65
	5	8.00000	270	270	96
	10	18.0000	198	198	71
	20	38.0000	215	215	80
	50	98.0000	202	202	79
	100	198.000	208	208	69
MPBNGC	2	2.00001	10,000	10,000	9999
	5	8.00000	35	35	34
	10	18.0000	46	46	45
	20	38.0000	42	42	41
	50	98.0000	60	60	59
	100	198.000	41	41	40

**Table 12** Results for MAXQUAD

	$n$	$f^*$	$\#f$	$\#\nabla f$	Iter
LiPsMin	10	-8.414083	48	426	47
HANSO	10	0	32	32	1
HANSO with GS	10	-8.41397	2943	2943	1 + 3 GS
MPBNGC	10	-8.414083	40	40	39

value reduction was applied for test problem MAXQ. A large number of optimization runs detected successfully minimal solutions. The bundle method MPBNGC stopped once because the maximal number of function and gradient evaluations was reached, see Table 11. HANSO failed once, see Table 12. By enabling the gradient sampling mode, the minimal point could be detected as indicated in the additional row of that table. The number of function value and gradient evaluations are hardly comparable as a consequence of the varying underlying information. However, several optimization runs performed by HANSO, see, e.g., Table 10, and LiPsMin, see, e.g., Table 10 resulted in a higher number of function and gradient evaluations than the other two respective routines, whereas the iteration numbers of all three routines are of comparable order of magnitude.

In Tables 13 and 14 the results of the piecewise smooth and nonconvex test problems are presented. From general theory one can expect that these problems are difficultly to solve and indeed the results are not as clear as those results of the previously considered test problems. The results of test problems Chained Crescent I and Number of active faces given in Tables 13 and 14 are encouraging. However, the final function values generated by HANSO are partly less accurate than expected and MPBNGC terminates several times because the maximum number of function and gradient evaluation was reached. Comparing test problems Chained Crescent I and II one can see how minor

**Table 13** Results for Chained Cresent II,  $q^0 = 0.1$ , and Chained Cresent I,  $q^0 = 1$

	$n$	Chained Cresent II				Chained Cresent I			
		$f^*$	$\#f$	$\#\nabla f$	Iter	$f^*$	$\#f$	$\#\nabla f$	Iter
LiPsMin	2	6.4e-13	53	304	52	7.0e-13	56	327	55
	5	8.3e-13	62	695	61	8.0e-13	61	347	60
	10	5.8e-13	64	1228	63	9.1e-13	64	375	63
	20	9.1e-13	64	2231	63	9.5e-13	65	381	64
	50	7.0e-13	65	5260	64	1.1e-13	149	875	148
HANSO	100	7.9e-13	65	10,251	64	2.1e-13	92	543	91
	2	0	175	175	51	0	175	175	51
	5	8.9e-16	346	346	120	0	170	170	57
	10	5.2e-15	626	626	238	0	171	171	49
	20	1.1e-14	1155	1155	292	0	178	178	48
MPBNGC	50	7.4e-7	453	453	92	1.7e-15	180	180	52
	100	4.3e-7	457	457	111	1.7e-15	145	145	46
	2	1.4e-8	52	52	45	1.4e-8	52	52	45
	5	4.1e-9	80	80	79	4.2e-9	78	78	57
	10	5.1e-9	196	196	195	1.1e-8	49	49	40
	20	7.0e-9	488	488	485	4.5e-9	65	65	50
	50	7.1e-9	519	519	518	5.7e-9	167	167	83
	100	8.0e-9	733	733	684	4.2e-9	96	96	66

changes in the objective function influence the optimization results. The required number of iterations of all three optimization methods increased.

As in the piecewise linear case the Chebyshev–Rosenbrock function seems to be more difficult than the other test problems. According to [9] the function has only one Clarke stationary point which is the minimizer as well. Only a few optimization runs detected the minimizer  $f^* = 0$  with sufficient accuracy. In the test with  $n \geq 10$  none of the methods seem to reach the unique stationary point within the given computational limitations. LiPsMin and HANSO often get stuck at a pseudo stationary value of 0.81814. This effects warrants further investigation.

The above results illustrate that in the piecewise smooth case the introduced optimization method LiPsMin compares well with the state of the art optimization software HANSO and MPBNGC. These results can be confirmed by Figs. 4 and 5 that give an idea of the convergence rate of LiPsMin. Each figure corresponds to one test problem and it shows how the function value  $f(x_k)$  of the  $k$ -th iteration decreases during an optimization run for  $n = 10$  and  $n = 100$ . On PS problems with  $s = n$  active switches we can expect superlinear convergence. Otherwise curvature becomes important, which is not reflected in the piecewise linear models. In particular for unconstrained smooth problems our method reduces to steepest descent with a particular line-search. Hence the convergence rate can only be linear as indicated in Figs. 4 and 5. A superlinear rate can possibly be achieved by generalizing the proximal term to an ellipsoidal norm defined by a positive definite approximation of the Hessian of a suitable Lagrangian.

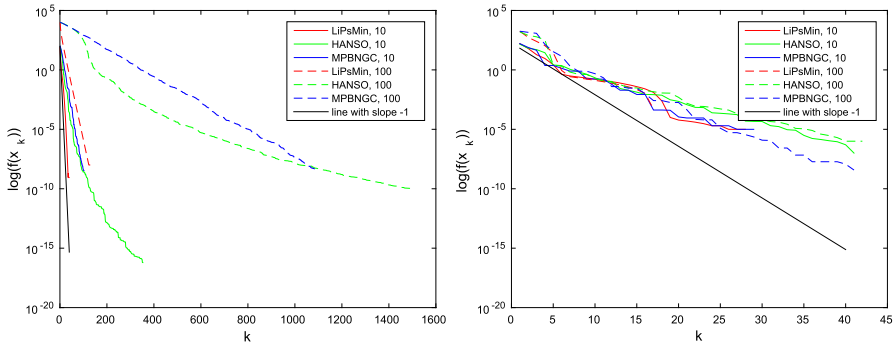
### 5 Conclusion and Outlook

In [8] we proposed a method for the optimization of Lipschitzian piecewise smooth functions by successive piecewise linearization. The central part of that previous article

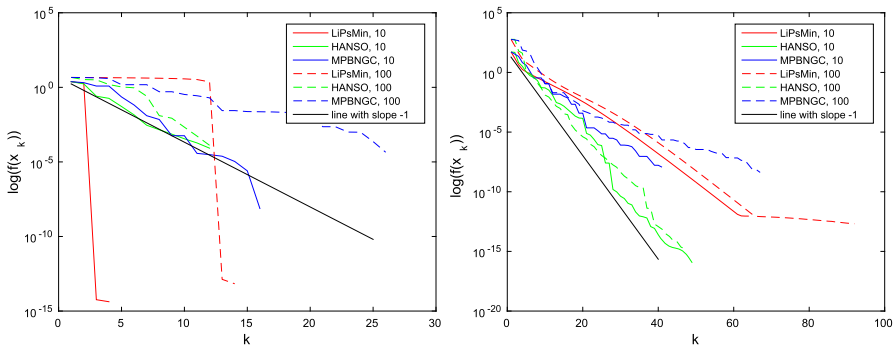


**Table 14** Results for 1st Chebyshev–Rosenbrock,  $q^0 = 0.1$ , and Number of Active Faces,  $q^0 = 0.1$

	$n$	1st Chebyshev–Rosenbrock				Number of active faces			
		$f^*$	$\#f$	$\#\nabla f$	Iter	$f^*$	$\#f$	$\#\nabla f$	Iter
LiPsMin	2	2.2e-14	308	1814	307	6.7e-16	3	7	2
	5	6.4e-2	1001	15,095	1000	2.2e-16	4	8	3
	10	0.81744	1001	30,371	1000	4.2e-15	4	8	3
	20	0.81814	1001	42,150	1000	8.2e-15	5	9	4
	50	0.81814	6	256	5	2.5e-14	9	13	8
HANSO	100	0.81814	6	542	5	6.8e-14	14	18	13
	2	9.6e-15	594	594	255	2.1e-6	23	23	8
	5	7.0e-2	27,980	27,980	10,000	1.3e-5	24	24	11
	10	0.81707	18,753	18,753	10,000	8.4e-5	23	23	11
	20	0.81814	2232	2232	1093	3.2e-5	25	25	9
MPBNGC	50	0.81814	2654	2654	482	2.4e-5	27	27	11
	100	0.81814	5173	5173	757	1.3e-4	29	29	11
	2	8.7e-9	143	143	75	8.2e-13	20	20	14
	5	0.16995	10,000	10,000	3509	2.2e-10	12	12	10
	10	0.62647	10,000	10,000	9786	7.6e-9	20	20	15
	20	0.81814	201	201	200	2.0e-7	10,000	10,000	9995
	50	0.63027	353	353	352	6.4e-7	10,000	10,000	6428
	100	0.06109	2283	2283	1842	4.4e-5	10,000	10,000	9993



**Fig. 4** Convergence behavior of LiPsMin, HANSO and MPBNGC for MAXQ (left) and Chained CB3 II (right) with  $n = 10$  and  $n = 100$



**Fig. 5** Convergence behavior of LiPsMin, HANSO and MPBNGC for number of active faces (left) and Chained Crescent I (right) with  $n = 10$  and  $n = 100$

was the concept of piecewise linearization and the exploitation of the therewith gained information. An approach of the outer routine LiPsMin was introduced and tested. In the current work we gave an overview of a refined version of the LiPsMin method.

While computing the results of the piecewise linear examples by the inner routine PLMin in [8], it became obvious that the computation of the critical step multiplier during the line-search was not as numerically accurate as required. Therefore, it was replaced by the more efficient solution of the quadratic subproblem introduced in Sect. 3.2. For this adapted inner routine of LiPsMin we confirmed convergence in finitely many iteration. A first version of the LiPsMin algorithm was introduced in [4]. In the current version a more aggressive updating strategy of the penalty multiplier  $q$  was applied. In Sect. 3.3 it was proven that LiPsMin combined with the new updating strategy maintains its global convergence to a stationary point. This property holds for HANSO in a probabilistic sense if the gradient sampling is switched on, which does however considerably reduce the efficiency on most problems.

The performance results of LiPsMin in the piecewise linear case, see Sect. 4.3, confirmed well our expectation. The incorporation of additional information gained by the abs-normal form leads to a pointed and predictable descent trajectory. In the

piecewise smooth case the performance results generated by LiPsMin also compared well with the state of the art optimization software tools MPBNGC and HANSO. Both the tables and the figures of Sect. 4 affirmed this conclusion. The results also indicated that LiPsMin may converge superlinearly under certain conditions. These conditions should be analyzed in future work.

Nevertheless, the results also illustrated that it is useful to check if stationary points are also minimal points. That is why it should be an important question of future work how to incorporate the optimality conditions of [7] such that minimizers can be uniquely identified. Hence, it would be beneficial to gain more information about the polyhedral decomposition of the domain, such as convexity properties of the function. The additional information can be used to identify the subsequent polyhedron more efficiently which is especially relevant when the considered function is high dimensional or includes a higher number of absolute value function evaluations.

**Acknowledgements** The authors are indebted to the two referees for their careful reading of the first submission and their many objections and suggestions, which helped to greatly improve the consistency and the readability of the paper.

## References

1. Burke, J., Lewis, A., Overton, M.: A robust gradient sampling algorithm for nonsmooth nonconvex optimization. *SIAM J. Optim.* **15**(3), 751–779 (2005)
2. Clarke, F.: *Optimization and Nonsmooth Analysis*. SIAM, Philadelphia (1990)
3. de Oliveira, W., Sagastizábal, C.: Bundle methods in the XXIst century: a birds'-eye view. *Pesqui. Oper.* **34**(3), 647–670 (2014)
4. Griewank, A.: On stable piecewise linearization and generalized algorithmic differentiation. *Optim. Methods Softw.* **28**(6), 1139–1178 (2013)
5. Griewank, A., Bernt, J.U., Radons, M., Streubel, T.: Solving piecewise linear systems in abs-normal form. *Linear Algebra Appl.* **471**, 500–530 (2015)
6. Griewank, A., Walther, A.: *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. SIAM, Philadelphia (2008)
7. Griewank, A., Walther, A.: First and second order optimality conditions for piecewise smooth objective functions. *Optim. Methods Softw.* **31**(5), 904–930 (2016)
8. Griewank, A., Walther, A., Fiege, S., Bosse, T.: On Lipschitz optimization based on gray-box piecewise linearization. *Math. Program. Ser. A* **158**, 383–415 (2016)
9. Gurbuzbalaban, M., Overton, M.L.: On Nesterov's nonsmooth Chebyshev–Rosenbrock functions. *Nonlinear Anal. Theory Methods Appl.* **75**(3), 1282–1289 (2012)
10. Haarala, M., Miettinen, K., Mäkelä, M.: New limited memory bundle method for large-scale nonsmooth optimization. *Optim. Methods Softw.* **19**(6), 673–692 (2004)
11. Hiriart-Urruty, J.B., Lemaréchal, C.: *Convex Analysis and Minimization Algorithms I*. Springer, Berlin (1993)
12. Karmita, N., Mäkelä, M.: Limited memory bundle method for large bound constrained nonsmooth optimization: convergence analysis. *Optim. Methods Softw.* **25**(6), 895–916 (2010)
13. Lemaréchal, C., Sagastizábal, C.: Variable metric bundle methods: from conceptual to implementable forms. *Math. Program.* **76**(3), 393–410 (1997)
14. Lewis, A., Overton, M.: Nonsmooth optimization via quasi-Newton methods. *Math. Program.* **141**(1–2), 135–163 (2013)
15. Lukšan, L., Vlček, J.: Test problems for nonsmooth unconstrained and linearly constrained optimization. Technical report 798, Institute of Computer Science, Academy of Sciences of the Czech Republic (2000)
16. Mäkelä, M., Neittaanmäki, P.: *Nonsmooth Optimization: Analysis and Algorithms with Applications to Optimal Control*. World Scientific Publishing Co., Singapore (1992)

17. Mäkelä, M.M.: Multiobjective proximal bundle method for nonconvex nonsmooth optimization: Fortran subroutine MPBNGC 2.0. Reports of the Department of Mathematical Information Technology, Series B, Scientific computing No. B 13/2003, University of Jyväskylä (2003)
18. Nesterov, Y.: Lexicographic differentiation of nonsmooth functions. *Math. Program.* **104**(2–3), 669–700 (2005)
19. Scholtes, S.: *Introduction to Piecewise Differentiable Functions*. Springer, Berlin (2012)
20. Shor, N.: *Nondifferentiable Optimization and Polynomial Problems*. Kluwer, Amsterdam (1998)