J Glob Optim (2018) 71:37–55
https://doi.org/10.1007/s10898-017-0568-z

Minimizing nonsmooth DC functions via successive DC piecewise-affine approximations

Manlio Gaudioso[1] · **Giovanni Giallombardo**[1] · **Giovanna Miglionico**[1] · **Adil M. Bagirov**[2]

Received: 13 December 2016 / Accepted: 6 September 2017 / Published online: 13 September 2017

Abstract We introduce a proximal bundle method for the numerical minimization of a nonsmooth difference-of-convex (DC) function. Exploiting some classic ideas coming from cutting-plane approaches for the convex case, we iteratively build two separate piecewise-affine approximations of the component functions, grouping the corresponding information in two separate bundles. In the bundle of the first component, only information related to points close to the current iterate are maintained, while the second bundle only refers to a global model of the corresponding component function. We combine the two convex piecewise-affine approximations, and generate a DC piecewise-affine model, which can also be seen as the pointwise maximum of several concave piecewise-affine functions. Such a nonconvex model is locally approximated by means of an auxiliary quadratic program, whose solution is used to certify approximate criticality or to generate a descent search-direction, along with a predicted reduction, that is next explored in a line-search setting. To improve the approximation properties at points that are far from the current iterate a supplementary quadratic program is also introduced to generate an alternative more promising search-direction. We discuss the main convergence issues of the line-search based proximal bundle method, and provide computational results on a set of academic benchmark test problems.

Keywords DC optimization · Nonconvex nonsmooth optimization · Cutting plane · Piecewise concave · Bundle method

✉ Giovanni Giallombardo
giovanni.giallombardo@unical.it

Manlio Gaudioso
gaudioso@dimes.unical.it

Giovanna Miglionico
gmiglionico@dimes.unical.it

Adil M. Bagirov
a.bagirov@federation.edu.au

1 DIMES, Università della Calabria, Rende, Italy

2 Federation University Australia, Ballarat, Australia

Mathematics Subject Classification 90C26 · 65K05

1 Introduction

Consider the problem:

$$\min_{x \in \mathbb{R}^n} f(x) \tag{1}$$

where we assume that $f : \mathbb{R}^n \mapsto \mathbb{R}$ is a DC function, according to the following

Definition 1 A function $f : \mathbb{R}^n \to \mathbb{R}$ is a DC function if there exist convex functions $f_1,\ f_2 : \mathbb{R}^n \to \mathbb{R}$ such that:

$$f(x) = f_1(x) - f_2(x).$$

Here $f_1 - f_2$ is called a DC decomposition of f, while f_1 and f_2 are called DC components of f. A function f is locally DC if for any $x_0 \in \mathbb{R}^n$, there exist $\epsilon > 0$ such that f is DC on the ball $B(x_0, \epsilon)$. It is well known that every locally DC function is DC.

Problem (1) is known as a DC program and some typical problems which can be modeled in such form are the bridge location problem, the design centering problem, the packing problem [1], the production-transportation planning [18], the location planning problem [23], cluster analysis [5,22], clusterwise linear regression analysis [7] and supervised data classification [2]. To date, DC programming has been mostly considered as a part of global optimization. Several algorithms are available to solve this problem globally [19,27].

Nonsmooth DC programming is an important subclass of DC programming problems. Several algorithms have been developed to solve such problems. An algorithm based on quasidifferentials of DC functions and discrete gradients is developed in [3]. A codifferential method is introduced in [6] and a proximal linearized algorithm is proposed in [24]. The paper [21] introduces a proximal bundle method that utilizes nonconvex cutting planes. A gradient splitting method introduced in [14] can be modified for minimizing DC functions. General references on bundle methods are [4,17], while possible variants are [8,12,15].

In this paper, we develop a proximal bundle method for the numerical minimization of nonsmooth DC functions by iteratively building two separate piecewise-affine approximations of the component functions. Combining these two convex piecewise-affine approximations we generate a DC piecewise-affine model, which can also be seen as the pointwise maximum of several concave piecewise-affine functions. Such a nonconvex model is then tackled by means of two auxiliary quadratic programs, that have different local approximation properties.

The rest of the paper is organized as follows. Section 2 presents some definitions and preliminary results on DC functions and DC programs. A cutting plane model for DC functions is introduced in Sect. 3. The description of the proposed method is given in Sect. 4. Termination properties of the algorithm, as well as some results on its convergence are discussed in Sect. 5. Computational results are reported in Sect. 6, and Sect. 7 contains some concluding remarks.

2 Some preliminary results

In the following we focus only on a few theoretical results regarding some relevant stationarity conditions that will be useful for later developments. We refer the reader to [1,16,25,26], and the references therein, for a thorough analysis of DC optimality conditions.

Definition 2 A point x^* is called a local minimizer of the problem (1) if $f_1(x^*) - f_2(x^*)$ is finite and there exists a neighborhood $\mathcal{N}$ of x^* such that

$$f_1(x^*) - f_2(x^*) \leq f_1(x) - f_2(x), \quad \forall x \in \mathcal{N}. \tag{2}$$

In general, nonsmooth DC functions are not regular, and the Clarke subdifferential calculus exists for such functions in the form of inclusions. Such a calculus cannot be used to compute subgradients of the DC function since

$$\partial_{cl} f(x) \subseteq \partial f_1(x) - \partial f_2(x), \tag{3}$$

where the symbol $\partial_{cl} f(\cdot)$ denotes Clarke's subdifferential.

Nonsmooth DC functions are quasidifferentiable. Depending on generalized subgradients used to approximate nonsmooth DC functions, different stationary points can be defined for them. A point x^* is called inf-stationary for problem (1) if

$$\emptyset \neq \partial f_2(x^*) \subseteq \partial f_1(x^*). \tag{4}$$

A point x^* is called Clarke stationary for problem (1) if

$$0 \in \partial_{cl} f(x^*). \tag{5}$$

Finally, a point x^* is called a critical point of the function f if

$$\partial f_2(x^*) \cap \partial f_1(x^*) \neq \emptyset. \tag{6}$$

It is interesting to analyze the relationships between different stationary points. Consider the special case where the function f_1 is nonsmooth convex and the function f_2 is smooth convex.

Proposition 1 *The generalized subdifferential of the DC function f, obtained as the difference between a nonsmooth convex function f_1 and a smooth convex function f_2, can be computed as*

$$\partial_{cl} f(x) = \partial f_1(x) - \nabla f_2(x).$$

Proof Let $D(x) = \partial f_1(x) - \nabla f_2(x)$. The inclusion $\partial_{cl} f(x) \subseteq D(x)$ follows from the calculus for generalized subdifferentials. Therefore we prove only the opposite inclusion. As convex functions f_1 and f_2 are directionally differentiable, then f is also directionally differentiable and

$$f'(x, d) = f_1'(x, d) - f_2'(x, d), \; d \in \mathbb{R}^n.$$

It is known that

$$f_1'(x, d) = \max_{\xi \in \partial f_1(x)} \xi^\top d, \quad f_2'(x, d) = \nabla f_2(x)^\top d.$$

Then

$$f'(x, d) = f_1'(x, d) - f_2'(x, d) = \max_{\xi \in \partial f_1(x)} (\xi - \nabla f_2(x))^\top d \tag{7}$$

for all $d \in \mathbb{R}^n$. This means that $f'(x, d) \geq u^\top d$ for all $d \in \mathbb{R}^n$ and $u \in D(x)$, which in turn, due to the convexity and compactness of the set $D(x)$, implies that $D(x) \subseteq \partial_{cl} f(x)$. □

Now assume that f_1 is smooth and f_2 is any convex nonsmooth function.

Proposition 2 *The generalized subdifferential of the DC function f, obtained as the difference between a smooth convex function f_1 and a nonsmooth convex function f_2, can be computed as*

$$\partial_{cl} f(x) = \nabla f_1(x) - \partial f_2(x).$$

Proof The proof follows from Proposition 1 and from the fact if f is a DC function, then $-f$ is also DC. □

Proposition 3 *Let S_{inf} be a set of inf-stationary points, S_{cl} a set of Clarke stationary points and S_{cr} a set of critical points of the function f. Then*

(1) $S_{inf} \subseteq S_{cl} \subseteq S_{cr}$;
(2) *if the function f_1 is continuously differentiable in $\mathbb{R}^n$ then $S_{cl} = S_{cr}$;*
(3) *if the function f_2 is continuously differentiable in $\mathbb{R}^n$ then $S_{inf} = S_{cl} = S_{cr}$.*

Proof (1) First we show that $S_{inf} \subseteq S_{cl}$. Take any $x \in S_{inf}$. Let $f^0(x, d)$ be a generalized directional derivative of the function f at the point x in a direction $d \in \mathbb{R}^n$ [4]:

$$f^0(x, d) = \limsup_{y \to x, \alpha \downarrow 0} \frac{f(y + \alpha d) - f(y)}{\alpha}.$$

Since both functions f_1 and f_2 are directionally differentiable, taking into account inf-stationarity of the point x, we get

$$f^0(x, d) \geq f'(x, d) = f_1'(x, d) - f_2'(x, d) \geq 0$$

for all $d \in \mathbb{R}^n$. This implies that $0 \in \partial_{cl} f(x)$, hence that $x \in S_{cl}$. The inclusion $S_{cl} \subseteq S_{cr}$ is obvious since according to (3) for any $x \in S_{cl}$ one gets $0 \in \partial f_1(x) - \partial f_2(x)$ and therefore $\partial f_1(x) \cap \partial f_2(x) \neq \emptyset$.

(2) Now assume that the function f_1 is continuously differentiable in $\mathbb{R}^n$, then $\nabla f_1(x) \in \partial f_2(x)$ for any $x \in S_{cr}$. In this case Proposition 2 implies that $0 \in \partial_{cl} f(x)$ that is $x \in S_{cl}$.

(3) Finally, if the function f_2 is continuously differentiable in $\mathbb{R}^n$, then $\nabla f_2(x) \in \partial f_1(x)$ for any $x \in S_{cr}$. In this case, according to Proposition 1, $0 \in \partial_{cl} f(x)$. Therefore $x \in S_{inf}$ and $x \in S_{cl}$.

□

In the next sections we will introduce a method requiring no differentiability assumptions of both f_1 and f_2, for which we will prove finite termination at a point approximately satisfying the criticality condition (6).

3 The model function

Our approach is based on some ideas coming from the classic cutting plane method for minimizing convex functions. In particular we assume that, within an iterative procedure, a certain set of couples $(x_j, g_j^{(2)})$, $j = 1, \ldots, k$, with $x_j \in \mathbb{R}^n$ and $g_j^{(2)} \in \partial f_2(x_j)$ are given. We single out the point x_k from the set of points x_j, $j = 1, \ldots, k$, and calculate $g_k^{(1)} \in \partial f_1(x_k)$. Convexity of f_2 implies that the following inequalities hold for any $x \in \mathbb{R}^n$:

$$f_2(x) \geq f_2(x_j) + g_j^{(2)\top}(x - x_j), \quad j = 1, \ldots, k, \tag{8}$$

which, by introducing the change of variables $d \triangleq x - x_k$, can be rewritten in the form:

$$f_2(x_k + d) \geq f_2(x_k) + g_j^{(2)\top} d - \alpha_j^{(2)}, \quad j = 1, \dots, k, \tag{9}$$

where $\alpha_j^{(2)}$, $j = 1, \dots, k$, is the linearization error associated to the jth first order expansion of f_2 rooted at x_j and is defined as

$$\alpha_j^{(2)} = f_2(x_k) - \left(f_2(x_j) + g_j^{(2)\top}(x_k - x_j) \right). \tag{10}$$

It follows from the definition of the ε-subdifferential $\partial_\varepsilon f_i(\cdot)$ of the convex function $f_i(\cdot)$, with $i = 1, 2$ and $\varepsilon \geq 0$, that $g_j^{(2)} \in \partial_{\alpha_j^{(2)}} f_2(x_k)$. Similarly, for any given $x \in \mathbb{R}^n$ and $g^{(1)} \in \partial f_1(x)$, it holds that $g^{(1)} \in \partial_\alpha f_1(x_k)$, with $\alpha \triangleq f_1(x_k) - f_1(x) - g^{(1)\top}(x_k - x)$.

We observe that the following inequality

$$f_2(x_k + d) \geq f_2(x_k) + \max_{j=1,\dots,k} \left\{ g_j^{(2)\top} d - \alpha_j^{(2)} \right\} \tag{11}$$

follows from (9), and thus we obtain:

$$f(x_k + d) = f_1(x_k + d) - f_2(x_k + d) \leq f_1(x_k + d) - \left(f_2(x_k) + \max_{j=1,\dots,k} \left\{ g_j^{(2)\top} d - \alpha_j^{(2)} \right\} \right). \tag{12}$$

The right-hand side of (12) is still DC, with $f_2(x_k + d)$ being replaced by its lower approximation provided by the convex cutting plane function

$$f_2(x_k) + \ell_2^{(k)}(d),$$

with

$$\ell_2^{(k)}(d) \triangleq \max_{j=1,\dots,k} \left\{ g_j^{(2)\top} d - \alpha_j^{(2)} \right\}. \tag{13}$$

Now we define a concave model $h_k(d)$ for the difference function $f(x_k + d) - f(x_k)$. We first note that from (12) it follows

$$\begin{aligned} f(x_k + d) - f(x_k) &= f_1(x_k + d) - f_2(x_k + d) - f_1(x_k) + f_2(x_k) \\ &\leq f_1(x_k + d) - \left(f_2(x_k) + \ell_2^{(k)}(d) \right) - f_1(x_k) + f_2(x_k) \\ &= f_1(x_k + d) - f_1(x_k) - \ell_2^{(k)}(d). \end{aligned} \tag{14}$$

Next, by substituting $f_1(x_k + d)$ with the affine approximation $f_1(x_k) + g_k^{(1)\top} d$ in (14), we have:

$$h_k(d) \triangleq g_k^{(1)\top} d - \ell_2^{(k)}(d) = \min_{j=1,\dots,k} \left\{ (g_k^{(1)} - g_j^{(2)})^\top d + \alpha_j^{(2)} \right\}. \tag{15}$$

We observe that $h_k(\cdot)$ interpolates the difference function at $d = 0$, since $\alpha_j^{(2)} \geq 0$ for every $j = 1, \dots, k$, with, in particular, $\alpha_k^{(2)} = 0$.

The rationale of our approach is the following. Suppose any direction $\bar{d}$ is calculated such that $h_k(\bar{d}) < 0$. If for a given $m \in (0, 1)$ it is

$$f(x_k + \bar{d}) - f(x_k) < m h_k(\bar{d}),$$

then a decrease has been achieved and a new iterate can be generated by setting $x_{k+1} = x_k + \bar{d}$. Suppose, on the contrary, that

$$f(x_k + \bar{d}) - f(x_k) \geq m h_k(\bar{d}),$$

and consider any $g_+^{(1)} \in \partial f_1(x_k + \bar{d})$. Convexity of f_1 implies

$$f_1(x_k) \geq f_1(x_k + \bar{d}) - g_+^{(1)\top} \bar{d},$$

which, taking into account (11) and (13), leads to

$$\begin{aligned} g_+^{(1)\top} \bar{d} - \ell_2^{(k)}(\bar{d}) &\geq f_1(x_k + \bar{d}) - f_1(x_k) - \ell_2^{(k)}(\bar{d}) \\ &\geq f_1(x_k + \bar{d}) - f_1(x_k) - (f_2(x_k + \bar{d}) - f_2(x_k)) \\ &= f(x_k + \bar{d}) - f(x_k) \\ &\geq m h_k(\bar{d}). \end{aligned} \tag{16}$$

The latter formula is of relevant importance as it will be used in the sequel in order to get better and better models of the objective function whenever descent is not achieved. In fact, suppose that the condition

$$f(x_k + \bar{d}) - f(x_k) \geq m h_k(\bar{d}) \tag{17}$$

is satisfied with $x_k + \bar{d}$ very close to x_k, so that $g_+^{(1)}$ can be considered to approximately belong to $\partial f_1(x_k)$. More precisely, let

$$0 \leq \alpha_+^{(1)} \triangleq f_1(x_k) - f_1(x_k + \bar{d}) + g_+^{(1)\top} \bar{d}, \tag{18}$$

and, for some given $\varepsilon > 0$, assume that

$$\alpha_+^{(1)} \in [0, \varepsilon] \tag{19}$$

and

$$g_+^{(1)} \in \partial_{\alpha_+^{(1)}} f_1(x_k). \tag{20}$$

Then, a better model $H_k(\cdot)$ than $h_k(\cdot)$ at $\bar{d}$, can be obtained by letting

$$H_k(d) \triangleq \max \left\{ g_k^{(1)\top} d - \ell_2^{(k)}(d),\ g_+^{(1)\top} d - \alpha_+^{(1)} - \ell_2^{(k)}(d) \right\}. \tag{21}$$

Indeed, noting that (18) is equivalent to

$$g_+^{(1)\top} \bar{d} - \alpha_+^{(1)} = f_1(x_k + \bar{d}) - f_1(x_k), \tag{22}$$

and taking into account (21), (22), (11), (17), and $h_k(\bar{d}) < 0$, we obtain

$$\begin{aligned} H_k(\bar{d}) &\geq g_+^{(1)\top} \bar{d} - \alpha_+^{(1)} - \ell_2^{(k)}(\bar{d}) \\ &= f_1(x_k + \bar{d}) - f_1(x_k) - \ell_2^{(k)}(\bar{d}) \\ &\geq f_1(x_k + \bar{d}) - f_1(x_k) - (f_2(x_k + \bar{d}) - f_2(x_k)) \\ &= f(x_k + \bar{d}) - f(x_k) \\ &\geq m h_k(\bar{d}) \\ &> h_k(\bar{d}). \end{aligned} \tag{23}$$

Hence, at the point $\bar{d}$, if a large gap between $h_k(\bar{d})$ and $f(x_k + \bar{d}) - f(x_k)$ occurs, then $H_k(\bar{d})$ is significantly higher than $h_k(\bar{d})$.

More in general, given any set $\{g_1^{(1)}, \ldots, g_s^{(1)}\}$ of $\alpha_i^{(1)}$-subgradients of f_1 at x_k, with $\alpha_i^{(1)} \leq \varepsilon$, $i = 1 \ldots, s$, we redefine the nonconvex model $H_k(\cdot)$ as the following maximum of finitely many concave piecewise-affine functions

$$H_k(d) \triangleq \max_{i=1,\ldots,s} \left\{ g_i^{(1)\top} d - \alpha_i^{(1)} - \ell_2^{(k)}(d) \right\}. \tag{24}$$

Next, we examine some differential properties of the model function H_k. Note that, taking into account definitions (13) and (24), function $H_k(d)$ may be rewritten in the form

$$H_k(d) = \max_{i \in I} h_k^{(i)}(d), \tag{25}$$

where $I \triangleq \{1, \ldots, s\}$ and $h_k^{(i)}(d)$, $i \in I$, is a concave piecewise-affine function defined as

$$h_k^{(i)}(d) \triangleq \min_{j \in J} \left\{ (g_i^{(1)} - g_j^{(2)})^\top d - \alpha_i^{(1)} + \alpha_j^{(2)} \right\} \tag{26}$$

with $J \triangleq \{1, \ldots, k\}$. Thus, the function $H_k(d)$ is of the *maxmin* type, it is directionally differentiable, and its directional derivative $H_k'(d; \xi)$ along any direction $\xi \in \mathbb{R}^n$ depends on the directional derivatives of the functions $h_k^{(i)}(d)$, $i \in I$ as follows:

$$H_k'(d; \xi) = \max_{i \in I(d)} h_k^{(i)\prime}(d; \xi),$$

where

$$I(d) \triangleq \{i \in I | H_k(d) = h_k^{(i)}(d)\},$$

and

$$h_k^{(i)\prime}(d; \xi) = \min_{j \in J(d)} \{(g_i^{(1)} - g_j^{(2)})^\top \xi\}$$

with

$$J(d) \triangleq \{j \in J | g_j^{(2)\top} d - \alpha_j^{(2)} = \ell_2^{(k)}(d)\}.$$

For an in-depth analysis of the differential properties of functions of the maxmin type we refer the reader to the historical paper [9].

In the following remark an approximate criticality condition for x_k is highlighted as a result of the minimization of $H_k(d)$.

Remark 1 Assume that $d = 0$ is a minimizer for $H_k(d)$. This implies that $H'(0; \xi) \geq 0$, for every $\xi \in \mathbb{R}^n$, which, in turn, is equivalent to infeasibility of the following inequality:

$$\max_{i \in I(0)} \min_{j \in J(0)} \left\{ (g_i^{(1)} - g_j^{(2)})^\top \xi \right\} < 0. \tag{27}$$

Now observe that $I(0)$ and $J(0)$ contain each at least one index assigned to a subgradient of $f_1(x_k)$ and of $f_2(x_k)$, respectively. Hence, defining the following system $\mathcal{S}_j$ of $|I(0)|$ linear inequalities

$$(g_i^{(1)} - g_j^{(2)})^\top \xi < 0 \quad i \in I(0),$$

for each index $j \in J(0)$, we have that infeasibility of (27) implies infeasibility of all systems $\mathcal{S}_j$, $j \in J(0)$. By Gordan's theorem the latter is equivalent to

$$g_j^{(2)} \in \operatorname{conv}\{g_i^{(1)},\ i \in I(0)\}, \quad j \in J(0),$$

and, consequently,

$$\operatorname{conv}\{g_j^{(2)},\ j \in J(0)\} \subseteq \operatorname{conv}\{g_i^{(1)},\ i \in I(0)\},$$

which can be interpreted as an approximation of the criticality condition (6) at the point x_k.

4 The algorithm

The approximate criticality condition presented in Remark 1, along with the properties of $H_k(\cdot)$ summarized in (23), would suggest to develop a method for solving problem (1) which is based on iterative minimizations of $H_k(d)$. The model function $H_k(d)$ is, as previously mentioned, of the maxmin type and can also be seen as the pointwise maximum of a (finite) number of concave functions. Although there exist several algorithms dealing with minimization of such a family of functions, the idea of direct minimization, at each iteration k, of $H_k(d)$ does not seem viable in terms of computational effort. Thus, borrowing some ideas from the approach proposed in [13] for the minimization of piecewise-concave functions, we introduce an iterative algorithm that does not require the direct minimization of the model function $H_k(d)$. In particular, at each iteration of the algorithm, a displacement vector d_k is calculated, such that $H_k(d_k) < 0$, along which a line search for function f is performed. The calculation of the search direction d_k is made on the basis of the solution of an auxiliary quadratic program that locally approximates $H_k(d)$, hence next referred to as the *local* program. In fact, we note that $\partial h_k^{(i)}(0)$, the subdifferential of $h_k^{(i)}(d)$ at $d = 0$, has the following expression

$$\partial h_k^{(i)}(0) = \text{conv}\{(g_i^{(1)} - g_j^{(2)}) | j \in J(0)\},$$

where $J(0) = \{j \in J \mid \alpha_j^{(2)} = 0\}$. Hence, any subgradient u_i, $u_i \in \partial h_k^{(i)}(0)$, can be calculated by selecting any index $j_i \in J(0)$ and setting

$$u_i = g_i^{(1)} - g_{j_i}^{(2)}.$$

We observe that in case the set $J(0)$ is a singleton, that is $j_i = j$, $i \in I$, the function $H_k(d)$ has the form

$$H_k(d) = \max_{i \in I}\{(g_i^{(1)} - g_j^{(2)})^\top d - \alpha_i^{(1)}\},$$

which is convex piecewise-affine. To make its minimization well-posed we add to $H_k(d)$ the strictly convex term $\frac{1}{2}\|d\|^2$, thus coming out with the minimization of the maximum of finitely many strictly convex functions of d. Such a problem can be equivalently formulated as the following auxiliary *local* problem $QP(I)$

$$\begin{aligned} z = \min_{d \in \mathbb{R}^n, v \in \mathbb{R}} \; & v + \frac{1}{2}\|d\|^2 \\ & v \geq u_i^\top d - \alpha_i^{(1)} \;\; \forall i \in I. \end{aligned} \tag{QP(I)}$$

Denoting by $(\bar{d}, \bar{v})$ the unique optimal solution of ($QP(I)$), a standard duality argument ensures that

$$\bar{d} = -\sum_{i \in I} \bar{\lambda}_i u_i, \tag{28}$$

$$\bar{v} = -\left\| \sum_{i \in I} \bar{\lambda}_i u_i \right\|^2 - \sum_{i \in I} \bar{\lambda}_i \alpha_i^{(1)} \tag{29}$$

where $\bar{\lambda}_i \geq 0$, $i \in I$, with $\sum_{i \in I} \bar{\lambda}_i = 1$, are the optimal variables of the dual of $QP(I)$. Although, in principle, the set $J(0)$ may contain multiple indices (i.e., multiple affine pieces of f_2 may be active at x_k), next we will adopt the strategy of forcing $J(0)$ to be a singleton.

We focus now on the properties of the solution returned by $QP(I)$. We observe that when $|\bar{v}|$ is small, by taking into account equation (29), the condition (6) for the point x_k to be critical is approximately satisfied. On the contrary, when $|\bar{v}|$ is large, since

$$0 \geq \bar{v} = \max_{i \in I}\{u_i^\top \bar{d} - \alpha_i^{(1)}\} \geq \max_{i \in I} h_k^{(i)}(\bar{d}) = H_k(\bar{d}) \tag{30}$$

we observe that a "good agreement" between the predicted reductions $\bar{v}$ and $H_k(\bar{d})$, denotes that the *local* program is still a good approximation of H_k at $\bar{d}$, hence $\bar{d}$ can be considered as a possible descent direction for function f. In case a "poor agreement" between $\bar{v}$ and $H_k(\bar{d})$ is detected, we propose to refine calculation of the search direction, by constructing a more accurate model of function H_k at $\bar{d}$, i.e., *away* from $d = 0$. In particular, we calculate, for every $i \in I$, the index $\bar{j}_i \in J$ such that

$$h_k^{(i)}(\bar{d}) = (g_i^{(1)} - g_{\bar{j}_i}^{(2)})^\top \bar{d} - \alpha_i^{(1)} + \alpha_{\bar{j}_i}^{(2)}$$

and we introduce the following auxiliary *away* problem $QP(I, \bar{J})$

$$\begin{aligned} \min_{d \in \mathbb{R}^n, v \in \mathbb{R}} \quad & v + \frac{1}{2}\|d\|^2 \\ & v \geq \bar{u}_i^\top d - \alpha_i^{(1)} + \alpha_{\bar{j}_i}^{(2)} \quad \forall i \in I \end{aligned} \tag{$QP(I, \bar{J})$}$$

again a (strictly) convex quadratic program, where $\bar{u}_i \triangleq g_i^{(1)} - g_{\bar{j}_i}^{(2)}$ and $\bar{J}$ is the set of indices $\bar{j}_i$, $i \in I$. It is worth noting that while problem $QP(I, \bar{J})$ embeds the function

$$\max_{i \in I}\{\bar{u}_i^\top d - \alpha_i^{(1)} + \alpha_{\bar{j}_i}^{(2)}\} \tag{31}$$

which interpolates the maxmin model H_k at $\bar{d}$, the problem $QP(I)$ uses the function

$$\max_{i \in I}\{u_i^\top d - \alpha_i^{(1)}\}, \tag{32}$$

which, in turn, interpolates H_k at $d = 0$. Function (31) resembles H_k around $\bar{d}$, but it can be a poor representation for small values in norm of d, as it does not even interpolate H_k (and f) at $d = 0$, provided $\alpha_{\bar{j}_i}^{(2)} > 0$ for at least one index $i \in I$. As a consequence, denoting by $(\hat{d}, \hat{v})$ the (unique) optimal solution of $QP(I, \bar{J})$, we will adopt $\hat{d}$ as an alternative search direction in case the predicted reduction $H_k(\hat{d})$ is strictly lower than $H_k(\bar{d})$.

We are now ready to describe our DC Piecewise-Concave algorithm (DCPCA). A compact description of DCPCA is first given in Scheme 1, where its essential steps are highlighted. The comments therein refer to the actual steps of DCPCA, which is fully detailed in Algorithm 1.

DCPCA takes as an input any starting point $x_0 \in \mathbb{R}^n$, and returns an approximate critical point x^*. We assume that the set

$$S_0 \triangleq \{x \in \mathbb{R}^n | f(x) \leq f(x_0)\}$$

is bounded. The following parameters are to be set: the optimality parameter $\theta > 0$, the subgradient threshold $\eta > 0$, the linearization-error threshold $\varepsilon > 0$, the approximate line-search parameter $m \in (0, 1)$, the step-size reduction parameter $\sigma \in (0, 1)$, the agreement rate $\rho \in (0, 1)$.

Some explanatory comments about the structure of the algorithm are in order. Given the current iterate x_k, DCPCA first determines the *main search-direction* $\bar{d}$ and the *main predicted-reduction* $\bar{v}$ by solving $QP(I)$. Next, the condition $\bar{v} < \rho H_k(\bar{d})$ checked at step 7

Scheme 1 DCPCA at a glance

A1: Initialization;	▹ Steps 1–3
A2: Find the main search-direction and the main predicted-reduction at the current iterate by solving $QP(I)$, and if an approximate criticality condition holds then terminate;	▹ Steps 4–6
A3: Check, if appropriate, for an alternative more promising search-direction at the current iterate by solving $QP(I, \bar{J})$, then set the strong predicted-reduction;	▹ Steps 7–17
A4: As long as the displacement along the selected search-direction is not too small make a backtracking line-search looking for a sufficient descent, then update the current iterate and go to A4 (serious step achieved);	▹ Steps 18–28
A5: If appropriate restore the main search-direction, update the strong predicted-reduction and go to A5;	▹ Steps 29–32
A6: If appropriate restore the main predicted-reduction and go to A5;	▹ Steps 33–34
A7: Accumulate first order information in the f_1 bundle and go to A2 (null step execution).	▹ Steps 35–39

is meant to detect if a "good agreement" holds between $\bar{v}$ and the *strong predicted reduction* $H_k(\bar{d})$ provided by the model at $\bar{d}$. Then, in case of "poor agreement" between $\bar{v}$ and $H_k(\bar{d})$, DCPCA generates an *alternative* search-direction $\hat{d}$ by solving $QP(I, \bar{J})$, and selects the search direction d_k at steps 11–15 as the most promising one between $\bar{d}$ and $\hat{d}$. Independent of the selected search-direction, an Armijo-type line-search is executed adopting the *strong predicted-reduction* yielded by the nonconvex model at d_k, i.e., $H_k(d_k)$. As soon as the line-search is successful (a sufficient decrease is achieved at step 18), a serious step is made, namely, the current estimate of the minimum is updated and a new iteration takes place. It is worth noting that the failure of the line-search, at a point that is very close to the current iterate, does not immediately imply a null-step execution (i.e., information enrichment of $\mathcal{B}_1$ about the subdifferential of f_1 at x_k) at steps 35–39. In fact, the role of the preliminary steps 29–34 is to prevent the algorithm from certifying the line-search failure, until the main search-direction has been checked for sufficient descent with respect to the main predicted-reduction. We note that, whenever $g_+^{(1)}$ is inserted into $\mathcal{B}_1$ at step 36, then $t\|d_k\| \leq \eta$ implies that $g_+^{(1)} \in \partial_{\varepsilon_1} f_1(x_k)$ for $\varepsilon_1 \leq 2\eta L_1$, where L_1 is the Lipschitz constant of f_1 on the set

$$S_0(\eta) \triangleq \left\{x \in \mathbb{R}^n : \operatorname{dist}(x, S_0) \leq \eta\right\}.$$

Finally, the following remark aims at clarifying the role of the stopping condition in terms of an approximate stationarity condition for problem (1).

Remark 2 The stopping condition $\bar{v} \geq -\theta$, checked at step 5 of DCPCA, is an approximate θ-criticality condition for x^*. Indeed, taking into account (29), the stopping condition ensures that

$$\left\| \sum_{i\in I} \lambda_i^* g_i^{(1)} - \sum_{i\in I} \lambda_i^* g_{j_i}^{(2)} \right\| \leq \sqrt{\theta} \quad \text{and} \quad \left\| \sum_{i\in I} \lambda_i^* \alpha_i^{(1)} \right\| \leq \sqrt{\theta},$$

which in turn implies that there exist $g_*^{(1)} \in \partial_\theta f_1(x^*)$ and $g_*^{(2)} \in \partial f_2(x^*)$ such that

$$\|g_*^{(1)} - g_*^{(2)}\|^2 \leq \theta,$$

namely, that

$$\operatorname{dist}\left(\partial_\theta f_1(x^*), \partial f_2(x^*)\right) \leq \theta,$$

an approximate θ-criticality condition for x^*, see (6).

Algorithm 1 DCPCA

Input: a starting point $x_0 \in \mathbb{R}^n$, parameters $\theta > 0$, $\eta > 0$, $\varepsilon > 0$, $m \in (0, 1)$, $\sigma \in (0, 1)$, $\rho \in (0, 1)$
Output: an approximate critical point $x^* \in \mathbb{R}^n$

1: Calculate $g_0^{(1)} \in \partial f_1(x_0)$, $g_0^{(2)} \in \partial f_2(x_0)$ and set $\alpha_0^{(1)} = \alpha_0^{(2)} = 0$ ▷ Initialization
2: Set $\mathcal{B}_1 = \{(g_0^{(1)}, \alpha_0^{(1)})\}$ and $\mathcal{B}_2 = \{(g_0^{(2)}, \alpha_0^{(2)})\}$ ▷
3: Set $I = \{1\}$, $J = \{1\}$, and set $k = 0$ ▷
4: Solve $QP(I)$ and obtain $(\bar{d}, \bar{v})$ ▷ Find the main search-direction and the main predicted-reduction
5: **if** $|\bar{v}| \le \theta$ **then** ▷ Stopping test
6: set $x^* = x_k$ and **exit** ▷ Return x^*
7: **else if** $\bar{v} < \rho H_k(\bar{d})$ **then** ▷ Good agreement between $\bar{v}$ and $H_k(\bar{d})$
8: set $d_k = \bar{d}$ ▷ Select $\bar{d}$ as the descent search-direction
9: **else** ▷ Poor agreement between $\bar{v}$ and $H_k(\bar{d})$
10: solve $QP(I, \bar{J})$ and obtain $(\hat{d}, \hat{v})$ ▷ Find an alternative search-direction at x_k
11: **if** $H_k(\hat{d}) < H_k(\bar{d})$ **then** ▷ Select the most promising direction at x_k
12: set $d_k = \hat{d}$ ▷
13: **else** ▷
14: set $d_k = \bar{d}$ ▷
15: **end if** ▷
16: **end if**
17: Set $v_k = H_k(d_k)$ and $t = 1$ ▷ Set the strong predicted-reduction and start the line-search
18: **if** $f(x_k + td_k) - f(x_k) \le mtv_k$ **then** ▷ Descent test
19: set $x_{k+1} = x_k + td_k$ ▷ Make a serious step
20: calculate $g_{k+1}^{(1)} \in \partial f_1(x_{k+1})$ and $g_{k+1}^{(2)} \in \partial f_2(x_{k+1})$ ▷
21: update $\alpha_i^{(1)} := \alpha_i^{(1)} + f_1(x_{k+1}) - f_1(x_k) - t g_i^{(1)\top} d_k$ for all $i \in I$ ▷
22: update $\alpha_j^{(2)} := \alpha_j^{(2)} + f_2(x_{k+1}) - f_2(x_k) - t g_j^{(2)\top} d_k$ for all $j \in J$ ▷
23: set $\mathcal{B}_1 = \mathcal{B}_1 \setminus \{(g_i^{(1)}, \alpha_i^{(1)}) : \alpha_i^{(1)} > \varepsilon,\ i \in I\} \cup \{(g_{k+1}^{(1)}, 0)\}$ ▷
24: set $\mathcal{B}_2 = \mathcal{B}_2 \cup \{(g_{k+1}^{(2)}, 0)\}$ ▷
25: update appropriately I and J, set $k = k + 1$ and **go to 4** ▷
26: **else if** $t\|d_k\| > \eta$ **then** ▷ Closeness test
27: set $t = \sigma t$ and **go to 18** ▷ Reduce the step-size and iterate the line-search
28: **end if**
29: **if** $d_k = \hat{d}$ **then**
30: set $d_k = \bar{d}$ ▷ Restore the main search-direction
31: set $v_k = H_k(d_k)$ ▷ Update the strong predicted-reduction
32: set $t = 1$ and **go to 18** ▷ Restart the line-search
33: **else if** $v_k < \bar{v}$ **then**
34: set $v_k = \bar{v}$ and **go to 18** ▷ Restore the main predicted-reduction and proceed with the line-search
35: **else** ▷ The line-search has failed with $d_k = \bar{d}$ and $v_k = \bar{v}$
36: calculate $g_+^{(1)} \in \partial f_1(x_k + td_k)$ ▷ Make a null step
37: calculate $\alpha_+^{(1)} = f_1(x_k) - f_1(x_k + td_k) + t g_+^{(1)\top} d_k$ ▷
38: set $\mathcal{B}_1 = \mathcal{B}_1 \cup \{(g_+^{(1)}, \alpha_+^{(1)})\}$, update appropriately I, and **go to 4** ▷
39: **end if**

5 Termination properties

Before introducing the main theorem about convergence of Algorithm 1, in the following lemma we give a bound on the size of the search-directions.

Lemma 1 *Let L_1 and L_2 be the Lipschitz constants of f_1 and f_2, respectively, on the set $S_0(\eta)$. Then the following bound holds*

$$\|d_k\| \le 2L \tag{33}$$

where $L \triangleq \max\{L_1, L_2\}$.

Proof Assume first that $d_k = \bar{d}$, then the result follows by taking into account (28). Next, assuming that $d_k = \hat{d}$, the result follows by observing that $H_k(\hat{d}) < H_k(\bar{d}) < 0$ (see steps 11 and 12, and, consequently, that $\hat{d}$ belongs to the set $\hat{S}$ where the objective function of problem $QP(I, \bar{J})$ is negative. Then, the property follows by noting that $\hat{S} \subset S'$, with

$$S' = \{d|\bar{u}'^{\top} d + \frac{1}{2}\|d\|^2 \leq 0\}$$

where $\bar{u}' = g_i^{(1)} - g_{\bar{j}_i}^{(2)}$ for the index i with corresponding $\alpha_i^{(1)} = 0$, independently of the choice of the index $\bar{j}_i$. □

Next we present the convergence results. We start by proving finiteness of the null-step sequence.

Lemma 2 *At any given iterate $x_k \in S_0$, the sequence of null-step execution made by Algorithm 1 terminates after finitely many iterations either fulfilling the sufficient decrease condition at step 18 or satisfying the stopping condition at step 5.*

Proof We need to show that the algorithm cannot loop infinitely many times between 38 and step 4. Observe, indeed, that a null step is made, with consequent enrichment of bundle $\mathcal{B}_1$ and return to step 4, only when the main search-direction $d_k = \bar{d}$ reveals itself unsuccessful with respect to the main predicted-reduction $v_k = \bar{v}$, see step 35. Now suppose, for a contradiction, that an infinite sequence of null steps occurs, and, to simplify the notation, let $\{\bar{v}_h\}$, $\{\bar{d}_h\}$, $\{\bar{z}_h\}$ and $\{t_h\}$ be the corresponding sequences of values of $\bar{v}$, $\bar{d}$, $\bar{z}$ and of the step-size t, respectively, where

$$\bar{z}_h = \bar{v}_h + \frac{1}{2}\|\bar{d}_h\|^2.$$

Indexing by $h+1$ the bundle element $\{(g_+^{(1)}, \alpha_+^{(1)})\}$ inserted into $\mathcal{B}_1$ at the hth null-step iteration, we observe that from condition

$$f(x_k + t_h\bar{d}_h) - f(x_k) > mt_h\bar{v}_h, \tag{34}$$

taking into account (22) and applying the subgradient inequality (9) to f_2, it follows that

$$\left(g_{h+1}^{(1)} - g_j^{(2)}\right)^{\top} \bar{d}_h > m\bar{v}_h + \frac{1}{t_h}\alpha_{h+1}^{(1)}, \quad j \in J(0), \tag{35}$$

which in turn implies that

$$\left(g_{h+1}^{(1)} - g_j^{(2)}\right)^{\top} \bar{d}_h - \alpha_{h+1}^{(1)} > m\bar{v}_h + \left(\frac{1}{t_h} - 1\right)\alpha_{h+1}^{(1)} \geq m\bar{v}_h, \quad j \in J(0) \tag{36}$$

since $t_h \leq 1$. From (36) it follows that, once the element $\{(g_{h+1}^{(1)}, \alpha_{h+1}^{(1)})\}$ is inserted into $\mathcal{B}_1$, a *cut* is introduced in the feasible region of problem $QP(I)$ and, consequently, the sequence $\{\bar{z}_h\}$ is monotonically increasing and convergent, being negative, to a limit $\tilde{z} \leq 0$. Moreover, see (33), the sequence $\{\bar{d}_h\}$ admits a convergent subsequence, corresponding to a certain set of indices $\mathcal{H}$, and let $\tilde{d}$ be its limit point. Hence, the corresponding subsequence $\{\bar{v}_h\}_{h\in\mathcal{H}}$ is convergent as well to a limit $\tilde{v} \leq 0$. Consider now two successive indices $p, q \in \mathcal{H}$. From (36) and the definition of problem $QP(I)$ we have, for $j \in J(0)$, that

$$\left(g_{p+1}^{(1)} - g_j^{(2)}\right)^{\top} \bar{d}_p - \alpha_{p+1}^{(1)} > m\bar{v}_p,$$

and

$$\left(g_{p+1}^{(1)} - g_j^{(2)}\right)^\top \bar{d}_q - \alpha_{p+1}^{(1)} \le \bar{v}_q.$$

They in turn imply that

$$\bar{v}_q - m\bar{v}_p > \left(g_{p+1}^{(1)} - g_j^{(2)}\right)^\top (\bar{d}_q - \bar{d}_p)$$

which, passing to the limit, would imply

$$(1-m)\tilde{v} \ge 0,$$

a contradiction, since in an infinite sequence of null steps the stopping condition can never be satisfied. □

Now we can prove finite termination of the algorithm

Theorem 1 *For any starting point $x_0 \in \mathbb{R}^n$, Algorithm 1 terminates after finitely many iterations at a point x^* satisfying the stopping criterion at step 5.*

Proof Suppose for a contradiction that the stopping criterion at step 5 is never fulfilled. Lemma 2 ensures that, in such a case, an infinite sequence of serious steps can only occur, with the algorithm looping infinitely many times between step 25 and step 4.

Observe now that whenever the main search-direction $d_k = \bar{d}$ is adopted (see steps 8, 14, or 30), it follows from (30) that $v_k = H_k(\bar{d}) \le \bar{v}$ (see steps 17 and 31). On the other hand, whenever the alternative search-direction $d_k = \hat{d}$ is adopted (see steps 11 and 12), it holds $v_k = H_k(\hat{d}) < H_k(\bar{d}) \le \bar{v}$ (see step 17). Summing up, every time a serious step is entered at step 19, taking into account the failed stopping test at step 5, it is

$$v_k \le \bar{v} \le -\theta < 0. \tag{37}$$

Furthermore, it also holds that either $t = 1$ or $t\|d_k\| > \sigma\eta$. As a consequence, the fulfillment of the sufficient decrease condition at step 18 implies that either

$$f(x_k + td_k) - f(x_k) < -m\theta, \tag{38}$$

in case $t = 1$, or

$$f(x_k + td_k) - f(x_k) < -\frac{m\sigma\eta\theta}{2L}, \tag{39}$$

in case $t\|d_k\| > \sigma\eta$, since Lemma 1 implies that $-t < -\frac{\sigma\eta}{\|d_k\|} \le -\frac{\sigma\eta}{2L}$. Hence, inequalities (38) and (39) imply that the decrease in the objective function value is bounded away from zero every time a serious step is made. This contradicts the fact that an infinite number of descent steps occurs, under the assumption that S_0 is bounded. □

6 Numerical results

In order to evaluate the practical behavior of the proposed approach, in the following we report on the computational performance of DCPCA applied to the solution of a set of 46 academic test problems, presented in [21], that belong to 10 different instance classes, with size ranging from $n = 2$ to $n = 50000$. The 10 instance classes all refer to nonsmooth DC problems, at least one DC component being always nonsmooth. Moreover, a starting point is given as a feature of each instance. A summary of some relevant information regarding the test set is provided in Table 1, where for each instance we report

Table 1 Summary of test instances

ID	n	f^*	$f(x_0)$	ID	n	f^*	$f(x_0)$
1.01	2	2	20.0000	5.10	350	0	18.7973
2.01	2	0	22.2000	5.11	400	0	18.8226
3.01	4	0	402.2000	5.12	500	0	18.8581
4.01	2	0	1.0000	5.13	1000	0	18.9290
4.02	5	0	10.0000	5.14	1500	0	18.9527
4.03	10	0	45.0000	5.15	3000	0	18.9763
4.04	50	0	1225.0000	5.16	10,000	0	18.9929
4.05	100	0	4950.0000	5.17	15,000	0	18.9953
4.06	150	0	11,175.0000	5.18	20,000	0	18.9965
4.07	200	0	19,900.0000	5.19	50,000	0	18.9986
4.08	250	0	31,125.0000	6.01	2	−2.5	0.1000
4.09	350	0	61,075.0000	7.01	2	0.5	103.0000
4.10	500	0	124,750.0000	8.01	3	3.5	5.0000
4.11	750	0	280,875.0000	9.01	4	1.8333	43.0000
5.01	2	0	4.7500	10.01	2	−0.5	−0.0500
5.02	5	0	10.4592	10.02	4	−2.5	0.0000
5.03	10	0	13.6738	10.03	5	−3.5[a]	0.1500
5.04	50	0	17.6128	10.04	10	−8.5	2.9500
5.05	100	0	18.2916	10.05	20	−18.5	26.8000
5.06	150	0	18.5270	10.06	50	−48.5	424.3500
5.07	200	0	18.6452	10.07	100	−98.5	3373.6000
5.08	250	0	18.7162	10.08	150	−148.5	11,347.8500
5.09	300	0	18.7635	10.09	200	−198.5	26,847.1000

[a] An anonymous reviewer pointed out that this better estimate ($f^* = -3.5$) of the optimum exists than the one available in [21]

- ID, an instance identifier,
- n, the number of variables,
- f^*, the known best value of the objective function,
- $f(x_0)$, the objective function value at the starting point.

Algorithm DCPCA has been implemented in Java, and the tests have been executed on a 3.50 GHz Intel Core i7 computer. The QP solver of IBM ILOG CPLEX 12.6 [20] has been used to solve the quadratic subprograms. The following set of parameters has been adopted: $\theta = 10^{-6}$, $\eta = 0.7$, $m = 10^{-4}$, $\rho = 0.95$, and $\varepsilon = 0.95$. Furthermore, the setting of the step-size reduction parameter σ depends on the instance size as follows: $\sigma = 0.05$ if $n < 10$, and $\sigma = 0.6$ if $\sigma \geq 10$.

We have compared the computational behavior of DCPCA against NCVX, a state-of-the-art general-purpose solver for nonconvex nonsmooth optimization introduced in [12], and against PBDC, a proximal bundle method for nonsmooth DC programs introduced in [21] where its practical efficiency has been thoroughly verified. We remark that PBDC terminates as soon as the distance between the convex hulls of subgradients of f_1 and f_2, in a neighborhood of the current iterate, is below a given threshold, while the stopping criterion of NCVX is based on the distance from zero of the Goldstein ϵ-subdifferential of f at the current point, as it does not exploit at all the DC structure of f. Since the stopping criteria and the related tolerance parameters are significantly different between the adopted algorithms, the rationale of the tolerance tuning of each algorithm has been to guarantee similar level of precision for as many instances as possible across the three algorithms. We have adopted a double-precision Fortran 95 implementation of PBDC, and a Java implementation of NCVX, running the experiments on the same machine.

A summary of the computational behavior of the three solvers is presented in Table 2 where, for each instance and each solver, we report an appropriate subset of the following results:

– $f(x^*)$, the objective function value returned by the algorithm upon termination,
– N_f, the number of the objective function evaluations,
– cpu, the CPU execution time measured in seconds,
– N_{g_1}, the number of subgradient evaluations for f_1,
– N_{g_2}, the number of subgradient evaluations for f_2,
– N_g, the number of subgradient evaluations for f.

The results show the good performance of DCPCA, both in terms of effectiveness, as it attains the known best value of the objective function for 45 out of 46 instances, and in terms of efficiency, since very good precision is attained for almost all cases, and the number of function evaluations is always kept to reasonably low levels, along with the corresponding computational time.

Focusing on the comparison against PBDC, we report only the number of function evaluations and the number of subgradient evaluations for f_1 and f_2, being the cpu execution times not comparable due to the different implementation languages adopted. We observe a slight advantage of DCPCA over PBDC in terms of effectiveness, as PBDC attains the known best value of the objective function for 43 out of 46 instances, while DCPCA and PBDC have comparable performance in terms of the number of function evaluations, with the former seemingly more suitable for large-scale instances. A sharper difference in favor of DCPCA can be observed considering the total number of subgradient evaluations, since the simpler structure of DCPCA allows to achieve remarkable performance even for large-scale instances by using fewer subgradients and, hence, by solving quadratic subproblems of smaller size.

Focusing, finally, on the comparison against NCVX, we observe that although DCPCA seems to have lower efficiency than NCVX in terms of the number of function evaluations, still DCPCA has a clear advantage in terms of effectiveness, as NCVX can find the known best value of the objective function for only 38 out of 46 instances. Making a fair comparison in terms of the number of subgradient evaluations is not straightforward, nevertheless we observe that for large instances DCPCA performs better than NCVX even comparing N_g with the sum of $N_{g_1} + N_{g_2}$. A similar trend can also be seen analyzing the cpu execution times. In summary, the simpler structure of the bundle management phase, coupled with the line-search approach, makes DCPCA more suited than NCVX to deal even with large-scale instances, although at the expenses of a possibly larger number of function evaluations for small-scale instances.

Table 2 Computational results with DCPCA, PBDC, and NCVX

	DCPCA					PBDC				NCVX			
ID	$f(x^*)$	N_f	cpu	N_{g_1}	N_{g_2}	$f(x^*)$	N_f	N_{g_1}	N_{g_2}	$f(x^*)$	N_f	cpu	N_g
1.01	2.0000005657	35	0.140	19	16	2.0000000198	22	17	16	2.0000034317	20	0.125	20
2.01	2.798508E−09	16	0.015	10	9	1.109770E−12	21	15	15	1.0000000004*	18	0.016	18
3.01	8.401510E−08	37	0.015	16	12	2.268880E−12	25	13	11	7.506135E−10	26	0.016	26
4.01	3.957674E−09	21	0.140	11	11	8.437690E−15	6	3	3	9.192281E−09	5	0.000	5
4.02	1.048708E−08	81	0.031	41	41	0.0000000000	13	6	5	8.546053E−12	17	0.016	17
4.03	1.544977E−08	20	0.031	11	11	5.684340E−14	16	11	9	5.968559E−13	31	0.031	31
4.04	5.798438E−08	100	0.046	51	51	9.094940E−13	52	51	12	1.136868E−12	105	0.188	105
4.05	2.819434E−11	201	0.484	102	102	9.094940E−13	102	102	30	2.643537E−08	211	1.250	211
4.06	7.457857E−11	301	1.609	152	152	1.670920E−05	198	198	70	1.628177E−08	312	4.094	312
4.07	3.835048E−07	400	4.203	201	201	3.637970E−12	339	339	132	1.184599E−07	412	11.078	412
4.08	1.309672E−10	501	9.156	252	252	7.275950E−12	502	502	187	6.803020E−10	513	25.984	513
4.09	1.520675E−09	701	30.031	352	352	7.275950E−12	705	705	331	3.067609E−06	1081	126.125	1081
4.10	2.371962E−09	1001	118.750	502	502	5.384200E−10	1357	1357	677	3.354216E−08	1012	312.047	1012
4.11	1.478475E−08	1501	552.265	752	752	3.550660E−09	2258	2258	1489	1.053559E−07	1511	1663.844	1511
5.01	1.695888E−11	6	0.078	4	3	0.0000000000	10	4	4	8.353496E−11	8	0.094	8
5.02	1.637899E−08	25	0.015	12	10	6.128430E−14	15	7	6	8.665868E−11	9	0.000	9
5.03	6.652157E−08	57	0.093	14	11	3.541610E−13	21	14	11	8.254246E−10	13	0.016	13
5.04	5.722661E−07	59	0.062	20	17	8.101060E−13	103	94	69	1.662895E−08	15	0.031	15
5.05	2.074663E−05	129	0.203	56	48	8.565920E−13	47	28	23	3.405707E−08	19	0.031	19
5.06	9.476286E−07	165	0.140	38	22	2.580520E−12	76	49	40	4.028033E−08	14	0.031	14
5.07	7.571011E−06	78	0.062	32	26	4.633650E−12	106	55	46	2.917597E−08	14	0.063	14
5.08	5.856034E−06	74	0.031	26	21	3.050580E−11	55	49	40	2.327263E−08	14	0.078	14
5.09	4.331377E−06	73	0.171	24	19	1.849930E−11	36	27	24	1.945210E−08	14	0.078	14

Table 2 continued

ID	DCPCA $f(x^*)$	N_f	cpu	N_{g_1}	N_{g_2}	PBDC $f(x^*)$	N_f	N_{g_1}	N_{g_2}	NCVX $f(x^*)$	N_f	cpu	N_g
5.10	3.678621E−06	71	0.140	22	17	1.273910E−10	43	34	27	1.661732E−08	14	0.094	14
5.11	3.425311E−06	70	0.140	21	16	7.887420E−11	44	36	24	1.659315E−08	14	0.094	14
5.12	1.782611E−06	70	0.140	21	16	2.104870E−10	29	22	18	3.365508E−08	16	0.172	16
5.13	1.331532E−06	71	0.281	19	14	4.747840E−10	32	26	22	1.378079E−10	40	1.547	40
5.14	9.211257E−07	71	0.468	19	14	6.268130E−10	25	20	16	8.609769E−08	46	2.438	46
5.15	5.083641E−07	76	1.281	20	16	1.941500E−09	29	24	16	1.150208E−06	13	0.563	13
5.16	7.439088E−07	78	5.437	19	15	5.934490E−09	22	19	11	5.277515E−09	14	2.281	14
5.17	5.948349E−07	80	8.578	19	15	2.656980E−08	178	54	48	5.844288E−09	15	3.891	15
5.18	4.899043E−07	81	12.484	19	15	0.0014116200	152	43	39	4.101435E−09	15	5.234	15
5.19	3.597117E−07	84	33.203	19	15	0.0005714210	99	26	26	3.208146E−08	14	12.797	14
6.01	−2.4999983682	33	0.125	33	33	2.4999997300	27	19	15	−2.4999134396	14	0.094	14
7.01	0.5000000055	57	0.046	46	29	0.5000000035	53	44	26	1.0000000001⋆	16	0.016	16
8.01	3.5000002914	33	0.015	26	20	3.5000001579	84	64	40	3.5004916999	19	0.016	19
9.01	1.8333335218	39	0.015	25	21	1.8333334317	90	78	41	1.8336086581	27	0.016	27
10.01	−0.5000000000	17	0.000	10	9	−0.4999999820	21	13	7	−0.4988362511⋆	8	0.016	8
10.02	−2.5000000000	6	0.015	6	5	−2.4999997880	23	16	9	−2.4999660639	133	0.078	133
10.03	−2.5000000000⋆	32	0.031	18	17	−2.4999998760⋆	22	13	10	−0.4998867020⋆	9	0.000	9
10.04	−8.5000000000	11	0.125	9	8	−8.4999996180	59	45	23	−8.4963254799⋆	72	0.063	72
10.05	−18.5000000000	16	0.000	14	13	−18.4999880100	60	44	20	−18.4997478111	50	0.047	50
10.06	−48.5000000000	33	0.031	30	29	−48.4998942300	140	121	54	−45.6587653013⋆	373	0.781	373
10.07	−98.5000000000	58	0.093	54	53	−98.4999766300	352	343	92	−98.4990145435	803	4.188	803
10.08	−148.5000000000	82	0.125	79	78	−126.4893240000⋆	491	490	117	−146.4995478794⋆	1105	11.922	1105
10.09	−198.5000000000	108	0.140	103	102	−90.4958107000⋆	476	475	131	−186.4996529410⋆	1122	21.563	1122

⋆The known best value of the objective function has not been found

7 Conclusions

We have introduced a proximal bundle method for the unconstrained minimization of a nonsmooth DC function. The main novelty of the approach is represented by the model adopted to approximate the function reduction, which is of the piecewise concave type, pieces being piecewise-affine, hence it is nonconvex. In particular, the model is built by using two separate piecewise-affine approximations of the DC components, which combine themselves into a DC piecewise-affine model. Rather than directly tackling the minimization of the nonconvex model, we resort to an auxiliary convex quadratic program, whose solution returns a descent search-direction, or certifies stationarity. Furthermore, in order to cope with possible poor approximation properties of the former model, at points that are far from the current estimate of the minimizer, we have introduced another auxiliary convex quadratic program, with improved approximation properties far from the current iterate, that can possibly provide an alternative more promising search-direction. Such two search-directions are then explored via a line-search approach aiming at finding a sufficient descent or a model improvement. We have proved finite termination of the method at points that satisfy an approximate criticality condition, and have tested the approach on a set of academic instances, with sizes ranging from extra-small to extra-large, obtaining rather encouraging performance in terms of effectiveness and efficiency. Future work would be focused on the possibility of adopting appropriate aggregation schemes to keep the bundle size limited.

Acknowledgements The authors are grateful to two anonymous reviewers for many helpful comments and suggestions.

References

1. An, L.T.H., Tao, P.D.: The DC (difference of convex functions) programming and DCA revisited with DC models of real world nonconvex optimization problems. J. Glob. Optim. **133**, 23–46 (2005)
2. Astorino, A., Fuduli, A., Gaudioso, M.: DC models for spherical separation. J. Glob. Optim. **48**(4), 657–669 (2010)
3. Bagirov, A.M.: A method for minimization of quasidifferentiable functions. Optim. Methods Softw. **17**(1), 31–60 (2002)
4. Bagirov, A.M., Karmitsa, N., Mäkelä, M.M.: Introduction to Nonsmooth Optimization: Theory, Practice and Software. Springer, Berlin (2014)
5. Bagirov, A.M., Taheri, S., Ugon, J.: Nonsmooth DC programming approach to the minimum sum-of-squares clustering problems. Pattern Recognit. **53**, 12–24 (2016)
6. Bagirov, A.M., Ugon, J.: Codifferential method for minimizing nonsmooth DC functions. J. Glob. Optim. **50**(1), 3–22 (2011)
7. Bagirov, A.M., Ugon, J.: Nonsmooth DC programming approach to clusterwise linear regression: optimality conditions and algorithms. Optim. Methods Softw. (in press). doi:10.1080/10556788.2017.1371717
8. Demyanov, A.V., Fuduli, A., Miglionico, G.: A bundle modification strategy for convex minimization. Eur. J. Oper. Res. **180**, 38–47 (2007)
9. Demyanov, V.F., Malozemov, V.N.: On the theory of non-linear minimax problems. Russ. Math. Surv. **26**(3) (1971). http://iopscience.iop.org/0036-0279/26/3/R02
10. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**, 201–213 (2002)
11. Fuduli, A., Gaudioso, M., Giallombardo, G.: Minimizing nonconvex nonsmooth functions via cutting planes and proximity control. SIAM J. Optim. **14**(3), 743–756 (2004)
12. Fuduli, A., Gaudioso, M., Giallombardo, G., Miglionico, G.: A partially inexact bundle method for convex semi-infinite minmax problems. Commun. Nonlinear Sci. Numer. Simul. **21**(1–3), 172–180 (2014)
13. Gaudioso, M., Giallombardo, G., Miglionico, G.: Minimizing piecewise-concave functions over polytopes. Math. Op. Res. (in press). doi:10.1287/moor.2017.0873

14. Gaudioso, M., Gorgone, E.: Gradient set splitting in nonconvex nonsmooth numerical optimization. Optim. Methods Softw. **25**(1), 59–74 (2010)
15. Gaudioso, M., Monaco, M.F.: Variants to the cutting plane approach for convex nondifferentiable optimization. Optimization **25**, 65–75 (1992)
16. Hiriart-Urruty, J.B.: From Convex Optimization to Nonconvex Optimization: Necessary and Sufficient Conditions for Global Optimality, in Nonsmooth Optimization and Related Topics, pp. 219–240. Plenum, New York/London (1989)
17. Hiriart-Urruty, J.B., Lemaréchal, C.: Convex Analysis and Minimization Algorithms I-II. Springer, Berlin (1993)
18. Holmberg, K., Tuy, H.: A production-transportation problem with stochastic demand and concave production costs. Math. Program. **85**, 157–179 (1999)
19. Horst, R., Thoai, N.V.: DC programming: overview. J. Optim. Theory Appl. **103**, 1–43 (1999)
20. Ibm Ilog Cplex Optimizer: http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/
21. Joki, K., Bagirov, A.M., Karmitsa, N., Mäkelä, M.M.: A proximal bundle method for nonsmooth DC optimization utilizing nonconvex cutting planes. J. Global Optim. **68**(3), 501–535 (2017). doi:10.1007/s10898-016-0488-3
22. Ordin, B., Bagirov, A.M.: A heuristic algorithm for solving the minimum sum-of-squares clustering problems. J. Glob. Optim. **61**(2), 341–361 (2015)
23. Pey-Chun, C., Hansen, P., Jaumard, B., Tuy, H.: Solution of the multisource Weber and conditional Weber problems by DC programming. Oper. Res. **46**(4), 548–562 (1998)
24. Souza, J.C.O., Oliveira, P.R., Soubeyran, A.: Global convergence of a proximal linearized algorithm for difference of convex functions. Optim. Lett. **10**, 1529–1539 (2016)
25. Strekalovsky, A.S.: On Global Optimality Conditions for D.C. Programming Problems. Irkutsk State University, Russia (1997)
26. Strekalovsky, A.S.: Global optimality conditions for nonconvex optimization. J. Glob. Optim. **12**, 415–434 (1998)
27. Tuy, H.: Convex Analysis and Global Optimization. Kluwer Academic Publishers, Dordrescht (1998)