Bundle methods for nonsmooth DC optimization

Kaisa Joki and Adil M. Bagirov

Abstract This chapter is devoted to algorithms for solving nonsmooth unconstrained difference of convex optimization problems. Different types of stationarity conditions are discussed and the relationship between sets of different stationary points (critical, Clarke stationary and inf-stationary) is established. Bundle methods are developed based on a nonconvex piecewise linear model of the objective function and the convergence of these methods is studied. Numerical results are presented to demonstrate the performance of the methods.

1 Introduction

Nonsmooth functions represented as a difference of two convex (DC) functions constitute an important subclass of nonsmooth functions. DC functions preserve some important properties of convex functions. Many practical problems such as supervised data classification and clustering problems in machine learning [5], clusterwise linear regression [7], piecewise linear regression [4], image restoration problems in artificial intelligence [26, 27] and statistical estimation problems [9] can be formulated as an unconstrained or constrained DC optimization problem.

The first important problem when dealing with DC optimization is how to represent a given function as a DC function. In many situations, it is

Kaisa Joki
Department of Mathematics and Statistics, University of Turku, FI-20014 Turku, Finland, e-mail: kaisa.joki@utu.fi

Adil M. Bagirov
School of Science, Engineering and Information Technology, Federation University Australia, University Drive, Mount Helen, PO Box 663, Ballarat, VIC 3353, Australia e-mail: a.bagirov@federation.edu.au

easy to find such representations without using any special techniques. For example, in all above mentioned application areas the objective functions can be represented as a DC function using simple operations. For some other cases, like polynomials, special algorithms have been developed to find DC representations (see e.g. [1, 11, 12]).

Traditionally, over the long period since 1980s, DC optimization has been considered as a branch of global optimization and various algorithms have been developed to globally solve these problems [38]. The first local search method, the DCA (DC Algorithm), for solving DC optimization problems was introduced in 1985 [33].

Over the recent years DC optimization problems, especially nonsmooth DC optimization problems, have attracted a significant attention. Several methods have been developed [2, 6, 16, 21, 23, 31, 32, 36]. Most successful among these methods are those based on the extension of the bundle method for minimizing convex functions. In this chapter, we present two versions of the bundle method for solving unconstrained DC optimization problems. The convergence properties of these methods are studied and they are tested using some academic test problems. In addition, we briefly discuss the relation between these methods and another bundle method for DC optimization.

The rest of the chapter is organized as follows. In Section 2, the unconstrained DC optimization problem is formulated and optimality conditions are discussed. Cutting plane models of convex functions are studied in Section 3. The proximal bundle method for DC optimization is presented in Section 4 and the double bundle method is given in Section 5. The piecewise-concave bundle method is briefly discussed in Section 6. Numerical results are reported in Section 7. Section 8 concludes the chapter.

2 Optimality conditions in DC optimization

In this chapter, the interest is in unconstrained DC optimization. Therefore, we start with formally defining DC functions.

Definition 1. A function $f : \mathbb{R}^n \to \mathbb{R}$ is a *DC function* if it can be represented in the form

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

where functions $f_1,\ f_2 : \mathbb{R}^n \to \mathbb{R}$ are convex and finite valued.

In this definition, the convex functions f_1 and f_2 used to define a DC function f are called *DC components* whereas $f_1 - f_2$ is a *DC decomposition* of f. DC functions, defined on $\mathbb{R}^n$, are LLC and they can be nonsmooth. From Definition 1, we notice that, even though DC functions are typically nonconvex, they have a structure separating their convex and concave behaviour.

The unconstrained DC optimization problem is of the form

$$\begin{cases} \text{minimize} & f(\mathbf{x}) \\ \text{subject to} & \mathbf{x} \in \mathbb{R}^n, \end{cases} \tag{1}$$

where the objective function $f : \mathbb{R}^n \to \mathbb{R}$ is a DC function.

Definition 2. For the problem (1), a point $\mathbf{x}^* \in \mathbb{R}^n$ is a local minimizer if $f(\mathbf{x}^*)$ is finite and there exists $\varepsilon > 0$ such that

$$f(\mathbf{x}^*) \le f(\mathbf{x}), \quad \text{for all } \mathbf{x} \in B(\mathbf{x}^*; \varepsilon).$$

Below we summarize commonly used necessary optimality conditions in DC optimization. More conditions are presented, for example, in [18, 26, 34].

Theorem 1. *[8, 18, 26, 37] Let functions $f_1, f_2 : \mathbb{R}^n \to \mathbb{R}$ be convex. If a point $\mathbf{x}^* \in \mathbb{R}^n$ is a local minimizer of a DC function $f = f_1 - f_2$ then the following conditions hold*

$$\partial f_2(\mathbf{x}^*) \subseteq \partial f_1(\mathbf{x}^*), \tag{2}$$

$$\mathbf{0} \in \partial f(\mathbf{x}^*) \quad \textit{and} \tag{3}$$

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

From Theorem 1, we obtain definitions of three types of stationary points. First, the points satisfying the condition (2) are called *inf-stationary points.* Second, if a point fulfils the condition (3) then it is *Clarke stationary.* Finally, the condition (4) gives a definition for a *critical point.* The next proposition presents relationships between the sets of different stationary points, which are also illustrated in Figure 1.

Proposition 1. *[16] Let S_{inf} be a set of inf-stationary points, S_{cl} be a set of Clarke stationary points and S_{cr} be a set of critical points of the DC function $f = f_1 - f_2$ with convex DC components $f_1, f_2 : \mathbb{R}^n \to \mathbb{R}$. Then*

(i) $S_{inf} \subseteq S_{cl} \subseteq S_{cr}$;
(ii) *if f_1 is differentiable in $\mathbb{R}^n$ then $S_{cl} = S_{cr}$;*
(iii) *if f_2 is differentiable in $\mathbb{R}^n$ then $S_{inf} = S_{cl} = S_{cr}$.*

Proposition 1 shows that inf-stationarity is the strongest condition among those presented in Theorem 1. Furthermore, this condition guarantees local optimality if the second DC component f_2 is a polyhedral convex function of the form

$$f_2(\mathbf{x}) = \max_{i=1,\dots,m} \{\mathbf{a}_i^\top \mathbf{x} + b_i\},$$

where $\mathbf{a}_i \in \mathbb{R}^n$, $b_i \in \mathbb{R}$ and $m \in \mathbb{N}$. Therefore, it would be worthwhile to use inf-stationarity as a stopping condition to design algorithms. Unfortunately, this condition is hard to verify in practice as it requires the whole subdifferentials of DC components to be known. This is a strong requirement since the calculation of the whole subdifferential, in general, may be

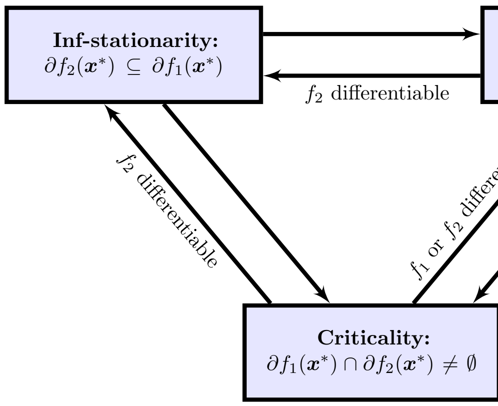


Fig. 1 Relationships between different stationary points

time consuming and in some cases even impossible. Most methods of nonsmooth optimization typically require that only one arbitrary subgradient of an objective function can be calculated at any $\mathbf{x} \in \mathbb{R}^n$. In this chapter, this assumption appears in the form that at $\mathbf{x} \in \mathbb{R}^n$ at most one arbitrary subgradient for both DC components can be calculated and this information may not be sufficient to validate inf-stationarity.

Proposition 1 implies that Clarke stationarity is stronger than criticality. The former condition is widely used in algorithms for minimizing nonsmooth nonconvex functions. In the convex case, this condition is also sufficient and guarantees global optimality. Even though this condition is often utilized, it can be hard to verify for a DC function using subgradents of DC components. This follows from the subdifferential calculus rule yielding [3]

$$\partial f(\mathbf{x}) \subseteq \partial f_1(\mathbf{x}) - \partial f_2(\mathbf{x}). \tag{5}$$

Thus, the difference of the subdifferentials of DC components is an estimate for the subdifferential of f. If f_1 or f_2 is differentiable then the equality holds in (5) and the estimate coincides with the subdifferential of f. However, by selecting the DC decomposition in a special way this estimate can always be made as rough as desired [17]. Thus, subgradients of DC components cannot be used to verify Clarke stationarity without some extra assumptions.

The condition (4) defining criticality is a relaxation of inf-stationarity. Criticality is commonly used as a stopping condition in DC optimization. Unlike the conditions (2) and (3) it is quite easy to verify and it typically provides good solutions. However, one major drawback of a critical point is that it does not need to be a local optimizer or even a saddle point. In the worst case, the algorithm may stop in a point, where the original DC function f is differentiable and the direction opposite to the gradient of f would provide a descent direction decreasing the value of f [23].

As already said, inf-stationarity is the strongest condition presented in Theorem 1 and it always implies Clarke stationarity and criticality. In addition, a Clarke stationary point always satisfies the criticality condition. However, inverse relationships between these necessary conditions are not obtained in a general case. This means that criticality is the weakest condition. In the following, we provide examples, where the inverse relationships do not hold. We start with three examples showing that not all critical points of DC functions are Clarke stationary points. The final example gives an illustration of the case, where a Clarke stationary point is not inf-stationary.

Example 1. [23] Consider a linear function $f(x) = x,\ x \in \mathbb{R}$ for which one possible DC decomposition $f = f_1 - f_2$ can be stated by selecting

$$f_1(x) = \max\{-x, 2x\} \quad \text{and} \quad f_2(x) = \max\{-2x, x\}.$$

The subdifferentials of DC components at a point $x^* = 0$ are $\partial f_1(0) = [-1, 2]$ and $\partial f_2(0) = [-2, 1]$. From this we obtain $\partial f_1(0) \cap \partial f_2(0) = [-1, 1] \neq \emptyset$. This implies that the point x^* is a critical point. However, the function f is differentiable at $x^* = 0$ and $\partial f(0) = \{1\}$. Therefore, the point x^* is not a Clarke stationary point and does not provide any interesting feature for the function f.

Example 2. Consider the function $f(x) = f_1(x) - f_2(x),\ x \in \mathbb{R}$, where

$$f_1(x) = \max\{0, x\} \quad \text{and} \quad f_2(x) = \max\{0, -2x\}.$$

It is obvious that $f(x) = x$ if $x \geq 0$ and $f(x) = 2x$ if $x < 0$. Due to this the subdifferential of the function f at $x^* = 0$ is $\partial f(0) = [1, 2]$. This means that the point $x^* = 0$ is not Clarke stationary as $0 \notin \partial f(0)$. On the other hand, the subdifferentials of the DC components f_1 and f_2 at $x^* = 0$ are $\partial f_1(0) = [0, 1]$ and $\partial f_2(0) = [-2, 0]$. Since $\partial f_1(0) \cap \partial f_2(0) = \{0\} \neq \emptyset$, the point $x^* = 0$ is a critical point. However, it is not Clarke stationary.

Example 3. [23] Consider the function $f(x) = f_1(x) - f_2(x)$, $x \in \mathbb{R}$, where DC components are selected to be

$$f_1(x) = \max\{x^2, x\} \quad \text{and} \quad f_2(x) = \max\{0.5x^2, -x\}.$$

At the point $x^* = 0$, the DC components are not differentiable and their subdifferentials are $\partial f_1(0) = [0, 1]$ and $\partial f_2(0) = [-1, 0]$. Since $\partial f_1(0) \cap \partial f_2(0) = \{0\} \neq \emptyset$ the point $x^* = 0$ is a critical point. However, the original DC function f is differentiable at x^* and $\partial f(0) = \{1\}$. This shows that x^* is not Clarke stationary.

Example 4. Consider the following function

$$f(x_1, x_2) = f_1(x_1, x_2) - f_2(x_1, x_2),$$

where $f_1(x_1, x_2) = |x_1|$ and $f_2(x_1, x_2) = |x_2|$. The subdifferentials of the DC components at $\mathbf{x}^* = \mathbf{0}$ are

$$\partial f_1(\mathbf{0}) = \operatorname{conv}\{(1,0)^\top, (-1,0)^\top\} \quad \text{and} \quad \partial f_2(\mathbf{0}) = \operatorname{conv}\{(0,1)^\top, (0,-1)^\top\}.$$

However, the subdifferential of the function f at $\mathbf{x}^* = \mathbf{0}$ is

$$\partial f(\mathbf{0}) = \operatorname{conv}\{(1,1)^\top, (-1,1)^\top, (1,-1)^\top, (-1,-1)^\top\}.$$

From this we easily deduce that $\partial f_2(\mathbf{0}) \not\subset \partial f_1(\mathbf{0})$. However, $\mathbf{0} \in \partial f(\mathbf{0})$ meaning that the point $\mathbf{x}^* = \mathbf{0}$ is Clarke stationary, but not inf-stationary.

It is worth noting, that instead of criticality it is also possible to test its generalization, the so-called ε-*criticality*, requiring that at a point $\mathbf{x}^* \in \mathbb{R}^n$ the condition

$$\partial_\varepsilon f_1(\mathbf{x}^*) \cap \partial_\varepsilon f_2(\mathbf{x}^*) \neq \emptyset$$

holds for $\varepsilon \geq 0$. The smaller the parameter $\varepsilon > 0$ is, the more accurate approximation of criticality is obtained. Naturally, ε-criticality coincides with criticality with the selection $\varepsilon = 0$.

Using ε-subdifferentials of DC components one can get the following necessary and sufficient condition for global optimality.

Theorem 2. [18] *Let functions $f_1, f_2 : \mathbb{R}^n \to \mathbb{R}$ be convex. A point $\mathbf{x}^* \in \mathbb{R}^n$ is a global minimizer of a DC function $f = f_1 - f_2$, if and only if*

$$\partial_\varepsilon f_2(\mathbf{x}^*) \subseteq \partial_\varepsilon f_1(\mathbf{x}^*) \quad \textit{for all } \varepsilon \geq 0.$$

Unfortunately, this condition is hard to utilize in practise, since it requires availability of the entire ε-subdifferentials of the DC components for all $\varepsilon \geq 0$. The calculation of the ε-subdifferential is a challenging task even for not very complex nonsmooth functions.

3 Convex cutting plane model of a DC component

Before presenting bundle methods for nonsmooth DC optimization, we introduce a classical convex cutting plane model used in bundle methods (see, e.g., [19, 24, 28, 30, 35]). This model serves as a basic tool for building an approximation of a DC function by treating DC components separately. This enables us to utilize explicitly the DC structure and to capture knowledge about both the convexity and concavity of the objective.

We require that at any point $\mathbf{x} \in \mathbb{R}^n$ one can calculate the values of the DC components f_1 and f_2. In addition, as already said, we assume that whenever necessary we can compute arbitrary subgradients $\boldsymbol{\xi}_1 \in \partial f_1(\mathbf{x})$ and $\boldsymbol{\xi}_2 \in \partial f_2(\mathbf{x})$ for both DC components or just for one of them.

In bundle methods, the basic idea is to approximate the subdifferential of the objective function by storing subgradients from the previous iterations into a bundle. We utilize a similar idea, but instead of one bundle, we construct separate bundles for both DC components at the current iteration point $\mathbf{x}_k \in \mathbb{R}^n$, namely, $\mathcal{B}_1^k$ and $\mathcal{B}_2^k$. More precisely, these sets are defined in the form

$$\mathcal{B}_i^k = \left\{ (\mathbf{y}_j, f_i(\mathbf{y}_j), \boldsymbol{\xi}_{i,j}) \,|\, j \in J_i^k \right\}, \quad \text{for } i = 1, 2,$$

where $\mathbf{y}_j \in \mathbb{R}^n$ is an auxiliary point from a previous iteration, $\boldsymbol{\xi}_{i,j} \in \partial f_i(\mathbf{y}_j)$ is an arbitrary subgradient and J_i^k is a nonempty set of indices. In general, the index sets J_1^k and J_2^k need not to be the same, but the current iteration point $\mathbf{x}_k$ has to be always included in both of them.

Elements of $\mathcal{B}_1^k$ and $\mathcal{B}_2^k$ can be used to linearize the DC components f_1 and f_2, respectively. The classical *cutting plane model* for the function $f_i,\ i = 1, 2$ at the point $\mathbf{x}_k \in \mathbb{R}^n$ is given by

$$\hat{f}_i^k(\mathbf{x}) = \max_{j \in J_i^k} \left\{ f_i(\mathbf{y}_j) + \boldsymbol{\xi}_{i,j}^\top (\mathbf{x} - \mathbf{y}_j) \right\} \quad \text{for } \mathbf{x} \in \mathbb{R}^n. \tag{6}$$

By introducing the variable $\mathbf{d} = \mathbf{x} - \mathbf{x}_k$, defining the *linearization error*

$$\alpha_{i,j}^k = f_i(\mathbf{x}_k) - f_i(\mathbf{y}_j) - \boldsymbol{\xi}_{i,j}^\top (\mathbf{x}_k - \mathbf{y}_j) \quad \text{for all } \ j \in J_i^k \tag{7}$$

and denoting by

$$\Delta_i^k(\mathbf{d}) = \max_{j \in J_i^k} \left\{ \boldsymbol{\xi}_{i,j}^\top \mathbf{d} - \alpha_{i,j}^k \right\}, \tag{8}$$

the model (6) can be rewritten as

$$\hat{f}_i^k(\mathbf{x}_k + \mathbf{d}) = \max_{j \in J_i^k} \left\{ f_i(\mathbf{x}_k) + \boldsymbol{\xi}_{i,j}^\top \mathbf{d} - \alpha_{i,j}^k \right\} = f_i(\mathbf{x}_k) + \Delta_i^k(\mathbf{d}). \tag{9}$$

The important properties of the cutting plane model are presented in the next proposition.

Proposition 2. *Let $f_i : \mathbb{R}^n \to \mathbb{R}$ be a convex function. Then for any $\mathbf{d} \in \mathbb{R}^n$ and $j \in J_i^k$ the following properties hold:*

(i) $\hat{f}_i^k(\mathbf{x}_k + \mathbf{d}) \le f_i(\mathbf{x}_k + \mathbf{d})$*;*
(ii) $\Delta_i^k(\mathbf{d}) \le f_i(\mathbf{x}_k + \mathbf{d}) - f_i(\mathbf{x}_k)$*;*
(iii) $\hat{f}_i^k(\mathbf{y}_j) = f_i(\mathbf{y}_j)$ *and* $\alpha_{i,j}^k \ge 0$*;*
(iv) $\hat{f}_i^k$ *is convex.*

Proof. The properties (i) and (iii) follow directly from Definition XX of the subdifferential of f_i. By combining the property (i) with (9) we obtain the property (ii). It is obvious that a function represented as a maximum of finitely many linear functions is convex which implies convexity of the function $\hat{f}_i^k$. □

From the reformulation (9) of the cutting plane model, we see that the main ingredients to build the model are the subgradients and linearization errors. In addition, whenever a new point $\mathbf{x}_{k+1}$ differing from the current iteration point $\mathbf{x}_k$ is produced, linearization errors need to be updated. Such update can be performed easily using the formula

$$\alpha_{i,j}^{k+1} = \alpha_{i,j}^{k} + f_i(\mathbf{x}_{k+1}) - f_i(\mathbf{x}_k) - \boldsymbol{\xi}_{i,j}^{\top}(\mathbf{x}_{k+1} - \mathbf{x}_k)$$

requiring no knowledge about the auxiliary points $\mathbf{y}_j$ and function values $f_i(\mathbf{y}_j)$ and it is sufficient to store only subgradients and linearization errors. Therefore the bundle $\mathcal{B}_i^k$ is presented in the form

$$\mathcal{B}_i^k = \left\{(\boldsymbol{\xi}_{i,j},\, \alpha_{i,j}^k) \,|\, j \in J_i^k\right\} \quad \text{for } i = 1, 2.$$

4 Proximal bundle method PBDC

In this section, we describe the proximal bundle method for unconstrained nonsmooth DC optimization (PBDC) originally introduced in [21] and also presented in [20]. This method guarantees approximate ε-criticality for the obtained solution. The idea in the model construction is to build the convex cutting plane model for both DC components. Due to this, the original DC function is approximated with a nonconvex DC cutting plane model obtained by combining the separate approximations of the DC components. This way the model takes explicitly into account the DC structure of the objective and incorporates both the convex and concave behaviour.

4.1 Model and direction finding

We start with introducing the model of the DC function f, where the idea is to substitute both DC components with their convex cutting plane models (6). Therefore, the *nonconvex DC cutting plane model* of f at the current iteration point $\mathbf{x}_k \in \mathbb{R}^n$ is given by

$$\tilde{f}^k(\mathbf{x}) = \hat{f}_1^k(\mathbf{x}) - \hat{f}_2^k(\mathbf{x}). \tag{10}$$

Even though this approximation is nonconvex, it is piecewise linear and models both convex and concave behaviour of the objective. Using the variable $\mathbf{d} = \mathbf{x} - \mathbf{x}_k$ and applying (9) we get

$$\tilde{f}^k(\mathbf{x}_k + \mathbf{d}) = f(\mathbf{x}_k) + \Delta_1^k(\mathbf{d}) - \Delta_2^k(\mathbf{d}).$$

One illustration of the model is given in Example 5.

Example 5. Consider the DC function $f(x) = f_1(x) - f_2(x)$, $x \in \mathbb{R}$ with

$$f_1(x) = \max\{0.85x^2 + x + 2, -x + 4.5\} \quad \text{and} \quad f_2(x) = \max\{0.5x^2, x + 1.5\}.$$

The convex cutting plane models of the DC components are constructed using the function values and subgradients of the DC components at $x = -6, -2$ and 2 and depicted in Figure 2. The overall approximation of f is given in Figure 3.

The model is used to determine a search direction. A quadratic stabilizing term is added into the model to keep it local enough and to guarantee the existence of the solution [28]. This leads to the following global optimization problem

$$\begin{cases} \text{minimize} & P^k(\mathbf{d}) = \varDelta_1^k(\mathbf{d}) - \varDelta_2^k(\mathbf{d}) + \frac{1}{2t}\|\mathbf{d}\|^2 \\ \text{subject to} & \mathbf{d} \in \mathbb{R}^n, \end{cases} \tag{11}$$

where $t > 0$ is a proximity parameter. The solution to this problem is denoted by $\mathbf{d}_t^k \in \mathbb{R}^n$.

From Proposition 2*(ii)*, we obtain that the terms $\varDelta_1^k(\mathbf{d}_t^k)$ and $\varDelta_2^k(\mathbf{d}_t^k)$ estimate the changes in the values of f_1 and f_2, respectively. In addition, we show in the next lemma that the term $\varDelta_1^k(\mathbf{d}_t^k) - \varDelta_2^k(\mathbf{d}_t^k)$ is always nonpositive and, thus, it can be seen as a predicted descent for the actual decrease in the objective function value.

Lemma 1. *For any $t > 0$, we obtain $\varDelta_1^k(\mathbf{d}_t^k) - \varDelta_2^k(\mathbf{d}_t^k) \leq -\dfrac{1}{2t}\|\mathbf{d}_t^k\|^2 \leq 0$.*

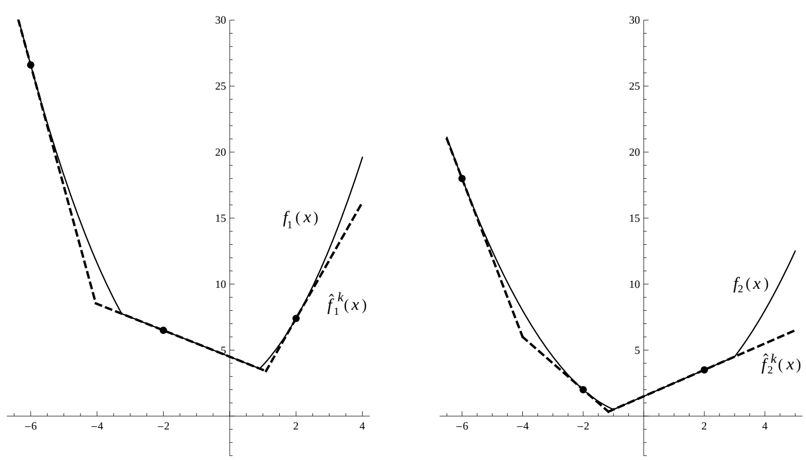


Fig. 2 The convex cutting plane models $\hat{f}_1^k(x)$ and $\hat{f}_2^k(x)$ of the DC components $f_1(x)$ and $f_2(x)$

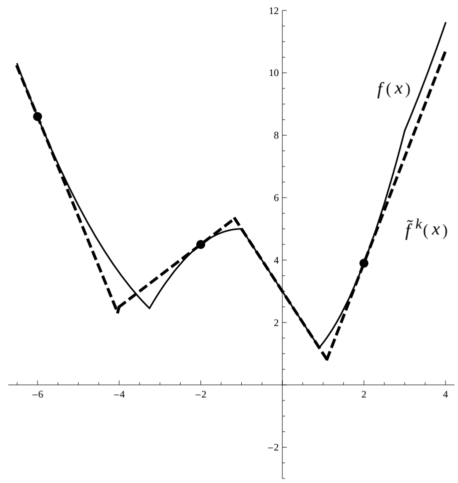


Fig. 3 The nonconvex DC cutting plane model $\tilde{f}^k(x)$ of the objective function $f(x)$

Proof. The direction $\mathbf{d}' = \mathbf{0}$ is a feasible solution for the problem (11). Since both bundles $\mathcal{B}_1^k$ and $\mathcal{B}_2^k$ contain a subgradient calculated at $\mathbf{x}_k$ it follows that among nonnegative linearization errors there is at least one whose value is zero. Thus, we get

$$P^k\left(\mathbf{d}'\right) = \Delta_1^k(\mathbf{0}) - \Delta_2^k(\mathbf{0}) + \frac{1}{2t}\|\mathbf{0}\|^2 = \max_{j \in J_1^k}\left\{-\alpha_{1,j}^k\right\} - \max_{j \in J_2^k}\left\{-\alpha_{2,j}^k\right\} = 0,$$

which implies that the global solution $\mathbf{d}_t^k$ of the problem (11) satisfies

$$P^k\big(\mathbf{d}_t^k\big) = \Delta_1^k\big(\mathbf{d}_t^k\big) - \Delta_2^k\big(\mathbf{d}_t^k\big) + \frac{1}{2t}\|\mathbf{d}_t^k\|^2 \le P\big(\mathbf{d}'\big) = 0.$$

This proves the claim. □

Another useful property is that the search direction $\mathbf{d}_t^k$ is always bounded.

Lemma 2. *For any $t > 0$, it holds that*

$$\|\mathbf{d}_t^k\| \le 2t\left(\|\boldsymbol{\xi}_1(\mathbf{x}_k)\| + \|\boldsymbol{\xi}_{2,\max}\|\right),$$

where $\boldsymbol{\xi}_1(\mathbf{x}_k) \in \partial f_1(\mathbf{x}_k)$ and $\|\boldsymbol{\xi}_{2,\max}\| = \max_{j \in J_2^k} \|\boldsymbol{\xi}_{2,j}\|$.

Proof. Using the definition (8), we deduce the following inequalities

$$\Delta_2^k(\mathbf{d}) \le \max_{j \in J_2^k} \boldsymbol{\xi}_{2,j}^\top \mathbf{d} \le \|\boldsymbol{\xi}_{2,\max}\|\|\mathbf{d}\| \quad \text{for all } \mathbf{d} \in \mathbb{R}^n$$

and

$$\Delta_1^k(\mathbf{d}) \geq \boldsymbol{\xi}_{1,j}^\top \mathbf{d} - \alpha_{1,j}^k \quad \text{for all } j \in J_1^k.$$

By combining them and taking into account that the bundle $\mathcal{B}_1^k$ contains the element $(\boldsymbol{\xi}_1(\mathbf{x}_k), 0)$, where $\boldsymbol{\xi}_1(\mathbf{x}_k) \in \partial f_1(\mathbf{x}_k)$, we obtain

$$\Delta_1^k(\mathbf{d}) - \Delta_2^k(\mathbf{d}) \geq \boldsymbol{\xi}_1(\mathbf{x}_k)^\top \mathbf{d} - \|\boldsymbol{\xi}_{2,\max}\| \|\mathbf{d}\| \geq -(\|\boldsymbol{\xi}_1(\mathbf{x}_k)\| - \|\boldsymbol{\xi}_{2,\max}\|)\|\mathbf{d}\|$$

for all $\mathbf{d} \in \mathbb{R}^n$. Then Lemma 1 implies that

$$-\frac{1}{2t}\|\mathbf{d}_t^k\|^2 \geq \Delta_1^k(\mathbf{d}_t^k) - \Delta_2^k(\mathbf{d}_t^k) \geq -(\|\boldsymbol{\xi}_1(\mathbf{x}_k)\| - \|\boldsymbol{\xi}_{2,\max}\|)\|\mathbf{d}_t^k\|.$$

This completes the proof. □

The problem (11) is nonconvex when $|J_2^k| \geq 2$ and it is not straightforward to distinguish its global solution among the local ones. However, the objective function P^k is still DC with the DC components $\Delta_1^k(\mathbf{d}) + \frac{1}{2t}\|\mathbf{d}\|^2$ and $\Delta_2^k(\mathbf{d})$ and the second DC component is polyhedral convex. This enables us to use a specific approach utilized in [25, 26, 34] to find the global solution.

The main observation in the approach is that the objective P^k can be rewritten in the form

$$P^k(\mathbf{d}) = \min_{i \in J_2^k} \left\{ P_i^k(\mathbf{d}) = \Delta_1^k(\mathbf{d}) - \boldsymbol{\xi}_{2,i}^\top \mathbf{d} + \alpha_{2,i}^k + \frac{1}{2t}\|\mathbf{d}\|^2 \right\}$$

and, thus, we obtain

$$\min_{\mathbf{d} \in \mathbb{R}^n} \min_{i \in J_2^k} \left\{ P_i^k(\mathbf{d}) \right\} = \min_{i \in J_2^k} \min_{\mathbf{d} \in \mathbb{R}^n} \left\{ P_i^k(\mathbf{d}) \right\}$$

showing that the original nonconvex problem can be replaced by a collection of convex subproblems. Due to this, for each $i \in J_2^k$ we solve a convex nonsmooth subproblem

$$\begin{cases} \text{minimize} & P_i^k(\mathbf{d}) = \Delta_1^k(\mathbf{d}) - \boldsymbol{\xi}_{2,i}^\top \mathbf{d} + \alpha_{2,i}^k + \frac{1}{2t}\|\mathbf{d}\|^2 \\ \text{subject to} & \mathbf{d} \in \mathbb{R}^n, \end{cases} \tag{12}$$

whose solution is denoted by $\mathbf{d}_t^k(i)$ and the global solution $\mathbf{d}_t^k$ of the problem (11) is obtained by setting

$$\mathbf{d}_t^k = \mathbf{d}_t^k(i^*), \quad \text{where } i^* = \arg\min_{i \in J_2^k} \left\{ P_i^k\big(\mathbf{d}_t^k(i)\big) \right\}.$$

This means that to solve the problem (11) one needs to solve $|J_2^k|$ subproblems and select the best solution. The bigger the size of the bundle $\mathcal{B}_2^k$ is, the more time-consuming it is to obtain the search direction. In practice, the computational burden can be controlled by restricting the size of $\mathcal{B}_2^k$, since the only requirement is that $|J_2^k| \geq 1$.

For each $i \in J_2^k$ the subproblem (12) can be reformulated as a quadratic programming problem

$$\begin{cases} \text{minimize} & v + \frac{1}{2t}\|\mathbf{d}\|^2 \\ \text{subject to} & (\boldsymbol{\xi}_{1,j} - \boldsymbol{\xi}_{2,i})^\top \mathbf{d} - (\alpha_{1,j}^k - \alpha_{2,i}^k) \leq v, \quad j \in J_1^k, \\ & v \in \mathbb{R},\ \mathbf{d} \in \mathbb{R}^n. \end{cases} \tag{13}$$

Therefore, the global solution $\mathbf{d}_t^k$ can be obtained by solving only smooth subproblems (13) or their dual counterparts

$$\begin{cases} \text{minimize} & \frac{1}{2}t\| \sum_{j \in J_1^k} \lambda_j \boldsymbol{\xi}_{1,j} - \boldsymbol{\xi}_{2,i}\|^2 + \sum_{j \in J_1^k} \lambda_j \alpha_{1,j}^k - \alpha_{2,i}^k \\ \text{subject to} & \sum_{j \in J_1^k} \lambda_j = 1 \\ & \lambda_j \geq 0, \quad j \in J_1^k. \end{cases} \tag{14}$$

The relationship between the optimal solutions $(v_t^k(i),\ \mathbf{d}_t^k(i))$ and $(\lambda_{t,j}^k(i),\ j \in J_1^k)$ to the primal problem (13) and the dual problem (14), respectively, is given as [21]:

$$\mathbf{d}_t^k(i) = -t\left(\sum_{j \in J_1^k} \lambda_{t,j}^k(i) \boldsymbol{\xi}_{1,j} - \boldsymbol{\xi}_{2,i} \right), \tag{15}$$

$$v_t^k(i) = -\frac{1}{t}\|\mathbf{d}_t^k(i)\|^2 - \sum_{j \in J_1^k} \lambda_{t,j}^k(i) \alpha_{1,j}^k + \alpha_{2,i}^k. \tag{16}$$

4.2 Algorithm

The PBDC method for unconstrained DC optimization is presented in Algorithm 1 and its basic structure contains all the characteristic steps used in standard bundle methods including the direction finding, descent test, serious steps and null steps.

To guarantee ε-criticality, two different stopping criteria are used. The first criterion is tested in Step 1 after each serious step and it is formulated using the difference of subgradients of DC components. If this difference is small enough, then the current iteration point $\mathbf{x}_k$ satisfies approximate criticality. The second stopping criterion in Step 4 compares the approximations of the ε-subdifferential of the DC components and, if the difference between ε-subgradients is small enough, then approximate ε-criticality is achieved.

Remark 1. In Algorithm 1, we first define the enlargement parameter $\theta > 0$ depending on the (overestimated) Lipschitz constants $L_1 > 0$ and $L_2 > 0$ of the DC components f_1 and f_2 on the set $\mathcal{F}_\varepsilon = \{\mathbf{x} \in \mathbb{R}^n \mid d(\mathbf{x}, \mathcal{F}_0) \leq \varepsilon\}$,

Algorithm 1: Proximal bundle method for DC optimization (PBDC)

Data: The stopping tolerance $\delta \in (0,1)$, the proximity measure $\varepsilon > 0$, the decrease parameters $r, c \in (0,1)$, the increase parameter $R > 1$, the descent parameter $m \in (0,1)$ and the (overestimated) Lipschitz constants $L_1 > 0$ and $L_2 > 0$ of f_1 and f_2, respectively.

Step 0. (*Initialization*) Select $\mathbf{x}_0 \in \mathbb{R}^n$ and set $\theta = \varepsilon / \max\{2L_1, 2L_2, 1\}$. Calculate $\boldsymbol{\xi}_1(\mathbf{x}_0) \in \partial f_1(\mathbf{x}_0)$ and $\boldsymbol{\xi}_2(\mathbf{x}_0) \in \partial f_2(\mathbf{x}_0)$. Initialize $\mathcal{B}_1^0 = \{(\boldsymbol{\xi}_1(\mathbf{x}_0), 0)\}$ and $\mathcal{B}_2^0 = \{(\boldsymbol{\xi}_2(\mathbf{x}_0), 0)\}$, $\boldsymbol{\xi}_{2,\max} = \mathbf{0}$, and $k = 0$.

Step 1. (*Criticality*) If $\|\boldsymbol{\xi}_1(\mathbf{x}_k) - \boldsymbol{\xi}_2(\mathbf{x}_k)\| < \delta$, then stop with $\mathbf{x}^* = \mathbf{x}_k$ as the final solution.

Step 2. (*Proximity parameter*) If $\|\boldsymbol{\xi}_2(\mathbf{x}_k)\| > \|\boldsymbol{\xi}_{2,\max}\|$, then $\boldsymbol{\xi}_{2,\max} = \boldsymbol{\xi}_2(\mathbf{x}_k)$. Set

$$t_{\min} = \frac{r\theta}{2(\|\boldsymbol{\xi}_1(\mathbf{x}_k)\| + \|\boldsymbol{\xi}_{2,\max}\|)}, \tag{17}$$

$t_{\max} = R t_{\min}$ and $\eta = r\, t_{\min}\, \delta$. Choose $t \in [t_{\min}, t_{\max}]$.

Step 3. (*Search direction*) Calculate the search direction $\mathbf{d}_t^k$ as a solution of (11).

Step 4. (ε-*criticality*) If $\|\mathbf{d}_t^k\| < \eta$, then set

$$J_1^k = J_1^k \setminus \{j \in J_1^k \mid \alpha_{1,j}^k > \varepsilon\}, \quad J_2^k = J_2^k \setminus \{j \in J_2^k \mid \alpha_{2,j}^k > \varepsilon\}$$

and calculate values $\boldsymbol{\xi}_1^*$ and $\boldsymbol{\xi}_2^*$ such that

$$\|\boldsymbol{\xi}_1^* - \boldsymbol{\xi}_2^*\| = \begin{cases} \text{minimize} & \|\boldsymbol{\xi}_1 - \boldsymbol{\xi}_2\| \\ \text{subject to} & \boldsymbol{\xi}_1 \in \operatorname{conv}\{\boldsymbol{\xi}_{1,j} \mid j \in J_1^k\} \\ & \boldsymbol{\xi}_2 \in \operatorname{conv}\{\boldsymbol{\xi}_{2,j} \mid j \in J_2^k\}. \end{cases} \tag{18}$$

If $\|\boldsymbol{\xi}_1^* - \boldsymbol{\xi}_2^*\| < \delta$ then STOP with $\mathbf{x}^* = \mathbf{x}_k$ as the final solution. Otherwise set $t_{\max} = t_{\max} - r\,(t_{\max} - t_{\min})$, select the value $t \in [t_{\min}, t_{\max}]$ and go back to Step 3.

Step 5. (*Descent test*) Set $\mathbf{y} = \mathbf{x}_k + \mathbf{d}_t^k$. If

$$f(\mathbf{y}) - f(\mathbf{x}_k) \le m\big(\Delta_1^k(\mathbf{d}_t^k) - \Delta_2^k(\mathbf{d}_t^k)\big), \tag{19}$$

then set $\mathbf{x}_{k+1} = \mathbf{y}$ and go to Step 8.

Step 6. (*Parameter update*) If $f(\mathbf{y}) > f(\mathbf{x}_0)$ and $\|\mathbf{d}_t^k\| > \theta$, then set $t = t - r(t - t_{\min})$ and go to Step 3.

Step 7. (*Bundle update*) If

$$f(\mathbf{y}) - f(\mathbf{x}_k) \ge -m\big(\Delta_1^k(\mathbf{d}_t^k) - \Delta_2^k(\mathbf{d}_t^k)\big),$$

then set $t = t - c(t - t_{\min})$. Calculate $\boldsymbol{\xi}_1 \in \partial f_1(\mathbf{y})$ and $\boldsymbol{\xi}_2 \in \partial f_2(\mathbf{y})$ together with $\alpha_1 = f_1(\mathbf{x}_k) - f_1(\mathbf{y}) + \boldsymbol{\xi}_1^\top \mathbf{d}_t^k$ and $\alpha_2 = f_2(\mathbf{x}_k) - f_2(\mathbf{y}) + \boldsymbol{\xi}_2^\top \mathbf{d}_t^k$. Update the bundle $\mathcal{B}_1^k = \mathcal{B}_1^k \cup \{(\boldsymbol{\xi}_1, \alpha_1)\}$. If $\Delta_2^k(\mathbf{d}_t^k) \le 0$ then update the bundle $\mathcal{B}_2^k = \mathcal{B}_2^k \cup \{(\boldsymbol{\xi}_2, \alpha_2)\}$ and test if $\|\boldsymbol{\xi}_2\| > \|\boldsymbol{\xi}_{2,\max}\|$ in which case set $\boldsymbol{\xi}_{2,\max} = \boldsymbol{\xi}_2$, update $t_{\min}$ using (17) and set $\eta = r t_{\min} \delta$. Go to Step 3.

Step 8. (*Serious step*) Select $\mathcal{B}_1^{k+1} \subseteq \mathcal{B}_1^k$ and $\mathcal{B}_2^{k+1} \subseteq \mathcal{B}_2^k$ and update the linearization errors in $\mathcal{B}_1^{k+1}$ and $\mathcal{B}_2^{k+1}$. Calculate $\boldsymbol{\xi}_1(\mathbf{x}_{k+1}) \in \partial f_1(\mathbf{x}_{k+1})$ and $\boldsymbol{\xi}_2(\mathbf{x}_{k+1}) \in \partial f_2(\mathbf{x}_{k+1})$. Set $\mathcal{B}_1^{k+1} = \mathcal{B}_1^{k+1} \cup \{(\boldsymbol{\xi}_1(\mathbf{x}_{k+1}), 0)\}$ and $\mathcal{B}_2^{k+1} = \mathcal{B}_2^{k+1} \cup \{(\boldsymbol{\xi}_2(\mathbf{x}_{k+1}), 0)\}$. Update $k = k + 1$ and go to Step 1.

respectively. Here $\mathcal{F}_0 = \{\mathbf{x} \in \mathbb{R}^n \mid f(\mathbf{x}) \leq f(\mathbf{x}_0)\}$, $d(\mathbf{x}, \mathcal{F}_0) = \inf\{\, \|\mathbf{x} - \mathbf{s}\| \mid \mathbf{s} \in \mathcal{F}_0\}$ and $\mathbf{x}_0 \in \mathbb{R}^n$ is a starting point. The local parameter, the so-called local proximity measure $\eta > 0$, is updated each time Step 2 is entered and possibly also in Step 7. In addition, the proximity parameter $t \in [t_{\min}, t_{\max}]$ is selected anew after each serious step and decreased during null steps (Steps 4, 6 and 7) to improve the model. Overall, the parameter selections together with parameter updates are inspired by [13, 14, 15].

As we have already seen in the previous subsection, the size of the bundle $\mathcal{B}_2^k$ affects the complexity of the search direction problem (11). Therefore, the size of $\mathcal{B}_2^k$ needs to be kept limited. The only requirement is that the current iteration point needs to always belong to this bundle. Thus, the size of $\mathcal{B}_2^k$ should satisfy the condition $|J_2^k| \geq 1$. In addition, in Step 8 of Algorithm 1 the bundles $\mathcal{B}_1^k$ and $\mathcal{B}_2^k$ can be chosen independently. This means that we can also decide to omit the previous information and continue with empty bundles. This is possible since before going back to Step 1 we always insert the element calculated at the new iteration point into both bundles.

4.3 Convergence analysis

Next, we prove the finite convergence of the PBDC method to a point satisfying the approximate ε-criticality. To show this the following assumptions are needed:

Assumption 0.1 *The set* $\mathcal{F}_0 = \{\mathbf{x} \in \mathbb{R}^n \mid f(\mathbf{x}) \leq f(\mathbf{x}_0)\}$ *is compact.*

Assumption 0.2 *Lipschitz constants of* f_1 *and* f_2 *or their overestimates are known on the set* $\mathcal{F}_\varepsilon = \{\mathbf{x} \in \mathbb{R}^n \mid d(\mathbf{x}, \mathcal{F}_0) \leq \varepsilon\}$*, where* $\varepsilon > 0$ *is the proximity measure. These constants are denoted by* $L_1 > 0$ *and* $L_2 > 0$*, respectively.*

First, we prove the following two useful lemmas.

Lemma 3. *If the condition* (19) *in Step 5 of Algorithm 1 is not satisfied, then the new element* $(\boldsymbol{\xi}_1, \alpha_1)$ *of* $\mathcal{B}_1^k$ *computed in Step 7 satisfies*

$$\boldsymbol{\xi}_1^\top \mathbf{d}_t^k - \alpha_1 > m\Delta_1^k(\mathbf{d}_t^k) + (1-m)\Delta_2^k(\mathbf{d}_t^k).$$

Proof. Whenever the condition (19) is not satisfied, we obtain

$$\begin{aligned} f_1(\mathbf{y}) - f_1(\mathbf{x}_k) &> m\Big(\Delta_1(\mathbf{d}_t^k) - \Delta_2(\mathbf{d}_t^k)\Big) + \Big(f_2(\mathbf{y}) - f_2(\mathbf{x}_k)\Big) \\ &\geq m\Delta_1^k(\mathbf{d}_t^k) + (1-m)\Delta_2^k(\mathbf{d}_t^k), \end{aligned}$$

where the second inequality follows from Proposition 2*(ii)*. The result can be obtained from this by noticing that $f_1(\mathbf{y}) - f_1(\mathbf{x}_k) = \boldsymbol{\xi}_1^\top \mathbf{d}_t^k - \alpha_1$, where $\boldsymbol{\xi}_1 \in \partial f_1(\mathbf{y})$ and $\alpha_1 = f_1(\mathbf{x}_k) - f_1(\mathbf{y}) + \boldsymbol{\xi}_1^\top \mathbf{d}_t^k$. ◻

Lemma 4. *Let Assumption 0.1 be valid. During each iteration k of Algorithm 1*

(i) $\mathbf{x}_k \in \mathcal{F}_\varepsilon$ *and* $\mathbf{y}_j \in \mathcal{F}_\varepsilon$ *for all* $j \in J_1^k \cup J_2^k$*.*
(ii) *there exists* $K > 0$ *such that* $\|\mathbf{x}_k - \mathbf{y}_j\| \leq K$ *for all* $j \in J_1^k \cup J_2^k$*.*
(iii) $\boldsymbol{\xi}_{i,j} \in \partial f_i(\mathbf{y}_j)$ *and* $\alpha_{i,j}^k$ *for all* $j \in J_i^k$ *and* $i = 1, 2$ *are bounded.*
(iv) t_{min} *is monotonically decreasing and bounded from below with a positive threshold.*
(v) $t_{\max}$ *is bounded from above.*

Proof. *(i)* Since each new iteration point decreases the value of the objective it follows that points $\mathbf{x}_k$ belong to the set $\mathcal{F}_\varepsilon$. In addition, Step 6 of Algorithm 1 guarantees that each point $\mathbf{y}_j$ inserted into the bundles $\mathcal{B}_1^k$ and $\mathcal{B}_2^k$ belongs to $\mathcal{F}_\varepsilon$.
(ii) The claim follows from *(i)*, since Assumption 0.1 ensures that the set $\mathcal{F}_\varepsilon$ is compact.
(iii) Since a DC component f_i is convex on $\mathbb{R}^n$ it is LLC and has a Lipschitz constant $L_i > 0$ on a compact set. By taking into account *(i)*, this yields that $\|\boldsymbol{\xi}_{i,j}\| \leq L_i$ for each $j \in J_i^k$ and $i = 1, 2$. This together with the property *(ii)* shows the boundedness of linearization errors, since

$$|\alpha_{i,j}^k| \leq |f_i(\mathbf{x}_k) - f_i(\mathbf{y}_j)| + \|\boldsymbol{\xi}_{i,j}\|\|\mathbf{x}_k - \mathbf{y}_j\| \leq L_i\|\mathbf{x}_k - \mathbf{y}_j\| + L_i K \leq 2L_i K$$

for each $j \in J_i^k$ and $i = 1, 2$.
(iv) During the iteration k of Algorithm 1, parameter $t_{\min}$ is either decreased or kept the same and this shows the first property. From the proof of the property *(iii)* we can obtain that

$$t_{\min} \geq \bar{t}_{\min} = \frac{r\theta}{2L_1 + 2L_2} > 0 \tag{20}$$

yielding the positive lower bound.
(v) Whenever $t_{\max}$ is set in Step 2 of Algorithm 1, the stopping condition in Step 1 cannot hold. Therefore, we have that

$$\|\boldsymbol{\xi}_1(\mathbf{x}_k)\| + \|\boldsymbol{\xi}_{2,\max}\| \geq \|\boldsymbol{\xi}_1(\mathbf{x}_k)\| + \|\boldsymbol{\xi}_2(\mathbf{x}_k)\| \geq \|\boldsymbol{\xi}_1(\mathbf{x}_k) - \boldsymbol{\xi}_2(\mathbf{x}_k)\| \geq \delta.$$

From this we can deduce that

$$t_{\max} \leq \bar{t}_{\max} = \frac{Rr\theta}{2\delta} \tag{21}$$

which proves the claim. □

In order to prove the finite convergence of Algorithm 1, we need to show that there does not exist any infinite loop. Therefore, we start with showing that it is impossible to execute an infinite sequence of consecutive null steps during one iteration.

Proposition 3. *Let Assumptions 0.1 and 0.2 be valid. At any iteration k, for any $\delta > 0$ and $\varepsilon > 0$, Algorithm 1 can pass through Steps 3-7 only finitely many times before entering Step 8 or fulfilling the stopping condition in Step 4.*

Proof. To prove the proposition we show that there does not exist an infinite sequence of consecutive null steps, containing Steps from 3 to 7, during the k-th iteration of Algorithm 1. We index by $i \in \mathcal{I}$ all the quantities obtained during the i-th null step. Next, we consider separately all cases, which could possibly generate the infinite sequence.

Case 1: Step 4 cannot be performed infinitely many times. Indeed, in this case according to Lemma 4*(iv)* the safeguard parameter $t_{\max}$ will be decreased infinitely many times and as a result will become arbitrarily close to $t_{\min}$. Thus, the proximity parameter t becomes arbitrarily close to $t_{\min}$, since $t \in [t_{\min}, t_{\max}]$, meaning that asymptotically t falls below the threshold

$$\rho = \frac{\theta}{2(\|\boldsymbol{\xi}_1(\mathbf{x}_k)\| + \|\boldsymbol{\xi}_{2,\max}\|)}. \tag{22}$$

This together with Lemma 2 yields that from some point on we always have $\|\mathbf{d}_t^k\| \leq \theta$. Therefore, if Step 6 of Algorithm 1 is entered, it is not executed and we move to Step 7, where the obtained subgradients $\boldsymbol{\xi}_1$ and $\boldsymbol{\xi}_2$ belong to $\partial_\varepsilon f_1(\mathbf{x}_k)$ and $\partial_\varepsilon f_2(\mathbf{x}_k)$, respectively. This is due to the selection $\theta = \varepsilon / \max\{2L_1, 2L_2, 1\}$ and Theorem ??. Altogether, the above considerations mean that there exists an iteration after which the bundles $\mathcal{B}_1^k$ and $\mathcal{B}_2^k$ contain only ε-subgradients when Step 4 is entered. Therefore, no subgradients are removed from the bundles in Step 4. In addition, according to (15) the solution $\mathbf{d}_t$ of the problem (11) is always of the form

$$\mathbf{d}_t^k = -t\left(\sum_{j \in J_1^k} \lambda_{t,j}^k(i^*)\boldsymbol{\xi}_{1,j} - \boldsymbol{\xi}_{2,i^*}\right),$$

where $i^* \in J_2^k$ and $\sum_{j \in J_1^k} \lambda_{t,j}^k(i^*) = 1$. Thus, $\mathbf{d}_t^k / t$ is a feasible solution of the problem (18) and

$$\frac{1}{t}\|\mathbf{d}_t^k\| < \frac{\eta}{t} = \frac{t_{\min}}{t} r\delta < \delta$$

since $\|\mathbf{d}_t^k\| < \eta$ whenever Step 4 is executed. Therefore, Step 4 cannot be called infinitely many times.

Case 2: Step 6 cannot be executed infinitely many times. The proximity parameter t decreases at each execution of Step 6. Thus, t converges to $t_{\min}$ due to Lemma 4*(iv)*. This means that after a finite number of steps t falls below the threshold ρ given in (22). According to Lemma 2, $\|\mathbf{d}_t^k\| < \theta$ when $t < \rho$. Thus, Step 6 can be executed only finitely many times.

Case 3: Step 7 cannot be performed infinitely many times. To simplify the notation let $\{\mathbf{d}_t^i\}$, $\{t_i\}$ and $\{\eta_i\}$ be the sequences obtained during null steps $i \in \mathcal{I}$. First, $\{t_i\}$ and $\{\eta_i\}$ converge to positive limits $\hat{t} > 0$ and $\hat{\eta} > 0$, since by Lemma 4*(iv)* both sequences are nonincreasing and bounded from below with a positive threshold. Second, Lemma 2 and Lemma 4*(iii)* imply that $\{\mathbf{d}_t^i\}$ is bounded and, thus, it has a convergent subsequence for $i \in \mathcal{I}' \subseteq \mathcal{I}$ converging to a limit $\hat{\mathbf{d}}$. In addition, we get that the sequences $\{\Delta_1^k(\mathbf{d}_t^i)\}$ and $\{\Delta_2^k(\mathbf{d}_t^i)\}$ are bounded. Therefore, they also have convergent subsequences for $i \in \mathcal{I}'' \subseteq \mathcal{I}'$ and these limits are denoted by $\hat{\Delta}_1^k$ and $\hat{\Delta}_2^k$. As a consequence of Lemma 1 and $\|\mathbf{d}_t^i\| > \eta_i$, we get the inequality

$$\Delta_1^k\big(\mathbf{d}_t^i\big) - \Delta_2^k\big(\mathbf{d}_t^i\big) \le -\frac{1}{2t_i}\|\mathbf{d}_t^i\|^2 \le -\frac{\eta_i^2}{2t_i} < 0 \quad \text{for all } i \in \mathcal{I} \tag{23}$$

implying

$$\hat{\Delta}_1^k - \hat{\Delta}_2^k \le -\frac{\hat{\eta}^2}{2\hat{t}} < 0. \tag{24}$$

Next, consider two successive indices $r, s \in \mathcal{I}''$. During the null step r we obtain a new element $(\boldsymbol{\xi}_1^r, \alpha_1^r)$ for the DC component f_1 and according to Lemma 1 it satisfies $(\boldsymbol{\xi}_1^r)^\top \mathbf{d}_t^r - \alpha_1^r > m\Delta_1^k\big(\mathbf{d}_t^r\big) + (1-m)\Delta_2^k\big(\mathbf{d}_t^r\big)$. In addition, it follows from (8) that $\Delta_1^k\big(\mathbf{d}_t^s\big) \ge (\boldsymbol{\xi}_1^r)^\top \mathbf{d}_t^s - \alpha_1^r$, which together with the previous inequality gives

$$\Delta_1^k\big(\mathbf{d}_t^s\big) - m\Delta_1^k\big(\mathbf{d}_t^r\big) + (m-1)\Delta_2^k\big(\mathbf{d}_t^r\big) > (\boldsymbol{\xi}_1^r)^\top(\mathbf{d}_t^s - \mathbf{d}_t^r).$$

Passing to the limit yields $(1-m)(\hat{\Delta}_1^k - \hat{\Delta}_2^k) > 0$ and as $m \in (0,1)$, this contradicts (24). Thus, Step 7 cannot be entered infinitely many times. □

Now, we are ready to prove the finite convergence of the PBDC method which means that serious steps cannot be repeated infinitely many times.

Theorem 3. *Let Assumptions 0.1 and 0.2 be valid. For any $\delta > 0$ and $\varepsilon > 0$, Algorithm 1 terminates after a finite number of iterations at a point $\mathbf{x}^*$ satisfying the approximate ε-criticality condition*

$$\|\boldsymbol{\xi}_1^* - \boldsymbol{\xi}_2^*\| \le \delta \quad \textit{with } \boldsymbol{\xi}_1^* \in \partial_\varepsilon f_1(\mathbf{x}^*) \textit{ and } \boldsymbol{\xi}_2^* \in \partial_\varepsilon f_2(\mathbf{x}^*).$$

Proof. Algorithm 1 terminates when the stopping criteria either in Step 1 or in Step 4 are satisfied. Both conditions guarantee approximate ε-criticality. Assume the contrary, that is Algorithm 1 never terminates. Then, we have an infinite sequence $\{\mathbf{x}_k\}$ and due to Proposition 3 each iteration point $\mathbf{x}_k$ is obtained after a finite number of null steps. In addition, at each iteration the sufficient descent condition (19) is satisfied guaranteeing that

$$f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k) \le m\big(\Delta_1^k(\mathbf{d}_t^k) - \Delta_2^k(\mathbf{d}_t^k)\big).$$

By combining this with (23) and taking into account the bounds (20) and (21) we obtain

$$f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k) \leq -\frac{m\eta_k^2}{2t_k} \leq -\frac{m(r\bar{t}_{\min}\delta)^2}{2\bar{t}_{\max}} < 0.$$

By summing up the first k of the above inequalities we get

$$f(\mathbf{x}_k) - f(\mathbf{x}_0) \leq -k\sigma,$$

where $\sigma = m(r\bar{t}_{\min}\delta)^2/(2\bar{t}_{\max}) > 0$. Therefore, we can conclude

$$\lim_{k\to\infty} f(\mathbf{x}_k) - f(\mathbf{x}_0) \leq -\infty.$$

This is a contradiction since local Lipschitz continuity of f and Assumption 0.1 guarantee that the objective f is bounded from below. $\square$

5 Double bundle method DBDC

The PBDC method, described above, terminates at solutions satisfying the approximate ε-criticality condition. This condition is easily tested by utilizing the DC structure, but as we have already demonstrated critical points have some drawbacks. Therefore, the PBDC method may terminate at a point which is not a local minimizer or even a saddle point. To avoid this undesirable feature, we now recall the double bundle method for unconstrained DC optimization (DBDC) introduced in [23] and presented also in [20].

The DBDC method is the successor of the PBDC method. It is based on the same DC cutting plane model that is used in the PBDC and the main structure of the DBDC method resembles to that of the PBDC. A significant difference between these two methods is that the DBDC method involves a special escape procedure to find descent directions at critical points which are not Clarke stationary. The use of such a procedure allows one to get stronger convergence results than those obtained for the PBDC method.

5.1 Calculating a subgradient of a DC function

When the DC structure is available, we typically want to use it through the whole algorithm. However, to guarantee Clarke stationarity we need to calculate subgradients of the original DC function. It follows from (5) that, in general, the subdifferentials of the DC components cannot be used for this purpose and, therefore, subgradients of the DC components cannot be utilized to verify Clarke stationarity. However, the careful selection of the subgradients of the DC components enables us to bypass this difficulty and to obtain subgradients of the DC function.

Finite valued convex functions defined on $\mathbb{R}^n$ are directionally differentiable. Under this assumption the DC function is also directionally differentiable. For a convex DC component f_i, the directional derivative at $\mathbf{x} \in \mathbb{R}^n$ in the direction $\mathbf{d} \in \mathbb{R}^n$ can be given in the form (see, e.g., [3])

$$f_i'(\mathbf{x}; \mathbf{d}) = \max \left\{ \boldsymbol{\xi}^\top \mathbf{d} \,|\, \boldsymbol{\xi} \in \partial f_i(\mathbf{x}) \right\} \quad \text{for } i = 1, 2.$$

For any $\mathbf{d} \in \mathbb{R}^n$, $\mathbf{d} \neq \mathbf{0}$, we define the set

$$G_i(\mathbf{x}; \mathbf{d}) = \left\{ \boldsymbol{\xi} \in \partial f_i(\mathbf{x}) \mid \boldsymbol{\xi}^\top \mathbf{d} = f_i'(\mathbf{x}; \mathbf{d}) \right\} \quad \text{for } i = 1, 2,$$

which consists of subgradients yielding the directional derivative of f_i. Since the directional derivative of a finite valued convex function defined on $\mathbb{R}^n$ is LLC it is differentiable almost everywhere. Then at a point $\mathbf{x} \in \mathbb{R}^n$ there exists a set $T_{DC}(\mathbf{x})$ of full measure such that both sets $G_1(\mathbf{x}; \mathbf{d})$ and $G_2(\mathbf{x}; \mathbf{d})$ are singletons for all $\mathbf{d} \in T_{DC}(\mathbf{x})$. Next, we show that directions $\mathbf{d} \in T_{DC}(\mathbf{x})$ have a central role when the DC components are used to determine a subgradient of the DC function at a point $\mathbf{x}$.

Theorem 4. *Let $f = f_1 - f_2$ be a DC function, $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{d} \in T_{DC}(\mathbf{x})$, $G_1(\mathbf{x}; \mathbf{d}) = \{\boldsymbol{\xi}_1\}$ and $G_2(\mathbf{x}; \mathbf{d}) = \{\boldsymbol{\xi}_2\}$. Then*

$$\boldsymbol{\xi}_1 - \boldsymbol{\xi}_2 \in \partial f(\mathbf{x}).$$

Proof. We start with defining the set

$$\begin{aligned} U_i(\mathbf{x}; \mathbf{d}) = \{ \boldsymbol{\xi} \in \mathbb{R}^n \,|\, \exists \{\mathbf{v}_k\} \text{ and } \{t_k\}, \mathbf{v}_k \in \partial f_i(\mathbf{x} + t_k \mathbf{d}), \\ \mathbf{v}_k \to \boldsymbol{\xi} \text{ and } t_k \downarrow 0 \text{ as } k \to \infty \} \end{aligned}$$

for $i = 1, 2$. Since both f_1 and f_2 are convex they are weakly semismooth. Then applying Definition XX (weakly semismooth) we get

$$U_i(\mathbf{x}; \mathbf{d}) \subseteq G_i(\mathbf{x}; \mathbf{d}) \quad \text{for } i = 1, 2.$$

By combining this with the definition of $U_i(\mathbf{x}; \mathbf{d})$ we obtain that for any $\varepsilon > 0$ there exists $t_0 > 0$ such that

$$\partial f_i(\mathbf{x} + t\mathbf{d}) \subset U_i(\mathbf{x}; \mathbf{d}) + B(\mathbf{0}; \varepsilon) \subseteq G_i(\mathbf{x}; \mathbf{d}) + B(\mathbf{0}; \varepsilon) = \{\boldsymbol{\xi}_i\} + B(\mathbf{0}; \varepsilon)$$

for $i = 1, 2$ and any $t \in (0, t_0)$. This implies that

$$\|\mathbf{v} - \boldsymbol{\xi}_1\| < \varepsilon \quad \text{and} \quad \|\mathbf{w} - \boldsymbol{\xi}_2\| < \varepsilon \tag{25}$$

for all $\mathbf{v} \in \partial f_1(\mathbf{x} + t\mathbf{d})$, $\mathbf{w} \in \partial f_2(\mathbf{x} + t\mathbf{d})$ and $t \in (0, t_0)$. In addition, any $\boldsymbol{\xi}_t \in \partial f(\mathbf{x} + t\mathbf{d})$ can be expressed as $\boldsymbol{\xi}_t = \mathbf{v}_t - \mathbf{w}_t$ where $\mathbf{v}_t \in \partial f_1(\mathbf{x} + t\mathbf{d})$

and $\mathbf{w}_t \in \partial f_2(\mathbf{x}+t\mathbf{d})$. Therefore, by taking into account inequalities (25) we obtain

$$\|\boldsymbol{\xi}_t - (\boldsymbol{\xi}_1 - \boldsymbol{\xi}_2)\| \leq \|\mathbf{v}_t - \boldsymbol{\xi}_1\| + \|\mathbf{w}_t - \boldsymbol{\xi}_2\| < 2\varepsilon$$

for all $\boldsymbol{\xi}_t \in \partial f(\mathbf{x}+t\mathbf{d})$ and $t \in (0, t_0)$. This shows that

$$\boldsymbol{\xi}_1 - \boldsymbol{\xi}_2 \in \partial f(\mathbf{x}+t\mathbf{d}) + B(\mathbf{0}; 2\varepsilon) \quad \text{for all } t \in (0, t_0). \tag{26}$$

On the other hand, upper semicontinuity of ∂f [8] means that for any $\varepsilon > 0$ there exists $t_1 > 0$ such that

$$\partial f(\mathbf{x}+t\mathbf{d}) \subset \partial f(\mathbf{x}) + B(\mathbf{0}; \varepsilon) \quad \text{for all } t \in (0, t_1) \tag{27}$$

which together with (26) implies that for any $\varepsilon > 0$

$$\boldsymbol{\xi}_1 - \boldsymbol{\xi}_2 \in \partial f(\mathbf{x}) + B(\mathbf{0}; 3\varepsilon).$$

This proves that $\boldsymbol{\xi}_1 - \boldsymbol{\xi}_2 \in \partial f(\mathbf{x})$. □

Theorem 4 shows that if the direction is selected with care, then a subgradient of the DC function can be obtained by using the subgradients of the DC components. The difficulty in applying Theorem 4 is that its claim does not necessarily hold for an arbitrary direction. Next, we show that at a point $\mathbf{x} \in \mathbb{R}^n$ for any direction $\mathbf{d} \in \mathbb{R}^n$ one can construct a direction $\bar{\mathbf{d}} \in T_{DC}(\mathbf{x})$ which is sufficiently close to $\mathbf{d}$.

Theorem 5. *Let $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{d} \in \mathbb{R}^n$ be any direction such that $\mathbf{d} \neq \mathbf{0}$, and assume that for a DC function $f = f_1 - f_2$ the subdifferentials $\partial f_1(\mathbf{x})$ and $\partial f_2(\mathbf{x})$ are polytopes. Then for a given $\bar{\mathbf{g}} \in V = \{\mathbf{g} \in \mathbb{R}^n \,|\, \mathbf{g} = (g_1, \dots, g_n),\, |g_i| = 1,\, \forall i\}$, there exists $\alpha_0 \in (0,1]$ such that for all $\alpha \in (0, \alpha_0]$:*

(i) $\bar{\mathbf{d}}(\alpha) = \mathbf{d} + \mathbf{e}^n(\alpha) \in T_{DC}(\mathbf{x})$, *where* $\mathbf{e}^n(\alpha) = (\alpha\bar{g}_1, \alpha^2\bar{g}_2, \dots, \alpha^n\bar{g}_n)$*;*
(ii) $G_1(\mathbf{x}; \bar{\mathbf{d}}(\alpha)) \subseteq G_1(\mathbf{x}; \mathbf{d})$ *and* $G_2(\mathbf{x}; \bar{\mathbf{d}}(\alpha)) \subseteq G_2(\mathbf{x}; \mathbf{d})$*;*
(iii) $f'(\mathbf{x}; \mathbf{d}) = (\boldsymbol{\xi}_1 - \boldsymbol{\xi}_2)^\top \mathbf{d}$ *for* $\boldsymbol{\xi}_1 \in G_1(\mathbf{x}; \bar{\mathbf{d}}(\alpha))$ *and* $\boldsymbol{\xi}_2 \in G_2(\mathbf{x}; \bar{\mathbf{d}}(\alpha))$*;*
(iv) $\boldsymbol{\xi}_1 - \boldsymbol{\xi}_2 \in \partial f(\mathbf{x})$ *for* $\boldsymbol{\xi}_1 \in G_1(\mathbf{x}; \bar{\mathbf{d}}(\alpha))$ *and* $\boldsymbol{\xi}_2 \in G_2(\mathbf{x}; \bar{\mathbf{d}}(\alpha))$.

Proof. The proof of (i) and (ii) is technical and it is given in [23]. To prove the case (iii) notice that $f_1'(\mathbf{x}; \mathbf{d}) = \boldsymbol{\xi}_1^\top \mathbf{d}$ for $\boldsymbol{\xi}_1 \in G_1(\mathbf{x}; \bar{\mathbf{d}}(\alpha))$, $f_2'(\mathbf{x}; \mathbf{d}) = \boldsymbol{\xi}_2^\top \mathbf{d}$ for $\boldsymbol{\xi}_2 \in G_1(\mathbf{x}; \bar{\mathbf{d}}(\alpha))$ and $f'(\mathbf{x}; \mathbf{d}) = f_1'(\mathbf{x}; \mathbf{d}) - f_2'(\mathbf{x}; \mathbf{d})$. The property (iv) is obtained directly from Theorem 4 by taking into account (i). □

The above theorem shows that at a point $\mathbf{x} \in \mathbb{R}^n$ we can always make a small controlled change into any direction $\mathbf{d}$ to obtain $\bar{\mathbf{d}} \in T_{DC}(\mathbf{x})$. Moreover, if for the altered direction $\bar{\mathbf{d}}$ the sets $G_1(\mathbf{x}; \bar{\mathbf{d}})$ and $G_2(\mathbf{x}; \bar{\mathbf{d}})$ are singletons, then the elements in those sets can be used to define a subgradient of f. In addition, the property *(iii)* of Theorem 5 guarantees that the calculated subgradients of the DC components define also the directional derivative of the DC function in the direction $\mathbf{d}$. Thus, the altered direction $\bar{\mathbf{d}}$ is sufficiently close to $\mathbf{d}$. We also notice that the previous theorem requires that at a point $\mathbf{x} \in \mathbb{R}^n$ the subdifferentials $\partial f_1(\mathbf{x})$ and $\partial f_2(\mathbf{x})$ of the DC components are polytopes. This is not very restrictive assumption and in practical applications it is nearly always fulfilled.

5.2 Algorithm

The DBDC method is described in Algorithm 3 and this method resembles the PBDC method (Algorithm 1). The significant difference is the stopping condition, since whenever a candidate solution is obtained, we execute the escape procedure. This procedure either guarantees Clarke stationarity or produces a descent direction yielding a sufficient decrease in the value of the objective. Thus, whenever necessary we are able to escape from non-Clarke stationary points of a DC function.

In the escape procedure, the goal is to identify whether the point $\mathbf{x} \in \mathbb{R}^n$ is Clarke stationary or not. For this reason, we need to approximate the Goldstein ε-subdifferential $\partial_\varepsilon^G f(\mathbf{x})$ of f (see Definition XX) for some small $\varepsilon > 0$. Moreover, the smaller the parameter ε is, the more accurate approximation of $\partial f(\mathbf{x})$ is obtained. Let U_k denote this approximation during iteration k and we have that $U_k \subseteq \partial_\varepsilon^G f(\mathbf{x})$ for all $k \geq 1$. In order to detect Clarke stationarity, we need to solve the problem

$$\begin{cases} \text{minimize} & \frac{1}{2}\|\mathbf{u}\|^2 \\ \text{subject to} & \mathbf{u} \in U_k, \end{cases} \tag{28}$$

providing the solution $\mathbf{u}_k$. If the norm of $\mathbf{u}_k$ is smaller than the stopping tolerance $\delta > 0$, then the point $\mathbf{x}$ satisfies approximate Clarke stationarity. Otherwise, we need to continue the verification procedure. In addition, if the point $\mathbf{x}$ is non-Clarke stationary then the final solution $\mathbf{x}^+$ provided by Algorithm 2 decreases the value of f.

5.3 Convergence analysis

Next, we prove that the DBDC method terminates after a finite number of steps at a point satisfying an approximate Clarke stationarity. In order to show this, we need Assumption 0.1 and also the following assumption:

Assumption 0.3 *The subdifferentials $\partial f_1(\mathbf{x})$ and $\partial f_2(\mathbf{x})$ of the DC components f_1 and f_2 are polytopes at any $\mathbf{x} \in \mathbb{R}^n$.*

First, we show that the escape procedure in Algorithm 2 has a finite convergence. We start with the following useful auxiliary result.

Algorithm 2: Escape procedure

Data: The point $\mathbf{x} \in \mathbb{R}^n$ under consideration, the stopping tolerance $\delta > 0$, the proximity measure $\varepsilon > 0$ and the descent parameter $\hat{m} \in (0,1)$

Step 0. (*Initialization*) Select a direction $\mathbf{d}_1 \in \{\mathbf{d} \in \mathbb{R}^n \mid \|\mathbf{d}\| = 1\}$. Set $\tilde{\mathbf{x}} = \mathbf{x}$, $U_0 = \emptyset$ and $k = 1$.

Step 1. (*New subgradient*) Find $\bar{\mathbf{d}}_k(\alpha) \in T_{DC}(\tilde{\mathbf{x}})$ using $\mathbf{d}_k$. Compute subgradients $\boldsymbol{\xi}_{1,k} \in \partial f_1(\tilde{\mathbf{x}})$ and $\boldsymbol{\xi}_{2,k} \in \partial f_2(\tilde{\mathbf{x}})$ such that $\boldsymbol{\xi}_{1,k} \in G_1(\tilde{\mathbf{x}}; \bar{\mathbf{d}}_k(\alpha))$ and $\boldsymbol{\xi}_{2,k} \in G_2(\tilde{\mathbf{x}}; \bar{\mathbf{d}}_k(\alpha))$. Set $\boldsymbol{\xi}_k = \boldsymbol{\xi}_{1,k} - \boldsymbol{\xi}_{2,k}$ and $U_k = \operatorname{conv}\{U_{k-1} \bigcup \{\boldsymbol{\xi}_k\}\}$.

Step 2. (*Clarke stationarity*) Compute $\mathbf{u}_k$ by solving the problem (28). If

$$\|\mathbf{u}_k\| \le \delta \tag{29}$$

then EXIT with $\mathbf{x}^+ = \mathbf{x}$.

Step 3. (*Search direction*) Compute the search direction $\mathbf{d}_{k+1} = -\mathbf{u}_k / \|\mathbf{u}_k\|$.

Step 4. (*Descent test*) If

$$f'(\mathbf{x}; \mathbf{d}_{k+1}) > -\hat{m}\|\mathbf{u}_k\| \tag{30}$$

then set $\tilde{\mathbf{x}} = \mathbf{x}$ and $k = k+1$ and go to Step 1.

Step 5. (*Step-length*) Calculate the step-length

$$\beta^* = \arg\max \{\beta > 0 \mid f(\mathbf{x} + \beta \mathbf{d}_{k+1}) - f(\mathbf{x}) \le -\hat{m}\beta\|\mathbf{u}_k\|\}. \tag{31}$$

If $\beta^* \ge \varepsilon$ then EXIT with $\mathbf{x}^+ = \mathbf{x} + \beta^* \mathbf{d}_{k+1}$. Otherwise, set $\tilde{\mathbf{x}} = \mathbf{x} + \beta^* \mathbf{d}_{k+1}$ and $k = k+1$ and go to Step 1.

Lemma 5. *Let Assumption 0.3 be valid and assume that the level set $\mathcal{F}_{\mathbf{x}} = \{\mathbf{y} \in \mathbb{R}^n \mid f(\mathbf{y}) \le f(\mathbf{x})\}$ is compact for $\mathbf{x} \in \mathbb{R}^n$. If at the k-th iteration Algorithm 2 does not terminate in Step 5, then*

$$f'(\mathbf{x} + \beta^* \mathbf{d}_{k+1}; \mathbf{d}_{k+1}) > -\bar{m}\|\mathbf{u}_k\| \quad \textit{for all } \bar{m} > \hat{m}.$$

Proof. The compactness of the set $\mathcal{F}_{\mathbf{x}}$ guarantees that the formula (31) for finding the step-length is well-defined and $\beta^* < \infty$. If Algorithm 2 does not terminate in Step 5, then $\beta^* < \varepsilon$ and

$$f(\mathbf{x} + \beta^* \mathbf{d}_{k+1}) - f(\mathbf{x}) \le -\hat{m}\beta^*\|\mathbf{u}_k\|. \tag{34}$$

In addition, no $\beta > \beta^*$ can satisfy the inequality in (31). Assume the contrary, that is

$$f'(\mathbf{x} + \beta^* \mathbf{d}_{k+1}; \mathbf{d}_{k+1}) \le -\bar{m}\|\mathbf{u}_k\| \quad \text{for some } \bar{m} > \hat{m}.$$

This inequality can be rewritten as

$$\lim_{t \downarrow 0} \frac{f(\mathbf{x} + (\beta^* + t)\mathbf{d}_{k+1}) - f(\mathbf{x} + \beta^* \mathbf{d}_{k+1})}{t} \le -\bar{m}\|\mathbf{u}_k\|.$$

Algorithm 3: Double bundle method for DC optimization (DBDC)

Data: The stopping tolerance $\delta \in (0,1)$, the proximity measure $\varepsilon > 0$, the enlargement parameter $\theta > 0$, the decrease parameters $r, c \in (0,1)$, the increase parameter $R > 1$ and the descent parameters $m, \hat{m} \in (0,1)$.

Step 0. (*Initialization*) Select $\mathbf{x}_0 \in \mathbb{R}^n$. Calculate $\boldsymbol{\xi}_1(\mathbf{x}_0) \in \partial f_1(\mathbf{x}_0)$ and $\boldsymbol{\xi}_2(\mathbf{x}_0) \in \partial f_2(\mathbf{x}_0)$. Initialize $\mathcal{B}_1^0 = \{(\boldsymbol{\xi}_1(\mathbf{x}_0), 0)\}$ and $\mathcal{B}_2^0 = \{(\boldsymbol{\xi}_2(\mathbf{x}_0), 0)\}$, $\boldsymbol{\xi}_{2,\max} = \mathbf{0}$, and $k = 0$.

Step 1. (*Criticality*) If $\|\boldsymbol{\xi}_1(\mathbf{x}_k) - \boldsymbol{\xi}_2(\mathbf{x}_k)\| < \delta$, then $\mathbf{d}_t^k = \mathbf{0}$ and go to Step 4.

Step 2. (*Proximity parameter*) If $\|\boldsymbol{\xi}_2(\mathbf{x}_k)\| > \|\boldsymbol{\xi}_{2,\max}\|$, then $\boldsymbol{\xi}_{2,\max} = \boldsymbol{\xi}_2(\mathbf{x}_k)$. Set

$$t_{\min} = \frac{r\theta}{2(\|\boldsymbol{\xi}_1(\mathbf{x}_k)\| + \|\boldsymbol{\xi}_{2,\max}\|)} \tag{32}$$

and $t_{\max} = R t_{\min}$. Choose $t \in [t_{\min}, t_{\max}]$.

Step 3. (*Search direction*) Calculate the search direction $\mathbf{d}_t^k$ as a solution of (11).

Step 4. (*Escape procedure*) If $\|\mathbf{d}_t^k\| < \delta$, then execute the escape procedure Algorithm 2 for the point $\mathbf{x}_k$ to obtain $\mathbf{x}^+$. Set $\mathbf{x}_{k+1} = \mathbf{x}^+$ and go to Step 8.

Step 5. (*Descent test*) Set $\mathbf{y} = \mathbf{x}_k + \mathbf{d}_t^k$. If

$$f(\mathbf{y}) - f(\mathbf{x}_k) \le m\big(\varDelta_1(\mathbf{d}_t^k) - \varDelta_2(\mathbf{d}_t^k)\big), \tag{33}$$

then set $\mathbf{x}_{k+1} = \mathbf{y}$ and go to Step 8.

Step 6. (*Parameter update*) If $f(\mathbf{y}) > f(\mathbf{x}_0)$ and $\|\mathbf{d}_t^k\| > \theta$, then set $t = t - r(t - t_{\min})$. Go to Step 3.

Step 7. (*Bundle update*) If

$$f(\mathbf{y}) - f(\mathbf{x}_k) \ge -m\big(\varDelta_1(\mathbf{d}_t^k) - \varDelta_2(\mathbf{d}_t^k)\big),$$

then set $t = t - c(t - t_{\min})$. Calculate $\boldsymbol{\xi}_1 \in \partial f_1(\mathbf{y})$ and $\boldsymbol{\xi}_2 \in \partial f_2(\mathbf{y})$ together with $\alpha_1 = f_1(\mathbf{x}_k) - f_1(\mathbf{y}) + \boldsymbol{\xi}_1^\top \mathbf{d}_t^k$ and $\alpha_2 = f_2(\mathbf{x}_k) - f_2(\mathbf{y}) + \boldsymbol{\xi}_2^\top \mathbf{d}_t^k$. Update the bundle $\mathcal{B}_1^k = \mathcal{B}_1^k \cup \{(\boldsymbol{\xi}_1, \alpha_1)\}$. If $\varDelta_2(\mathbf{d}_t^k) \le 0$ then update the bundle $\mathcal{B}_2^k = \mathcal{B}_2^k \cup \{(\boldsymbol{\xi}_2, \alpha_2)\}$ and test if $\|\boldsymbol{\xi}_2\| > \|\boldsymbol{\xi}_{2,\max}\|$ in which case set $\boldsymbol{\xi}_{2,\max} = \boldsymbol{\xi}_2$ and update $t_{\min}$ using (32). Go to Step 3.

Step 8. (*Clarke stationarity*) If $\mathbf{x}_{k+1} = \mathbf{x}_k$, then Clarke stationarity is achieved and STOP with $\mathbf{x}^* = \mathbf{x}_k$ as the final solution.

Step 9. (*Serious step*) Select $\mathcal{B}_1^{k+1} \subseteq \mathcal{B}_1^k$ and $\mathcal{B}_2^{k+1} \subseteq \mathcal{B}_2^k$ and update the linearization errors in the selected sets. Calculate $\boldsymbol{\xi}_1(\mathbf{x}_{k+1}) \in \partial f_1(\mathbf{x}_{k+1})$ and $\boldsymbol{\xi}_2(\mathbf{x}_{k+1}) \in \partial f_2(\mathbf{x}_{k+1})$. Set $\mathcal{B}_1^{k+1} = \mathcal{B}_1^{k+1} \cup \{(\boldsymbol{\xi}_1(\mathbf{x}_{k+1}), 0)\}$, $\mathcal{B}_2^{k+1} = \mathcal{B}_2^{k+1} \cup \{(\boldsymbol{\xi}_2(\mathbf{x}_{k+1}), 0)\}$ and $k = k+1$. Go to Step 1.

By Definition XX of the directional derivative this means that for any $\rho > 0$ there exits $t^* > 0$ such that

$$\frac{f(\mathbf{x} + (\beta^* + t^*)\mathbf{d}_{k+1}) - f(\mathbf{x} + \beta^* \mathbf{d}_{k+1})}{t^*} \le -\bar{m}\|\mathbf{u}_k\| + \rho.$$

Since $\bar{m} > \hat{m}$, we can select $\rho = (\bar{m} - \hat{m})\|\mathbf{u}_k\| > 0$. Then we obtain

$$f(\mathbf{x} + (\beta^* + t^*)\mathbf{d}_{k+1}) - f(\mathbf{x} + \beta^* \mathbf{d}_{k+1}) \le -\hat{m} t^* \|\mathbf{u}_k\|.$$

This together with (34) yields

$$f(\mathbf{x} + (\beta^* + t^*)\mathbf{d}_{k+1}) - f(\mathbf{x}) \le -\hat{m}(\beta^* + t^*)\|\mathbf{u}_k\|$$

showing that the inequality in (31) holds for $\beta = \beta^* + t^*$. This is a contradiction since $\beta^* + t^* > \beta^*$. □

Theorem 6. *Let Assumption 0.3 be valid and assume that the level set* $\mathcal{F}_{\mathbf{x}} = \{\mathbf{y} \in \mathbb{R}^n \mid f(\mathbf{y}) \le f(\mathbf{x})\}$ *is compact for* $\mathbf{x} \in \mathbb{R}^n$. *For any* $\delta > 0$ *and* $\varepsilon > 0$, *Algorithm 2 terminates after at most*

$$N_{max} = \left\lceil \frac{\ln(\delta^2/L^2)}{\ln\left(1 - \frac{(1-\hat{m})^2\delta^2}{8L^2}\right)} \right\rceil + 1$$

iterations, where $\lceil\cdot\rceil$ *is a ceiling of a number,* $\hat{m} \in (0,1)$ *and* $L > \delta$ *is the Lipschitz constant of* f *at* $\mathbf{x} \in \mathbb{R}^n$.

Proof. Algorithm 2 terminates if a new better iteration point is found in Step 5 or the stopping condition (29) is fulfilled. Therefore, to prove the theorem it is sufficient to show that one of these two stopping criteria will be satisfied after a finite number of steps.

Assume that the escape procedure does not terminate at the k-th iteration. This means that we end up calculating a new subgradient $\boldsymbol{\xi}_{k+1}$ in Step 1 at the start of the $(k+1)$-th iteration. Next, we show that this subgradient does not belong to $U_k \subset \partial_\varepsilon^G f(\mathbf{x})$ and, therefore, in this case the approximation of the Goldstein ε-subdifferential is improved. The necessary and sufficient condition for the quadratic problem (28) implies that $\mathbf{u}_k^\top \mathbf{u} \ge \|\mathbf{u}_k\|^2$ for all $\mathbf{u} \in U_k$ which yields that

$$\mathbf{u}^\top \mathbf{d}_{k+1} \le -\|\mathbf{u}_k\| \quad \text{for all } \mathbf{u} \in U_k. \tag{35}$$

Now the new point $\tilde{\mathbf{x}}$, where the subgradient $\boldsymbol{\xi}_{k+1}$ is calculated, is either $\mathbf{x}$ or $\mathbf{x} + \beta^*\mathbf{d}_{k+1}$. If the first case occurs, then $\tilde{\mathbf{x}} = \mathbf{x}$ and the condition (30) is satisfied providing that

$$f'(\tilde{\mathbf{x}}; \mathbf{d}_{k+1}) = f'(\mathbf{x}; \mathbf{d}_{k+1}) > -\hat{m}\|\mathbf{u}_k\| > -\bar{m}\|\bar{\mathbf{u}}_k\| \tag{36}$$

for $\bar{m} \in (\hat{m}, 1)$. In the latter case, $\tilde{\mathbf{x}} = \mathbf{x} + \beta^*\mathbf{d}_{k+1}$ for $\beta^* < \varepsilon$ and, thus this point can also be used to calculate a subgradient from the set $\partial_\varepsilon^G f(\mathbf{x})$. Furthermore, Lemma 5 guarantees that

$$f'(\tilde{\mathbf{x}}; \mathbf{d}_{k+1}) = f'(\mathbf{x} + \beta^*\mathbf{d}_{k+1}; \mathbf{d}_{k+1}) > -\bar{m}\|\mathbf{u}_k\| \tag{37}$$

for $\bar{m} \in (\hat{m}, 1)$. The inequalities (36), (37) and Theorem 5*(iii)*, *(iv)* imply that

$$\boldsymbol{\xi}_{k+1}^\top \mathbf{d}_{k+1} > -\bar{m}\|\mathbf{u}_k\| \quad \text{for all } \bar{m} \in (\hat{m}, 1). \tag{38}$$

This means that (35) cannot hold for $\boldsymbol{\xi}_{k+1}$. Thus, the approximation of $\partial_\varepsilon^G f(\mathbf{x})$ is significantly improved, since $\boldsymbol{\xi}_{k+1} \notin U_k$.

Next, we show that the stopping condition (29) is always fulfilled after a finite number of steps if a new better iteration point is never found. This proves the finite termination of Algorithm 2. First, we notice that $t\boldsymbol{\xi}_{k+1} + (1-t)\mathbf{u}_k \in U_{k+1}$ for any $t \in (0,1)$ meaning that

$$\begin{aligned}\|\mathbf{u}_{k+1}\|^2 &\le \|t\boldsymbol{\xi}_{k+1} + (1-t)\mathbf{u}_k\|^2 = \|\mathbf{u}_k + t(\boldsymbol{\xi}_{k+1} - \mathbf{u}_k)\|^2 \\ &= \|\mathbf{u}_k\|^2 + 2t\mathbf{u}_k^\top \boldsymbol{\xi}_{k+1} - \mathbf{u}_k + t^2\|\boldsymbol{\xi}_{k+1} - \bar{\mathbf{u}}_k\|^2.\end{aligned}$$

In addition, local Lipschitz continuity of a DC function guarantees that the Goldstein ε-subdifferential $\partial_\varepsilon^G f(\mathbf{x})$ is bounded at $\mathbf{x}$ with a Lipschitz constant $L > \delta$ determined at that point and thus $\|\boldsymbol{\xi}\| \le L$ for all $\boldsymbol{\xi} \in \partial_\varepsilon^G f(\mathbf{x})$. On the other hand, inequality (38) yields that $\boldsymbol{\xi}_{k+1}^\top \mathbf{u}_k < \bar{m}\|\bar{\mathbf{u}}_k\|^2$ and we obtain

$$\begin{aligned}\|\mathbf{u}_{k+1}\|^2 &\le \|\mathbf{u}_k\|^2 + 2t\mathbf{u}_k^\top(\boldsymbol{\xi}_{k+1} - \mathbf{u}_k) + 4t^2L^2 \\ &< \|\bar{\mathbf{u}}_k\|^2 - 2t(1-\bar{m})\|\mathbf{u}_k\|^2 + 4t^2L^2\end{aligned}$$

for each $\bar{m} \in (\hat{m},1)$. Since $t \in (0,1)$, we can select $t = \frac{(1-\bar{m})\|\mathbf{u}_k\|^2}{4L^2}$ and this gives the approximation

$$\|\mathbf{u}_{k+1}\|^2 < \|\mathbf{u}_k\|^2 \left(1 - \frac{(1-\bar{m})^2\|\mathbf{u}_k\|^2}{4L^2}\right) \qquad \text{for all } \bar{m} \in (\hat{m},1).$$

In addition, there exists $\bar{m} \in (\hat{m},1)$ such that $2(1-\bar{m})^2 > (1-\hat{m})^2$. Therefore, by selecting such $\bar{m}$ and taking into account that $\|\mathbf{u}_k\| > \delta$ for each $k > 0$ we get

$$\|\mathbf{u}_{k+1}\|^2 < \|\mathbf{u}_k\|^2 \left(1 - \frac{(1-\hat{m})^2\delta^2}{8L^2}\right).$$

From this and $\|\mathbf{u}_1\| \le L$ we can conclude that

$$\|\mathbf{u}_k\|^2 < \|\mathbf{u}_1\|^2 \left(1 - \frac{(1-\hat{m})^2\delta^2}{8L^2}\right)^{k-1} \le L^2 \left(1 - \frac{(1-\hat{m})^2\delta^2}{8L^2}\right)^{k-1}$$

showing that when

$$k \ge \left\lceil \frac{\ln(\delta^2/L^2)}{\ln\left(1 - \frac{(1-\hat{m})^2\delta^2}{8L^2}\right)} \right\rceil + 1$$

the stopping criterion $\|\mathbf{u}_k\| \le \delta$ is satisfied. □

Next, similar to the PBDC method, we show that during one iteration of the DBDC method the number of null steps is finite.

Proposition 4. *Let Assumption 0.1 be valid. At any iteration k, for any $\delta > 0$, Algorithm 3 can pass through Steps 3-7 only finitely many times before entering Step 8 or the escape procedure in Step 4.*

Proof. The proof is similar to that of Proposition 3. Nevertheless, we do not need to consider *Case 1*, since the execution of Step 4 terminates the iteration k. Thus, we consider only *Cases 2* and *3*. Furthermore, in *Case 3* we do not have the sequence $\{\eta_i\}$, since η_i is replaced everywhere with the parameter $\delta > 0$. This means that $\hat{\eta}$ is also replaced with this parameter. □

The final result establishes the finite convergence of the DBDC method.

Theorem 7. *Let Assumptions 0.1 and 0.3 be valid. For any $\delta > 0$ and $\varepsilon > 0$, Algorithm 3 terminates after a finite number of iterations at a point $\mathbf{x}^*$ satisfying the approximate Clarke stationarity condition*

$$\|\boldsymbol{\xi}^*\| \leq \delta \quad \textit{with} \;\; \boldsymbol{\xi}^* \in \partial_\varepsilon^G f(\mathbf{x}^*).$$

Proof. Algorithm 3 terminates when the stopping condition in Step 8 is satisfied and this condition guarantees approximate Clarke stationarity. Similarly to the proof of Theorem 3 we next show that if the DBDC method never terminates then the objective function f is not bounded from below. This is a contradiction, since the boundedness of f follows from LLC and Assumption 0.1.

First, notice that if the stopping condition is never fulfilled then Algorithm 3 generates an infinite sequence $\{\mathbf{x}_k\}$. Proposition 4 and Theorem 6 guarantee that a new iteration point $\mathbf{x}_{k+1}$ is always found after a finite number of null steps. Moreover, a new point is obtained either from Step 4 or Step 5. In the first case, Algorithm 2 produces this point and therefore the inequality

$$f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k) \leq -\hat{m}\varepsilon\delta < 0$$

is always satisfied. In the latter case, the descent condition (33) holds and similarly to the proof of Theorem 3 we can show that

$$f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k) \leq -\frac{m\delta^2}{2\bar{t}_{\max}},$$

by taking into account (21) and the fact that the parameter η_i is replaced everywhere with $\delta > 0$. Thus, after each iteration of the DBDC method $f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k) \leq -\sigma < 0$, where $\sigma = \min\{\hat{m}\varepsilon\delta, m\delta^2/(2\bar{t}_{\max})\}$. This enables us to write

$$f(\mathbf{x}_k) - f(\mathbf{x}_0) \leq -k\sigma$$

and a contradiction follows by passing to the limit. □

6 Piecewise-concave bundle method DCPCA

In this section, we briefly describe the main idea of the piecewise concave bundle method for unconstrained DC optimization (DCPCA), which terminates at points satisfying approximate ε-criticality. We also compare the model of the DCPCA with the one used in the PBDC and DBDC methods. For details of the DCPCA, we refer to [16].

The idea in the model construction is to substitute the DC components with their convex cutting plane models (6). Thus, the *nonconvex DC cutting plane model* of a DC function f is given by

$$\tilde{f}^k(\mathbf{x}) = \hat{f}_1^k(\mathbf{x}) - \hat{f}_2^k(\mathbf{x})$$

and in this general form the model is the same as the one used in the PBDC and DBDC methods. However, the bundles of the DC components used in the DCPCA differ from the ones used in the previous methods. In the PBDC and DBDC methods, both bundles of DC components refer to a global model, since the points included into the bundles are collected from the current and past iterations. Nevertheless, in the DCPCA the bundle of the first DC component f_1 only contains information from points which are close to the current iteration point $\mathbf{x}_k$, while the bundle of f_2 refers to the global model. Thus, during each iteration we maintain a local approximation for the DC component f_1.

In order to obtain a search direction in the DCPCA method, the nonconvex cutting plane model is rewritten as

$$\tilde{f}^k(\mathbf{x}) = \max_{j \in J_1^k} \left\{ f(\mathbf{x}_k) + \boldsymbol{\xi}_{1,j}^\top \mathbf{d} - \alpha_{1,j}^k - \Delta_2^k(\mathbf{d}) \right\}. \tag{39}$$

Therefore, the model can be seen as a pointwise maximum of concave piecewise-affine functions. Different from the previous bundle methods, the search direction problem constructed from the nonconvex model (39) is not solved exactly, but instead it is locally approximated with an auxiliary convex quadratic problem. In addition, DCPCA utilizes a supplementary convex quadratic problem to improve the search direction at points which are far away from the current iteration point $\mathbf{x}_k$.

After the search direction is obtained, a line search procedure is executed in the DCPCA method to decide whether a serious step or a null step is performed. If a null step occurs the current iteration point does not change and only the bundle $\mathcal{B}_1^k$ of the DC component f_1 is updated with a new bundle element improving the model. However, if a serious step occurs a new better iteration point $\mathbf{x}_{k+1}$ is obtained and both bundles are updated. Therefore, the bundle $\mathcal{B}_2^k$ needs to be updated only whenever a serious step is done. On the other hand, the bundle $\mathcal{B}_1^k$ is updated in every step. However, in every serious step we are able to reduce the bundle $\mathcal{B}_1^k$, since for the DC component

f_1 we maintain a local approximation, and thus each bundle element of $\mathcal{B}_1^k$ far away from $\mathbf{x}_{k+1}$ can be removed at a serious step.

7 Numerical results

In this section, we illustrate the performance of the PBDC and DBDC methods with nonsmooth DC test problems. We start with demonstration of the main difference between these two methods using an illustrative example. Then, we compare the performance of the PBDC and DBDC methods with that of the proximal bundle method MPBNGC (see Section XX) which is designed for a general nonsmooth objective function. With this comparison the goal is to justify the usage of the DC bundle methods when the DC structure of the problem is available.

7.1 Illustrative example

The PBDC method converge to a critical point and due to this the obtained solution may be located in an unfavourable place being unable to describe an interesting feature for the objective. However, this disadvantage is overcome in the DBDC method with the escape procedure which is illustrated in the following example.

Example 6. Consider the functions f and their DC components f_1 and f_2 from Examples 1–3. Note that in each of these examples the point $x^* = 0$ is a critical point but not Clarke stationary. Due to this, the PBDC method can stop at this critical point. On the other hand, in the DBDC method the fulfilment of the criticality condition at x^* leads to the escape procedure presented in Algorithm 2.

The escape procedure starts by calculating a subgardient ξ of f at $x^* = 0$ by using either a direction $d_1 = 1$ or $d_1 = -1$. In Examples 1 and 3, we always obtain $\xi = 1$ regardless of the selection of d_1 whereas in Example 2 the result is either $\xi = 1$ or $\xi = 2$ depending on whether we use $d_1 = 1$ or $d_1 = -1$, respectively. However, both $\xi = 1$ and $\xi = 2$ yield the search direction $d_2 = -1$. Furthermore, in each example we obtain the same value of the directional derivative of f, that is, $f'(x^*; d_2) = f(0; -1) = -1$. Thus, with the selection of $\hat{m} \in (0, 1/2)$ the condition (30) is not satisfied, since for both $\xi = 1$ and $\xi = 2$ we deduce that

$$f'(x^*; d_2) = -1 \leq -\hat{m}\|\xi\|.$$

This shows that in all three examples d_2 is the descent direction. Thus, the DBDC is able to generate a better iteration point and bypass the problematic critical point x^*.

7.2 Numerical experiments

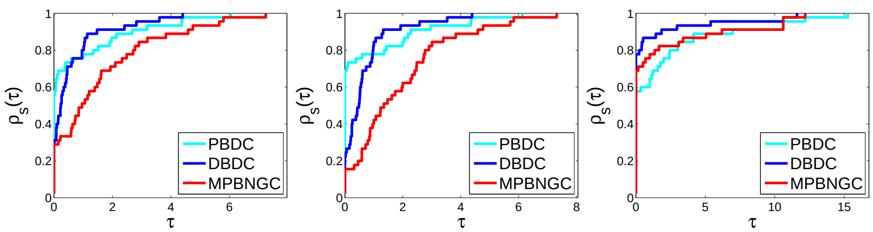
The numerical experiments were carried out using 53 instances of 16 academic DC test problems from [23]. More specifically, Problems 1–10 are introduced in [21] whereas Problems 11–16 can be found from [22]. The selection of the input parameters of the PBDC and DBDC is shown in Table 1. In addition, in both DC bundle methods the size of $\mathcal{B}_1$ is set to $\min\{n+5, 1000\}$ and the size of $\mathcal{B}_2$ is *three*. In Algorithm 2, the size of the set U_k is restricted to $2n$. Furthermore, in MPBNGC the bundle size is fixed to $\min\{n+3, 1000\}$ and the final accuracy is set to 10^{-10}. Other input parameters of the MPBNGC are selected as default values [29].

Both PBDC and DBDC methdos are implemented using double precision Fortran 95 whereas the MPBNGC is implemented in double precision Fortran 77. The numerical tests are performed on an Intel® Core™ i5-2400 CPU (3.10GHz, 3.10GHz) running under Windows 7 and `gfortran` is used as a compiler.

Table 1 The input parameters of the PBDC and DBDC methods

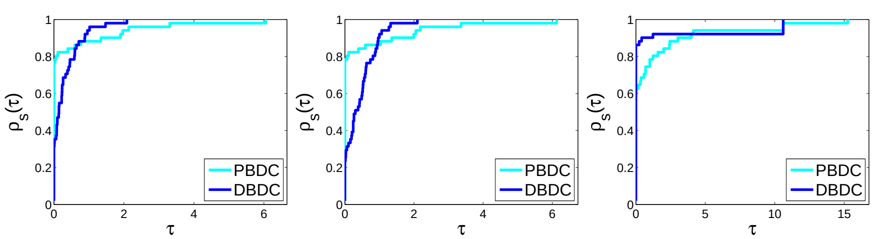
PBDC	DBDC
$\delta = \begin{cases} 0.005n, & \text{if } n < 150 \\ 0.015n, & \text{if } 150 \le n \le 200 \\ 0.05n, & \text{if } n > 200, \end{cases}$	$\delta = \begin{cases} 10^{-5}, & \text{if } n \le 200 \\ 10^{-4}, & \text{if } n > 200 \end{cases}$
$\varepsilon = 0.1$	$\varepsilon = \begin{cases} 10^{-6}, & \text{if } n \le 50 \\ 10^{-5}, & \text{if } n > 50 \end{cases}$
$L_1 = 1000$	$\hat{m} = 0.01$
$L_2 = 1000$	$\theta = 5 \cdot 10^{-5}$
The same input parameters in both methods	
$r = \begin{cases} 0.75, & \text{if } n < 10 \\ \text{the first two desimals of } n/(n+5), & \text{if } 10 \le n < 300 \\ 0.99, & \text{if } n \ge 300, \end{cases}$	
$R = 10^7$	
$c = 0.1$	
$m = 0.2$	

To illustrate the numerical results we use performance profiles [10] which are presented in Figures 4–7. Figure 4 contains the performance profiles with all three solvers and Figures 5–7 present pairwise comparisons. In all cases the number of function evaluations, the number of subgradient evaluations and CPU time are used to draw performance profiles. Furthermore, in each performance profile we include from 53 instances of the test problems only those where the compared methods yield the same (local) solution. That is, 45 instances of the test problems were used for the performance profiles in Figures 4, 6 and 7 whereas 51 instances were used in Figure 5.



(a) (b) (c)

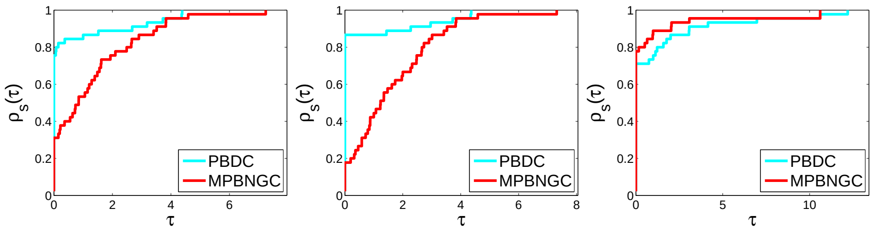
Fig. 4 The performance profiles for the PBDC, DBDC and MPBNGC with 45 instances. (a) Function evaluations, (b) Subgradient evaluations and (c) CPU time.



(a) (b) (c)

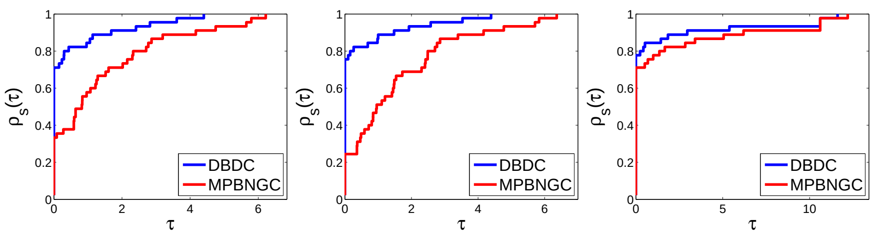
Fig. 5 The performance profiles for the PBDC and DBDC with 51 instances. (a) Function evaluations, (b) Subgradient evaluations and (c) CPU time.

Results show that both the DC bundle methods PBDC and DBDC outperform the MPBNGC method in the sense of the number of function and subgradient evaluations. However, the pairwise comparison of the PBDC and DBDC methods, presented in Figures 5(a) and 5(b), does not clearly show superiority of either of them. Figure 4(c) shows that the DBDC method performs slightly better than the other two methods in the sense of required CPU time. If we restrict our consideration to the pairwise comparison of the PBDC and DBDC methods (see Figure 5(c)) then the differences in CPU time are not so significant. Furthermore, Figure 6(c) demonstrates that the



(a) (b) (c)

Fig. 6 The performance profiles for the PBDC and MPBNGC with 45 instances. (a) Function evaluations, (b) Subgradient evaluations and (c) CPU time.



(a) (b) (c)

Fig. 7 The performance profiles for the DBDC and MPBNGC with 45 instances. (a) Function evaluations, (b) Subgradient evaluations and (c) CPU time.

MPBNGC method is slightly better than the PBDC method in terms of CPU time.

One interesting feature of the DC bundle methods PBDC and DBDC is that they often find the global or best known solution for test problems, even though these methods are only local search methods. For example, in 48 out of 53 instances of the test problems both DC bundle methods find the best known solutions whereas the bundle method MPBNGC obtains the best known solutions only in 44 instances. This indicates that the DC cutting plane model of the DC objective function is able to capture some relevant information about the objective in order to avoid local optimizers.

8 Conclusions

In this chapter, we concentrate on the problem of unconstrained minimization of nonsmooth functions represented as the difference of two convex functions. Necessary and sufficient optimality conditions for such problems are presented and the relationship between sets of different stationary points are discussed in detail. Two different bundle methods are presented to solve DC problems together with their convergence analysis. In addition, the cutting plane model used in these methods is compared to a different DC model. The performance of the methods are demonstrated using nonsmooth test problems with DC objective functions. The results clearly show that the use of DC structure in nonsmooth optimization leads to the design of efficient and accurate methods in comparison with general purposed nonsmooth optimization methods. Furthermore, DC optimization methods based on the extension of the bundle methods for convex optimization are highly successful in finding global solutions to DC optimization problems.

Acknowledgements This work was financially supported by the University of Turku, the Academy of Finland (Projects 294002 and 319274) and the Australian Government through the Australian Research Council's Discovery funding scheme (Project No. DP190100580).

References

1. Ahmadi, A.A., Hall, G.: DC decomposition of nonconvex polynomials with algebraic techniques. Mathematical Programming **169**(1), 69–94 (2018)
2. Bagirov, A.M.: A method for minimization of quasidifferentiable functions. Optimization Methods and Software **17**(1), 31–60 (2002)
3. Bagirov, A.M., Karmitsa, N., Mäkelä, M.M.: Introduction to Nonsmooth Optimization: Theory, Practice and Software. Springer, Cham, Heidelberg (2014)
4. Bagirov, A.M., Taheri, S., Asadi, S.: A difference of convex optimization algorithm for piecewise linear regression. Journal of Industrial and Management Optimization **15**(2), 909–932 (2019)
5. Bagirov, A.M., Taheri, S., Ugon, J.: Nonsmooth DC programming approach to the minimum sum-of-squares clustering problems. Pattern Recognition **53**(1), 12–24 (2016)
6. Bagirov, A.M., Ugon, J.: Codifferential method for minimizing nonsmooth DC functions. Journal of Global Optimization **50**(1), 3–22 (2011)
7. Bagirov Adil M. and Ugon, J.: Nonsmooth DC programming approach to clusterwise linear regression: optimality conditions and algorithms. Optimization Methods and Software **33**(1), 194–219 (2018)
8. Clarke, F.H.: Optimization and Nonsmooth Analysis. Wiley-Interscience, New York (1983)
9. Cui, Y., Pang, J.-S., Sen, B.: Composite difference-max programs for modern statistical estimation problems. SIAM Journal on Optimization **28**(4), 3344–3374 (2018)
10. Dolan, E., Moré, J.: Benchmarking optimization software with performance profiles. Mathematical Programming **91**(2), 201–213 (2002)
11. Ferrer, A.: Representation of a polynomial function as a difference of convex polynomials, with an application. In: N. Hadjisavvas, J.E. Martínez-Legaz, J.P. Penot (eds.) Generalized Convexity and Generalized Monotonicity, vol. 502, pp. 189–207. Springer, Berlin (2001)
12. Ferrer, A., Martínez-Legaz, J.E.: Improving the efficiency of DC global optimization methods by improving the DC representation of the objective function. Journal of Global Optimization **43**(4), 513–531 (2009)

13. Fuduli, A., Gaudioso, M., Giallombardo, G.: A DC piecewise affine model and a bundling technique in nonconvex nonsmooth minimization. Optimization Methods and Software **19**(1), 89–102 (2004)
14. Fuduli, A., Gaudioso, M., Giallombardo, G.: Minimizing nonconvex nonsmooth functions via cutting planes and proximity control. SIAM Journal on Optimization **14**(3), 743–756 (2004)
15. Fuduli, A., Gaudioso, M., Nurminski, E.A.: A splitting bundle approach for nonsmooth non-convex minimization. Optimization **64**(5), 1131–1151 (2015)
16. Gaudioso, M., Giallombardo, G., Miglionico, G., Bagirov, A.M.: Minimizing nonsmooth DC functions via successive DC piecewise-affine approximations. Journal of Global Optimization **71**(1), 37–55 (2018)
17. Hiriart-Urruty, J-.B.: Generalized differentiability, duality and optimization for problems dealing with differences of convex functions. In: J. Ponstein (ed.) Convexity and Duality in Optimization, vol. 256, pp. 37–70. Springer, Berlin (1985)
18. Hiriart-Urruty, J.-B.: From convex optimization to nonconvex optimization. Necessary and sufficient conditions for global optimality. In: F.H. Clarke, V.F. Demyanov, F. Giannessi (eds.) Nonsmooth Optimization and Related Topics, Ettore Majorana International Sciences Series 43, pp. 219–239. Springer, Boston (1989)
19. Hou, L., Sun, W.: On the global convergence of a nonmonotone proximal bundle method for convex nonsmooth minimization. Optimization Methods and Software **23**(2), 227–235 (2008)
20. Joki, K.: Bundle methods in nonsmooth DC optimization. PhD thesis, University of Turku (2018)
21. Joki, K., Bagirov, A.M., Karmitsa, N., Mäkelä, M.M.: A proximal bundle method for nonsmooth DC optimization utilizing nonconvex cutting planes. Journal of Global Optimization **68**(3), 501–535 (2017)
22. Joki, K., Bagirov, A.M., Karmitsa, N., Mäkelä, M.M., Taheri, S.: Double bundle method for nonsmooth DC optimization. TUCS Technical Report No. 1173, Turku Centre for Computer Science, Turku (2017)
23. Joki, K., Bagirov, A.M., Karmitsa, N., Mäkelä, M.M., Taheri, S.: Double bundle method for finding Clarke stationary points in nonsmooth DC programming. SIAM Journal on Optimization **28**(2), 1892–1919 (2018)
24. Kiwiel, K.C.: Proximity control in bundle methods for convex nondifferentiable minimization. Mathematical Programming **46**(1–3), 105–122 (1990)
25. Le Thi, H.A., Pham Dinh, T.: Solving a class of linearly constrained indefinite quadratic problems by D.C. algorithms. Journal of Global Optimization **11**(3), 253–285 (1997)
26. Le Thi, H.A., Pham Dinh, T.: The DC (difference of convex functions) programming and DCA revisited with DC models of real world nonconvex optimization problems. Annals of Operations Research **133**(1–4), 23–46 (2005)
27. Le Thi, H.A., Pham Dinh, T.: Difference of convex functions algorithms (DCA) for image restoration via a Markov random field model. Optimization and Engineering **18**(4), 873–906 (2017)
28. Mäkelä, M.M.: Survey of bundle methods for nonsmooth optimization. Optimization Methods and Software **17**(1), 1–29 (2002)
29. 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 B 13/2003, University of Jyväskylä, Jyväskylä (2003)
30. Mäkelä, M.M., Neittaanmäki, P.: Nonsmooth Optimization: Analysis and Algorithms with Applications to Optimal Control. World Scientific Publishing Co., Singapore (1992)
31. Oliveira, W.: Proximal bundle methods for nonsmooth DC programming. Journal of Global Optimization (2019). DOI 10.1007/s10898-019-00755-4

32. Pang, J.-S., Tao, M.: Decomposition methods for computing directional stationary solutions of a class of nonsmooth nonconvex optimization problems. SIAM Journal on Optimization **28**(2), 1640–1669 (2018)
33. Pham Dinh, T., Le Thi, H.A.: Convex analysis approach to DC programming: Theory, algorithm and applications. Acta Mathematica Vietnamica **22**(1), 289–355 (1997)
34. Pham Dinh, T., Le Thi, H.A.: Convex analysis approach to D.C. programming: Theory, algorithms and applications. Acta Mathematica Vietnamica **22**(1), 289–355 (1997)
35. Schramm, H., Zowe, J.: A version of the bundle idea for minimizing a nonsmooth function: Conceptual idea, convergence analysis, numerical results. SIAM Journal on Optimization **2**(1), 121–152 (1992)
36. Souza, J.C.O., Oliveira, P.R., Soubeyran, A.: Global convergence of a proximal linearized algorithm for difference of convex functions. Optimization Letters **10**(7), 1529–1539 (2016)
37. Toland, J.F.: On subdifferential calculus and duality in nonconvex optimization. Mémoires de la Société Mathématique de France **60**, 177–183 (1979)
38. Tuy, H.: Convex Analysis and Global Optimization, 1st edn. Kluwer Academic Publishers, Dordrecht (1998)