



Proximal bundle methods for nonsmooth DC programming

Wellington de Oliveira¹

Received: 23 October 2017 / Accepted: 19 February 2019 / Published online: 27 February 2019
© Springer Science+Business Media, LLC, part of Springer Nature 2019

Abstract

We consider the problem of minimizing the difference of two nonsmooth convex functions over a simple convex set. To deal with this class of nonsmooth and nonconvex optimization problems, we propose new proximal bundle algorithms and show that the given approaches generate subsequences of iterates that converge to critical points. Trial points are obtained by solving strictly convex master programs defined by the sum of a convex cutting-plane model and a freely-chosen Bregman function. In the unconstrained case with the Bregman function being the Euclidean distance, new iterates are solutions of strictly convex quadratic programs of limited sizes. Stronger convergence results (d -stationarity) can be achieved depending on (a) further assumptions on the second DC component of the objective function and (b) solving possibly more than one master program at certain iterations. The given approaches are validated by encouraging numerical results on some academic DC programs.

Keywords DC programming · Nonsmooth optimization · Bundle methods

1 Introduction

In this work, we are concerned with local optimization methods to address the following class of nonsmooth and nonconvex problems:

$$\min_{x \in X} f(x) \quad \text{with} \quad f(x) := f_1(x) - f_2(x), \quad (1)$$

where $\emptyset \neq X \subset \mathbb{R}^n$ is a simple closed convex set (typically, but not necessarily a polyhedral one) contained in an open set $\Omega \subset \mathbb{R}^n$, and $f_1 : \Omega \rightarrow \mathbb{R}$ and $f_2 : \Omega \rightarrow \mathbb{R}$ are both convex and possibly nonsmooth functions. The mapping f is the *difference of two convex functions* and it is called a *DC function* while f_1 and f_2 are its *DC components*. Accordingly, problem (1) is a convex-constrained DC program.

The past few years have witnessed a substantial development in the area of DC programming. This class of problems forms an important sub-field of nonconvex programming and has been receiving much attention from the mathematical programming community

✉ Wellington de Oliveira
wellington.oliveira@mines-paristech.fr

¹ MINES ParisTech, PSL – Research University, CMA – Centre de Mathématiques Appliquées, Sophia Antipolis, France

[14,15,22,27,31,32,38,41–43]. We refer to [43, Part II] for a comprehensive presentation of several algorithms designed to find global solutions. Yet, local optimization methods play an important role in global optimization because algorithms of the latter class typically employ local methods to find stationary/critical points that feed a certain search strategy for global solutions. A non-exhaustive list of applications of DC programming fitting the above formulation includes production–transportation planning problems [23], location planning [38], cluster analysis [3,29], sensor covering [1], and engineering problems [15,43].

A well-known method for dealing with the optimization problem (1) is the *DC Algorithm*—DCA—of [31,42], which iteratively linearizes the second DC component yielding convex optimization subproblems that are solved to define trial points. Another important algorithm for DC programming is the *Proximal Linearized Method*—PLM—[7,38,41], which can be seen as a regularized variant of DCA: the convex optimization subproblem is augmented with a Bregman function to prevent tailing-off effect that makes calculations unstable as the iteration process progresses.

The main disadvantage of both DCA and PLM is the need of solving exactly a convex nonsmooth program per iteration. As an attempt to overcome this issue, the recent work [41] investigates an inexact version of PLM by requiring the convex subproblems to be asymptotically solved up to optimality and subgradients of f_1 satisfying a certain proximity property with respect to the previously computed subgradient of f_2 [41, Equation (15)].

In this work, we propose more sophisticated and implementable variants of PLM for handling problem (1). The new algorithms belong to the bundle method family [21, Chapter XV], a class of methods proposed by C. Lemaréchal in the 70s [33]. Bundle methods are among the most efficient algorithms for solving nonsmooth *convex* optimization problems. This class of methods constitutes a very active area of research in the nonsmooth optimization community [8–10,12,13,30,48]. Extensions of proximal bundle algorithms to nonsmooth nonconvex programs have been investigated by different authors in [14,17,18,36,37] and references therein.

As far as the DC setting is concerned, the insightful paper [27] proposes a proximal bundle method for finding critical points of unconstrained DC programs, i.e., problem (1) with $X = \mathbb{R}^n$. The authors consider a DC piecewise linear model approximating the objective function f and compute trial points by globally minimizing such a model over \mathbb{R}^n : this task amounts to solving a fixed number of quadratic programs (QPs) per iteration. In general terms, the number of QPs solved per iteration increases with the DC model's size [27, p. 514].

The recent paper [16] also proposes a bundle method employing a DC piecewise model to deal with unconstrained DC programs. Differently from [27] that globally minimizes the DC model to define new iterates, the DC model in [16] is tackled by means of two auxiliary QPs that have different local approximation properties. These quadratic programs must be solved at every iteration to compute descent directions in which a line-search is performed to define trial points.

Inspired by these methods for unconstrained DC programs, we propose in this work two proximal bundle methods for finding critical and d (irectional)-stationary points (see definitions in Sect. 2 below) of DC problems (1). The given variants relate to both proximal linearized methods of [7,41] and proximal bundle algorithms of [16,27]. In contrast to a more difficult (but possibly better) nonconvex approximation of f , we employ a simple convex piecewise linear model. No line-search nor estimates of Lipschitz constants of the DC component are required. Moreover, both variants employ a reliable straightforward stopping test. We care to mention that the master program defining trial points in our algorithms is

not necessarily a QP problem but a more general strictly convex program exploiting possible compelling structures of the feasible set X . This is computationally useful when X has a particular structure (for instance a ball, a simplex, a spectahedron, and other domains [4, Sect. 2.3]) and a specialized solver for handling the master problem is at disposal. We assume throughout this paper that such a specialized solver is available.

As standard in the DC literature, our first algorithm is shown to generate a subsequence of points that converges to a critical point of (1). Under the hypothesis that f_2 in (1) is the pointwise maximum of N known differentiable functions our second algorithm is ensured to compute a d -stationary point, which is the sharpest stationarity definition in DC programming [38, Sect. 3.2]. The price to obtain this stronger result is the solution of possibly several (but no more than N) master programs at certain iterations. We care to mention that the bundle method of [26] is ensured to compute Clarke stationary points of unconstrained DC programs. To this end, the authors of [26] assume that the subdifferentials of the DC components at any point $x \in \mathbb{R}^n$ are polytopes. This is, for instance, the case when both f_1 and f_2 are the pointwise maximum of finitely many differentiable functions. Clarke stationarity is a stronger property than criticality, but weaker than d -stationarity [38].

The remainder of this work is organized as follows: some well-known results and properties of DC programming are recalled in Sect. 2. The first proximal bundle method algorithm for finding critical points is presented in Sect. 3. Its convergence analysis is given in Sect. 4. The second proximal bundle algorithm for finding d -stationary points of a particular class of problem (1) is introduced in Sect. 5 together with its convergence analysis. We report some numerical experiments in Sects. 6, and 7 closes the paper with some final remarks.

2 Basic concepts and properties of DC programming

As mentioned in [31], DC programming is an extension of convex programming that is vast enough to cover almost all nonconvex optimization problems, but still allows the use of powerful tools from convex analysis and convex optimization. As in nonconvex nonsmooth optimization, many definitions of stationary points exist. Below we list some of them.

A point $\bar{x} \in X$ is called a *local minimizer* of problem (1) if there exists a neighborhood $V \subset X$ of \bar{x} such that $f(\bar{x}) \leq f(x)$ for all $x \in V$. As convex functions are directionally differentiable in the interior of their domains [6, Proposition 2.2.7], the *directional derivative*

$$f'_i(x; d) := \lim_{t \downarrow 0} \frac{f_i(x + td) - f_i(x)}{t}$$

for the DC component f_i , $i = 1, 2$, is well defined for all x in the open set $\Omega \subset \text{dom } f_i$ and all $d \in \mathbb{R}^n$. It is well known that $f'_i(x; d) = \max_{g \in \partial f_i(x)} \langle g, d \rangle$, where

$$\partial f_i(x) := \{g \in \mathbb{R}^n : f_i(y) \geq f_i(x) + \langle g, y - x \rangle \quad \forall y \in \Omega\}$$

is the *subdifferential* of f_i at point x . For $\epsilon \geq 0$, the *inexact subdifferential* is denoted by

$$\partial_\epsilon f_i(x) := \{g \in \mathbb{R}^n : f_i(y) \geq f_i(x) + \langle g, y - x \rangle - \epsilon \quad \forall y \in \Omega\}.$$

Since DC components are locally Lipschitz continuous, DC functions satisfy also local Lipschitz continuity. Thus, the directional and Clarke directional derivatives of a DC function are well defined for all $x \in \Omega$:

$$\begin{aligned} \text{(directional derivative)} \quad & f'(x; d) = f'_1(x; d) - f'_2(x; d) \\ \text{(Clarke directional derivative)} \quad & f^0(x; d) := \limsup_{t \downarrow 0} \sup_{y \rightarrow x} \frac{f(y+td) - f(y)}{t}. \end{aligned}$$

It can be shown that if $\bar{x} \in X \subset \Omega$ is a local minimizer of f over X , then $\bar{x} \in X$ is a *d(directional)-stationary point*, i.e., $f'(\bar{x}; (x - \bar{x})) \geq 0$ for all $x \in X$. For DC programs this definition is equivalent to

$$f'_1(\bar{x}; (x - \bar{x})) \geq f'_2(\bar{x}; (x - \bar{x})) \quad \forall x \in X.$$

The above inequality is equivalent to $f'_1(\bar{x}; (x - \bar{x})) \geq \max_{g_2 \in \partial f_2(\bar{x})} \langle g_2, x - \bar{x} \rangle$ for all $x \in X$. In other words, $\bar{x} \in X$ is a *d-stationary point* of (1) if for all $g_2 \in \partial f_2(\bar{x})$

$$f'_1(\bar{x}; (x - \bar{x})) \geq \langle g_2, x - \bar{x} \rangle \quad \forall x \in X.$$

It follows from convexity of f_1 that $\bar{x} \in X$ satisfies the above inequality if and only if

$$\bar{x} \in \arg \min_{x \in X} [f_1(x) - \langle g_2, x \rangle] \quad \forall g_2 \in \partial f_2(\bar{x}).$$

This shows that $\bar{x} \in X$ is a *d-stationary point* of (1) if

$$\partial f_2(\bar{x}) \subseteq \partial f_1(\bar{x}) + N_X(\bar{x}) \quad (= \partial[f_1(\bar{x}) + i_X(\bar{x})]), \tag{2}$$

where $N_X(\bar{x})$ is the normal cone of X at \bar{x} and i_X is the *indicator function* of set X . The equality $\partial f_1(\bar{x}) + N_X(\bar{x}) = \partial[f_1(\bar{x}) + i_X(\bar{x})]$ follows from convexity of $X \subset \Omega \subset \text{dom}(f_1)$ and the convexity of f_1 [40]. Notice that verifying the above characterization of *d-stationarity* is impossible in many cases of interest. Hence, one generally employs a weaker notion of stationarity: a point $\bar{x} \in X$ is called a *critical point* of problem (1) if

$$\emptyset \neq \partial f_2(\bar{x}) \cap \partial[f_1(\bar{x}) + i_X(\bar{x})], \quad \text{or equivalently} \quad \emptyset \neq \partial f_2(\bar{x}) \cap \{\partial f_1(\bar{x}) + N_X(\bar{x})\}. \tag{3}$$

Another useful notion is *Clarke stationarity*: a point $\bar{x} \in X$ is a *c(larke)-stationary point* of problem (1) if

$$f^0(\bar{x}; (x - \bar{x})) \geq 0 \quad \forall x \in X.$$

It follows from the inequality $f^0(x; d) \geq f'(x; d)$ that every *d-stationary point* is also *c-stationary*. The reverse implication is not always true [38, Example 2]. The following sequence of implications related to problem (1) can be found in the DC programming literature (see for instance [16,22,26,31,38,43]):

$$\text{local minimizer} \Rightarrow d - \text{stationarity} \Rightarrow c - \text{stationarity} \Rightarrow \text{criticality}.$$

If f_1 is a continuously differentiable function on Ω , then

$$c - \text{stationarity} \Leftrightarrow \text{criticality}.$$

If f_2 is a continuously differentiable function on Ω , then

$$d - \text{stationarity} \Leftrightarrow c - \text{stationarity} \Leftrightarrow \text{criticality}.$$

If f_2 is a polyhedral function $f_2(x) = \max_{i=1, \dots, N} \{ \langle a_i, x \rangle + b_i \}$ (where $a_i \in \mathbb{R}^n$ and $b_i \in \mathbb{R}$), then $\bar{x} \in X$ is a local minimizer if and only if \bar{x} is *d-stationary point* [31, Theorem 1]. Moreover,

$$\begin{aligned} \min_{x \in X} [f_1(x) - f_2(x)] &= \min_{x \in X} [f_1(x) - \max_{i=1, \dots, N} \{ \langle a_i, x \rangle + b_i \}] \\ &= \min_{x \in X} [f_1(x) + \min_{i=1, \dots, N} - \{ \langle a_i, x \rangle + b_i \}] \\ &= \min_{i=1, \dots, N} [\min_{x \in X} \{ f_1(x) - \langle a_i, x \rangle - b_i \}], \end{aligned}$$

showing that a global solution of the DC program $\min_{x \in X} [f_1(x) - f_2(x)]$ can be obtained by solving N convex programs $\min_{x \in X} [f_1(x) - \langle a_i, x \rangle] - b_i$.

3 Proximal bundle method for convex constrained DC programming

Let $\omega : \Omega \rightarrow \mathbb{R}$ be a twice differentiable and strongly convex function on X with parameter $\rho > 0$, w.r.t. the norm $\| \cdot \|_p$ ($p \in [1, \infty]$), that is

$$\omega(x) \geq \omega(y) + \langle \nabla \omega(y), x - y \rangle + \frac{\rho}{2} \|x - y\|_p^2 \quad \forall x, y \in X. \tag{4}$$

We can rewrite the DC function $f = f_1 - f_2$ in (1) for a given ω as

$$f(x) = f_1(x) + \omega(x) - [f_2(x) + \omega(x)].$$

Let $y \in X$, $g_2 \in \partial f_2(y)$ and $\nabla \omega(y)$ be given. We can then overestimate f by the following convex function

$$\begin{aligned} f(x) &\leq f_1(x) + \omega(x) - [f_2(y) + \omega(y) + \langle g_2 + \nabla \omega(y), x - y \rangle] \\ &= f_1(x) - f_2(y) - \langle g_2, x - y \rangle + D(x, y), \end{aligned}$$

where $D(\cdot, \cdot)$ is the *Bregman function*

$$D(x, y) := \omega(x) - [\omega(y) + \langle \nabla \omega(y), x - y \rangle] \left(\geq \frac{\rho}{2} \|x - y\|_p^2 \right). \tag{5}$$

It is shown in [7] that the following Proximal Linearized Method generates a sequence of trial points x^{k+1} whose cluster points (if any) are critical to problem (1).

PROXIMAL LINEARIZED ALGORITHM WITH A BREGMAN FUNCTION

Initialization: choose $x^0 \in X$ and compute $g_2^0 \in \partial f_2(x^0)$
Repeat: for $k = 0, 1, 2, \dots$

$$\begin{aligned} x^{k+1} &= \arg \min_{x \in X} [f_1(x) - \langle g_2^k, x - x^k \rangle + D(x, x^k)] \\ g_2^{k+1} &\in \partial f_2(x^{k+1}) \end{aligned} \tag{6}$$

Stopping test: If $x^{k+1} = x^k$ stop.

The work [41] studies the above algorithm with the standard choice $\omega(\cdot) = \| \cdot \|_2^2/2$ (Euclidean norm). With this same regularizing function the authors of [38] propose two algorithms (akin to the above one) to find d -stationary points of (1). Both results in [41] and [38] can be extended to a more general strongly convex function ω without much difficulty [7].

Notice that the above proximal algorithm requires solving exactly a constrained convex nonsmooth program per iteration. This can be a difficult task when dealing with real-life DC programs, mainly if f_1 is only assessed via a black-box/oracle. In order to accelerate the optimization process, the authors of [41] investigate an inexact version of the above algorithm with $\omega(\cdot) = \| \cdot \|_2^2/2$. In what follows we propose two proximal bundle algorithms that do not require solving exactly subproblem (6) and differently from [41] no assumption on the computed subgradient $g_1^{k+1} \in \partial f_1(x^{k+1})$ is made. The first of these methods yields critical points whereas the second algorithm presented in Sect. 5 finds d -stationary points under the more restrictive assumption that f_2 is the pointwise maximum of finitely many convex differentiable functions.

3.1 A proximal bundle method for DC programs

Let k denote an iteration counter and let $g_1^j \in \partial f_1(x^j)$ and $g_2^j \in \partial f_2(x^j)$ be subgradients of the DC components calculated during an iteration $j \in \{0, 1, 2, \dots\}$. Convexity of f_1 implies that the linearization

$$\bar{f}_1^j(x) := f_1(x^j) + \langle g_1^j, x - x^j \rangle$$

approximates $f_1(x)$ from below. As a result, we can construct a *cutting-plane model*

$$\check{f}_1^k(x) := \max_{j \in \mathcal{B}_1^k} \bar{f}_1^j(x) \leq f_1(x) \quad \forall x \in \Omega, \tag{7}$$

where $\mathcal{B}_1^k \subset \{0, 1, \dots, k\}$ is the index set containing the *bundle of information* of f_1 . By following the general ideas of bundle methods we replace f_1 in the master program (6) with its cutting-plane model \check{f}_1^k . Since \check{f}_1^k can be a rough approximation of f_1 at certain iterations k , the trial point x^{k+1} obtained from (6) with f_1 replaced by \check{f}_1^k can be far away from the solution of (6). In order to diminish the impact of coarse approximations of f_1 along the iterative process, we shall regularize the resulting master program by keeping trial points near to a certain stability center $x^{k(\ell)} \in X$, where the index ℓ counts the number of times that such a center has been updated and $k(\ell)$ is the iteration in which the center is obtained. The stability center $x^{k(\ell)}$ is some previous iterate, usually the “best” point generated by the iterative process so far. Accordingly, we replace subproblem (6) with

$$\min_{x \in X} \check{f}_1^k(x) - \langle g_2^{k(\ell)}, x - x^{k(\ell)} \rangle + \mu_k D(x, x^{k(\ell)}), \tag{8}$$

where $\mu_k > 0$ is a prox-parameter determining the influence of the Bregman function D on the next trial point x^{k+1} . In terms of optimal solution x^{k+1} subproblem (8) is equivalent to

$$\begin{cases} \min_{x \in X, r \in \mathbb{R}} r + \mu_k \omega(x) - \langle g_2^{k(\ell)} + \mu_k \nabla \omega(x^{k(\ell)}), x \rangle \\ \text{s.t.} \quad \bar{f}_1^j(x) \leq r, \quad j \in \mathcal{B}_1^k. \end{cases} \tag{9}$$

Notice that if X is a polyhedron and $\omega(\cdot) = \|\cdot\|_2^2/2$, then $D(x, x^{k(\ell)}) = \|x - x^{k(\ell)}\|_2^2/2$ and subproblem (9) is a convex quadratic problem (QP).

Proposition 1 *Given a stability center $x^{k(\ell)} \in X$ and a prox-parameter $\mu_k > 0$, let x^{k+1} be the unique solution of subproblem (9). Assume that either X is polyhedral or an appropriate constraint qualification [21] holds in (9), then there exist $s^{k+1} \in N_X(x^{k+1})$ and $\alpha_j \geq 0$ with $\sum_{j \in \mathcal{B}_1^k} \alpha_j = 1$ such that $\sum_{j \in \mathcal{B}_1^k} \alpha_j g_1^j := p^{k+1} \in \partial \check{f}_1^k(x^{k+1})$ and*

$$p^{k+1} + s^{k+1} - g_2^{k(\ell)} + \mu_k \left(\nabla \omega(x^{k+1}) - \nabla \omega(x^{k(\ell)}) \right) = 0. \tag{10}$$

Moreover, the aggregate linearization

$$\bar{f}_1^{-k}(x) := \check{f}_1^k(x^{k+1}) + \langle p^{k+1}, x - x^{k+1} \rangle \text{ satisfies } \bar{f}_1^{-k}(x) \leq f_1(x) \text{ for all } x \in \mathbb{R}^n. \tag{11}$$

Proof The assumption on X ensures the existence of Lagrange multipliers $\alpha_j \geq 0$ associated to the constraints $\bar{f}_1^j(x) = f_1(x^j) + \langle g_1^j, x - x^j \rangle \leq r, j \in \mathcal{B}_1^k$. Hence, the optimality conditions of (9) read as:

$$-\left(-g_2^{k(\ell)} + \mu_k \begin{pmatrix} \nabla \omega(x^{k+1}) - \nabla \omega(x^{k(\ell)}) \\ 1 \end{pmatrix} \right) - \sum_{j \in \mathcal{B}_1^k} \alpha_j \begin{pmatrix} g_1^j \\ -1 \end{pmatrix} \in \begin{pmatrix} N_X(x^{k+1}) \\ 0 \end{pmatrix}.$$

The above inclusion implies that $\sum_{j \in \mathcal{B}_1^k} \alpha_j = 1$ and that there exists $s^{k+1} \in N_X(x^{k+1})$ such that $s^{k+1} = g_2^{k(\ell)} - \mu_k(\nabla\omega(x^{k+1}) - \nabla\omega(x^{k(\ell)})) - \sum_{j \in \mathcal{B}_1^k} \alpha_j g_1^j$, which is exactly (10) with $p^{k+1} = \sum_{j \in \mathcal{B}_1^k} \alpha_j g_1^j$. Note that p^{k+1} is a convex combination of active subgradients of \check{f}_1^k at x^{k+1} . As a result, $p^{k+1} \in \partial \check{f}_1^k(x^{k+1})$ [5, Lemma 10.8]. The inequality $\bar{f}_1^{-k}(x) \leq f_1(x)$ holds because $p^{k+1} \in \partial \check{f}_1^k(x^{k+1})$ and $\check{f}_1^k(\cdot) \leq f_1(\cdot)$. \square

If $X = \mathbb{R}^n$ and $\omega(x) = \|x\|_2^2/2$ then the solution x^{k+1} of (8) is, from Eq. (10), $x^{k+1} = x^{k(\ell)} + \frac{1}{\mu_k}(g_2^{k(\ell)} - \sum_{j \in \mathcal{B}_1^k} \alpha_j g_1^j)$. Furthermore, the Lagrange multipliers α_j (with $j \in \mathcal{B}_1^k$) can be obtained by solving the dual QP of (8), that has dimension $|\mathcal{B}_1^k|$ (see [5, Lemma 10.8]). Therefore, by keeping the size of \mathcal{B}_1^k bounded we also keep the method’s memory (number of elements in the bundle) limited.

Once the trial point x^{k+1} is computed by a specialized solver (for QP, quadratically constrained QP, conic programming, etc.) a classification rule decides when to update $x^{k(\ell)}$. For a given $\kappa \in (0, 1)$ and a lower bound $\underline{\mu} > 0$ of the prox-parameter μ_k , a possible rule is as follows: if

$$f(x^{k+1}) \leq f(x^{k(\ell)}) - \kappa \underline{\mu} D(x^{k+1}, x^{k(\ell)}) \tag{12}$$

then a *serious step* is performed and we set $x^{k(\ell+1)} := x^{k+1}$ and $\ell := \ell + 1$. Otherwise, a *null step* is performed and both stability center and counter ℓ remain unchanged. A more economical rule in terms of evaluations of f_2 is to test the condition

$$f_1(x^{k+1}) \leq f_1(x^{k(\ell)}) + \langle g_2^{k(\ell)}, x^{k+1} - x^{k(\ell)} \rangle - \kappa \underline{\mu} D(x^{k+1}, x^{k(\ell)}). \tag{13}$$

If it holds we do a serious step. Otherwise we perform a null step. Note that (13) takes into account only the first DC component f_1 and not the DC function f . Moreover, the above inequality does not necessarily imply that $f_1(x^{k+1}) \leq f_1(x^{k(\ell)})$ due to term $\langle g_2^{k(\ell)}, x^{k+1} - x^{k(\ell)} \rangle$. However, when (13) holds the DC function is decreased by an amount of at least $\kappa \underline{\mu} D(x^{k(\ell+1)}, x^{k(\ell)})$.

Lemma 1 *If (13) holds then $f(x^{k(\ell)}) \geq f(x^{k(\ell+1)}) + \kappa \underline{\mu} D(x^{k(\ell+1)}, x^{k(\ell)})$.*

Proof When inequality (13) is satisfied we obtain $x^{k(\ell+1)} = x^{k+1}$ and thus

$$f_1(x^{k(\ell)}) \geq f_1(x^{k(\ell+1)}) - \langle g_2^{k(\ell)}, x^{k(\ell+1)} - x^{k(\ell)} \rangle + \kappa \underline{\mu} D(x^{k(\ell+1)}, x^{k(\ell)}).$$

Convexity of f_2 yields $f_2(x^{k(\ell+1)}) \geq f_2(x^{k(\ell)}) + \langle g_2^{k(\ell)}, x^{k(\ell+1)} - x^{k(\ell)} \rangle$. By summing these two inequalities we obtain

$$f_1(x^{k(\ell)}) + f_2(x^{k(\ell+1)}) \geq f_1(x^{k(\ell+1)}) + f_2(x^{k(\ell)}) + \kappa \underline{\mu} D(x^{k(\ell+1)}, x^{k(\ell)}),$$

i.e., $f_1(x^{k(\ell)}) - f_2(x^{k(\ell)}) \geq f_1(x^{k(\ell+1)}) - f_2(x^{k(\ell+1)}) + \kappa \underline{\mu} D(x^{k(\ell+1)}, x^{k(\ell)})$ as stated. \square

We are now in the position to present our first algorithm, which makes use of (13) to update stability centers. As a result, there is no need to evaluate f_2 at x^{k+1} : only subgradients of f_2 are computed during serious steps, i.e., when (13) is satisfied. During null steps (when (13) does not hold) only the first DC component f_1 needs to be assessed through an oracle that returns at x^{k+1} the value of the function and one of its subgradients. A similar algorithm can be stated with rule (12) instead, but with evaluations of f_2 at every iteration.

Algorithm 1 PROXIMAL BUNDLE ALGORITHM FOR DC PROGRAMMING

- 1: Let $x^0 \in X, \kappa \in (0, 1), 0 < \underline{\mu} \leq \mu_0 \leq \bar{\mu} < \infty$ and a tolerance $\delta_{\text{Tot}} \geq 0$ be given. ▷ Initialization
- 2: Compute $(f_1(x^0), g_1^0 \in \partial f_1(x^0)), g_2^0 \in \partial f_2(x^0)$, and define $\mathcal{B}_1^0 := \{0\}$ and $k(0) := \ell := 0$. ▷ Loop
- 3: **for** $k = 0, 1, 2, \dots$ **do**
- 4: Solve (9) to obtain x^{k+1} and the Lagrange multipliers $\alpha_j \geq 0$ ▷ Stopping test
- 5: **if** $\|x^{k+1} - x^{k(\ell)}\|_2 \leq \delta_{\text{Tot}}$ **then**
- 6: Stop and return $x^{k(\ell)}$
- 7: **end if** ▷ Oracle for f_1
- 8: Call the oracle to compute $(f_1(x^{k+1}), g_1^{k+1} \in \partial f_1(x^{k+1}))$
- 9: **if** (13) holds true **then**
- 10: Compute $g_2^{k+1} \in \partial f_2(x^{k+1})$ ▷ Serious step
- 11: Define $x^{k(\ell+1)} := x^{k+1}, k(\ell + 1) := k + 1$ and $\ell := \ell + 1$
- 12: Choose $\mathcal{B}_1^{k+1} \supset \{k + 1\}$ and $\mu_{k+1} \in [\underline{\mu}, \mu_k]$
- 13: **else**
- 14: Choose $\mathcal{B}_1^{k+1} \supset \{-k, k(\ell), k + 1\}$ with $-k$ given in (11) and $\mu_{k+1} \in [\mu_k, \bar{\mu}]$ ▷ Null step
- 15: **end if**
- 16: **end for**

After a null step, the bundle of information \mathcal{B}_1^k can have as few as three linearizations: the new linearization calculated at x^{k+1} , the linearization formed at the current stability center $x^{k(\ell)}$ and the aggregate linearization \tilde{f}_1^{-k} defined in Proposition 1. Right after a serious step the bundle size can be reset to only one linearization, as is standard in proximal bundle methods for convex optimization. Notice also that the proximal parameter μ_k is forbidden to decrease after a null step. This is crucial to prove convergence of the algorithm, which stops when the next iterate approximately coincides with the current stability center. This is a cheap and reliable stopping test as shown in Theorem 1 below.

4 Convergence analysis

Let $\mathcal{L} \subset \{0, 1, 2, \dots\}$ denote the index set gathering the serious steps: $\ell \in \mathcal{L}$ implies that $x^{k(\ell)}$ is the ℓ th stability center. Throughout this section we will use the notation $i(\ell) = k(\ell + 1) - 1$ to refer to the algorithm’s iteration yielding a serious step. Then for such iterations subproblem (8) reads as

$$x^{k(\ell+1)} = \arg \min_{x \in X} \tilde{f}_1^{i(\ell)}(x) - \langle g_2^{k(\ell)}, x - x^{k(\ell)} \rangle + \mu_{i(\ell)} D(x, x^{k(\ell)}).$$

Our goal is to show that any cluster point $\bar{x} \in X$ of the sequence of stability centers $\{x^{k(\ell)}\}_\ell$ generated by Algorithm 1 is a critical point of (1), i.e., a point satisfying (3).

We start with the following result showing that if the algorithm performs only finitely many steps, then the last stability center is a critical point if $\delta_{\text{Tot}} = 0$.

Lemma 2 Assume that $\delta_{\text{Tot}} = 0$ and suppose that Algorithm 1 stops at iteration k . Then the last stability center $x^{k(\ell)}$ is a critical point of (1).

Proof Convexity of f_1 and feasibility of $x^{k(\ell)}$ imply that

$$\begin{aligned} \min_{x \in X} \check{f}_1^k(x) - \langle g_2^{k(\ell)}, x - x^{k(\ell)} \rangle + \mu_k D(x, x^{k(\ell)}) &\leq \\ \min_{x \in X} f_1(x) - \langle g_2^{k(\ell)}, x - x^{k(\ell)} \rangle + \mu_k D(x, x^{k(\ell)}) &\leq f_1(x^{k(\ell)}). \end{aligned} \tag{14}$$

In addition, the point x^{k+1} solves the first subproblem (8) and the optimal value of this problem is $\check{f}_1^k(x^{k+1}) - \langle g_2^{k(\ell)}, x^{k+1} - x^{k(\ell)} \rangle + \mu_k D(x^{k+1}, x^{k(\ell)})$. Hence, if the algorithm stops at iteration k we have that $x^{k+1} = x^{k(\ell)}$ and

$$\check{f}_1^k(x^{k+1}) - \langle g_2^{k(\ell)}, x^{k+1} - x^{k(\ell)} \rangle + \mu_k D(x^{k+1}, x^{k(\ell)}) = \check{f}_1^k(x^{k(\ell)}) = f_1(x^{k(\ell)}),$$

where the last equality follows from the assumption that $k(\ell) \in \mathcal{B}_1^k$. We have thus shown that

$$f_1(x^{k(\ell)}) \leq \min_{x \in X} f_1(x) - \langle g_2^{k(\ell)}, x - x^{k(\ell)} \rangle + \mu_k D(x, x^{k(\ell)}) \leq f_1(x^{k(\ell)}),$$

i.e., the point $x^{k+1} = x^{k(\ell)}$ also solves $\min_{x \in X} f_1(x) - \langle g_2^{k(\ell)}, x - x^{k(\ell)} \rangle + \mu_k D(x, x^{k(\ell)})$. The optimality condition of this reads as

$$0 \in \partial f_1(x^{k+1}) - g_2^{k(\ell)} + \mu_k \left(\nabla \omega(x^{k+1}) - \nabla \omega(x^{k(\ell)}) \right) + N_X(x^{k+1}).$$

Since $x^{k+1} = x^{k(\ell)}$ the above inclusion is equivalent to $g_2^{k(\ell)} \in \partial f_1(x^{k(\ell)}) + N_X(x^{k(\ell)})$, which gives (3) because $g_2^{k(\ell)} \in \partial f_2(x^{k(\ell)})$. □

From now on we assume that $\delta_{\text{Tot}} = 0$ and that the algorithm loops indefinitely.

4.1 Infinitely many serious steps

In what follows we assume that the algorithm generates infinitely many serious steps, i.e., $|\mathcal{L}| = \infty$. The following result shows that $p^{k(\ell+1)} + s^{k(\ell+1)}$ defined in Proposition 1 is an approximate subgradient of the function $f_1(x) + i_X(x)$ at the point $x = x^{k(\ell+1)}$.

Lemma 3 *Let $k(\ell + 1)$ be the iteration index of the $(\ell + 1)$ th stability center and $i(\ell) = k(\ell + 1) - 1$ be the iteration index in which $x^{k(\ell+1)}$ is determined. Assume $\mu_k \leq \bar{\mu} < \infty$ and that $\{x^{k(\ell)}\}_{\ell \in \mathcal{L}}$ is a bounded sequence. Let $p^{k(\ell+1)} \in \partial \check{f}_1^{i(\ell)}(x^{k(\ell+1)})$ and $s^{k(\ell+1)} \in N_X(x^{k(\ell+1)})$ be as in (10), and denote $\beta^{k(\ell+1)} := p^{k(\ell+1)} + s^{k(\ell+1)}$. Then there exist constants $M, L > 0$ such that*

$$\|\beta^{k(\ell)}\|_2 \leq M \quad \text{and} \quad \|g_1^{k(\ell)}\|_2 \leq L \text{ for all } \ell \in \mathcal{L}.$$

Moreover, $f_1(x^{k(\ell+1)}) \geq \check{f}_1^{i(\ell)}(x^{k(\ell+1)}) \geq f_1(x^{k(\ell)}) - L \|x^{k(\ell+1)} - x^{k(\ell)}\|_2$ and

$$\beta^{k(\ell+1)} \in \partial_{e_{i(\ell)}} [f_1(x^{k(\ell)}) + i_X(x^{k(\ell)})] \quad \text{with} \quad e_{i(\ell)} = (M + L) \|x^{k(\ell)} - x^{k(\ell+1)}\|_2.$$

Proof It follows from (10) and the triangular inequality that

$$\begin{aligned} \|p^{k(\ell+1)} + s^{k(\ell+1)}\|_2 &= \|g_2^{k(\ell)} + \mu_k \left(\nabla \omega(x^{k(\ell)}) - \nabla \omega(x^{k(\ell+1)}) \right)\|_2 \leq \|g_2^{k(\ell)}\|_2 \\ &\quad + \bar{\mu} \|\nabla \omega(x^{k(\ell)}) - \nabla \omega(x^{k(\ell+1)})\|_2. \end{aligned}$$

Since $\{x^{k(\ell)}\}_{\ell \in \mathcal{L}}$ is bounded and the DC components f_1 and f_2 are finite-valued convex functions on $\Omega \supset X$, then [20, Theorem 3.1.2] ensures that $\{g_1^{k(\ell)}\}_{\ell \in \mathcal{L}}$ and $\{g_2^{k(\ell)}\}_{\ell \in \mathcal{L}}$ are

bounded sequences as well. In particular, there exists $L > 0$ such that $\|g_1^{k(\ell)}\|_2 \leq L$ for all $\ell \in \mathcal{L}$. Moreover, the existence of a finite constant $M > 0$ bounding $\|p^{k(\ell+1)} + s^{k(\ell+1)}\|_2$ is ensured because $\bar{\mu} < \infty$ and $\nabla\omega$ is a continuous map.

Recall that $N_X(x^{k(\ell+1)}) = \partial i_X(x^{k(\ell+1)})$ because $x^{k(\ell+1)} \in X$. Then $s^{k(\ell+1)} \in \partial i_X(x^{k(\ell+1)})$ and therefore $i_X(x) \geq i_X(x^{k(\ell+1)}) + \langle s^{k(\ell+1)}, x - x^{k(\ell+1)} \rangle$ for all $x \in \mathbb{R}^n$. Since $p^{k(\ell+1)} \in \partial \check{f}_1^{i(\ell)}(x^{k(\ell+1)})$ we get

$$\begin{aligned} f_1(x) + i_X(x) &\geq \check{f}_1^{i(\ell)}(x) + i_X(x) \\ &\geq [\check{f}_1^{i(\ell)}(x^{k(\ell+1)}) + \langle p^{k(\ell+1)}, x - x^{k(\ell+1)} \rangle] \\ &\quad + [i_X(x^{k(\ell+1)}) + \langle s^{k(\ell+1)}, x - x^{k(\ell+1)} \rangle] \\ &= \check{f}_1^{i(\ell)}(x^{k(\ell+1)}) + i_X(x^{k(\ell+1)}) + \langle \beta^{k(\ell+1)}, x - x^{k(\ell+1)} \rangle. \end{aligned}$$

As $x^{k(\ell)} \in X$ for all ℓ , then $i_X(x^{k(\ell+1)}) = 0$. By utilizing this in the above inequality we obtain

$$\begin{aligned} f_1(x) + i_X(x) &\geq \check{f}_1^{i(\ell)}(x^{k(\ell+1)}) + \langle \beta^{k(\ell+1)}, x - x^{k(\ell+1)} \rangle \\ &= f_1(x^{k(\ell)}) + \langle \beta^{k(\ell+1)}, x - x^{k(\ell)} \rangle + \\ &\quad + \langle \beta^{k(\ell+1)}, x^{k(\ell)} - x^{k(\ell+1)} \rangle + \check{f}_1^{i(\ell)}(x^{k(\ell+1)}) - f_1(x^{k(\ell)}), \end{aligned}$$

which gives by the Cauchy–Schwarz inequality

$$\begin{aligned} f_1(x) + i_X(x) &\geq f_1(x^{k(\ell)}) + \langle \beta^{k(\ell+1)}, x - x^{k(\ell)} \rangle \\ &\quad - M \|x^{k(\ell)} - x^{k(\ell+1)}\|_2 + \check{f}_1^{i(\ell)}(x^{k(\ell+1)}) - f_1(x^{k(\ell)}). \end{aligned} \tag{15}$$

Note that Algorithm 1 keeps in the bundle the index $k(\ell)$ of the current stability center. Therefore,

$$\check{f}_1^{i(\ell)}(x^{k(\ell+1)}) = \max_{j \in \mathcal{B}_1^{i(\ell)}} \{f_1(x^j) + \langle g_1^j, x^{k(\ell+1)} - x^j \rangle\} \geq f_1(x^{k(\ell)}) + \langle g_1^{k(\ell)}, x^{k(\ell+1)} - x^{k(\ell)} \rangle.$$

Again by the Cauchy–Schwarz inequality we get $\check{f}_1^{i(\ell)}(x^{k(\ell+1)}) \geq f_1(x^{k(\ell)}) - L \|x^{k(\ell+1)} - x^{k(\ell)}\|_2$. Thus it follows from (15) that (because $i_X(x^{k(\ell)}) = 0$)

$$\begin{aligned} f_1(x) + i_X(x) &\geq f_1(x^{k(\ell)}) + \langle \beta^{k(\ell+1)}, x - x^{k(\ell)} \rangle - (M + L) \|x^{k(\ell)} - x^{k(\ell+1)}\|_2 \\ &= f_1(x^{k(\ell)}) + i_X(x^{k(\ell)}) + \langle \beta^{k(\ell+1)}, x - x^{k(\ell)} \rangle - e_{i(\ell)}. \end{aligned}$$

Since $x \in \mathbb{R}^n$ is an arbitrary point we conclude that $\beta^{k(\ell+1)} \in \partial_{e_{i(\ell)}}[f_1(x^{k(\ell)}) + i_X(x^{k(\ell)})]$. The inequalities $f_1(x^{k(\ell+1)}) \geq \check{f}_1^{i(\ell)}(x^{k(\ell+1)}) \geq f_1(x^{k(\ell)}) - L \|x^{k(\ell+1)} - x^{k(\ell)}\|_2$ follow trivially from the already shown inequality $\check{f}_1^{i(\ell)}(x^{k(\ell+1)}) \geq f_1(x^{k(\ell)}) - L \|x^{k(\ell+1)} - x^{k(\ell)}\|_2$. \square

Suppose that $\{x^{k(\ell)}\}_{\ell \in \mathcal{L}}$ is bounded, $\bar{x} \in X$ is one of its cluster points, and $\|x^{k(\ell+1)} - x^{k(\ell)}\|_2 \rightarrow 0$. Then the above lemma ensures that $\{p^{k(\ell)} + s^{k(\ell)}\}_{\ell \in \mathcal{L}}$ is a bounded sequence and that any cluster point $\bar{\beta}$ of $\{p^{k(\ell)} + s^{k(\ell)}\}_{\ell \in \mathcal{L}}$ satisfies $\bar{\beta} \in \partial[f_1(\bar{x}) + i_X(\bar{x})]$ as a consequence of [21, Proposition 4.1.1]. This crucial property is employed in the following proposition to establish that any cluster point $\bar{x} \in X$ of $\{x^{k(\ell)}\}_{\ell \in \mathcal{L}}$ is a critical point of (1).

Proposition 2 *Assume that the level set $\{x \in X : f(x) \leq f(x^0)\}$ is bounded and that Algorithm 1 performs infinitely many serious steps, i.e., $|\mathcal{L}| = \infty$ and $\ell \rightarrow \infty$. Then any cluster point $\bar{x} \in X$ of the sequence $\{x^{k(\ell)}\}_{\ell \in \mathcal{L}}$ is a critical point of problem (1).*

Proof Lemma 1 shows that $f(x^{k(\ell)}) - f(x^{k(\ell+1)}) \geq \kappa \underline{\mu} D(x^{k(\ell+1)}, x^{k(\ell)})$ and therefore the sequence $\{f(x^{k(\ell)})\}_{\ell \in \mathcal{L}}$ is strictly decreasing, which in turn implies that $\{x^{k(\ell)}\}_{\ell \in \mathcal{L}}$ is a bounded sequence by the assumption of having a bounded level set. Continuity of f ensures that the level set is also closed, and hence compact. The Weierstrass theorem implies that the optimal value of (1) is finite. Therefore, by summing the above inequality with respect to ℓ we get

$$\begin{aligned} \infty > f(x^{k(0)}) - \min_X f(x) &\geq f(x^{k(0)}) - \lim_{\ell \rightarrow \infty} f(x^{k(\ell+1)}) = \sum_{\ell=0}^{\infty} [f(x^{k(\ell)}) - f(x^{k(\ell+1)})] \\ &\geq \kappa \underline{\mu} \sum_{\ell=0}^{\infty} D(x^{k(\ell+1)}, x^{k(\ell)}). \end{aligned}$$

Then it follows from the definition of D in (5) and the equivalence of norms in \mathbb{R}^n that

$$\infty > \sum_{\ell=0}^{\infty} \|x^{k(\ell+1)} - x^{k(\ell)}\|_2^2, \quad \text{and consequently} \quad \lim_{\ell \rightarrow \infty} \|x^{k(\ell+1)} - x^{k(\ell)}\|_2 = 0.$$

Lemma 3 ensures that $\beta^{k(\ell+1)} = p^{k(\ell+1)} + s^{k(\ell+1)} \in \partial_{e_{i(\ell)}} [f_1(x^{k(\ell)}) + i_X(x^{k(\ell)})]$ with $e_{i(\ell)} = (M + L)\|x^{k(\ell+1)} - x^{k(\ell)}\|_2$ for two (possibly unknown) constants $M, L > 0$. Note also that $\lim_{\ell \rightarrow \infty} e_{i(\ell)} = 0$ because $\lim_{\ell \rightarrow \infty} \|x^{k(\ell+1)} - x^{k(\ell)}\|_2 = 0$. Thus, there exist subsets $\mathcal{L}'' \subset \mathcal{L}' \subset \mathcal{L} = \{0, 1, \dots\}$ such that $\{x^{k(\ell)}\}_{\ell \in \mathcal{L}'}$ converges to a point $\bar{x} \in X$ (because $\{x^{k(\ell)}\}_{\ell \in \mathcal{L}}$ is bounded and X is closed), $\{\beta^{k(\ell+1)}\}_{\ell \in \mathcal{L}''}$ converges to a point $\bar{\beta} \in \partial[f_1(\bar{x}) + i_X(\bar{x})]$ (see Proposition 4.1.1 in [21] for more details). In order to show that \bar{x} is a critical point of problem (1), we only need to prove that $\lim_{\ell \in \mathcal{L}''} g_2^{k(\ell)} = \bar{\beta}$. This latter result follows directly from Eq. (10), continuity of $\nabla \omega$ and inequality $\mu_k \leq \bar{\mu} < \infty$:

$$\begin{aligned} \bar{\beta} = \lim_{\ell \in \mathcal{L}''} [p^{k(\ell+1)} + s^{k(\ell+1)}] &= \lim_{\ell \in \mathcal{L}''} g_2^{k(\ell)} \\ &\quad - \lim_{\ell \in \mathcal{L}''} \mu_{i(\ell)} \left(\nabla \omega(x^{k(\ell)}) - \nabla \omega(x^{k(\ell+1)}) \right) = \lim_{\ell \in \mathcal{L}''} g_2^{k(\ell)}, \end{aligned}$$

showing that $\bar{\beta}$ also belongs to $\partial f_2(\bar{x})$. This concludes the proof. □

4.2 Finitely many serious steps (infinitely many null steps)

In this section, we assume that after the $\hat{\ell}$ th-stability center $x^{k(\hat{\ell})} = \hat{x}$ only null steps are performed. Notice that in this case $g_2^{k(\hat{\ell})} = \hat{g}_2$ is fixed and therefore Algorithm 1 behaves exactly as a convex bundle algorithm with the master program given by

$$\min_{x \in X} f_k^{\text{model}}(x) + \mu_k D(x, \hat{x}),$$

where $f_k^{\text{model}}(x) := \check{f}_1^k(x) - \langle \hat{g}_2, x \rangle$ is a cutting-plane model for the convex function $f_1(x) - \langle \hat{g}_2, x \rangle$. Thus it follows from the convergence analysis of convex bundle methods that the sequence of iterates generated after the last serious step converges to the last stability center: $\lim_{k \rightarrow \infty} x^{k+1} = \hat{x}$. Indeed, if $D(x, y) = \|x - y\|_2^2$ this claim follows directly from [45, Proposition 4.4]. For the more general setting of a Bregman function $D(\cdot, \cdot)$ but with $X = \mathbb{R}^n$, the result $\lim_{k \rightarrow \infty} x^{k+1} = \hat{x}$ can be justified by [12, Theorem 5.8] if the prox-parameter $\mu_k > 0$ is fixed after finitely many steps of the algorithm. Overall, we have the following result.

Lemma 4 Let $\hat{x} = x^{k(\hat{\ell})}$ be the last stability center generated by Algorithm 1 and assume that $\{\mu_k\}_{k \geq k(\hat{\ell})}$ is a nondecreasing sequence contained in $[\underline{\mu}, \bar{\mu}]$. Then

$$\lim_{k \rightarrow \infty} x^{k+1} = \hat{x}.$$

Since Algorithm 1 does not consider the same assumptions of either [45, Proposition 4.4] nor [12, Theorem 5.8]), we provide for the sake of completeness the proof of Lemma 4 at “Appendix”. The following result is a version of Lemma 3 for the sequence of null iterates.

Lemma 5 Let $\hat{x} = x^{k(\hat{\ell})}$ be the last stability center generated by Algorithm 1, and for $k \geq k(\hat{\ell})$ $p^{k+1} \in \partial \check{f}_1^k(x^{k+1})$ and $s^{k+1} \in N_X(x^{k+1})$. If $\{\mu_k\}_{k \geq k(\hat{\ell})}$ is nondecreasing, then there exist constants $M, L > 0$ such that $\|p^{k+1} + s^{k+1}\|_2 \leq M$ and $\|p^{k+1}\|_2 \leq L$ for all $k \geq k(\hat{\ell})$. Moreover, $\beta^{k+1} = p^{k+1} + s^{k+1} \in \partial_{e_k}[f_1(\hat{x}) + i_X(\hat{x})]$, where $e_k = (M + L)\|x^{k+1} - \hat{x}\|_2$ for all $k \geq k(\hat{\ell})$.

Proof It follows from (10) that $\|p^{k+1} + s^{k+1}\|_2 \leq \|\hat{g}_2\|_2 + \bar{\mu}\|\nabla\omega(\hat{x}) - \nabla\omega(x^{k+1})\|_2$ for $k \geq k(\hat{\ell})$. Since $\{\mu_k\}_{k \geq k(\hat{\ell})}$ is nondecreasing, Lemma 8 (in the “Appendix”) ensures that $\{x^k\}_{k > k(\hat{\ell})}$ is a bounded sequence (therefore $\{x^k\}_k$ is bounded itself). Then there exists a finite constant $M > 0$ bounding $\|p^{k+1} + s^{k+1}\|_2$ (because $\nabla\omega$ is continuous). As f_1 is a finite-valued convex function on $\Omega \supset X$, its subdifferential is locally bounded [20, Theorem 3.1.2]. Then there exists a constant $L > 0$ such that $\|g_1^k\|_2 \leq L$ for all k . We recall the subdifferential of \check{f}_1^k at x is the convex hull of the set $\{g_1^j : \check{f}_1^k(x) = f_1(x^j) + \langle g_1^j, x - x^j \rangle\}$. Thus every subdifferential of \check{f}_1^k is also bounded by L , which in turn implies that \check{f}_1^k (regardless the iteration k) has L as a Lipschitz constant.

Recall that $N_X(x^{k+1}) = \partial i_X(x^{k+1})$ because $x^{k+1} \in X$. Then $s^{k+1} \in \partial i_X(x^{k+1})$ and therefore $i_X(x) \geq i_X(x^{k+1}) + \langle s^{k+1}, x - x^{k+1} \rangle$ for all $x \in \mathbb{R}^n$. Since $p^{k+1} \in \partial \check{f}_1^k(x^{k+1})$ and $i_X(x^k) = 0$ for all k we get (for $k \geq k(\hat{\ell})$)

$$\begin{aligned} f_1(x) + i_X(x) &\geq \check{f}_1^k(x) + i_X(x) \\ &\geq \check{f}_1^k(x^{k+1}) + \langle p^{k+1}, x - x^{k+1} \rangle + [i_X(x^{k+1}) + \langle s^{k+1}, x - x^{k+1} \rangle] \\ &= \check{f}_1^k(x^{k+1}) + \langle \beta^{k+1}, x - x^{k+1} \rangle \\ &= f_1(\hat{x}) + \langle \beta^{k+1}, x - \hat{x} \rangle + \langle \beta^{k+1}, \hat{x} - x^{k+1} \rangle + \check{f}_1^k(x^{k+1}) - f_1(\hat{x}) \\ &\geq f_1(\hat{x}) + \langle \beta^{k+1}, x - \hat{x} \rangle - M\|\hat{x} - x^{k+1}\|_2 - [f_1(\hat{x}) - \check{f}_1^k(x^{k+1})]. \end{aligned}$$

Since by construction $k(\hat{\ell}) \in \mathcal{B}_1^k$ for all $k \geq k(\hat{\ell})$, we have that $\check{f}_1^k(\hat{x}) = f_1(\hat{x})$. Then $f_1(\hat{x}) - \check{f}_1^k(x^{k+1}) = \check{f}_1^k(\hat{x}) - \check{f}_1^k(x^{k+1}) \leq L\|\hat{x} - x^{k+1}\|_2$. This shows that

$$f_1(x) + i_X(x) \geq f_1(\hat{x}) + i_X(\hat{x}) + \langle \beta^{k+1}, x - \hat{x} \rangle - e_k \quad \forall x \in \mathbb{R}^n,$$

because $i_X(\hat{x}) = 0$. Therefore, $\beta^{k+1} \in \partial_{e_k}[f_1(\hat{x}) + i_X(\hat{x})]$ for all $k \geq k(\hat{\ell})$. □

We are now in the position to prove that the last stability center \hat{x} is a critical point of problem (1).

Proposition 3 Let $\hat{x} = x^{k(\hat{\ell})}$ be the last stability center generated by Algorithm 1 and assume that $\{\mu_k\}_{k \geq k(\hat{\ell})}$ is a nondecreasing sequence contained in $[\underline{\mu}, \bar{\mu}]$. Then the sequence $\{x^{k+1}\}_k$ converges to \hat{x} and $\bar{x} = \hat{x}$ is a critical point of problem (1).

Proof Lemma 4 gives $\lim_{k \rightarrow \infty} x^{k+1} = \hat{x}$. It follows from (10), continuity of the mapping $\nabla\omega$ and $\mu_k \leq \bar{\mu}$ that

$$\lim_{k \rightarrow \infty} \beta^{k+1} = \lim_{k \rightarrow \infty} [p^{k+1} + s^{k+1}] = \hat{g}_2 - \lim_{k \rightarrow \infty} \mu_k (\nabla\omega(x^{k+1}) - \nabla\omega(\hat{x})) = \hat{g}_2 \in \partial f_2(\hat{x}).$$

Lemma 5 ensures that $\beta^{k+1} \in \partial_{e_k}[f_1(\hat{x}) + i_X(\hat{x})]$ with $e_k = (M + L)\|x^{k+1} - \hat{x}\|_2$. Since $\lim_{k \rightarrow \infty} e_k = 0$ due to $\lim_{k \rightarrow \infty} x^{k+1} = \hat{x}$, then $\hat{g}_2 = \lim_{k \rightarrow \infty} \beta^{k+1} \in \partial[f_1(\hat{x}) + i_X(\hat{x})]$ and (3) is satisfied with $\bar{x} = \hat{x}$. □

4.3 Convergence analysis: main result

Convergence analysis of Algorithm 1 is summarized in the following theorem.

Theorem 1 *Consider Algorithm 1 and suppose that the level set $\{x \in X : f(x) \leq f(x^0)\}$ is bounded. If the stopping-test tolerance $\delta_{\text{Tot}} = 0$, then any cluster point \bar{x} of the sequence of stability centers $\{x^{k(\ell)}\}_{\ell \in \mathcal{L}}$ generated by the algorithm satisfies (3). Moreover, if the stopping-test tolerance $\delta_{\text{Tot}} > 0$ then the algorithm stops after finitely many steps with an approximate critical point $\bar{x} = x^{k(\ell)}$. In addition, if $\nabla\omega(\cdot)$ is a locally Lipschitz continuous map, then the approximate critical point $\bar{x} = x^{k(\ell)}$ satisfies*

$$\partial_{s_1\delta_{\text{Tot}}}[f_1(\bar{x}) + i_X(\bar{x})] \cap [\partial f_2(\bar{x}) + B(0; s_2\delta_{\text{Tot}})] \neq \emptyset,$$

where $s_1, s_2 > 0$ are two constants and $B(0; s_2\delta_{\text{Tot}})$ is the closed ball in \mathbb{R}^n with center at zero and with radius $s_2\delta_{\text{Tot}}$.

Proof First, suppose that $\delta_{\text{Tot}} = 0$. If Algorithm 1 stops at iteration k then Lemma 2 ensures that the last stability center is a critical point of (1). Suppose now that Algorithm 1 does not stop. If infinitely many serious steps are generated, then Proposition 2 gives the result. Otherwise there will be a finite number of serious steps and Proposition 3 ensures that the last stability center is a critical point.

Moreover, suppose that $\delta_{\text{Tot}} > 0$. If infinitely many stability centers are generated then Proposition 2 shows that $\lim_{\ell \rightarrow \infty} \|x^{k(\ell+1)} - x^{k(\ell)}\|_2 = 0$. Otherwise, if there is only finitely many stability centers, then Proposition 3 ensures that $\lim_{k \rightarrow \infty} \|x^{k+1} - \hat{x}\|_2 = 0$ where $\hat{x} = x^{k(\ell)}$ is the last stability center. In any case, the stopping test of Algorithm 1 will be triggered after finitely many iterations if $\delta_{\text{Tot}} > 0$. In this case, $\|x^{k+1} - x^{k(\ell)}\|_2 \leq \delta_{\text{Tot}}$ and therefore Lemma 5 gives

$$p^{k+1} + s^{k+1} \in \partial_{(M+L)\delta_{\text{Tot}}}[f_1(x^{k(\ell)}) + i_X(x^{k(\ell)})].$$

Furthermore, Eq. (10) and assumption on $\nabla\omega$ yield

$$\|p^{k+1} + s^{k+1} - g_2^{k(\ell)}\|_2 = \mu_k \|\nabla\omega(x^{k+1}) - \nabla\omega(x^{k(\ell)})\|_2 \leq s_2 \|x^{k+1} - x^{k(\ell)}\|_2 \leq s_2\delta_{\text{Tot}}.$$

These properties show that $\bar{x} = x^{k(\ell)}$ is an approximate critical point of problem (1) with $s_1 = M + L$ and $s_2 = \bar{\mu} L_\omega$, where L_ω is the Lipschitz constant of $\nabla\omega$ on the bounded sequence $\{x^k\}_k$. □

4.4 Some comments on convex and DC models

Algorithm 1 employs the convex model $\check{f}_1^k(x) - \check{f}_2^{k(\ell)}(x)$, with

$$\check{f}_2^{k(\ell)}(x) := f_2(x^{k(\ell)}) + \langle g_2^{k(\ell)}, x - x^{k(\ell)} \rangle,$$

to approximate the DC function f , where \check{f}_1^k is a cutting-plane model of f_1 . Instead of considering only one linearization for the second DC component, one could also gather a bundle of information $\mathcal{B}_2^k \subset \{1, 2, \dots, k\}$ for f_2 and define a cutting-plane model

$$\check{f}_2^k(x) := \max_{j \in \mathcal{B}_2^k} \bar{f}_2^j(x).$$

This provides a DC model $\check{f}_1^k - \check{f}_2^k$ for f that is expected to be better than the convex one $\check{f}_1^k(x) - \bar{f}_2^{k(\ell)}(x)$. Both publications [16] and [27] consider the DC model $\check{f}_1^k(\cdot) - \check{f}_2^k(\cdot)$ in their proximal bundle algorithms for unconstrained DC programs. For instance, the method of [27] defines trial points by solving globally the following DC subproblem

$$x^{k+1} \in \arg \min_{x \in \mathbb{R}^n} \check{f}_1^k(x) - \check{f}_2^k(x) + \mu_k \|x - x^{k(\ell)}\|_2^2. \tag{16}$$

Since \check{f}_2^k is a polyhedral function, a global solution to the above subproblem can be obtained by solving $|\mathcal{B}_2^k|$ strictly convex QPs

$$\min_{j \in \mathcal{B}_2^k} \left\{ \min_{x \in X} \check{f}_1^k(x) - \bar{f}_2^j(x) + \mu_k \|x - x^{k(\ell)}\|_2^2 \right\}.$$

As a result, if the bundle of information \mathcal{B}_2^k is large then each iteration of the algorithm in [27] can be too time consuming. For this reason, the size of the bundle \mathcal{B}_2^k should be kept small. The proximal bundle method of [27] is shown to converge to a critical point of an unconstrained DC program even when \mathcal{B}_2^k has only two indexes, i.e., only two QPs are needed to be solved per iteration. In contrast, Algorithm 1 requires solving only one convex subproblem (a QP if $\omega(\cdot) = \|\cdot\|_2^2/2$ and X is a polyhedron) per iteration, which results in taking $\mathcal{B}_2^k = \{k(\ell)\}$ for all k . This is a feature of practical interest if X contains some conic/quadratic constraints.

5 A proximal bundle method for finding d -stationary points

In this section, we consider a particular case of problem (1) in which the second DC component is the pointwise maximum of finitely many differentiable convex functions $\psi_i : \Omega \rightarrow \mathbb{R}$:

$$\min_{x \in X} f(x), \quad \text{with } f(x) = f_1(x) - f_2(x) \quad \text{and} \quad f_2(x) := \max_{1 \leq i \leq N} \psi_i(x). \tag{17}$$

Inspired by the work [38], the following modification of Algorithm 1 may solve several subproblems of type (8) at certain iterations to generate a subsequence of stability centers $\{x^{k(\ell)}\}_{\ell \in \mathcal{L}}$ that converges to a d -stationary point \bar{x} of problem (17), i.e., \bar{x} satisfying (2). To this end, we explore the structure of f_2 . It is well known that the subdifferential of f_2 at any given point $x \in X \subset \Omega$ is the convex hull of gradients of functions ψ_i that are active:

$$\begin{aligned} \partial f_2(x) &= \text{conv}(\{\nabla \psi_i(x)\}_{i \in A(x)}), \quad \text{with } A(x) := \{1 \leq i \leq N : \psi_i(x) \\ &= f_2(x)\} \quad \text{for all } x \in X. \end{aligned} \tag{18}$$

As a result, if $\bar{x} \in X$ satisfies

$$\nabla \psi_i(\bar{x}) \in \partial[f_1(\bar{x}) + i_X(\bar{x})] \quad \text{for all } i \in A(\bar{x}),$$

then (2) holds from the convexity of $\partial[f_1(\bar{x}) + i_X(\bar{x})]$ and thus \bar{x} is a d -stationary point of (17). For technical reasons, Algorithm 2 below makes use of the following relaxation of $A(x)$:

$$A_\epsilon(X) = \{1 \leq i \leq N : \psi_i(x) \geq f_2(x) - \epsilon\}, \quad \text{with } \epsilon > 0.$$

To avoid critical points that are not d -stationary the parameter ϵ above must be strictly positive [38, Example 4].

Algorithm 2 A PROXIMAL BUNDLE ALGORITHM FOR COMPUTING d - STATIONARY POINTS

```

1: Let  $x^0 \in X, \kappa \in (0, 1), 0 < \underline{\mu} \leq \mu_0 \leq \bar{\mu} < \infty$  and tolerances  $\delta_{\text{Tot}} \geq 0$  and  $\epsilon > 0$  be given.
2: Compute  $(f_1(x^0), g_1^0 \in \partial f_1(x^0)), f_2(x^0)$ , and  $\nabla \psi_i(x^0)$  for all  $i \in A_\epsilon(x^0)$ .
3: Define  $\mathcal{B}_1^0 := \{0\}$  and  $k(0) := \ell := 0$ .
4: for  $k = 0, 1, 2, \dots$  do
5:   for all  $i \in A_\epsilon(x^{k(\ell)})$  do
6:     Compute the unique solution  $(r_i, y_i)$  and the multipliers  $\alpha_i \in \mathbb{R}_+^{|\mathcal{B}_1^k|}$  of
           
$$\begin{cases} \min_{x \in X, r \in \mathbb{R}} & r - \langle \nabla \psi_i(x^{k(\ell)}), x - x^{k(\ell)} \rangle + \mu_k D(x, x^{k(\ell)}) \\ \text{s.t.} & \tilde{f}_1^j(x) \leq r, \quad j \in \mathcal{B}_1^k. \end{cases} \quad (19)$$

7:   end for
8:   Define  $x^{k+1} := y_{i^*}$ , where
           
$$i^* \in \arg \min_{i \in A_\epsilon(x^{k(\ell)})} [r_i - \langle \nabla \psi_i(x^{k(\ell)}), y_i - x^{k(\ell)} \rangle + \mu_k D(y_i, x^{k(\ell)})]$$

9:   if  $\|x^{k+1} - x^{k(\ell)}\|_2 \leq \delta_{\text{Tot}}$  then
10:     Stop and return  $x^{k(\ell)}$ 
11:   end if
12:   Call the oracle to compute  $(f_1(x^{k+1}), g_1^{k+1} \in \partial f_1(x^{k+1}))$  and  $f_2(x^{k+1})$ 
13:   if (12) holds true then
14:     Compute  $\nabla \psi_i(x^{k+1})$  for all  $i \in A_\epsilon(x^{k+1})$ 
15:     Define  $x^{k(\ell+1)} := x^{k+1}, k(\ell+1) := k+1$  and  $\ell := \ell+1$ 
16:     Choose  $\mathcal{B}_1^{k+1} \supset \{k+1\}$  and  $\mu_{k+1} \geq \underline{\mu}$ 
17:   else
18:     Choose  $\mathcal{B}_1^{k+1} \supset \{-k, k(\ell), k+1\}$  and  $\mu_{k+1} \in [\mu_k, \bar{\mu}]$ 
19:   end if
20: end for

```

Note that if the number N of functions ψ_i in (17) is equal to one (i.e., $f_2 = \psi_1$), then Algorithm 2 becomes essentially Algorithm 1 (with the only difference in the descent test). However, if $N > 1$ then Algorithm 2 solves $|A_\epsilon(x^k)|$ subproblems per iteration.

Moreover, some vectors $\nabla \psi_i(x^{k(\ell)})$ with $i \in A_\epsilon(x^{k(\ell)})$ may not belong to $\partial f_2(x^{k(\ell)})$ because $\epsilon > 0$. As a result, the new trial point x^{k+1} solution of (19) for some $i^* \in A_\epsilon(x^{k(\ell)})$ may not be issued by a true subgradient $\nabla \psi_{i^*}(x^{k(\ell)})$ of f_2 at $x^{k(\ell)}$. Since Lemma 1 relies on the fact that $g_2^{k(\ell)} \in \partial f_2(x^{k(\ell)})$, we cannot employ the descent test (13) with $g_2^{k(\ell)}$ replaced with $\nabla \psi_{i^*}(x^{k(\ell)})$: even if

$$f_1(x^{k+1}) \leq f_1(x^{k(\ell)}) + \langle \nabla \psi_{i^*}, x^{k+1} - x^{k(\ell)} \rangle - \kappa \underline{\mu} D(x^{k+1}, x^{k(\ell)})$$

holds true, nothing ensures that $f(x^{k+1}) \leq f(x^{k(\ell)}) - \kappa \underline{\mu} D(x^{k+1}, x^{k(\ell)})$. This is why we have replaced the descent test (13) with the more direct one given in (12). A downside of

such replacement is that we need to call an oracle for f_2 at every trial point, in contrast to Algorithm 1 that never evaluates f_2 (but computes one of its subgradients at serious steps).

In order to analyze the convergence of Algorithm 2, we rely on the convergence analysis of Algorithm 1. We start with the following lemma.

Lemma 6 *Let x^{k+1} and i^* as be defined in Algorithm 2. Then for all $i \in A_\epsilon(x^{k(\ell)})$*

$$\begin{aligned} & \check{f}_1^k(x^{k+1}) - \langle \nabla \psi_{i^*}(x^{k(\ell)}), x^{k+1} - x^{k(\ell)} \rangle + \mu_k D(x^{k+1}, x^{k(\ell)}) \\ & \leq \min_{x \in X} f_1(x) - \langle \nabla \psi_i(x^{k(\ell)}), x - x^{k(\ell)} \rangle + \bar{\mu} D(x, x^{k(\ell)}) \leq f_1(x^{k(\ell)}). \end{aligned} \tag{20}$$

Proof It follows from the definition of x^{k+1} that $\check{f}_1^k(x^{k+1}) - \langle \nabla \psi_{i^*}(x^{k(\ell)}), x^{k+1} - x^{k(\ell)} \rangle + \mu_k D(x^{k+1}, x^{k(\ell)}) = \min_{x \in X} \check{f}_1^k(x) - \langle \nabla \psi_{i^*}(x^{k(\ell)}), x - x^{k(\ell)} \rangle + \mu_k D(x, x^{k(\ell)})$, which is in turn less than or equal to

$$\check{f}_1^k(y_i) - \langle \nabla \psi_i(x^{k(\ell)}), y_i - x^{k(\ell)} \rangle + \mu_k D(y_i, x^{k(\ell)}) \quad \forall i \in A_\epsilon(x^{k(\ell)})$$

due to the definition of i^* and y_i given in Algorithm 2. Again, the definition of y_i and the inequalities $\check{f}_1^k(x) \leq f_1(x)$ and $\mu_k \leq \bar{\mu}$ ensure the first inequality in (20). The last one follows from feasibility of $x^{k(\ell)}$. □

Theorem 2 *Consider Algorithm 2 applied to problem (17) and suppose that $\delta_{\text{Tot}} = 0$ and the level set $\{x \in X : f(x) \leq f(x^0)\}$ is bounded. Then any cluster point \bar{x} of the sequence of stability centers $\{x^{k(\ell)}\}_{\ell \in \mathcal{L}}$ generated by the algorithm is a d -stationary point, i.e., \bar{x} satisfies (2).*

Proof We split the proof in three main parts: the first part considers the case in which Algorithm 2 stops with $\delta_{\text{Tot}} = 0$, the second one analyzes the case of infinitely many serious steps, and finally the last part assumes that after a last serious step the algorithm loops forever generating only null steps.

First part Suppose that Algorithm 2 stops at iteration k . Then $x^{k+1} = x^{k(\ell)}$ and Lemma 6 gives

$$\begin{aligned} \check{f}_1^k(x^{k(\ell)}) & \leq \min_{x \in X} f_1(x) - \langle \nabla \psi_i(x^{k(\ell)}), x - x^{k(\ell)} \rangle + \mu_k D(x, x^{k(\ell)}) \\ & \leq f_1(x^{k(\ell)}) \quad \forall i \in A_\epsilon(x^{k(\ell)}). \end{aligned}$$

As $k(\ell) \in \mathcal{B}_1^k$, then $\check{f}_1^k(x^{k(\ell)}) = f_1(x^{k(\ell)})$ and the above relation ensures that $x^{k+1} = x^{k(\ell)}$ solves

$$\begin{aligned} & \min_{x \in X} f_1(x) - \langle \nabla \psi_i(x^{k(\ell)}), x - x^{k(\ell)} \rangle + \mu_k D(x, x^{k(\ell)}) \quad \forall i \in \overline{A}(x^{k(\ell)}) \subset A_\epsilon(x^{k(\ell)}), \quad \text{i.e.,} \\ & 0 \in \partial f_1(x^{k+1}) - \nabla \psi_i(x^{k(\ell)}) + \mu_k \left(\nabla \omega(x^{k+1}) - \nabla \omega(x^{k(\ell)}) \right) + N_X(x^{k+1}) \quad \forall i \in A(x^{k(\ell)}). \end{aligned}$$

Since $x^{k+1} = x^{k(\ell)}$, then $\nabla \psi_i(x^{k(\ell)}) \in \partial f_1(x^{k(\ell)}) + N_X(x^{k(\ell)})$ for all $i \in A(x^{k(\ell)})$, which by (18) implies that $\bar{x} = x^{k(\ell)}$ satisfies (2). In what follows we suppose that the algorithm does not stop.

Second part Let us suppose that Algorithm 2 generates an infinite sequence of serious steps. By summing the inequality $f(x^{k(\ell+1)}) \leq f(x^{k(\ell)}) - \kappa \mu D(x^{k(\ell+1)}, x^{k(\ell)})$ over $\ell \in \mathcal{L} = \{0, 1, 2, \dots\}$ and using the fact that $\{x \in X : f(x) \leq f(x^0)\}$ is a bounded set, we conclude that $\{x^{k(\ell)}\}_{\ell \in \mathcal{L}}$ is a bounded sequence. Moreover, $\sum_{\ell=0}^\infty \|x^{k(\ell+1)} - x^{k(\ell)}\|_2^2 < \infty$

by (5) and the equivalence of norms in \mathbb{R}^n . Boundedness of $\{x^{k(\ell)}\}_{\ell \in \mathcal{L}}$ ensures that there exists an index set $\mathcal{L}' \subset \mathcal{L}$ such that the sequence $\{x^{k(\ell)}\}_{\ell \in \mathcal{L}'}$ converges to a point $\bar{x} \in X$. Continuity of f_2 and the assumption that $\epsilon > 0$ yield the inclusion $A(\bar{x}) \subset A_\epsilon(x^{k(\ell)})$ for all $\ell \in \mathcal{L}'$ large enough. In what follows, let $i(\ell) = k(\ell + 1) - 1$. The definition of $x^{k(\ell+1)} (= x^{k+1} = y_{i^*})$ and (20) give

$$\begin{aligned} & \check{f}_1^{i(\ell)}(x^{k(\ell+1)}) - \langle \nabla \psi_{i^*}(x^{k(\ell)}), x^{k(\ell+1)} - x^{k(\ell)} \rangle + \mu_{i(\ell)} D(x^{k(\ell+1)}, x^{k(\ell)}) \\ & \leq f_1(x) - \langle \nabla \psi_i(x^{k(\ell)}), x - x^{k(\ell)} \rangle + \bar{\mu} D(x, x^{k(\ell)}) \end{aligned} \tag{21}$$

for all $x \in X, i \in A(\bar{x})$ and $\ell \in \mathcal{L}'$ large enough. Algorithm 2 keeps in the bundle the index $k(\ell)$ of the last serious step. As a result,

$$\check{f}_1^{i(\ell)}(x^{k(\ell+1)}) = \max_{j \in \mathcal{B}_1^{i(\ell)}} \{f_1(x^j) + \langle g_1^j, x^{k(\ell+1)} - x^j \rangle\} \geq f_1(x^{k(\ell)}) + \langle g_1^{k(\ell)}, x^{k(\ell+1)} - x^{k(\ell)} \rangle.$$

By the Cauchy–Schwarz inequality (and remembering that $\{x^{k(\ell)}\}_{\ell \in \mathcal{L}}$ is bounded and f_1 is convex) we get $\check{f}_1^{i(\ell)}(x^{k(\ell+1)}) \geq f_1(x^{k(\ell)}) - \|g_1^{k(\ell)}\|_2 \|x^{k(\ell+1)} - x^{k(\ell)}\|_2 \geq f_1(x^{k(\ell)}) - L \|x^{k(\ell+1)} - x^{k(\ell)}\|_2$. Combining this inequality with (21) yields

$$\begin{aligned} & f_1(x^{k(\ell)}) - L \|x^{k(\ell+1)} - x^{k(\ell)}\|_2 - \langle \nabla \psi_{i^*}(x^{k(\ell)}), x^{k(\ell+1)} - x^{k(\ell)} \rangle + \mu_{i(\ell)} D(x^{k(\ell+1)}, x^{k(\ell)}) \\ & \leq f_1(x) - \langle \nabla \psi_i(x^{k(\ell)}), x - x^{k(\ell)} \rangle + \bar{\mu} D(x, x^{k(\ell)}) \end{aligned} \tag{22}$$

for all $x \in X, i \in A(\bar{x})$ and $\ell \in \mathcal{L}'$ large enough. By going to the limit with $\ell \in \mathcal{L}'$ tending to infinity (and recalling that $\lim_{\ell \rightarrow \infty} \|x^{k(\ell+1)} - x^{k(\ell)}\|_2 = 0$ because $\sum_{\ell=0}^\infty \|x^{k(\ell+1)} - x^{k(\ell)}\|_2^2 < \infty$) we obtain

$$f_1(\bar{x}) \leq f_1(x) - \langle \nabla \psi_i(\bar{x}), x - \bar{x} \rangle + \bar{\mu} D(x, \bar{x}), \quad \forall x \in X, \forall i \in A(\bar{x}).$$

This shows that \bar{x} solves all the $|A(\bar{x})|$ subproblems $\min_{x \in X} f_1(x) - \langle \nabla \psi_i(\bar{x}), x - \bar{x} \rangle + \bar{\mu} D(x, \bar{x})$, i.e., $\nabla \psi_i(\bar{x}) \in \partial f_1(\bar{x}) + N_X(\bar{x})$ for all $i \in A(\bar{x})$, implying thus that \bar{x} is a d -stationary point of (17).

Third part Let us consider the case of finitely many serious steps. Accordingly, there exists a last stability center $\hat{x} = x^{k(\hat{\ell})}$ and $f(\hat{x}) < f(x^{k+1}) + \kappa \bar{\mu} D(x^{k+1}, \hat{x})$ for all $k \geq k(\hat{\ell})$. As mentioned right before Lemma 6 in page 13, this implies that a point x^{k+1} for $k \geq k(\hat{\ell})$ does not satisfy test (13) with $g_2^{k(\hat{\ell})} = \nabla \psi_i(\hat{x})$ for any $i \in A(\hat{x})$ [although (13) may hold for some $g_2^{k(\hat{\ell})} = \nabla \psi_j(\hat{x})$ with $j \in A_\epsilon(\hat{x}) \setminus A(\hat{x})$]. Hence, by seeing the null iterates $x^{k+1} = y_{i^*}$ with $i^* \in A_\epsilon(\hat{x}) \setminus A(\hat{x})$ as mere points enriching the bundle \mathcal{B}_1^k , Lemma 8 and Proposition 3 apply: $\lim_{k \rightarrow \infty} [f_1(x^{k+1}) - \check{f}_1^k(x^{k+1})] = 0$, $\lim_{k \rightarrow \infty} x^{k+1} = \hat{x}$ and \hat{x} is a critical point. It remains to show that \hat{x} is indeed a d -stationary point of (17). To this end, consider the following inequality extracted from (20):

$$\begin{aligned} & \check{f}_1^k(x^{k+1}) - \langle \nabla \psi_{i^*}(\hat{x}), x^{k+1} - \hat{x} \rangle + \mu_k D(x^{k+1}, \hat{x}) \\ & \leq f_1(x) - \langle \nabla \psi_i(\hat{x}), x - \hat{x} \rangle + \bar{\mu} D(x, \hat{x}), \end{aligned}$$

for all $x \in X$ and all $i \in A(\hat{x})$. By passing to the limit with k tending to infinity we obtain

$$\begin{aligned} f_1(\hat{x}) & = \lim_{k \rightarrow \infty} [\check{f}_1^k(x^{k+1}) - \langle \nabla \psi_{i^*}(\hat{x}), x^{k+1} - \hat{x} \rangle + \mu_k D(x^{k+1}, \hat{x})] \\ & \leq f_1(x) - \langle \nabla \psi_i(\hat{x}), x - \hat{x} \rangle + \bar{\mu} D(x, \hat{x}) \end{aligned}$$

for all $x \in X$ and $i \in A(\hat{x})$. This shows that \hat{x} solves all the $|A(\hat{x})|$ subproblems

$$\min_{x \in X} f_1(x) - \langle \nabla \psi_i(\hat{x}), x - \hat{x} \rangle + \bar{\mu} D(x, \hat{x}) \quad \text{for } i \in A(\hat{x}),$$

implying thus that $\bar{x} = \hat{x}$ is a d -stationary point of (17) (see 18).

In all the cases, if $\delta_{\text{Tot}} > 0$ the algorithm stops after finitely many steps. \square

6 Numerical experiments

We assess the numerical performance of Algorithms 1 and 2 on some academic DC problems. For comparison reasons, we consider two variants of Algorithm 1 with the Bregman function $D(x, y) = \|x - y\|_2^2/2$, two implementations of the classic DC algorithm (DCA) of [42], one implementation of the proximal linearized algorithm [41, Algorithm 1], the DC proximal bundle method of [27] and an algorithm for nonsmooth and nonconvex optimization [35]. We also compare the computational behavior of Algorithm 1 against the results reported in the paper [16]. The considered solvers are as follows:

- PBM1: Algorithm 1.
- PBM-d: Algorithm 2.
- PBM3: Algorithm 1 with subproblem (9) replaced with the DC subproblem (16), where the model for f_2 has three pieces: three linearizations computed with the last three stability centers.
- DCA-CPM: this is an implementation of the DCA algorithm of [42], with trial points defined as

$$x^{k+1} \in \arg \min_{x \in X} f_1(x) - \langle g_2^k, x \rangle \quad \text{with } g_2^k \in \partial f_2(x^k). \quad (23)$$

The convex subproblem (23) is solved by an implementation of the Kelley's cutting-plane method [28]. We have made use of warm start, meaning that when a new subgradient $g_2^{k+1} \in \partial f_2(x^{k+1})$ is computed the solver re-uses the cutting-plane model for f_1 constructed at the previous iteration.

- DCA-LBM: as DCA-CPM, but with the convex subproblem (23) solved by an implementation of the level bundle method of [34].
- PLM: this is an implementation of the proximal linearized algorithm in [41, Algorithm 1], with trial points defined as in (6). The convex subproblem (6) is solved by replacing f_1 with its cutting-plane approximation, which is iteratively improved the same way as the classical proximal bundle method does. As in DCA-CPM, we have employed warm start.
- HANSO: Hybrid Algorithm for Non-Smooth Optimization [35]. This is a MATLAB package based on the BFGS and gradient sampling methods for nonsmooth and nonconvex unconstrained optimization problems. The package (version 2.2) is freely available by its developers at the link: www.cs.nyu.edu/overton/software/hanso.
- PBDC: the proximal bundle method of [27], whose Fortran code is freely available by its developers at the link: <http://napsu.karmitsa.fi/pbdc/>.

Except for solver HANSO and PBDC that are open source packages, we have implemented the considered solvers in MATLAB (version 2017a) using the Gurobi (version 7.5.1, www.gurobi.com) solver for LP, QP and quadratically-constrained problems. Solver HANSO was employed with its default parameters, except the memory of its BFGS algorithm that was set to 50 for problems where dimension is greater or equal to 1000. Solver PBDC was used with its default parameters, except its stopping-test tolerance (`user_crit_tol`) that was

increased to $0.06 \cdot n$. The other solvers employ the stopping test $\|x^{k+1} - x^{k(\ell)}\|_2 \leq \delta_{\text{Tol}}$, with tolerance $\delta_{\text{Tol}} = 10^{-4}$. In our implementation, the solvers PBM1 and PBM3 consider the test (12) with $\kappa = 0.1$ and $\mu = 10^{-5}$ for updating stability centers. This implies that the number of evaluations of f_1 is the same as for f_2 . The maximum size of the bundle of information for solvers PBM1, PBM3 and PBM-d was set to $\max\{100, \min\{n + 5, 1000\}\}$. Moreover, only active indexes were kept in the bundle after a serious step. The MATLAB codes as well as the considered test problems are freely available at the link: <http://www.oliveira.mat.br/solvers>.

Numerical experiments were performed on a computer with Intel(R) Core(TM), i7-6820HQ, CPU @ 2.70 GHz, 32G (RAM), under Windows 7, 64Bits. The Fortran code of PBDC was compiled and executed in a Virtual Machine running Linux Ubuntu. The total CPU time allowed for each solver on each test problem was set to 3600 s.

For the sake of comparison, we also present some results from the paper [16]. Since we have not ran the solver DCPCA, it does not make sense to compare CPU time. However, the solution quality and number of subgradients evaluations can be compared, keeping in mind that DCPCA, PBDC and the MATLAB solvers employ different stopping tests.

6.1 Unconstrained DC programs

We consider all the unconstrained DC programs reported in [2,16,27]. Their formulation, optimal values and initial points are given in [27].

Since the inner algorithms employed by solvers DCA-CPM and DCA-LBM require the feasible set to be compact, we have added the bounds $-100 \leq x \leq 100$ to the test problems when running these two solvers (the other solvers did not consider this change).

Table 1 reports numerical experiments on several instances of the considered unconstrained DC problems. The first column indicates the problem, the second states the dimension of the problem and the third column reports the optimal value. The remaining columns present the obtained function value $f(x^{k(\ell)})$, number of subgradient evaluations for DC components and CPU time (in seconds) for all the considered solvers. We emphasize that:

- PBM1 and PBM3 The number of subgradient evaluations $\#g_1$ is equal to the number of function evaluations of f_1 . It also coincides with the number of iterations performed by the algorithm. The number of subgradient evaluations $\#g_2$ of the second DC component coincides with the number of serious steps of the algorithm.
- DCA-CPM, DCA-LBM and PLM The number of subgradient evaluations $\#g_1$ is equal to the number of function evaluations of f_1 . Moreover, $\#g_2$ coincides with the number of iterations.
- HANSO The number of subgradient evaluations is equal to the number of function evaluations. The solver does not exploit the DC decomposition of the objective function.
- PBDC The number of function f evaluations can be slightly bigger than $\#g_1$.
- DCPCA The number of evaluations of f_1 is greater than that number of subgradient evaluations due to the line-search.

Notice that the solution quality of solvers PBM1 and PBM3 is comparable to solvers PBDC and DCPCA. These four DC bundle solvers provided more precise solutions than the solvers DCA-CPM, DCA-LBM, PLM and the nonconvex solver HANSO. Moreover, the bundle solvers were successful in computing a critical point of problem (1) in all instances (except solver PBDC that exceeded the CPU time limit when dealing with problem 4, $n = 750$). Except for problem 10 with $n = 5$, solvers PBM1, PBM3 and DCPCA found the best known function values of the considered problems.

Table 1 continued

| Problem data | | PBM1 | | | PBM3 | | | DCA-CPM | | | DCA-LBM | | | |
|--------------|--------|-----------|---------------|-----|------|------|---------------|---------|-----|-------|---------------|-----|-----|------|
| Pbm | n | \bar{f} | $f(x^{k(l)})$ | #g1 | #g2 | CPU | $f(x^{k(l)})$ | #g1 | #g2 | CPU | $f(x^{k(l)})$ | #g1 | #g2 | CPU |
| 5 | 500 | 0 | 2.40E-08 | 45 | 33 | 0.46 | 4.00E-08 | 50 | 36 | 1.05 | 3.92E-04 | 24 | 2 | 0.16 |
| 5 | 1000 | 0 | 2.33E-08 | 47 | 34 | 0.44 | 1.93E-08 | 46 | 34 | 0.96 | 1.41E-03 | 22 | 2 | 0.16 |
| 5 | 1500 | 0 | 1.89E-08 | 47 | 34 | 0.45 | 3.75E-08 | 49 | 35 | 1.01 | 1.97E-03 | 24 | 2 | 0.19 |
| 5 | 3000 | 0 | 5.60E-08 | 44 | 32 | 0.55 | 2.40E-08 | 45 | 35 | 1.14 | 1.97E-03 | 24 | 2 | 0.32 |
| 5 | 10,000 | 0 | 7.18E-08 | 44 | 32 | 1.08 | 4.19E-08 | 45 | 35 | 1.95 | 1.97E-03 | 24 | 2 | 0.77 |
| 5 | 20,000 | 0 | 1.46E-07 | 37 | 25 | 2.08 | 1.39E-07 | 40 | 30 | 2.48 | 1.97E-03 | 24 | 2 | 1.7 |
| 5 | 50,000 | 0 | 1.23E-07 | 51 | 33 | 6.83 | 8.17E-08 | 46 | 33 | 5.77 | 1.97E-03 | 24 | 2 | 4.08 |
| 6 | 2 | -2.5 | -2.5 | 24 | 23 | 0.11 | -2.5 | 24 | 23 | 0.21 | -2.5 | 30 | 2 | 0.05 |
| 7 | 2 | 0.5 | 0.5 | 104 | 65 | 0.51 | 0.5 | 66 | 46 | 0.85 | 1.0* | 527 | 62 | 1.19 |
| 8 | 3 | 3.5 | 3.5 | 22 | 11 | 0.1 | 3.5 | 18 | 9 | 0.14 | 3.77* | 73 | 2 | 0.15 |
| 9 | 4 | 1.83 | 1.83 | 36 | 13 | 0.21 | 1.83 | 34 | 12 | 0.42 | 9.2* | 108 | 2 | 0.22 |
| 10 | 2 | -0.5 | -0.5 | 3 | 1 | 0.01 | -0.5 | 3 | 1 | 0.01 | -0.5 | 41 | 2 | 0.07 |
| 10 | 4 | -2.5 | -2.5 | 6 | 4 | 0.02 | -2.5 | 6 | 4 | 0.04 | -2.5 | 174 | 3 | 0.32 |
| 10 | 5 | -2.5 | -2.5* | 6 | 4 | 0.02 | -2.5* | 6 | 4 | 0.04 | -2.5* | 268 | 3 | 0.63 |
| 10 | 10 | -8.5 | -8.5 | 20 | 12 | 0.08 | -8.5 | 21 | 12 | 0.18 | -8.5 | 7 | 7 | 0.06 |
| 10 | 50 | -48.5 | -48.5 | 66 | 34 | 0.48 | -48.5 | 63 | 32 | 0.91 | -48.5 | 27 | 27 | 0.09 |
| 10 | 100 | -98.5 | -98.5 | 124 | 64 | 1.3 | -98.5 | 149 | 73 | 3.71 | -98.5 | 52 | 52 | 0.22 |
| 10 | 150 | -148.5 | -148.5 | 225 | 114 | 3.23 | -148.5 | 215 | 103 | 9.05 | -148.5 | 77 | 77 | 0.11 |
| 10 | 200 | -198.5 | -198.5 | 263 | 134 | 4.86 | -198.5 | 305 | 147 | 17.26 | -198.5 | 102 | 102 | 0.12 |

Table 1 continued

| Problem data | | PLM | | | HANSO | | | PBDC | | | DCPCA | | | | | |
|--------------|-----|-----------|------------------|-----|-------|------|------------------|--------|-------|------------------|-------|-----|---------|------------------|-----|-----|
| Pbm | n | \bar{f} | $f(x^{k(\ell)})$ | #g1 | #g2 | CPU | $f(x^{k(\ell)})$ | #f | CPU | $f(x^{k(\ell)})$ | #g1 | #g2 | CPU | $f(x^{k(\ell)})$ | #g1 | #g2 |
| 1 | 2 | 2 | 2 | 269 | 19 | 0.81 | 2 | 886 | 0.41 | 2 | 16 | 15 | 0.00 | 2 | 19 | 16 |
| 2 | 2 | 0 | 3.66E-10 | 14 | 5 | 0.05 | 1.32E-06 | 308 | 0.06 | 1.11E-12 | 15 | 15 | 0.00 | 2.80E-09 | 10 | 9 |
| 3 | 4 | 0 | 7.74E-11 | 9 | 2 | 0.03 | 2.22E-05 | 1091 | 0.15 | 2.27E-12 | 15 | 11 | 0.01 | 8.40E-08 | 16 | 12 |
| 4 | 2 | 0 | 5.15E-09 | 4 | 2 | 0.01 | 1.04E-05 | 164 | 0.03 | 8.44E-15 | 3 | 3 | 0.01 | 3.96E-09 | 11 | 11 |
| 4 | 5 | 0 | 9.55E-12 | 8 | 2 | 0.01 | 6.34E-06 | 862 | 0.14 | 7.16E-13 | 6 | 5 | 0.02 | 1.05E-08 | 41 | 41 |
| 4 | 10 | 0 | 2.57E-08 | 13 | 2 | 0.02 | 8.08E-06 | 1990 | 0.22 | 5.68E-14 | 11 | 9 | 0.03 | 1.54E-08 | 11 | 11 |
| 4 | 50 | 0 | 3.31E-07 | 55 | 3 | 0.1 | 1.63E-13 | 26,661 | 4.05 | 9.09E-13 | 51 | 12 | 0.08 | 5.80E-08 | 51 | 51 |
| 4 | 100 | 0 | 0.00E+00 | 186 | 43 | 0.42 | 1.29E-11 | 45,619 | 9.76 | 9.09E-13 | 102 | 30 | 0.45 | 2.82E-11 | 102 | 102 |
| 4 | 150 | 0 | 1.69E-11 | 290 | 70 | 0.78 | 1.12E-07 | 46,905 | 9.41 | 2.17E-05 | 191 | 69 | 3.27 | 7.46E-11 | 152 | 152 |
| 4 | 200 | 0 | 1.49E-09 | 207 | 4 | 0.48 | 1.36E-04 | 47,527 | 8.3 | 7.28E-12 | 369 | 120 | 17.09 | 3.84E-07 | 201 | 201 |
| 4 | 250 | 0 | 3.14E-10 | 257 | 4 | 0.68 | 3.17E-03 | 48,880 | 8.7 | 1.82E-11 | 481 | 194 | 45.93 | 1.31E-10 | 252 | 252 |
| 4 | 500 | 0 | 0.00E+00 | 507 | 4 | 2.65 | 1.24E+00* | 59,599 | 45.27 | 2.18E-10 | 1443 | 687 | 1044.84 | 2.37E-09 | 502 | 502 |
| 4 | 750 | 0 | 8.52E-11 | 786 | 18 | 6.91 | 1.68E+01* | 72,002 | 98.26 | - | - | - | - | 1.48E-08 | 752 | 752 |
| 5 | 2 | 0 | 1.08E-12 | 5 | 2 | 0.01 | 1.59E-05 | 228 | 0.06 | 0.00E+00 | 4 | 4 | 0.00 | 1.70E-11 | 4 | 3 |
| 5 | 5 | 0 | 1.19E-07 | 8 | 2 | 0.01 | 2.67E-10 | 2831 | 0.6 | 6.13E-14 | 7 | 6 | 0.01 | 1.64E-08 | 12 | 10 |
| 5 | 10 | 0 | 3.36E-08 | 80 | 7 | 0.14 | 3.25E-07 | 24,217 | 5.8 | 3.54E-13 | 14 | 11 | 0.02 | 6.65E-08 | 14 | 11 |
| 5 | 50 | 0 | 2.11E-07 | 284 | 12 | 0.89 | 7.03E-06 | 27,477 | 14.08 | 2.59E-12 | 37 | 29 | 0.08 | 5.72E-07 | 20 | 17 |
| 5 | 100 | 0 | 3.08E-07 | 53 | 6 | 0.18 | 1.05E-05 | 27,695 | 26.43 | 2.23E-12 | 26 | 21 | 0.17 | 2.07E-05 | 56 | 48 |
| 5 | 200 | 0 | 2.56E-07 | 194 | 9 | 1.12 | 1.32E-05 | 28,172 | 48.12 | 6.24E-11 | 56 | 46 | 0.24 | 7.57E-06 | 32 | 26 |
| 5 | 300 | 0 | 1.15E-06 | 54 | 7 | 0.34 | 4.68E-06 | 27,457 | 67.43 | 1.38E-11 | 26 | 23 | 0.23 | 4.33E-06 | 24 | 19 |

Table 1 continued

| Problem data | | PLM | | | HANSO | | | PBDC | | | DCPCA | | | | | |
|--------------|--------|-----------|---------------|------|-------|---------|---------------|--------|--------|---------------|-------|-----|------|---------------|-----|-----|
| Pbm | n | \bar{f} | $f(x^{k(l)})$ | #g1 | #g2 | CPU | $f(x^{k(l)})$ | #f | CPU | $f(x^{k(l)})$ | #g1 | #g2 | CPU | $f(x^{k(l)})$ | #g1 | #g2 |
| 5 | 400 | 0 | 1.31E-06 | 82 | 17 | 0.63 | 7.11E-06 | 27,595 | 78.34 | 1.05E-10 | 42 | 32 | 0.37 | 3.43E-06 | 21 | 16 |
| 5 | 500 | 0 | 5.77E-07 | 198 | 17 | 2.01 | 2.20E-05 | 27,952 | 140.63 | 2.17E-10 | 18 | 14 | 0.17 | 1.78E-06 | 21 | 16 |
| 5 | 1000 | 0 | 7.14E-07 | 120 | 22 | 1.39 | 2.03E-05 | 28,235 | 243.95 | 3.92E-10 | 39 | 34 | 0.71 | 1.33E-06 | 19 | 14 |
| 5 | 1500 | 0 | 1.74E-06 | 181 | 32 | 3.35 | 3.02E-06 | 24,786 | 98.08 | 6.27E-10 | 20 | 16 | ‡ | 9.21E-07 | 19 | 14 |
| 5 | 3000 | 0 | 4.96E-07 | 166 | 11 | 5.8 | 2.82E-05 | 22,364 | 167.34 | 1.94E-09 | 24 | 16 | ‡ | 5.08E-07 | 20 | 16 |
| 5 | 10,000 | 0 | 1.58E-05 | 17 | 3 | 1.14 | 2.36E-06 | 26,688 | 492.4 | 5.93E-09 | 19 | 11 | ‡ | 7.44E-07 | 19 | 15 |
| 5 | 20,000 | 0 | 1.16E-05 | 16 | 2 | 1.97 | 2.87E-06 | 19,187 | 589.03 | 1.41E-03 | 43 | 39 | ‡ | 4.90E-07 | 19 | 15 |
| 5 | 50,000 | 0 | 3.74E-05 | 18 | 2 | 5.45 | 7.90E-06 | 8954 | 655.08 | 5.71E-04 | 26 | 26 | ‡ | 3.60E-07 | 19 | 15 |
| 6 | 2 | -2.5 | -2.5 | 204 | 9 | 0.52 | -2.5 | 222 | 0.02 | -2.5 | 10 | 8 | 0.00 | -2.5 | 33 | 33 |
| 7 | 2 | 0.5 | 0.5 | 1017 | 83 | 10.1 | 0.5 | 1716 | 0.2 | 0.5 | 62 | 29 | 0.01 | 0.5 | 46 | 29 |
| 8 | 3 | 3.5 | 3.77* | 71 | 7 | 0.16 | 3.5 | 605 | 0.08 | 3.5 | 32 | 15 | 0.01 | 3.5 | 26 | 20 |
| 9 | 4 | 1.83 | 9.2* | 597 | 4 | 3.88 | 1.83 | 118 | 0.02 | 1.83 | 71 | 36 | 0.01 | 1.83 | 25 | 21 |
| 10 | 2 | -0.5 | -0.5 | 4 | 4 | 0.16 | -0.5 | 38 | 0.01 | -0.5 | 4 | 3 | 0.01 | -0.5 | 10 | 9 |
| 10 | 4 | -2.5 | -0.5* | 4 | 4 | 0.45 | -2.5 | 56 | 0.01 | -2.5 | 12 | 5 | 0.01 | -2.5 | 6 | 5 |
| 10 | 5 | -2.5 | -0.5* | 4 | 4 | 0.5 | -3.5 | 42 | 0.01 | -2.5* | 8 | 6 | 0.02 | -2.5* | 18 | 17 |
| 10 | 10 | -8.5 | -0.5* | 5 | 5 | 5.3 | -8.5 | 74 | 0.01 | -8.5 | 32 | 17 | 0.01 | -8.5 | 9 | 8 |
| 10 | 50 | -48.5 | -0.5* | 6 | 6 | 54.35 | -43.5* | 81 | 0.01 | -48.5 | 120 | 53 | 0.66 | -48.5 | 30 | 29 |
| 10 | 100 | -98.5 | -0.5* | 6 | 6 | 293.08 | -84.5* | 90 | 0.02 | -98.5 | 340 | 92 | 1.46 | -98.5 | 54 | 53 |
| 10 | 150 | -148.5 | -0.5* | 6 | 6 | 709.69 | -129.5* | 90 | 0.02 | -126.49* | 442 | 103 | 1.30 | -148.5 | 79 | 78 |
| 10 | 200 | -198.5 | -0.5* | 6 | 6 | 1284.65 | -169.5* | 95 | 0.02 | -150.49* | 466 | 124 | 1.20 | -198.5 | 103 | 102 |

The asterisk (*) means that the known best value of the objective function has not been reached. The symbol (‡) means that compilation was not possible and data has been taken from [27]. “-” means that the considered algorithm failed to compute a critical point in 1 h of processing. For problem 10 the DCA subproblem (23) has an explicit solution and therefore no auxiliary algorithm was employed by DCA-CPM, DCA-LBM and PLM

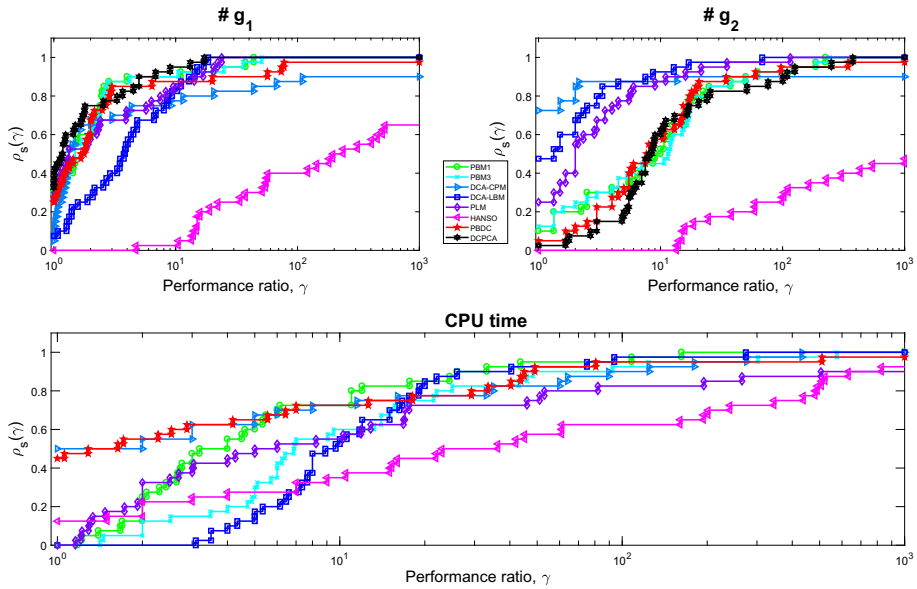


Fig. 1 Performance profile of the results presented in Table 1

Solver HANSO (for general nonconvex optimization problems) was the only solver that could solve globally problem 3 with $n = 5$. Overall, this solver performed a larger number of function/subgradient evaluations than the other solvers that exploited the DC structure of the objective function. Solver DCPCA required less subgradient evaluations of g_1 than solver PBM1 did, however the former needed more subgradient evaluations of g_2 .

Notice also that when employing a convex cutting-plane model (solver PBM1) the total number of iterations and computation burden do not necessary increase when compared to PBM3, PBDC and DCPCA. Moreover, the quality of final solution candidates is not deteriorated.

We summarize Table 1 in Fig. 1 by employing performance profiles [11]. For example, let the criterion be CPU time. For each solver, we plot the proportion of problems that is solved within a factor γ of the time required by the best algorithm. More specifically, denoting by $t_s(p)$ the time spent by solver s to solve problem p and by $t^*(p)$ the best time for the same problem among all the solvers, the proportion of problems solved by s within a factor γ is

$$\rho_s(\gamma) := \frac{\text{number of problems } p \text{ such that } t_s(p) \leq \gamma t^*(p)}{\text{total number of problems}}.$$

Therefore, the value $\rho_s(1)$ gives the probability of the solver s to be the best by a given criterion. Furthermore, unless $t_s(p) = \infty$ (which means that solver s failed to solve problem p), it follows that $\lim_{\gamma \rightarrow \infty} \rho_s(\gamma) = 1$. Thus, the higher is the line the better is the solver (by this criterion).

Figure 1 shows that the four proximal bundle solvers perform better than the other solvers in both number of subgradient evaluations of f_1 and robustness. However, in terms of subgradient evaluations of f_2 the DCA solvers DCA-CPM and DCA-LBM performed better. Overall, solver PBM1 presents a good compromise between CPU time and robustness.

6.2 Linearly-constrained DC programs

In this subsection, we consider two convex constrained DC programs obtained by approximating chance-constrained problems of the form

$$\begin{cases} \min_x \langle q, x \rangle \\ \text{s.t. } Ax = b \\ \mathbb{P}[c(x, \xi) \leq 0] \geq p \\ \underline{x} \leq x \leq \bar{x}, \end{cases} \tag{24}$$

where $\xi \in \mathcal{E} \subset \mathbb{R}^m$ is a random vector having probability measure \mathbb{P} , $A \in \mathbb{R}^{s \times n}$ and $b \in \mathbb{R}^s$. Parameter $p \in (0, 1)$ is a confidence level and $c : \mathbb{R}^n \times \mathcal{E} \rightarrow \mathbb{R}$ is a given DC function (not necessary differentiable). It is well known that the above problem may fail to be convex even when $c(\cdot, \cdot)$ is convex in both arguments. Moreover, evaluating the probability function for a given point involves computing a multidimensional integral. This is a difficult task when random variables have large dimension [47], and therefore Monte-Carlo simulation is an important alternative to approximate \mathbb{P} . We refer to [19,39,44,46] for more details on chance-constrained programming.

As discussed in [25], if $c(\cdot, \xi)$ is a DC function the probability constraint $\mathbb{P}[c(x, \xi) \leq 0] \geq p$ can be approximated by the DC constraint $\mathbb{E}[c(x, \xi) + t]^+ - \mathbb{E}[c(x, \xi)]^+ \leq t(1 - p)$, where $t \approx 0$ is a positive parameter, $[a]^+ := \max\{a, 0\}$ and $\mathbb{E}[\cdot]$ is the expected value operator w.r.t. \mathbb{P} . By penalizing this constraint with¹ $\rho > 0$ we get the following approximation of (24):

$$\begin{cases} \min_x \langle q, x \rangle + \rho \left[\mathbb{E}[c(x, \xi) + t]^+ - \mathbb{E}[c(x, \xi)]^+ - t(1 - p) \right]^+ \\ \text{s.t. } Ax = b \\ \underline{x} \leq x \leq \bar{x}, \end{cases} \tag{25}$$

that can be written as a DC program

$$\begin{cases} \min_x f_1(x) - f_2(x) \\ \text{s.t. } Ax = b \\ \underline{x} \leq x \leq \bar{x}, \end{cases}$$

with $f_1(x) = \langle q, x \rangle + \rho \max(\mathbb{E}[c(x, \xi) + t]^+ - t(1 - p), \mathbb{E}[c(x, \xi)]^+)$ and $f_2(x) = \rho \mathbb{E}[c(x, \xi)]^+$. The expectations $\mathbb{E}[c(x, \xi) + t]^+$ and $\mathbb{E}[c(x, \xi)]^+$ can be approximated by Monte-Carlo simulation [25]: in our numerical experiments we have used a fixed sample of 10,000 scenarios randomly generated accordingly to the distribution of ξ .

6.2.1 PlanToy: an academic chance-constrained planning problem

We consider a small example of a management problem coming from the industry of energy. The problem consists of planing two fictitious refineries for producing two types of fuel to meet a demand that is deterministic in the first month of planning, but uncertain in the second one. The decision maker wishes to make an optimal decision on the amount of processed petrol by the two refineries (x_1 and x_2 for the first month, and x_5 and x_6 for the second one), storing petrol (x_3 and x_7), and importation (x_4 and x_8). The storage and importation decisions must ensure that the second-month demand is satisfied with probability p . The fictitious planning problem (PlanToy) reads as

¹ We have used $\rho = 10^4 \times n$ in our numerical experiments.

$$\begin{cases} \min_{x \geq 0} & 2x_1 + 3x_2 + 0.5x_3 + 12x_4 + 2x_5 + 3x_6 + 0.5x_7 + 12.5x_8 \\ \text{s.t.} & 2x_1 + 6x_2 = 190 \\ & 3x_1 + 2.8x_2 = 168 \\ & x_1 + x_2 + x_3 - x_4 = 60 \\ & -x_3 + x_5 + x_6 + x_7 - x_8 = 47.16 \\ & \mathbb{P}[c(x, \xi) \leq 0] \geq p \\ & x_3 \leq 10, x_7 \leq 10, \end{cases} \tag{26}$$

where $c(x, \xi) = \max\{\xi_1 - 2x_5 - 6x_6, \xi_2 - 3x_5 - 2.8x_6\}$, and $\xi = (\xi_1, \xi_2)$ is a random vector (of fuel demand) following a normal distribution with mean $\mathbb{E}[\xi] = (193, 178)$ and covariance matrix

$$C = \begin{pmatrix} 9 & \text{Cov}(\xi_1, \xi_2) \\ \text{Cov}(\xi_2, \xi_1) & 10.24 \end{pmatrix}.$$

Under this assumption the probability constraint $\mathbb{P}[c(x, \xi) \leq 0] \geq p$ can be replaced with the convex one $\phi(x) \leq 0$ with $\phi(x) := \log(p) - \log(\mathbb{P}[c(x, \xi) \leq 0])$ [39]. For comparison purposes, the resulting convex optimization problem was solved by using the MATLAB optimization routine `fmincon`. We employed the Matlab’s statistical function `mvncdf` to evaluate the probability function:

$$\mathbb{P}[c(x, \xi) \leq 0] = \mathbb{P} \begin{bmatrix} \xi_1 \leq 2x_5 + 6x_6 \\ \xi_2 \leq 3x_5 + 2.8x_6 \end{bmatrix}.$$

Such a convex reformulation is possible thanks to the assumption on the probability distribution of the random vector ξ . For more general (and possibly more realistic) distributions, the PlanToy cannot be recast into a convex formulation. We refer the interested reader to textbook [39] for a comprehensive discussion on convexity of chance-constrained programs.

Regardless convexity, evaluating the probability function is a difficult task in general. Therefore, the Monte-Carlo approach and DC approximation discussed above become attractive in the large-scale setting.

By varying the confidence level $p \in \{0.5, 0.6, 0.7, 0.8, 0.9, 0.95\}$ and the covariance coefficient $\text{Cov} := \text{Cov}(\xi_1, \xi_2) \in \{-4.8, 0, 4.8\}$ we ended up with 18 variants of PlanToy. Table 2 reports the results obtained by applying the solvers PBM1, PBM3 and PLM on these instances. As PLM, both solvers DCA-CPM and DCA-LBM had a similar and unsatisfactory performance on these instances. Solvers HANSO and PBDC were not applied because they do not handle constrained problems.

As starting points for these solvers, we have considered the solution of the simpler individual chance-constrained program obtained from (26) by replacing the joint probability constraint $\mathbb{P}[c(x, \xi) \leq 0] \geq p$ with the individual ones $\mathbb{P}_1[\xi_1 \leq 2x_5 + 6x_6] \geq p$ and $\mathbb{P}_2[\xi_2 \leq 3x_5 + 2.8x_6] \geq p$, where $\xi_1 \sim N(193, 9)$ and $\xi_2 \sim N(178, 10.4)$ due to the assumptions on the joint distribution of $\xi = (\xi_1, \xi_2)$. By making use of p -quantiles (that can be easily computed by several statistical softwares), the above constraints are in fact linear ones:

$$2x_5 + 6x_6 \geq \mathbb{P}_1^{-1}[p] \quad \text{and} \quad 3x_5 + 2.8x_6 \geq \mathbb{P}_2^{-1}[p].$$

Therefore, problem (26) with $\mathbb{P}[c(x, \xi) \leq 0] \geq p$ replaced with the these two linear constraints is a mere linear programming problem approximating the nonlinear one (26) (see [39] for more details). This is why we consider such an approximation only for computing an initial point for our solvers.

Table 2 Linearly-constrained DC problem: numerical results on 18 instances of the PlanToy problem. All the solvers were initialized with the same starting point

| Problem data | | $f(x^{k(\ell)})$ | | | | | | $\mathbb{P}[c(\bar{x}, \xi) \leq 0]$ | | | | | | # g_1 | | | | | | # g_2 | | | | | | CPU time | | | | | |
|--------------|------|------------------|---------|---------|---------|-------|-------|--------------------------------------|------|------|-----|------|------|---------|------|------|-------|------|------|---------|------|------|-----|------|------|----------|--|--|--|--|--|
| p | Cov | \bar{f} | PBM1 | PBM3 | PLM | PBM1 | PBM3 | PLM | PBM1 | PBM3 | PLM | PBM1 | PBM3 | PLM | PBM1 | PBM3 | PLM | PBM1 | PBM3 | PLM | PBM1 | PBM3 | PLM | PBM1 | PBM3 | PLM | | | | | |
| 0.50 | -4.8 | 410.282 | 410.322 | 410.478 | 472.213 | 0.499 | 0.500 | 0.167 | 34 | 36 | 35 | 10 | 9 | 4 | 1.24 | 1.36 | 1.56 | | | | | | | | | | | | | | |
| 0.60 | -4.8 | 413.986 | 414.734 | 414.744 | 473.210 | 0.604 | 0.604 | 0.284 | 23 | 44 | 39 | 5 | 11 | 4 | 0.74 | 1.54 | 1.63 | | | | | | | | | | | | | | |
| 0.70 | -4.8 | 417.977 | 418.930 | 418.930 | 469.323 | 0.705 | 0.705 | 0.435 | 28 | 33 | 98 | 7 | 7 | 9 | 0.82 | 1.06 | 3.64 | | | | | | | | | | | | | | |
| 0.80 | -4.8 | 422.690 | 422.813 | 422.813 | 459.128 | 0.802 | 0.802 | 0.609 | 29 | 27 | 52 | 9 | 7 | 5 | 0.78 | 0.78 | 1.72 | | | | | | | | | | | | | | |
| 0.90 | -4.8 | 429.297 | 436.510 | 429.277 | 446.815 | 0.849 | 0.899 | 0.802 | 25 | 70 | 69 | 5 | 26 | 9 | 0.64 | 1.93 | 1.91 | | | | | | | | | | | | | | |
| 0.95 | -4.8 | 434.808 | 435.105 | 434.570 | 442.713 | 0.948 | 0.947 | 0.900 | 28 | 33 | 91 | 7 | 10 | 25 | 0.61 | 0.83 | 2.56 | | | | | | | | | | | | | | |
| 0.50 | 0 | 409.928 | 409.946 | 409.909 | 455.989 | 0.498 | 0.498 | 0.251 | 28 | 30 | 79 | 12 | 11 | 10 | 0.95 | 1.10 | 3.52 | | | | | | | | | | | | | | |
| 0.60 | 0 | 413.723 | 416.458 | 414.855 | 458.085 | 0.597 | 0.599 | 0.361 | 52 | 48 | 54 | 17 | 19 | 5 | 1.65 | 1.68 | 2.09 | | | | | | | | | | | | | | |
| 0.70 | 0 | 417.795 | 417.847 | 417.802 | 457.041 | 0.699 | 0.699 | 0.492 | 32 | 30 | 85 | 9 | 5 | 10 | 0.93 | 0.95 | 3.08 | | | | | | | | | | | | | | |
| 0.80 | 0 | 422.579 | 423.152 | 423.206 | 452.437 | 0.805 | 0.805 | 0.640 | 28 | 36 | 70 | 8 | 8 | 11 | 0.72 | 1.02 | 2.29 | | | | | | | | | | | | | | |
| 0.90 | 0 | 429.247 | 429.917 | 429.689 | 445.254 | 0.899 | 0.901 | 0.811 | 25 | 33 | 80 | 5 | 9 | 11 | 0.58 | 0.86 | 2.23 | | | | | | | | | | | | | | |
| 0.95 | 0 | 434.786 | 434.892 | 434.877 | 442.797 | 0.949 | 0.950 | 0.903 | 39 | 39 | 72 | 12 | 9 | 21 | 0.90 | 0.97 | 2.03 | | | | | | | | | | | | | | |
| 0.50 | 4.8 | 409.138 | 408.958 | 409.077 | 437.366 | 0.495 | 0.494 | 0.341 | 36 | 39 | 300 | 14 | 15 | 40 | 1.21 | 1.43 | 13.15 | | | | | | | | | | | | | | |
| 0.60 | 4.8 | 413.046 | 414.750 | 412.858 | 441.875 | 0.601 | 0.595 | 0.439 | 42 | 31 | 71 | 11 | 7 | 8 | 1.32 | 1.04 | 2.64 | | | | | | | | | | | | | | |
| 0.70 | 4.8 | 417.229 | 417.160 | 417.161 | 444.201 | 0.698 | 0.698 | 0.558 | 31 | 35 | 114 | 9 | 12 | 13 | 0.89 | 1.08 | 3.97 | | | | | | | | | | | | | | |
| 0.80 | 4.8 | 422.130 | 424.528 | 424.528 | 441.924 | 0.787 | 0.787 | 0.693 | 51 | 50 | 234 | 29 | 29 | 35 | 1.39 | 1.45 | 7.80 | | | | | | | | | | | | | | |
| 0.90 | 4.8 | 428.937 | 429.813 | 429.341 | 440.957 | 0.896 | 0.903 | 0.833 | 26 | 36 | 40 | 6 | 7 | 8 | 0.60 | 0.94 | 1.10 | | | | | | | | | | | | | | |
| 0.95 | 4.8 | 434.566 | 435.381 | 434.821 | 440.713 | 0.942 | 0.951 | 0.912 | 25 | 38 | 70 | 6 | 11 | 20 | 0.55 | 0.92 | 1.91 | | | | | | | | | | | | | | |

Solver PLM failed to globally solve all the considered instances of PlanToy: the method stopped with a critical point of the penalized DC program (25), which is not even a feasible one. On the other hand, the proximal bundle solvers were successful.

We care to mention that the obtained function value has three sources of inaccuracy: (1) the approximation of the probability distribution by a sample of 10,000 scenarios, (2) approximation of the characteristic function $\mathbf{1}_{[0, \infty)}(z)$ by $([z + t]^+ - [z]^+)/t$, and (3) the solvers tolerance. As a result, one cannot expect that the obtained function values coincide with the (approximate) optimal value \bar{f} . Nevertheless, the function values computed by solvers PBM1 and PBM3 are very close to the optimal values and the obtained solutions are (nearly) feasible: the columns 7–9 of Table 2 report the probability² of the computed point \bar{x} to satisfy the random constraint $c(x, \xi) \leq 0$.

Once again, PBM1 (with a convex model) was as precise as PBM3 (with a DC model) but faster than the latter on these instances of PlanToy.

6.2.2 A norm optimization problem with chance constraints

We now consider a DC approximation of the chance-constrained program

$$\begin{cases} \min_{x \in \mathbb{R}_+^n} & -\|X\|_1 \\ \text{s.t} & \mathbb{P}[\|\xi x\|_2 \leq 10] \geq p \end{cases} \tag{27}$$

given in [25, Sect. 5.1]. This problem fits (24) with $c(x, \xi) = \max_{i=1, \dots, 10} \{\sum_{j=1}^n \xi_{ij}^2 x_j^2 - 100\}$, where ξ_{ij} ($i = 1, \dots, n$ and $j = 1, \dots, 10$) are independent and identically distributed standard normal random variables. The DC reformulation of this problem is

$$\min_{x \in \mathbb{R}_+^n} f_1(x) - f_2(x),$$

with $f_1(x) = -\sum_{i=1}^n x_i + \rho \max(\mathbb{E}[c(x, \xi) + t]^+ - t(1 - p), \mathbb{E}[c(x, \xi)]^+)$ and $f_2(x) = \rho \mathbb{E}[c(x, \xi)]^+$.

In order to approximate the expectation \mathbb{E} , we have used Monte-Carlo simulation with $N = 10,000$ scenarios. As discussed in [25], when ξ_{ij} are independent and identically distributed standard normal random variables the global solution and optimal value of (27) are, respectively,

$$\bar{x}_i = \frac{10}{\sqrt{F_{\chi_n^2}^{-1}(p^{1/10})}}, \quad i = 1, \dots, n, \quad \text{and} \quad \bar{f} = \frac{10n}{\sqrt{F_{\chi_n^2}^{-1}(p^{1/10})}},$$

where $F_{\chi_n^2}^{-1}$ denotes the inverse distribution function of a Chi-square distribution with n degrees of freedom. Table 3 reports some results obtained by applying the six solvers on 18 instances of (27).

Since computing exactly the constraint $\mathbb{P}[c(x, \xi) \leq 0]$ is a difficult task, we do not report in Table 3 the probability value $\mathbb{P}[c(\bar{x}, \xi) \leq 0]$ at the obtained candidate solutions. However, we estimated this probability by the DC approximation mentioned above. We observed that in all the considered instances the value $(\mathbb{E}[c(\bar{x}, \xi) + t]^+ - \mathbb{E}[c(\bar{x}, \xi)]^+)/t$ coincides with the given confidence level p for the solvers PBM1 and PBM3. This ensures that the provided candidate solutions are feasible for the DC formulation above and, therefore, nearly feasible for the considered norm optimization problem with chance-constraints.

² Probability computed by using the mvncdf Matlab’s function.

Table 3 Linearly-constrained DC problem: numerical results on 18 instances of the norm optimization problem with chance constraint. All the solvers were initialized with the same starting points, chosen randomly from $[0, 0.2]^n$

| n | \bar{f} | PEM1 | | PEM3 | | DCA-CPM | | DCA-LBM | | FLM | | HANSO | | | | | | | |
|-----|-----------|------------------|--------------|------|------------------|--------------|-----|------------------|--------------|------|------------------|--------------|------|------------------|--------------|--------|--------|------|--------|
| | | $f(x^k(\theta))$ | \mathbb{P} | CPU | $f(x^k(\theta))$ | \mathbb{P} | CPU | $f(x^k(\theta))$ | \mathbb{P} | CPU | $f(x^k(\theta))$ | \mathbb{P} | CPU | $f(x^k(\theta))$ | \mathbb{P} | CPU | | | |
| 2 | -7.2 | -7.22 | 0.80 | 1.0 | -7.22 | 0.80 | 1.1 | -4.56 | 1.00 | 0.2 | -4.56 | 1.00 | 0.2 | -4.69 | 1.00 | 66.3 | -7.23 | 0.80 | 91.0 |
| 4 | -11.8 | -11.79 | 0.80 | 1.6 | -11.79 | 0.80 | 1.8 | -8.09 | 1.00 | 0.4 | -8.09 | 1.00 | 0.2 | -8.09 | 1.00 | 0.3 | -11.79 | 0.80 | 188.1 |
| 5 | -13.8 | -13.76 | 0.80 | 1.4 | -13.76 | 0.80 | 1.3 | -10.09 | 1.00 | 0.5 | -10.09 | 1.00 | 0.5 | -10.11 | 1.00 | 0.7 | -13.68 | 0.80 | 187.5 |
| 10 | -21.9 | -21.89 | 0.80 | 1.4 | -21.89 | 0.80 | 1.5 | -17.49 | 0.99 | 1.9 | -17.49 | 0.99 | 0.5 | -17.49 | 1.00 | 2.7 | -21.77 | 0.80 | 551.4 |
| 20 | -34.0 | -34.03 | 0.80 | 2.2 | -34.03 | 0.80 | 2.4 | -29.48 | 0.99 | 21.9 | -29.48 | 1.00 | 0.8 | -29.48 | 0.99 | 32.6 | -33.36 | 0.80 | 1551.4 |
| 50 | -58.9 | -58.89 | 0.80 | 3.1 | -58.89 | 0.80 | 3.3 | † | † | † | -53.66 | 1.00 | 1.8 | -53.65 | 0.99 | 299.8 | -46.82 | 0.80 | 2128.0 |
| 100 | -87.6 | -87.39 | 0.80 | 4.1 | -87.39 | 0.80 | 4.3 | † | † | † | -82.80 | 0.99 | 3.5 | -82.80 | 0.95 | 585.8 | - | - | - |
| 150 | -109.7 | -109.63 | 0.80 | 7.1 | -109.63 | 0.80 | 7.1 | † | † | † | -105.64 | 0.99 | 9.3 | -105.64 | 0.94 | 919.3 | - | - | - |
| 200 | -128.5 | -128.46 | 0.80 | 8.9 | -128.46 | 0.80 | 9.5 | † | † | † | -124.67 | 0.96 | 20.1 | -124.67 | 0.92 | 1572.4 | - | - | - |
| 2 | -6.6 | -6.59 | 0.90 | 0.8 | -6.59 | 0.90 | 0.9 | -4.56 | 1.00 | 0.1 | -4.56 | 1.00 | 0.3 | -4.71 | 1.00 | 80.2 | -6.60 | 0.90 | 105.7 |
| 4 | -11.0 | -11.02 | 0.90 | 1.8 | -11.02 | 0.90 | 2.1 | -8.09 | 1.00 | 0.4 | -8.09 | 1.00 | 0.2 | -8.09 | 1.00 | 0.3 | -10.93 | 0.90 | 184.2 |
| 5 | -12.9 | -12.92 | 0.90 | 1.4 | -12.92 | 0.90 | 1.8 | -10.09 | 1.00 | 0.5 | -10.09 | 1.00 | 0.5 | -10.09 | 1.00 | 0.6 | -12.90 | 0.90 | 231.9 |
| 10 | -20.8 | -20.79 | 0.90 | 1.9 | -20.79 | 0.90 | 2.1 | -17.49 | 0.99 | 2.0 | -17.49 | 0.99 | 0.5 | -17.49 | 0.99 | 2.1 | -20.51 | 0.90 | 521.4 |
| 20 | -32.7 | -32.75 | 0.90 | 2.8 | -32.75 | 0.90 | 4.7 | -29.48 | 0.99 | 21.7 | -29.48 | 1.00 | 0.9 | -29.49 | 0.99 | 90.5 | -32.24 | 0.90 | 1533.2 |
| 50 | -57.4 | -57.27 | 0.90 | 4.0 | -57.26 | 0.90 | 3.0 | † | † | † | -53.66 | 1.00 | 2.0 | -53.65 | 0.98 | 244.3 | -38.74 | 0.90 | 1990.8 |
| 100 | -85.9 | -85.88 | 0.90 | 5.1 | -85.88 | 0.90 | 5.4 | † | † | † | -82.80 | 1.00 | 4.6 | -82.80 | 0.96 | 667.4 | - | - | - |
| 150 | -108.0 | -107.94 | 0.90 | 6.1 | -107.94 | 0.90 | 6.3 | † | † | † | -105.64 | 0.99 | 7.1 | -105.64 | 0.94 | 1005.4 | - | - | - |
| 200 | -126.7 | -126.78 | 0.90 | 8.8 | -126.53 | 0.90 | 7.9 | † | † | † | -124.67 | 0.95 | 36.7 | -124.67 | 0.93 | 1640.4 | - | - | - |

The symbol “-” means that the considered algorithm failed to compute a critical point in 1 h of processing, and “†” means that the considered solver failed in computing a critical point. The first eight results were obtained with $p = 0.8$ and the last eight with $p = 0.9$

Once again, we cannot expect that the obtained function values coincide with the optimal value \bar{f} due to the reasons previously mentioned. However, we can see from Table 3 that the bundle solvers provided good estimates of the optimal values. Solvers DCA-LBM and PLM computed critical points for all problem instances, however they failed to compute global solutions. In order to apply HANSO, we dropped the (inactive) constraint $x \geq 0$. After this solver HANSO could solve up to optimality only the smaller instances and find a critical point for the medium size ones, but could not solve the larger instances in a time limit of 1 h. The reason is that HANSO requires many function evaluations: the considered function requires Monte-Carlo simulation and is therefore difficult to evaluate. Table 4 reports the number of subgradient evaluations.

6.3 Quadratically-constrained DC programs

We now consider the following quadratically-constrained DC program (QCDC)

$$\bar{f} := \begin{cases} \min_{x \in \mathbb{R}^n} \min_{j \in \{1, \dots, m\}} \left\{ \frac{1}{2} \|x - c^j\|_2^2 \right\} \\ \text{s.t. } \frac{1}{2} \sum_{i=1}^n \alpha_i x_i^2 \leq \frac{1}{2} K^2, \end{cases} \tag{28}$$

with given data $K \in \mathbb{R}_+$, $\alpha \in \mathbb{R}_+^n$ and $c^j \in \mathbb{R}^n$, $j = 1, \dots, m$. The optimal value (and a solution) of the above QCDC can be found by solving individually the m quadratically-constrained QP programs (QCQP)

Table 4 Norm optimization problem with chance constraint: number of subgradients evaluations

| n | PBM1 | | PBM3 | | DCA-CPM | | DCA-LBM | | PLM | | HANSO # f |
|-----|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|----------------|
| | # g_1 | # g_2 | # g_1 | # g_2 | # g_1 | # g_2 | # g_1 | # g_2 | # g_1 | # g_2 | |
| 2 | 32 | 8 | 33 | 8 | 33 | 2 | 44 | 3 | 2372 | 463 | 3239 |
| 4 | 49 | 10 | 49 | 10 | 88 | 2 | 38 | 3 | 69 | 2 | 6399 |
| 5 | 41 | 10 | 36 | 8 | 118 | 2 | 81 | 2 | 131 | 5 | 6008 |
| 10 | 37 | 6 | 37 | 6 | 348 | 2 | 71 | 2 | 300 | 3 | 16,125 |
| 20 | 49 | 10 | 49 | 10 | 1413 | 2 | 95 | 2 | 934 | 2 | 39,672 |
| 50 | 47 | 6 | 47 | 6 | † | † | 119 | 2 | 2306 | 4 | 37,020 |
| 100 | 41 | 5 | 41 | 5 | † | † | 138 | 2 | 2116 | 2 | – |
| 150 | 53 | 9 | 50 | 9 | † | † | 196 | 2 | 2317 | 2 | – |
| 200 | 53 | 10 | 53 | 10 | † | † | 244 | 2 | 2626 | 2 | – |
| 2 | 27 | 7 | 27 | 7 | 33 | 2 | 44 | 3 | 2699 | 535 | 4309 |
| 4 | 57 | 15 | 60 | 15 | 88 | 2 | 38 | 3 | 69 | 2 | 7116 |
| 5 | 45 | 7 | 51 | 9 | 122 | 2 | 81 | 2 | 108 | 2 | 8253 |
| 10 | 52 | 9 | 52 | 9 | 358 | 2 | 69 | 2 | 252 | 2 | 17,158 |
| 20 | 64 | 10 | 96 | 11 | 1397 | 2 | 95 | 2 | 1501 | 9 | 41,973 |
| 50 | 62 | 15 | 43 | 9 | † | † | 121 | 2 | 1887 | 3 | 35,239 |
| 100 | 51 | 7 | 51 | 7 | † | † | 152 | 2 | 2125 | 2 | – |
| 150 | 45 | 5 | 45 | 5 | † | † | 173 | 2 | 2299 | 2 | – |
| 200 | 53 | 9 | 45 | 6 | † | † | 326 | 2 | 2628 | 2 | – |

The symbol “–” means that the considered algorithm failed to compute a critical point in 1 h of processing, and “†” means that the considered solver failed in computing a critical point. The first eight results were obtained with $p = 0.8$ and the last eight with $p = 0.9$

$$v_j := \begin{cases} \min_{x \in \mathbb{R}^n} \frac{1}{2} \|x - c^j\|_2^2 \\ \text{s.t. } \frac{1}{2} \sum_{i=1}^n \alpha_i x_i^2 \leq \frac{1}{2} K^2, \end{cases}$$

and taking $\bar{f} = \min_{j \in \{1, \dots, m\}} v_j$. Problem (28) has the following DC representation

$$\begin{cases} \min_{x \in \mathbb{R}^n} f_1(x) - f_2(x) \\ \text{s.t. } \frac{1}{2} \sum_{i=1}^n \alpha_i x_i^2 \leq \frac{1}{2} K^2, \end{cases}$$

with

$$f_1(x) := \frac{1}{2} \sum_{j=1}^m \|x - c^j\|_2^2 \quad \text{and} \quad f_2(x) := \frac{1}{2} \max_{l \in \{1, \dots, m\}} \left\{ \sum_{j \neq l} \|x - c^j\|_2^2 \right\}.$$

Several instances of problem (28) were generated by the following scheme: we set $K = 10$, $m = 5$ and α_i drawn randomly and uniformly from $[0, 1]$ for all $i = 1, \dots, n$. Vectors c^j , $j = 1, \dots, m$, were constructed in such a way that no c^j is feasible for (28):

$$c^j := (1 + s_j)K \frac{\tilde{c}^j}{\sqrt{\sum_{i=1}^n \alpha_i (\tilde{c}_i^j)^2}},$$

where $\tilde{c}_i^j, i = 1, \dots, n, j = 1, \dots, m$, was randomly generated following a standard normal distribution and s_j uniformly from the set $\{1, 2, 3, 4, 5\}$.

For the QCDC problem (28), the proximal bundle method subproblem (9) becomes a QCQP (once again with the choice $D(x, \hat{x}) := \|x - \hat{x}\|_2^2/2$) that can be solved by specialized algorithms. In this study we applied GUROBI. Table 5 reports some numerical results obtained by applying solvers PBM1, PBM3, DCA-CPM and PLM to some instances of (28).

Solver DCA-CPM (respectively PLM) solves # g_2 linear (respectively quadratic) optimization subproblems with quadratic constraints per iteration. Solver PBM1 (respectively PBM3) requires solving # g_1 (respectively three times # g_1) QCQPs. This explains why PBM1 is faster than the other considered solvers. Once again, the proximal bundle solvers were successful in computing global solutions.

6.4 Computing d -stationary points

In this section, we illustrate the performance of Algorithm 2 on two additional test problems whose second DC component is the pointwise maximum of finitely many differentiable functions. We start with the following two-dimensional problem

$$\min_{x \in \mathbb{R}^2} f_1(x) - f_2(x) \quad \text{s.t.} \quad -x_1 + x_2 \leq 1, \tag{29}$$

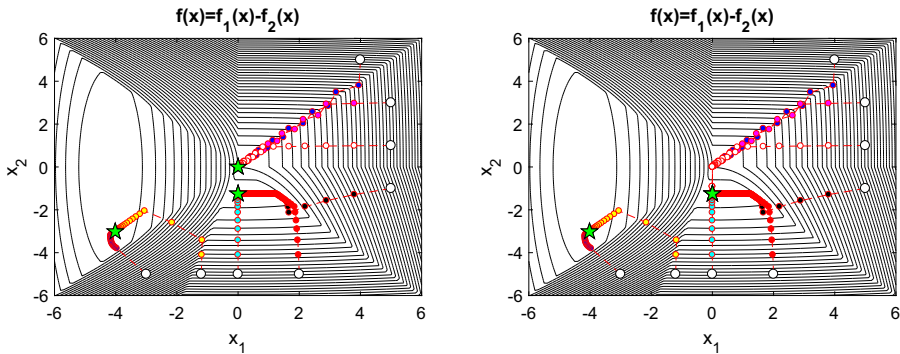
with $f_1(x) = 1.2 \max\{0.1x_1^2 + 0.005x_2^2, 0.005x_1^2 + 0.1x_2^2\}$ and $f_2(x) = \max\{-x_1, -0.3x_2, 0\}$. Note that the point $\tilde{x} = (0, 0)$ is critical for this problem, however \tilde{x} is not d -stationary since small negative perturbations of its components yield smaller function values. Figure 2a presents eight different sequences obtained by PBM1 initialized with eight different starting points. In Fig. 2b we consider PBM-d with the same initial points. Notice that PBM-d does not stop at the critical point $\tilde{x} = (0, 0)$, since near to \tilde{x} the algorithm solves two QPs (19) and can escape from \tilde{x} .

A comparison of PBM-d with other solvers is given in Table 6. We do not report CPU time because all the solvers computed a critical point in less than half a second. As PBM-d,

Table 5 Quadratically constrained DC programs: function value, subgradient evaluations and CPU time

| Data | PBML | | | PBM3 | | | DCA-CPM | | | PLM | | | | |
|--------|-----------|------------------|------------------|------|-----|--------|------------------|-----|-----|--------|------------------|-----|-----|---------|
| | \bar{f} | $f(x^{k(\ell)})$ | $f(x^{k(\ell)})$ | #g1 | #g2 | CPU | $f(x^{k(\ell)})$ | #g1 | #g2 | CPU | $f(x^{k(\ell)})$ | #g1 | #g2 | CPU |
| 2 | 366.20 | 366.2 | 366.2 | 2 | 1 | 0.05 | 366.2 | 2 | 1 | 0.04 | 366.44 | 9 | 38 | 0.09 |
| 5 | 215.06 | 215.06 | 215.06 | 2 | 1 | 0.01 | 215.06 | 2 | 1 | 0.01 | 216.44 | 6 | 32 | 0.07 |
| 10 | 376.58 | 376.58 | 376.58 | 29 | 6 | 0.19 | 376.58 | 17 | 2 | 0.16 | 378.33 | 7 | 156 | 0.79 |
| 50 | 257.94 | 257.94 | 257.94 | 5 | 1 | 0.03 | 257.94 | 5 | 1 | 0.04 | 258.77 | 10 | 426 | 18.95 |
| 100 | 852.54 | 857.67 | 857.67 | 24 | 6 | 0.31 | 857.67 | 23 | 8 | 0.69 | 859.69 | 7 | 356 | 10.87 |
| 200 | 837.69 | 837.69 | 837.69 | 17 | 4 | 0.3 | 837.69 | 18 | 4 | 1.15 | 839.59 | 7 | 359 | 22.64 |
| 300 | 302.45 | 302.45 | 302.45 | 17 | 4 | 0.52 | 302.45 | 18 | 4 | 1.4 | 303.53 | 8 | 381 | 27.71 |
| 350 | 281.12 | 281.12 | 281.12 | 29 | 7 | 0.94 | 281.12 | 20 | 5 | 1.87 | 282.1 | 8 | 384 | 34.54 |
| 500 | 270.79 | 270.79 | 270.79 | 19 | 4 | 0.64 | 270.79 | 18 | 3 | 1.49 | 271.79 | 8 | 391 | 44.99 |
| 1000 | 288.35 | 288.35 | 288.35 | 16 | 3 | 1.1 | 288.35 | 22 | 6 | 5.2 | 288.99 | 9 | 394 | 70.97 |
| 1500 | 282.14 | 282.14 | 282.14 | 19 | 3 | 3.12 | 282.14 | 26 | 6 | 11.93 | 283.73 | 8 | 380 | 92.62 |
| 3000 | 281.89 | 281.89 | 281.89 | 27 | 6 | 11.61 | 281.89 | 16 | 3 | 10.63 | 282.78 | 8 | 394 | 195.01 |
| 10,000 | 284.31 | 284.31 | 284.31 | 18 | 3 | 37.3 | 284.31 | 21 | 4 | 99.22 | 285.96 | 8 | 392 | 651.63 |
| 20,000 | 289.15 | 289.15 | 289.15 | 20 | 3 | 113.11 | 289.15 | 29 | 6 | 342.7 | 289.68 | 9 | 389 | 1290.82 |
| 50,000 | 284.91 | 284.91 | 284.91 | 18 | 4 | 384.52 | 284.91 | 24 | 4 | 965.91 | 286.08 | 8 | 386 | 3241.62 |

The symbol “-” means that the considered solver failed to compute a critical point in 1 h of processing. All solvers started with the same initial points randomly and sparsely generated from $[0, 1]^n$



(a) Results of Algorithm 1 (solver PBM1) initialized with eight different starting points: three initial points lead to the critical point (0, 0) that is not *d*-stationary, three initial points lead to the local minimum (0, -1.25) and two initial points lead to the global solution $\bar{x} \approx (-4.01582, -3.01582)$.

(b) Results of Algorithm 2 (solver PBM-d) initialized with eight different starting points: all the initial points lead to *d*-stationary points, which are the local and global solutions.

Fig. 2 Level curves of $f(x) = f_1(x) - f_2(x)$, with $f_1(x) = 1.2 \max\{0.1x_1^2 + 0.005x_2^2, 0.005x_1^2 + 0.1x_2^2\}$ and $f_2(x) = \max\{-x_1, -0.3x_2, 0\}$

solver DCA-LBM also computed *d*-stationary points for problem (29) regardless the initial point.

We have also considered an unconstrained variant of problem (29). For the same starting points the obtained results are similar to the ones presented in Table 6, with the difference that the global solution is $\bar{x} \approx (-4.1667, 0.000)$ (with optimal value approximately -2.08333). In this setting, we have also considered the solver PBDC that succeed in computing the global solution for every starting point of Table 6. However, if one starts nearly zero [e.g. $x^0 = (0.001, 0.001)$] then PBDC converges to the critical point (0, 0), whereas PBM-d always converges to a *d*-stationary point (either the global or the local solution of the problem).

Finally, we consider the following test problem

$$\min_{x \in \mathbb{R}^n} \sum_{i=1}^n (x_i + (-1)^i) + 2 \min_{i \in \{1,2,\dots,n\}} \{x_i\}, \tag{30}$$

which has the DC decomposition

$$f_1(x) = \sum_{i=1}^n (x_i + (-1)^i) + 2 \sum_{i=1}^n x_i \text{ and } f_2(x) = 2 \max_{i \in \{1,2,\dots,n\}} \left\{ \sum_{j=1, j \neq i}^n x_j \right\}.$$

Its optimal value is -4 obtained with $\bar{x}_i = (-1)^{i+1}$ for all but an even index j such that $\bar{x}_j = -3$. A critical point (but not a *d*-stationary one) is $\bar{x}_i = (-1)^{i+1}$ for all but an odd index j with $\bar{x}_j = -1$. At this point, the function value is zero. Table 7 reports some numerical experiments obtained by applying seven solvers to problem (30).

Note that solvers PBM1, PBM3 and DCA-LBM computed critical points that are not *d*-stationary for problem (30). On the other hand, the solvers PBM-d, PLM and HANSO succeeded in computing global solutions for all instances. A very good performance of solver PBDC is highlighted: PBDC succeeded in computing global solutions in six out of nine cases.

Table 6 Problem (29) with eight different starting points. Solvers PBM-d and DCA-LBM computed d -stationary points regardless the considered initial point x^0

| Problem data | | PBM1 | | | PBM3 | | | DCA-LBM | | | PLM | | | PBM-d | | | |
|--------------|---------|-----------|------------------|-----|------|------------------|-----|----------|------------------|-----|-----|------------------|-----|-------|------------------|-----|-----|
| x_1^0 | x_2^0 | \bar{f} | $f(x^{k(\ell)})$ | #g1 | #g2 | $f(x^{k(\ell)})$ | #g1 | #g2 | $f(x^{k(\ell)})$ | #g1 | #g2 | $f(x^{k(\ell)})$ | #g1 | #g2 | $f(x^{k(\ell)})$ | #g1 | #g2 |
| 4 | 5 | -2.02603 | 0 | 28 | 27 | 0 | 28 | 0 | -2.02603 | 56 | 3 | 0 | 189 | 14 | -0.1875 | 29 | 28 |
| 5 | 3 | -2.02603 | 0 | 28 | 27 | 0 | 28 | 0 | -0.18749 | 60 | 3 | 0 | 180 | 14 | -0.1875 | 29 | 28 |
| 5 | 1 | -2.02603 | 0 | 27 | 26 | 0 | 27 | 0 | -2.02603 | 55 | 3 | 0 | 181 | 14 | -0.1875 | 49 | 44 |
| 5 | -1 | -2.02603 | -0.1875 | 71 | 69 | -0.1875 | 71 | -0.1875 | -0.18749 | 37 | 2 | -0.1875 | 525 | 70 | -0.1875 | 71 | 69 |
| 2 | -5 | -2.02603 | -0.1875 | 71 | 68 | -0.1875 | 71 | -0.1875 | -0.1875 | 36 | 2 | -0.1875 | 529 | 68 | -0.1875 | 71 | 68 |
| 0 | -5 | -2.02603 | -0.1875 | 21 | 20 | -0.1875 | 21 | -0.1875 | -0.1875 | 35 | 2 | -0.1875 | 100 | 10 | -0.1875 | 21 | 20 |
| -1.2 | -5 | -2.02603 | -2.02603 | 30 | 28 | -2.02603 | 30 | -2.02603 | -0.1875 | 37 | 2 | -2.02603 | 129 | 15 | -2.02603 | 30 | 28 |
| -3 | -5 | -2.02603 | -2.02603 | 30 | 28 | -2.02603 | 30 | -2.02603 | -2.02603 | 28 | 2 | -2.02603 | 161 | 13 | -2.02603 | 30 | 28 |

Table 7 Problem (30) with nine different instances. All solvers were initialized with the vector $x_i^0 = 10, i = 1, \dots, n$. Except for PLM, all the solvers required less than one second of CPU time to compute a critical point

| Problem data | PBM1 | | PBM3 | | PBM-d | | DCA-LBM | | PLM | | HANSO | | PBDC | | | |
|--------------|-----------|------------------|------|-----|------------------|-----|---------|------------------|-----|-----|--------|------------------|------|------------------|-----|-----|
| | \bar{f} | $f(x^{k(\ell)})$ | #g1 | #g2 | $f(x^{k(\ell)})$ | #g1 | #g2 | $f(x^{k(\ell)})$ | #g1 | #g2 | CPU | $f(x^{k(\ell)})$ | #f | $f(x^{k(\ell)})$ | #g1 | #g2 |
| 2 | -4 | 0.00 | 2 | 1 | 0.00 | 2 | 1 | 0.01 | 28 | 2 | 0.91 | -4 | 27 | -4 | 20 | 15 |
| 4 | -4 | 0.00 | 2 | 1 | 0.00 | 2 | 1 | 0.00 | 24 | 2 | 1.22 | -4 | 41 | -4 | 22 | 16 |
| 5 | -4 | 0.00 | 3 | 2 | 0.00 | 2 | 1 | 0.00 | 26 | 2 | 1.22 | -4 | 44 | -4 | 18 | 14 |
| 10 | -4 | 0.00 | 2 | 1 | 0.00 | 2 | 1 | 0.00 | 27 | 2 | 1.87 | -4 | 38 | -4 | 30 | 18 |
| 20 | -4 | 0.00 | 2 | 1 | 0.00 | 2 | 1 | 0.00 | 29 | 2 | 6.29 | -4 | 47 | -4 | 30 | 23 |
| 50 | -4 | 0.00 | 2 | 1 | 0.00 | 2 | 1 | 0.00 | 32 | 2 | 12.14 | -4 | 51 | -4 | 23 | 16 |
| 100 | -4 | 0.00 | 2 | 1 | 0.00 | 2 | 1 | 0.00 | 31 | 2 | 61.18 | -4 | 44 | 9.90E-05 | 11 | 7 |
| 150 | -4 | 0.00 | 2 | 1 | 0.00 | 2 | 1 | 0.00 | 31 | 2 | 176.13 | -4 | 31 | 5.18E-03 | 10 | 8 |
| 200 | -4 | 0.00 | 2 | 1 | 0.00 | 2 | 1 | 0.00 | 30 | 2 | 711.96 | -4 | 43 | -3.99E+00 | 11 | 10 |

7 Concluding remarks

Inspired by Proximal Linearized Methods, this work proposes and analyzes two proximal bundle algorithms for dealing with convex constrained nonsmooth DC programs. No line-search nor estimates of Lipschitz constants of the DC components are required by the algorithms, which possess a reliable and straightforward stopping test. Moreover, the given algorithms consider convex models of the underlying DC function. The first algorithm is shown to generate a subsequence of points that converges to a critical point of the underlying DC problem. The second algorithm is proved to generate a sequence of descent steps converging to a d -stationary point, which is the sharpest stationary definition in DC programming. However, this requires us to assume that the second DC component f_2 is the pointwise maximum of N differentiable functions. The price to obtain this stronger result is the solution of possibly several (but no more than N) master programs at certain iterations.

Numerical experiments on approximately one hundred instances of different academic DC programs indicate that employing a convex model for approximating the DC objective function can be a simpler alternative to DC models for dealing with DC problems via proximal bundle methods.

Acknowledgements The author is grateful to the reviewers for their remarks and constructive suggestions that considerably improved the original version of this article.

A Appendix

A.1 A self-contained analysis of the sequence of infinitely many null steps generated after a last serious step

We assume that after the $\hat{\ell}$ th-stability center $x^{k(\hat{\ell})} = \hat{x}$ only null steps are performed, i.e.,

$$f_1(x^{k+1}) > f_1(\hat{x}) + \langle \hat{g}_2, x^{k+1} - \hat{x} \rangle - \kappa \underline{\mu} D(x^{k+1}, \hat{x}),$$

where $\hat{g}_2 = g_2^{k(\hat{\ell})}$ is the last subgradient computed for f_2 . Notice that in this case the sequence $\{\mu_k\}_{k \geq k(\hat{\ell})}$ is nondecreasing. In what follows we present some auxiliary results to prove Lemma 4. We start by defining the following two useful functions:

$$F^k(x) := \check{f}_1^k(x) - \langle \hat{g}_2, x - \hat{x} \rangle + \mu_k D(x, \hat{x}) \tag{31}$$

$$F^{-k}(x) = \check{f}_1^k(x^{k+1}) + \langle p^{k+1} + s^{k+1}, x - x^{k+1} \rangle - \langle \hat{g}_2, x - \hat{x} \rangle + \mu_k D(x, \hat{x}). \tag{32}$$

Notice that F^{-k} is twice differentiable (because ω defining D is so):

$$\nabla F^{-k}(x) = p^{k+1} + s^{k+1} - \hat{g}_2 + \mu_k (\nabla \omega(x) - \nabla \omega(\hat{x})) \quad \text{and} \quad \nabla^2 F^{-k}(x) = \mu_k \nabla^2 \omega(x),$$

where $\nabla^2 \omega(x) \in \mathbb{R}^{n \times n}$ is the Hessian of the function ω . Since ω is strongly convex then $\nabla^2 \omega(x)$ is positive definite for all $x \in \mathbb{R}^n$. It follows from (10) that $\nabla F^{-k}(x^{k+1}) = 0$, i.e., the point x^{k+1} is the unique minimizer of $F^{-k}(x)$ over \mathbb{R}^n . The Taylor and mean value Theorems [5, Sect. 13] give, for some $z = \lambda x^{k+1} + (1 - \lambda)\hat{x}$ and $\lambda \in [0, 1]$,

$$\begin{aligned} F^{-k}(x) &= F^{-k}(x^{k+1}) + \langle \nabla F^{-k}(x^{k+1}), x - x^{k+1} \rangle + \frac{1}{2} \langle \nabla^2 F^{-k}(z)(x - x^{k+1}), x - x^{k+1} \rangle \\ &= F^{-k}(x^{k+1}) + \langle 0, x - x^{k+1} \rangle + \frac{\mu_k}{2} \langle \nabla^2 \omega(z)(x - x^{k+1}), x - x^{k+1} \rangle \end{aligned}$$

$$\begin{aligned}
 &\geq F^{-k}(x^{k+1}) + \frac{\mu_k \rho}{2} \|x - x^{k+1}\|_p^2 \\
 &= F^k(x^{k+1}) + \frac{\mu_k \rho}{2} \|x - x^{k+1}\|_p^2,
 \end{aligned}
 \tag{33}$$

where the inequality is due to the assumption that ω is strongly convex with parameter $\rho > 0$ and norm $\|\cdot\|_p$ in (5) ($\langle \nabla^2 \omega(z)(x - x^{k+1}), x - x^{k+1} \rangle \geq \rho \|x - x^{k+1}\|_p$), and the last equality follows from (31) and (32). The above development is crucial to show the following lemma, which is essentially a reformulation of [10, Lemma 6.3] to our setting.

Lemma 7 *Let $\hat{x} = x^{k(\hat{\ell})}$ be the last stability center generated by Algorithm 1 during the iteration $k(\hat{\ell})$ after which only null steps are performed. Assume also that for $k \geq k(\hat{\ell})$ the function F^k is the model given in (31) and x^{k+1} is an iterate obtained from a null step. If $\{\mu_k\}_{k \geq k(\hat{\ell})}$ is nondecreasing, then*

(i) *the sequence $\{F^k(x^{k+1})\}_{k \geq k(\hat{\ell})}$ is nondecreasing and satisfies*

$$F^k(x^{k+1}) + \frac{\mu_k \rho}{2} \|x^{k+2} - x^{k+1}\|_p^2 \leq F^{k+1}(x^{k+2}) \text{ for all } k \geq k(\hat{\ell});$$

(ii) *the sequence $\{F^k(x^{k+1})\}_{k \geq k(\hat{\ell})}$ is bounded from above:*

$$F^k(x^{k+1}) + \frac{\mu_k \rho}{2} \|\hat{x} - x^{k+1}\|_p^2 \leq f_1(\hat{x}) \text{ for all } k \geq k(\hat{\ell});$$

(iii) *the following inequality holds true for all $k \geq k(\hat{\ell})$*

$$\begin{aligned}
 \check{f}_1^k(x^{k+1}) - \check{f}_1^{k-1}(x^k) &\leq F^k(x^{k+1}) - F^{k-1}(x^k) + \mu_{k-1}[D(x^k, \hat{x}) \\
 &\quad - D(x^{k+1}, \hat{x})] - \langle \hat{g}_2, x^k - x^{k+1} \rangle.
 \end{aligned}$$

Proof Algorithm 1 ensures that the aggregate index $-k$ enters the bundle in every null step. In particular $-k \in \mathcal{B}_1^{k+1}$ for all $k \geq k(\hat{\ell})$. Then for all $x \in X$ and all $k \geq k(\hat{\ell})$ we have that

$$\begin{aligned}
 F^{-k}(x) &= [\check{f}_1^k(x^{k+1}) + \langle p^{k+1}, x - x^{k+1} \rangle] + \langle s^{k+1}, x - x^{k+1} \rangle - \langle \hat{g}_2, x - \hat{x} \rangle + \mu_k D(x, \hat{x}) \\
 &= \bar{f}_1^{-k}(x) + \langle s^{k+1}, x - x^{k+1} \rangle - \langle \hat{g}_2, x - \hat{x} \rangle + \mu_k D(x, \hat{x}) \\
 &\leq \bar{f}_1^{-k}(x) - \langle \hat{g}_2, x - \hat{x} \rangle + \mu_k D(x, \hat{x}) \\
 &\leq \check{f}_1^{k+1}(x) - \langle \hat{g}_2, x - \hat{x} \rangle + \mu_k D(x, \hat{x}) \\
 &\leq \check{f}_1^{k+1}(x) - \langle \hat{g}_2, x - \hat{x} \rangle + \mu_{k+1} D(x, \hat{x}) = F^{k+1}(x),
 \end{aligned}$$

where the first inequality is due to $s^{k+1} \in N_X(x^{k+1})$, the second follows from inequality $\bar{f}_1^{-k}(\cdot) \leq \check{f}_1^{k+1}(\cdot)$ ensured because $-k \in \mathcal{B}_1^{k+1}$, and the last inequality follows from the assumption $\mu_{k+1} \geq \mu_k$. Set $x = x^{k+2}$ in (33) to obtain (i), and $x = \hat{x}$ to obtain $F^k(x^{k+1}) + \frac{\mu_k \rho}{2} \|\hat{x} - x^{k+1}\|_p^2 \leq F^{-k}(\hat{x}) \leq F^{k+1}(\hat{x}) = \check{f}_1^{k+1}(\hat{x}) \leq f_1(\hat{x})$. To show (iii), note that for all $k \geq k(\hat{\ell})$

$$\begin{aligned}
 &F^k(x^{k+1}) - \check{f}_1^k(x^{k+1}) \\
 &= -\langle \hat{g}_2, x^{k+1} - \hat{x} \rangle + \mu_k D(x^{k+1}, \hat{x}) \\
 &\geq -\langle \hat{g}_2, x^{k+1} - \hat{x} \rangle + \mu_{k-1} D(x^{k+1}, \hat{x}) \\
 &= -\langle \hat{g}_2, x^{k+1} - \hat{x} \rangle + \mu_{k-1} D(x^k, \hat{x}) + \mu_{k-1}[D(x^{k+1}, \hat{x}) - D(x^k, \hat{x})] \\
 &= -\langle \hat{g}_2, x^k - \hat{x} \rangle + \mu_{k-1} D(x^k, \hat{x}) + \mu_{k-1}[D(x^{k+1}, \hat{x}) - D(x^k, \hat{x})] - \langle \hat{g}_2, x^{k+1} - x^k \rangle \\
 &= F^{k-1}(x^k) - \check{f}_1^{k-1}(x^k) + \mu_{k-1}[D(x^{k+1}, \hat{x}) - D(x^k, \hat{x})] - \langle \hat{g}_2, x^{k+1} - x^k \rangle,
 \end{aligned}$$

where the inequality is due to $\mu_k \geq \mu_{k-1}$ and the last equality follows from (31) (with k therein replaced with $k - 1$). The result thus follows. \square

Given the above properties, the following lemma shows that the cutting-plane model \check{f}_1^k asymptotically approximates the DC component f_1 on the sequence of null iterates.

Lemma 8 *Under the assumptions of Lemma 7, Algorithm 1 ensures that $\{x^k\}_k$ is a bounded sequence and*

$$\lim_{k \rightarrow \infty} [f_1(x^k) - \check{f}_1^{k-1}(x^k)] = 0.$$

Proof Lemma 7(i) ensures that the sequence $\{F^k(x^{k+1})\}_{k \geq k(\hat{\ell})}$ is nondecreasing. Thus, there exists a constant $C > 0$ such that $F^k(x^{k+1}) \geq -C$ for all $k \geq k(\hat{\ell})$. Using Lemma 7(ii) we conclude that

$$\frac{\mu_k \rho}{2} \|\hat{x} - x^{k+1}\|_p^2 \leq f_1(\hat{x}) - F^k(x^{k+1}) \leq f_1(\hat{x}) + C \quad \forall k \geq k(\hat{\ell}), \tag{34}$$

showing that the sequences $\{\|\hat{x} - x^k\|_p\}_{k \geq k(\hat{\ell})}$ is bounded because $\{\mu_k\}_{k > k(\hat{\ell})}$ is nondecreasing. Accordingly, $\{x^k\}_k$ is also bounded. It follows from Lemma 7(ii) that the sequence $\{F^k(x^{k+1})\}_{k \geq k(\hat{\ell})}$ is bounded from above by $f_1(\hat{x})$. Lemma 7(i) shows that $\{F^k(x^{k+1})\}_{k \geq k(\hat{\ell})}$ is nondecreasing and hence

$$\lim_{k \rightarrow \infty} [F^{k+1}(x^{k+2}) - F^k(x^{k+1})] = 0 \quad \text{and} \quad \lim_{k \rightarrow \infty} \|x^{k+2} - x^{k+1}\|_p^2 = 0, \tag{35}$$

where the second limit follows from Lemma 7(i) and the assumption that $\{\mu_k\}_{k \geq k(\hat{\ell})}$ is nondecreasing. Note that, by (5),

$$\begin{aligned} D(x^k, \hat{x}) - D(x^{k+1}, \hat{x}) &= [\omega(x^k) - \omega(\hat{x}) - \langle \nabla \omega(\hat{x}), x^k - \hat{x} \rangle \\ &\quad - [\omega(x^{k+1}) - \omega(\hat{x}) - \langle \nabla \omega(\hat{x}), x^{k+1} - \hat{x} \rangle] \\ &= \omega(x^k) - \omega(x^{k+1}) - \langle \nabla \omega(\hat{x}), x^k - x^{k+1} \rangle. \end{aligned}$$

Since $\{\mu_k\}_k$ is a bounded sequence and ω is a continuous function, we conclude that

$$\lim_{k \rightarrow \infty} \mu_{k-1} [D(x^k, \hat{x}) - D(x^{k+1}, \hat{x})] = 0. \tag{36}$$

The inclusion $k \in \mathcal{B}_1^k$ implies $f_1(x^k) + \langle g_1^k, x - x^k \rangle = \bar{f}_1^k(x) \leq \check{f}_1^k(x)$ for all $x \in \mathbb{R}^n$. Setting $x = x^{k+1}$ in this inequality yields $f_1(x^k) = \bar{f}_1^k(x^{k+1}) + \langle g_1^k, x^k - x^{k+1} \rangle$. Therefore, for $k > k(\hat{\ell})$

$$\begin{aligned} f_1(x^k) - \check{f}_1^{k-1}(x^k) &= \bar{f}_1^k(x^{k+1}) + \langle g_1^k, x^k - x^{k+1} \rangle - \check{f}_1^{k-1}(x^k) \\ &\leq \check{f}_1^k(x^{k+1}) + \langle g_1^k, x^k - x^{k+1} \rangle - \check{f}_1^{k-1}(x^k) \\ &\leq F^k(x^{k+1}) - F^{k-1}(x^k) + \mu_{k-1} [D(x^k, \hat{x}) \\ &\quad - D(x^{k+1}, \hat{x})] + \langle g_1^k - \hat{g}_2, x^k - x^{k+1} \rangle, \end{aligned}$$

where the last inequality is due to Lemma 7(iii). Applying the limit with $k \rightarrow \infty$ in the above inequalities and taking into account (35) and (36), remembering that $\{x^k\}_k$ and $\{g_1^k\}_k$ are bounded sequences, we conclude that $\limsup_{k \rightarrow \infty} [f_1(x^k) - \check{f}_1^{k-1}(x^k)] \leq 0$. Since f_1 is convex we have $f_1(x^k) \geq \check{f}_1^{k-1}(x^k)$ and the result follows. \square

Proof of Lemma 4

Suppose that \hat{x} is not a cluster point of $\{x^{k+1}\}_{k \geq k(\hat{\ell})}$. Then there would exist $\epsilon > 0$ and an index $\tilde{k} \geq k(\hat{\ell})$ such that

$$(1 - \kappa)\underline{\mu} D(x^{k+1}, \hat{x}) \geq \frac{(1 - \kappa)\underline{\mu}\rho}{2} \|x^{k+1} - \hat{x}\|_p^2 > \epsilon$$

for all index $k + 1 \geq \tilde{k}$. It follows from Lemma 8 that there exists an index $\bar{k} \geq k(\hat{\ell})$ such that

$$f_1(x^{k+1}) - \check{f}_1^k(x^{k+1}) \leq \epsilon \text{ for all } k + 1 \geq \bar{k}.$$

Definition of x^{k+1} , feasibility of \hat{x} and inequality $\check{f}_1^k(\cdot) \leq f_1(\cdot)$ yield the inequality

$$\check{f}_1^k(x^{k+1}) - \langle \hat{g}_2, x^{k+1} - \hat{x} \rangle + \mu_k D(x^{k+1}, \hat{x}) \leq f_1(\hat{x}).$$

As by assumption $\kappa \in (0, 1)$, $\mu_k \geq \underline{\mu}$ and assuming that $k + 1 > \max\{\tilde{k}, \bar{k}\}$ we get

$$\begin{aligned} f_1(\hat{x}) &\geq f_1(x^{k+1}) - \langle \hat{g}_2, x^{k+1} - \hat{x} \rangle + \kappa \underline{\mu} D(x^{k+1}, \hat{x}) \\ &\quad + (1 - \kappa)\underline{\mu} D(x^{k+1}, \hat{x}) + \check{f}_1^k(x^{k+1}) - f_1(x^{k+1}) \\ &> f_1(x^{k+1}) - \langle \hat{g}_2, x^{k+1} - \hat{x} \rangle + \kappa \underline{\mu} D(x^{k+1}, \hat{x}) + \epsilon - \epsilon, \end{aligned}$$

showing that x^{k+1} satisfies the descent test (13), i.e. x^{k+1} becomes the new stability center. But this contradicts the fact that \hat{x} is the last stability center. Hence, the sequence $\{x^{k+1}\}_k$ has a subsequence that converges to \hat{x} , i.e., $\lim_{k \in \mathcal{K}} x^k = \hat{x}$ for some index set $\mathcal{K} \subset \{k(\hat{\ell}) + 1, k(\hat{\ell}) + 2, \dots\}$. We now proceed to show that indeed the whole sequence converges to \hat{x} : it follows from (31) that

$$F^{k-1}(x^k) = \check{f}_1^{k-1}(x^k) - \langle \hat{g}_2, x^k - \hat{x} \rangle + \mu_{k-1} D(x^k, \hat{x})$$

and, therefore, $\lim_{k \in \mathcal{K}} F^{k-1}(x^k) = f_1(\hat{x})$ from Lemma 8. Lemma 7(i) shows that $\{F^{k-1}(x^k)\}_{k \geq k(\hat{\ell})}$ is nondecreasing and hence $\lim_{k \rightarrow \infty} F^k(x^{k+1}) = \lim_{k \in \mathcal{K}} F^{k-1}(x^k) = f_1(\hat{x})$. This property combined with (34) shows that the whole sequence $\{x^k\}_k$ converges to \hat{x} . □

References

1. Astorino, A., Miglionico, G.: Optimizing sensor cover energy via DC programming. *Optim. Lett.* **10**(2), 355–368 (2016)
2. Bagirov, A.M.: A method for minimization of quasidifferentiable functions. *Optim. Methods Softw.* **17**(1), 31–60 (2002)
3. Bagirov, A.M., Yearwood, J.: A new nonsmooth optimization algorithm for minimum sum-of-squares clustering problems. *Eur. J. Oper. Res.* **170**(2), 578–596 (2006)
4. Ben-Tal, A., Nemirovski, A.: Non-Euclidean restricted memory level method for large-scale convex optimization. *Math. Program.* **102**, 407–456 (2005)
5. Bonnans, J., Gilbert, J., Lemaréchal, C., Sagastizábal, C.: *Numerical Optimization: Theoretical and Practical Aspects*, 2nd edn. Springer, Berlin (2006)
6. Clarke, F.H.: *Optimisation and nonsmooth analysis*. Soc. Ind. Appl. Math. (1990). <https://doi.org/10.1137/1.9781611971309>
7. Cruz Neto, J.X., Oliveira, P.R., Soubeyran, A., Souza, J.C.O.: A generalized proximal linearized algorithm for DC functions with application to the optimal size of the firm problem. *Ann. Oper. Res.* (2018). <https://doi.org/10.1007/s10479-018-3104-8>

8. de Oliveira, W., Solodov, M.: A doubly stabilized bundle method for nonsmooth convex optimization. *Math. Program.* **156**(1), 125–159 (2016)
9. de Oliveira, W.: Target radius methods for nonsmooth convex optimization. *Oper. Res. Lett.* **45**(6), 659–664 (2017)
10. de Oliveira, W., Sagastizábal, C., Lemaréchal, C.: Convex proximal bundle methods in depth: a unified analysis for inexact oracles. *Math. Program. Ser. B* **148**, 241–277 (2014)
11. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. *Math. Program.* **91**, 201–213 (2002)
12. Frangioni, A.: Generalized bundle methods. *SIAM J. Optim.* **13**(1), 117–156 (2002)
13. Frangioni, A., Gorgone, E.: Generalized bundle methods for sum-functions with “easy” components: applications to multicommodity network design. *Math. Program.* **145**(1), 133–161 (2014)
14. Fuduli, A., Gaudioso, M., Giallombardo, G.: A DC piecewise affine model and a bundling technique in nonconvex nonsmooth minimization. *Optim. Methods Softw.* **19**(1), 89–102 (2004)
15. Gaudioso, M., Giallombardo, G., Miglionico, G.: Minimizing piecewise-concave functions over polyhedra. *Math. Oper. Res.* **43**(2), 580–597 (2018)
16. Gaudioso, M., Giallombardo, G., Miglionico, G., Bagirov, A.M.: Minimizing nonsmooth DC functions via successive DC piecewise-affine approximations. *J. Glob. Optim.* **71**(1), 37–55 (2018)
17. Hare, W., Sagastizábal, C.: A redistributed proximal bundle method for nonconvex optimization. *SIAM J. Optim.* **20**(5), 2442–2473 (2010)
18. Hare, W., Sagastizábal, C., Solodov, M.: A proximal bundle method for nonsmooth nonconvex functions with inexact information. *Comput. Optim. Appl.* **63**(1), 1–28 (2016)
19. Henrion, R.: A Critical Note on Empirical (Sample Average, Monte Carlo) Approximation of Solutions to Chance Constrained Programs (Chapter 3 in [24]). Springer, Berlin (2013)
20. Hiriart-Urruty, J.B., Lemaréchal, C.: Convex Analysis and Minimization Algorithms I. *Grundlehren der mathematischen Wissenschaften*, vol. 305, 2nd edn. Springer, Berlin (1996)
21. Hiriart-Urruty, J.B., Lemaréchal, C.: Convex Analysis and Minimization Algorithms II. *Grundlehren der mathematischen Wissenschaften*, vol. 306, 2nd edn. Springer, Berlin (1996)
22. Hiriart-Urruty, J.B.: Generalized Differentiability/Duality and Optimization for Problems Dealing with Differences of Convex Functions, pp. 37–70. Springer, Berlin, Heidelberg (1985)
23. Holmberg, K., Tuy, H.: A production–transportation problem with stochastic demand and concave production costs. *Math. Program.* **85**(1), 157–179 (1999)
24. Hömberg, D., Tröltzsch, F. (eds.): System Modeling and Optimization. IFIP Advances in Information and Communication, vol. 391. Springer, Berlin (2013)
25. Hong, L.J., Yang, Y., Zhang, L.: Sequential convex approximations to joint chance constrained programs: A Monte Carlo approach. *Oper. Res.* **59**(3), 617–630 (2011)
26. Joki, K., Bagirov, A., Karmitsa, N., Mäkelä, M.M., Taheri, S.: Double bundle method for finding Clarke stationary points in nonsmooth DC programming. *SIAM J. Optim.* **28**(2), 1892–1919 (2018)
27. Joki, K., Bagirov, A.M., Karmitsa, N., Mäkelä, M.M.: A proximal bundle method for nonsmooth DC optimization utilizing nonconvex cutting planes. *J. Glob. Optim.* **68**(3), 501–535 (2017)
28. Kelley, J.E.: The cutting-plane method for solving convex programs. *J. Soc. Ind. Appl. Math.* **8**(4), 703–712 (1960)
29. Khalaf, W., Astorino, A., d’Alessandro, P., Gaudioso, M.: A DC optimization-based clustering technique for edge detection. *Optim. Lett.* **11**(3), 627–640 (2017)
30. Kiwiel, K.C.: A proximal bundle method with approximate subgradient linearizations. *SIAM J. Optim.* **16**(4), 1007–1023 (2006)
31. Le Thi, H.A., Tao, P.D.: DC programming in communication systems: challenging problems and methods. *Vietnam J. Comput. Sci.* **1**(1), 15–28 (2014)
32. Le Thi, H.A., Pham Dinh, T., Ngai, H.V.: Exact penalty and error bounds in DC programming. *J. Glob. Optim.* **52**(3), 509–535 (2012)
33. Lemaréchal, C.: An algorithm for minimizing convex functions. *Inf. Process.* **1**, 552–556 (1974)
34. Lemaréchal, C., Nemirovskii, A., Nesterov, Y.: New variants of bundle methods. *Math. Program.* **69**(1), 111–147 (1995)
35. Lewis, A.S., Overton, M.L.: Nonsmooth optimization via quasi-Newton methods. *Math. Program.* **141**(1), 135–163 (2013)
36. Mäkelä, M.M., Miettinen, M., Lukšan, L., Vlček, J.: Comparing nonsmooth nonconvex bundle methods in solving hemivariational inequalities. *J. Glob. Optim.* **14**(2), 117–135 (1999)
37. Noll, D., Apkarian, P.: Spectral bundle methods for non-convex maximum eigenvalue functions: first-order methods. *Math. Program.* **104**(2), 701–727 (2005)
38. Pang, J.S., Razaviyayn, M., Alvarado, A.: Computing B-stationary points of nonsmooth DC programs. *Math. Oper. Res.* **42**(1), 95–118 (2017)

39. Prékopa, A.: *Stochastic Programming*. Kluwer, Dordrecht (1995)
40. Rockafellar, R.: *Convex Analysis*, 1st edn. Princeton University Press, Princeton (1970)
41. Souza, J.C.O., Oliveira, P.R., Soubeyran, A.: Global convergence of a proximal linearized algorithm for difference of convex functions. *Optim. Lett.* **10**(7), 1529–1539 (2016)
42. Tao, P.D., Le Thi, H.A.: Convex analysis approach to DC programming: theory, algorithms and applications. *Acta Math. Vietnam.* **22**(1), 289–355 (1997)
43. Tuy, H.: *Convex Analysis and Global Optimization*. Springer Optimization and Its Applications, 2nd edn. Springer, Berlin (2016)
44. van Ackooij, W.: Eventual convexity of chance constrained feasible sets. *Optimization* **64**(5), 1263–1284 (2015)
45. van Ackooij, W., Cruz, J.B., de Oliveira, W.: A strongly convergent proximal bundle method for convex minimization in Hilbert spaces. *Optimization* **65**(1), 145–167 (2016)
46. van Ackooij, W., Henrion, R.: Gradient formulae for nonlinear probabilistic constraints with Gaussian and Gaussian-like distributions. *SIAM J. Optim.* **24**(4), 1864–1889 (2014)
47. van Ackooij, W., de Oliveira, W.: Level bundle methods for constrained convex optimization with various oracles. *Comput. Optim. Appl.* **57**(3), 555–597 (2014)
48. van Ackooij, W., Sagastizábal, C.: Constrained bundle methods for upper inexact oracles with application to joint chance constrained energy problems. *SIAM J. Optim.* **24**(2), 733–765 (2014)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.