manuscript No.
(will be inserted by the editor)

A progressive decoupling algorithm for minimizing the difference of convex and weakly convex functions over a linear subspace

Welington de Oliveira[1] **and João Carlos Oliveira Souza**[2]

July 1, 2024

Abstract Commonly, decomposition and splitting techniques for optimization problems strongly depend on convexity. Implementable splitting methods for nonconvex and nonsmooth optimization problems are scarce and often lack convergence guarantees. Among the few exceptions is the Progressive Decoupling Algorithm (PDA), which has local convergence should convexity be elicitable. In this work, we furnish PDA with a descent test and extend the method to accommodate a broad class of nonsmooth optimization problems with non-elicitable convexity. More precisely, we focus on the problem of minimizing the difference of convex and weakly convex functions over a linear subspace. This framework covers, in particular, a family of stochastic programs with nonconvex recourse and statistical estimation problems for supervised learning.

Keywords Nonconvex optimization · Decomposition · Splitting · Stochastic programming

Mathematics Subject Classification (2000) 49J52 · 49J53 · 49K99 · 90C26

1 Introduction.

This work builds upon the *Progressive Decoupling Algorithm (PDA)* of [1] to minimize a nonconvex function over the intersection of a convex set with a linear subspace. Differently from [1], convexity is not assumed to be *elicitable* (not even locally), and evaluating the function's proximal mapping need not be convenient to execute. The manuscript's main assumption is that the objective function can be expressed as the difference of convex and weakly convex functions, a broad setting that covers many practical problems. As pointed out in [2], this function class comprises the family of difference-of-convex (DC) functions and other composite functions.

For the class of problems of interest, the linear subspace represents the linkage constraint, meaning that if not for it, the underlying optimization problem (possibly with the weakly convex function linearized) would be decomposable and thus much easier to handle. This is the case in nonconvex stochastic programming and some statistical estimation problems for supervised learning, as detailed in § 2.2 below.

In such linkage problems, when the objective function is convex, several splitting algorithms can be applied depending on the problem's structure: the alternating direction method of multipliers (ADMM) [3, 4], the Spingarn's method of partial inverses [5], the progressive hedging algorithm (PHA) [6] as well the scenario decomposition with alternating projection [7] in stochastic programming, the progressive decoupling algorithm (PDA) [1], the Douglas Rachford (DR) splitting method [8, 9], and others. All these methods are particular instances of the celebrated proximal point method, and thus (elicitable) convexity plays an important role.

[1] Mines Paris, Université PSL, Centre de Mathématiques Appliquées (CMA), 06904 Sophia Antipolis, France
[2] Department of Mathematics, CCN, Federal University of Piauí, Teresina, PI 64049-550, Brazil
E-mail: welington.oliveira@minesparis.psl.eu

Nonconvexity adds significant difficulty in so much that the proximal point theory can no longer be applied, at least directly. As a result, only a few splitting methods for nonconvex problems that explore the linkage structure of the problem exist in the literature. In addition to the PDA of [1] that requires elicitable convexity, the works [10–13] study the DR method for minimizing the sum of a differentiable function with Lipschitz continuous gradient (L-smooth function) and a proper lower semicontinuous function with an easily computable proximal mapping. By defining a merit function, global subsequential convergence to a critical point and eventual convergence rate are obtained under some extra assumptions. For instance, local convergence rates for weakly convex functions are derived in [13]. Concerning ADMM variants for addressing nonconvex problems, the study [14] examines several approaches under somewhat restrictive assumptions about the nonsmooth function.

We highlight that the methods in [1, 10–14] do not apply to our broader setting because the involved functions need not be L-smooth or have easily computable proximal mappings. To propose an implementable splitting method in this case, we linearize the nonconvex function at serious iterates (candidate solutions), yielding a convex optimization subproblem dealt with inexactly via a progressive decoupling scheme furnished with a descent test. The latter mechanism allows us to define the next serious iterate whenever the objective function improves by a certain amount. The nonconvex function is then linearized at such a point, and the process repeats. Our approach, which has convergence guarantees to critical points, can be seen as a splitting method for finding a zero of the sum of three specific operators: the subdifferential of a convex function, the subdifferential of a weakly concave function, and the normal cone to a linear subspace. Hence, the work contributes to broadening the range of applicability of splitting methods in nonsmooth and nonconvex optimization.

The remainder of this work is organized as follows. Section 2 recalls some key definitions and prerequisites. Our approach and its convergence analysis are presented in Section 3. Some numerical experiments reporting on the practical performance of our algorithm is presented in Section 4, and finally Section 5 concludes the paper.

Notation. The following notation is employed throughout the text. For any points $x, y \in \mathbb{R}^n$, $\langle x, y\rangle$ stands for the Euclidean inner product, and $\|\cdot\|$ for the associated norm, i.e., $\|x\| = \sqrt{\langle x, x\rangle}$. For a convex set X, $N_X(x)$ stands for its normal cone at the point x, i.e., the set $\{y : \langle y, z - x\rangle \leqslant 0 \text{ for all } z \in X\}$ if $x \in X$ and the empty set otherwise. The indicator function of $X \subset \mathbb{R}^n$ is defined as $\delta_X(x) = 0$ if $x \in X$ and $\delta_X(x) = +\infty$ otherwise. The convex hull of a set X is $\mathtt{co}X$ and the relative interior is denoted by $\mathtt{ri}\, X$. The domain of a function $\varphi : \mathbb{R}^n \to (-\infty, +\infty]$ is represented by $\mathcal{D}\mathrm{om}(\varphi) = \{x \in \mathbb{R}^n : \varphi(x) < +\infty\}$.

2 Preliminaries

This section presents the problem, examples in nonconvex stochastic programs and statistics-based learning models, key concepts in variational analysis, some necessary optimality conditions, and a brief description of the progressive decoupling algorithm of [1].

2.1 Problem statement

This work is concerned with the following class of nonsmooth and nonconvex optimization problems

$$\min_{x \in X \cap S} f(x), \quad \text{with} \quad f(x) := c(x) - w(x), \tag{1}$$

under the following assumptions:

i) $c : \mathbb{R}^n \to \mathbb{R}$ is a convex function;

ii) $w : \mathbb{R}^n \to \mathbb{R}$ is a weakly convex function, that is, there exists a (possibly unknown) parameter $\mu_w \geq 0$ such that $w(x) + \frac{\mu}{2}\|x\|^2$ is convex for all $\mu \geq \mu_w$;

iii) $X \subset \mathbb{R}^n$ is a closed convex set and $S \subset \mathbb{R}^n$ is a linear subspace such that $\mathtt{ri}\, X \cap S \neq \emptyset$. Furthermore, we assume that projecting onto S (or its orthogonal complement $S^\perp$) is relatively convenient to execute.

This function class enjoys favorable properties in so much as they can be recast as *Difference-of-Convex* (DC) functions [15]. Hence, problem (1) can, in theory, be recast as a DC program, a setting that proves *practical if explicit DC decompositions are available*; see for instance [16, 17] and references therein. However, if no DC decomposition is known for f, the DC machinery is unsuitable, and DC methods are not applicable. Compared to DC, the *convex-weakly convex* (CwC) structure in (1) appears more naturally in applications [2]. In considering the nonconvex function, we have in mind the following settings: $w(x) = \phi(x)$ is a convex function; $w(x) = -h(x)$, with h being L-smooth; $w(x) = \phi(x) - h(x)$, with ϕ and h as previously; $w(x) = \phi(G(x))$, with $\phi : \mathbb{R}^m \to \mathbb{R}$ convex and Lipschitz and $G : \mathbb{R}^n \to \mathbb{R}^m$ a smooth mapping with Lipschitz Jacobian. In all these settings, the resulting function w is weakly convex as justified in [2, § 2].

2.2 Examples

2.2.1 Nonconvex Stochastic Programs

Let $T \geq 2$ be the time horizon where decisions must be taken. For every stage $t = 1, \ldots, T$, the choice of a vector $x_t \in \mathbb{R}^{n_t}$ is followed by the uncovering of a random event ξ_t in some support set $\Xi_t \subset \mathbb{R}^{m_t}$. We denote by $\Xi = \Xi_1 \times \ldots \times \Xi_T$ the set of all scenarios $\xi = (\xi_1, \ldots, \xi_T)$ and assume for simplicity that there are only finitely many of them: $\Xi = \{\xi^1, \ldots, \xi^N\}$, and $p_i = p(\xi^i) > 0$, $i = 1, \ldots, N$, are the associated probabilities. In reacting to information provided by a scenario ξ, the decision $x_t(\xi) \in \mathbb{R}^{n_t}$ at stage t must be nonanticipative:

$$x_t(\xi) \text{ depends only on the past } \xi_{[t-1]} := (\xi_1, \ldots, \xi_{t-1}), \text{ not on } \xi.$$

We denote by $x(\cdot)$ a function that assigns to each scenarios $\xi \in \Xi$ a decision

$$x(\xi) := (x_1(\xi), \ldots, x_T(\xi)) \in \mathbb{R}^{n_1} \times \ldots \times \mathbb{R}^{n_T} := \mathbb{R}^n.$$

The vector $x := (x(\xi^1), \ldots, x(\xi^N)) \in \mathbb{R}^{nN}$ comprises all the decision variables, and is called a decision policy. Nonanticipative policies form a linear space denoted by $S \subset \mathbb{R}^{nN}$; see [18, § 3.1.4] for the equations representating this linear space (see also (19) for the particular example in the numerical section). Additional constraints are imposed by requiring

$$x(\xi^i) \in X(\xi^i) \subset \mathbb{R}^n \;\text{ where } X(\xi^i) \neq \emptyset \text{ is a closed convex set } i = 1, \ldots, N.$$

Each decision $x(\cdot)$ has a cost represented by the (random) function

$$f(x(\cdot), \cdot) : \xi \to f(x(\xi), \xi), \quad \text{from } \Xi \text{ to } \mathbb{R},$$

for which we assume the CwC decomposition $c(x(\cdot), \cdot) - w(x(\cdot), \cdot)$ is available. With this notation, the problem we are interested in solving is, for $X := X(\xi^1) \times \cdots \times X(\xi^N)$,

$$\min_{x \in X \cap S} \sum_{i=1}^{N} p_i [c(x(\xi^i), \xi^i) - w(x(\xi^i), \xi^i)]\,. \tag{2}$$

For the simpler two-stage setting, i.e., $T = 2$, some recent works [2,19] employ a decomposition per stage. In doing so, the main assumption is that the resulting second-stage subproblems are convex (although the recourse function can be nonconvex). For more general problems not satisfying this hypothesis, decomposition per stage is no longer a choice, and the method presented in Section 3 below is, to the best of our knowledge, the first decomposition approach capable of handling problems within such a challenging framework. (The work [13] also deals with nonconvex stochastic programs by handling simpler models consisting of relaxing the nonanticipativity constraints.)

The CwC structure above can appear in several ways. For instance, if $f(\cdot;\xi)$ is L-smooth (the setting in our numerical experiments), we can take $c(\cdot;\xi)\equiv 0$ and $w(\cdot;\xi)=-f(\cdot;\xi)$. Another situation appears when $f(\cdot;\xi):=\tilde{c}(\cdot;\xi)+\rho\,\mathtt{dist}^2_{W(\xi)}(\cdot)$, with $\tilde{c}(\cdot;\xi)$ convex, $\rho>0$ a penalty parameter, and $\mathtt{dist}^2_{W(\xi)}(\cdot)$ the squared distance to a complicating set $W(\xi)$ (e.g., modeling integrability constraints). As the distance function has a known DC decomposition [17, Ex. 4], we can write f as in (2).

2.2.2 Statistics-based Learning Models

Let ξ^i and $p_i>0$ be as above, and $M(y,\xi^i)$ be a piecewise affine model:

$$M(y,\xi):=\max_{j\in J_1}\left\{\langle a^j,\xi\rangle+\alpha_j\right\}-\max_{j\in J_2}\left\{\langle b^j,\xi\rangle+\beta_j\right\}\quad =:M_1(y,\xi)-M_2(y,\xi),$$

with J_1 and J_2 finite index sets and $y:=\left\{(a^j,\alpha^j)_{j\in J_1},(b^j,\beta^j)_{j\in J_2}\right\}$. Accordingly, a piecewise-affine statistical model for supervised learning is given by

$$z=M(y;\xi)+\varepsilon,$$

where $z\in\mathbb{R}$ is the output, $\xi\in\mathbb{R}^d$ the input, ε is the unobserved random error assumed to have (conditional) mean zero, and $y\in\mathbb{R}^n$ the model parameter to be estimated (see [20, Ch. 3] for more details). Given a training set (z^i,ξ^i), $i=1,\ldots,N$, and loss function $\mathfrak{L}:\mathbb{R}\to\mathbb{R}_+$, the population model minimizes the expected loss subject to prescribed constraints on y:

$$\min_{y\in Y}\ R(y)+\sum_{i=1}^{N}p_i\mathfrak{L}\big(M_1(y,\xi^i)-M_2(y,\xi^i)-z^i\big),$$

where $R:\mathbb{R}^n\to\mathbb{R}$ is a regularizing function (e.g., R is a multiple of the ℓ_1-norm, ℓ_2-norm, total variation, etc.). We stress that many loss functions $\mathfrak{L}:\mathbb{R}\to\mathbb{R}_+$ can be used in practice [20, § 3.1.3]. For instance, if $\mathfrak{L}(\cdot)=|\cdot|$ is the absolute function, then $|M_1(y,\xi^i)-M_2(y,\xi^i)-z^i|=2\max\left\{M_1(y,\xi^i)-z^i,\,M_2(y,\xi^i)\right\}-[M_1(y,\xi^i)-z^i+M_2(y,\xi^i)]$. By setting

$$\begin{aligned}
S&:=\left\{x=(x_1,\ldots,x_N)\in\mathbb{R}^{nN}:\,x_1=\cdots=x_N\right\},\\
c(x)&:=\sum_{i=1}^{N}p_i[R(x_i)+2\max\left\{M_1(x_i,\xi^i)-z^i,\,M_2(x_i,\xi^i)\right\},\\
w(x)&:=\sum_{i=1}^{m}p_i[M_1(y,\xi^i)-z^i+M_2(y,\xi^i)],\\
&\text{and}\quad X:=Y\times\cdots\times Y\subset\mathbb{R}^{nN},
\end{aligned}$$

it can be easily seen that the problem fits the general structure in (1) with w being indeed a convex function. Similar settings appear if the chosen loss function is the one with margin, or an arbitrary function belonging to the quantile family of loss functions, or the truncated hinge loss, or others (see [20, pages 125 and 126]).

2.3 Key Variational Concepts

This ubsections recalls some well known concepts in variational analysis; see, for instance, [21] for more information.

A function $f:\mathbb{R}^n\to\mathbb{R}$ is said to be *locally Lipschitz continuous* if for each $x'\in\mathbb{R}^n$ there is a neighborhood $V_{x'}\subset\mathbb{R}^n$ of x' such that, for some $L_{x'}\geq 0$,

$$|f(x)-f(y)|\leq L_{x'}\|x-y\|\quad\forall\,x,y\in V_{x'}.$$

Function f is said to be *Lipschitz continuous* if $L_{x'} = L$ can be taken independent of $x' \in \mathbb{R}^n$, and $V_{x'}$ in the above inequality is replaced with the entire space $\mathbb{R}^n$.

Let $c : \mathbb{R}^n \to \mathbb{R}$ be a convex function. Then c is locally Lipschitz continuous and, for each $x \in \mathbb{R}^n$, the *subdifferential* of c at x is denoted by

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

The elements of $\partial c(x)$ are referred to as the *subgradients* of c at x, and the set-valued operator ∂c is monotone maximal. Recall that a mapping $T : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is monotone (globally) if $\langle y'-y, x'-x\rangle \geq 0$ when $y \in T(x)$ and $y' \in T(x')$. It is maximal monotone if furthermore there is no monotone mapping T' with $\operatorname{gph} T' \supset \operatorname{gph} T$ and $\operatorname{gph} T' \neq \operatorname{gph} T$, where $\operatorname{gph} T = \{(x, y) \in \mathbb{R}\times\mathbb{R} : y \in T(x)\}$; see, for instance, [1].

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a locally Lipschitz continuous function. Then the generalized directional derivative defined by $f^\circ(x; d) := \limsup_{x' \to x,\ \tau\downarrow 0} \dfrac{f(x' + \tau d) - f(x')}{\tau}$ is finite for all $x \in \mathbb{R}^n$ in every direction $d \in \mathbb{R}^n$ [21, Prop. 2.1.1(a)]. Such a mathematical concept permits to define the *Clarke subdifferential* of f at x,

$$\partial^{\mathsf{C}} \mathtt{f}(x) := \{g \in \mathbb{R}^n : \langle g, d\rangle \leq f^\circ(x; d) \ \text{ for all } \ d \in \mathbb{R}^n\},$$

which is a nonempty, convex, and compact subset of $\mathbb{R}^n$ [21, Prop. 2.1.2(a)] satisfying $f^\circ(x; d) = \max_{g\in\partial^{\mathsf{c}} f(x)}\langle g, d\rangle$. The elements of $\partial^{\mathsf{C}} f(x)$ are referred to as *generalized (or Clarke) subgradients*, as they are the usual subgradients when f is convex [21, Prop. 2.2.7]; i.e., $\partial^{\mathsf{C}} f = \partial f$ if f is convex. Furthermore, when f is continuously differentiable, $\partial^{\mathsf{C}} f(x)$ reduces to the singleton $\{\nabla f(x)\}$. A fundamental result concerning the generalized subdifferential is the following one [21, Prop. 2.1.2]: the mapping $\partial^{\mathsf{C}} f$ is locally bounded in the interior of the domain of f. As a result, since in our setting $\mathcal{D}\mathrm{om}(f) = \mathbb{R}^n$ by assumption, the image $\partial^{\mathsf{C}} f(X)$ of every bounded set $X \subset \mathbb{R}^n$ is bounded in $\mathbb{R}^n$.

A function $w : \mathbb{R}^n \to \mathbb{R}$ is *weakly convex* if there exists a (possibly unknown) parameter $\mu_w \geq 0$ such that $w(x) + \frac{\mu}{2}\|x\|^2$ is convex for all $\mu \geq \mu_w$. Such a definition is equivalent to saying (see [22, Prop. 4.8]) that for all $g \in \partial^{\mathsf{C}} w(x)$, the following inequality holds

$$w(y) \geq w(x) + \langle g, y - x\rangle - \frac{\mu_w}{2}\|y - x\|^2 \quad \forall\, y \in \mathbb{R}^n. \tag{3}$$

Clearly, this inequality holds for all $\mu \geq \mu_w$.

A locally Lipschitz continuous function $f : \mathbb{R}^n \to \mathbb{R}$ is *subdifferentially regular* (or simply regular) at $x \in \mathbb{R}^n$ if for every $d \in \mathbb{R}^n$ the ordinary directional derivative at x exists and coincides with the generalized one:

$$f'(x; d) := \lim_{\tau\downarrow 0} \frac{f(x' + \tau d) - f(x')}{\tau} = f^\circ(x; d) \quad \forall\, d \in \mathbb{R}^n.$$

It holds that smooth functions, as well as weakly convex ones, are regular at every point in the interior of their domains; see [22, Prop. 4.5] and [23, Thm. 1]. Moreover, a finite linear combination (by nonnegative scalars) of regular functions at x is regular [21, Prop. 2.3.6]. However, although the convex c and weakly convex function w are regular, their difference $f = c - w$ need not be. The main inconvenient of non-regularity is the implication of weaker stationarity conditions for (1).

2.4 Necessary Optimality Conditions

It is well-known that the sharpest stationarity condition for nonsmooth and nonconvex optimization problems is d(irectional)-stationarity. As $X \cap S$ is a convex set, $\bar{x} \in X \cap S$ is a d-stationary point of problem (1) if

$$f'(\bar{x}; (x - \bar{x})) \geq 0 \quad \forall\, x \in X \cap S.$$

We recall that local optimal solutions are d-stationary, but the converse is not necessarily true. Another condition is Clarke stationarity

$$0 \in \partial^{\mathsf{C}} f(\bar{x}) + N_{X\cap S}(\bar{x}).$$

It is not difficult to see that this inclusion implies $f^\circ(\bar{x};(x-\bar{x})) \geq 0$ for all $x \in X \cap S$. Being f non-regular, $f^\circ(\bar{x};\cdot) \geq f'(\bar{x};\cdot)$ and therefore Clarke stationarity is, in general, a weaker condition than d-stationarity. Since in this work we deal with f through the decomposition $c-w$, we may not be able to numerically compute vectors in $\partial^{\mathsf{C}} f$ but in its enlargement $\partial c - \partial^{\mathsf{C}} w$ ($\supset \partial^{\mathsf{C}} f$). As a result, we may get a weaker optimality condition, denoted by criticality: $\bar{x} \in X \cap S$ is a *critical point* to problem (1) if

$$0 \in \partial c(\bar{x}) - \partial^{\mathsf{C}} w(\bar{x}) + N_{X\cap S}(\bar{x}). \tag{4}$$

If c is differentiable at $\bar{x}$, then criticality is equivalent to Clarke stationarity: in this case $\partial^{\mathsf{C}} f(\bar{x}) = \nabla c(\bar{x}) - \partial^{\mathsf{C}} w(\bar{x}) = \partial c(\bar{x}) - \partial^{\mathsf{C}} w(\bar{x})$. The three conditions are equivalent provided w is differentiable at $\bar{x}$: under this assumption, not only $\partial^{\mathsf{C}} f(\bar{x}) = \partial c(\bar{x}) - \nabla w(\bar{x}) = \partial c(\bar{x}) - \partial^{\mathsf{C}} w(\bar{x})$ but also $f'(\bar{x};\cdot) = f^\circ(\bar{x};\cdot)$, i.e., f is regular at $\bar{x}$. (The latter claim follows from the fact that $f'(\bar{x};d) = c'(\bar{x};d) - \langle \nabla w(\bar{x}), d\rangle = \max_{g\in\partial c(\bar{x})-\nabla w(\bar{x})}\langle g, d\rangle = \max_{g\in\partial^{\mathsf{c}} f(\bar{x})}\langle g, d\rangle = f^\circ(\bar{x};d)$.)

We now give a different, but equivalent, characterization of criticality in terms of a *linkage problem*. Observe that assumption iii) in Section 2.1 implies that $\mathtt{ri}\, X \cap \mathtt{ri}\, S \neq \emptyset$ because S is a linear subspace. Hence, [24, Cor. 23.8.1] ensures that for all $x \in X \cap S$,

$$\partial\delta_{X\cap S}(x) = N_{X\cap S}(x) = N_X(x) + N_S(x) = \partial\delta_X(x) + \partial\delta_S(x).$$

Moreover, as $\mathcal{D}\mathrm{om}(c) = \mathbb{R}^n$ from assumption i), we get that

$$\emptyset \neq \mathtt{ri}\, \mathcal{D}\mathrm{om}(c) \cap \mathtt{ri}\, X \cap \mathtt{ri}\, S, \quad \text{and thus} \tag{5a}$$

$$\partial[c+\delta_X+\delta_S](x) = \partial c(x) + \partial\delta_X(x) + \partial\delta_S(x) \quad \text{for all } x \in S. \tag{5b}$$

Accordingly, the criticality condition in (4) reads as

$$0 \in \partial c(\bar{x}) + N_X(\bar{x}) + N_S(\bar{x}) - \partial^{\mathsf{C}} w(\bar{x}) \quad = \quad \partial[c+\delta_X](\bar{x}) + N_S(\bar{x}) - \partial^{\mathsf{C}} w(\bar{x}).$$

As S is a linear subspace, it follows that $N_S(x) = S^\perp$ for all $x \in S$. Thus, the above system of generalized equations can be equivalently written in the form of a linkage problem:

$$\text{find } \bar{x} \in S \text{ and } \bar{y} \in S^\perp \text{ such that } \bar{y} \in T(\bar{x}), \quad \text{with} \quad T(x) := \partial[c+\delta_X](x) - \partial^{\mathsf{C}} w(x). \tag{6}$$

Note that in (6) $T : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is a set-valued operator that fails to be monotone due to $-\partial^{\mathsf{C}} w$. This fact precludes the application of methods based on the celebrated *proximal point algorithm* to solve (6). For the special class of nonmonotone operators with (local) elicitable maximal monotonicity, the progressive decoupling algorithm (PDA) of [1] is perhaps the sole proximal-point-like algorithm with local convegence for problem in this class.

2.5 The Progressive Decoupling Algorithm in a Nutshell

Let $U : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ be an operator, S a linear subspace of $\mathbb{R}^n$, and consider the linkage problem:

$$\text{find } \bar{x} \in S \text{ and } \bar{y} \in S^\perp \text{ such that } \bar{y} \in U(\bar{x}). \tag{7}$$

The progressive decoupling algorithm (PDA) of [1] is an iterative method designed to solve (7) provided maximal monotonicity of U is elicitable.

Definition 2.1 (Elicitation, [1] Def. 1) Maximal monotonicity is said to be elicitable globally at a level $e \geq 0$ if the mapping $U + e\,\mathtt{proj}_{S^\perp} : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is maximal monotone. (Here, $\mathtt{proj}_{S^\perp}$ is the projection operator onto the orthogonal complement of S.) □

Local elicitation requires $U + e\,\mathtt{proj}_{S^\perp}$ be maximal monotone on a neighborhood of a solution of (7); see [1, Def. 1]. It follows from definition that if monotonicity is elicitable at a level e it is elicitable at all higher levels e as well.

PDA is an iterative method that produces three sequences of points as follows, for given $x^1 \in S$, $y^1 \in S^\perp$ and $r > e \geq 0$:

$$\begin{cases} \hat{x}^k & \text{s.t. } 0 \in U^k(\hat{x}^k), \text{ where } U^k(x) := U(x) - y^k + r[x - x^k] \\ x^{k+1} & = \mathtt{proj}_S(\hat{x}^k) \\ y^{k+1} & = y^k - (r-e)(\hat{x}^k - x^{k+1}). \end{cases} \tag{8}$$

PDA's Convergence analysis is summarized in the following theorem [1].

Theorem 2.1 (Thm. 1 in [1]) *Suppose the linkage problem* (7) *is solvable, and that* $e \geq 0$ *gives a level at which maximal monotonicity is elicited globally. Then the iterations in* (8) *for any* $r > e$*, starting from any* x^1 *and* $y^1 \in S^\perp$*, will generate a sequence of pairs* (x^k, y^k) *that converges to a pair* $(\bar{x}, \bar{y})$ *solving* (7)*.*

Identifying whether maximal monotonicity of an operator is elicitable, and at what level $e \geq 0$, is a non-trivial task in general. Indeed, when applied to the linkage problem (6), global (local) elicitable monotonicity amounts to saying that there exists $e \geq 0$ such that $[f + \delta_X](x) + \frac{e}{2}d_S^2(x)$ is globally (locally) convex, where $d_S(x) := \min_{z \in S} \|z - x\| = \|\mathtt{proj}_{S^\perp}(x)\|$. Observe that inferring on the convexity of $[f + \delta_X](x) + \frac{e}{2}d_S^2(x)$ might be difficult for most applications. Furthermore, many practical problems involve operators whose maximal monotonicity cannot be elicitable. This are, for instance, the cases of the problems in Subsections 2.2.1 and 2.2.2. Figure 1 gives a simple example of an elicitable function whereas Figure 2 illustrates the non-elicitable case.

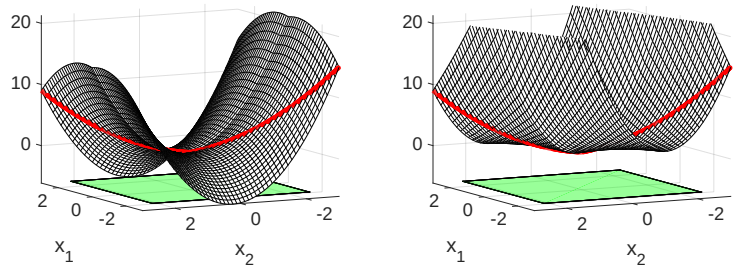

Fig. 1 Let $f(x) = c(x) - w(x)$, $c(x) = |x_1| + 2x_2^2$, $w(x) = x_1^2 + x_2$, and $S = \{x \in \mathbb{R}^2 : x_1 = x_2\}$. (a) Function f is nonconvex, while in (b), $f(x) + \frac{5}{2}d_S^2(x)$ is convex. Hence, convexity of f is elicitable globally at level $e = 5$. The red lines in both subfigures represent the same function $\phi(x) = [f + \delta_S](x)$, which is convex in this case.

In what follows, we do not assume that convexity in (1) can be elicitable and propose a new variant of the PDA for computing a critical point to (1), i.e., a pair of vectors solving the nonmonotone linkage problem (6).

3 CwC Progressive Decoupling Algorithm

This section presents the CwC Progressive Decoupling Algorithm and its convergence analysis.

3.1 The algorithm

Our goal in this section is to present an extension of the progressive decoupling algorithm of [1] for computing a critical point to the nonconvex problem (1). A naive variant consists of replacing, at

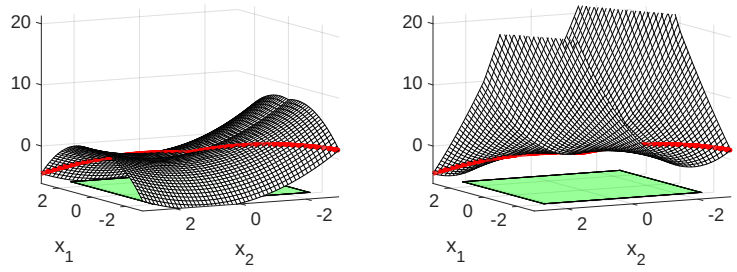

Fig. 2 Let $f(x) = c(x) - w(x)$, with $c(x) = |x_1| + \frac{1}{2}x_2^2$, and w and S be as in Figure 1. (a) Function f is nonconvex and, in (b), $f(x) + \frac{e}{2}d_S^2(x)$ remains nonconvex for all $e \geq 0$. Hence, convexity of f is not elicitable at any level. Note that $\phi(x) = [f + \delta_S](x)$, represented by the red lines, is nonconvex.

every iteration $\ell = 1, 2, \ldots$, the generalized subdifferential $\partial^{\mathtt{C}} w$ in (7) with an element $g^\ell \in \partial^{\mathtt{C}} w(x^\ell)$ and defining the next iterate as a solution to the following monotone (and thus simpler) linkage problem:

$$\text{find } x^{\ell+1} \in S \text{ and } y^{\ell+1} \in S^\perp \text{ such that } y^{\ell+1} \in T^\ell(x^{\ell+1}) \quad \text{with} \quad T^\ell(x) := \partial[c + \delta_X](x) - g^\ell.$$

In other words, in this simple strategy, iterates are produced as follows:

$$\text{compute } g^\ell \in \partial^{\mathtt{C}} w(x^\ell), \text{ and let } x^{\ell+1} \in \arg\min_{x \in X \cap S} \; c(x) - [w(x^\ell) + \langle g^\ell, x - x^\ell \rangle].$$

In pursuing this path, two technical difficulties arise:

- As w need not be convex, the convex function obtained by linearizing w need neither be an upper nor a lower approximation of $f = c - w$. As a result, the link between the above subproblem and (1) is fragile, and this arises some technical difficulties in analyzing the above scheme.
- Exactly solving the linearized subproblem to define the next iterate can be unaffordable.

To circumvent the first inconvenient, we add the quadratic term $\frac{\mu}{2}\|x - x^\ell\|^2$ to the subproblem's objective and update the prox-parameter $\mu \geq 0$ iteratively in order to estimate the unknown constant μ_w given in (3). We target μ such that

$$c(x^{\ell+1}) - [w(x^\ell) + \langle g^\ell, x^{\ell+1} - x^\ell \rangle] + \frac{\mu}{2}\|x^{\ell+1} - x^\ell\|^2 \geq c(x^{\ell+1}) - w(x^{\ell+1}),$$

a key inequality strengthening the link between the linearized problem and (1). We stress that μ need not be an upper bound for μ_w, but large enough so that this last inequality holds for consecutive iterates. Note that if w is convex, then $\mu = 0$ does the job.

To get around the practical inconvenient of solving a difficult convex subproblem per iteration, we propose to employ PDA with a safeguard permitting to stop the algorithm as soon as an *incumbent* point is found. Declaring an incumbent point can be a delicate task as PDA may not produce feasible iterates: note that when $U(x) = \partial[c + \delta_X](x)$ in (8), the algorithm defines $\hat{x}^k \in X$, $x^{k+1} \in S$, and the constraint $X \cap S$ is satisfied only asymptotically if no further structure is assumed. Therefore, to declare an incumbent point we furnish PDA with a descent test accompanied by a penalty function $P_X : \mathbb{R}^n \to \mathbb{R}_+$ associated with the convex closed set X, that is, $P_X(x) = 0$ if and only if $x \in X$. We are interested in nonsmooth penalty functions, so that exact penalization is possible. For instance, if X is simple enough to perform the projection with respect to a given norm $\|\cdot\|_\diamond$, then we may take $P_X(x) = \rho \min_{z \in X} \|z - x\|_\diamond$, with $\rho > 0$ a penalty parameter. If, instead, X is defined by a convex mapping $G : \mathbb{R}^n \to \mathbb{R}^m$, that is, $X = \{x \in \mathbb{R}^n : G(x) \leq 0\}$, then we may take $P_X(x) = \rho \sum_{i=1}^m \max\{G_i(x), 0\}$. We stress that the use of exact penalty functions is restricted to the descent test, and they do not enter the optimization subproblems. This is

an important feature worth highlighting as it avoids numerical errors frequently encountered in penalization techniques. For problems satisfying the favorable condition

$$\texttt{proj}_S(X) \subset X, \tag{9}$$

as the ones of Subsection 2.2.2 above, the use of a penalty function is needless. These ideas are detailed in Algorithm 1.

Algorithm 1 CwC Progressive Decoupling Algorithm (`CwC-PDA`)

Given $x^1 \in X \cap S$, choose $y^1 \in S^\perp$, compute $g^1 \in \partial^{\mathsf{C}} w(x^1)$ ▷ Step 0: Initialization
Choose $0 < r_{\min} \le r^1 \le r_{\max} < \infty$, $\kappa \in (0, \frac{1}{2})$, $\mu^1 > 0$, and $\texttt{Tol} > 0$
Given a penalty function P_X, define $\ell_1 \leftarrow 1$

for $k = 1, 2, 3, \ldots$ **do**
  Let $\hat{x}^k$ be the solution to the strongly convex program ▷ Step 1: Trial point

$$\min_{x \in X} \; c(x) - \langle g^{\ell_k} + y^k, x \rangle + \frac{\mu^k}{2}\|x - x^{\ell_k}\|^2 + \frac{r^{\ell_k}}{2}\|x - x^k\|^2 \tag{10}$$

  Define $x^{k+1} \leftarrow \texttt{proj}_S(\hat{x}^k)$ and $y^{k+1} \leftarrow y^k - r^{\ell_k}(\hat{x}^k - x^{k+1})$

  Set $v_k \leftarrow \max\{\|x^{k+1} - x^{\ell_k}\|^2, \|y^{k+1} - y^k\|^2, \|x^{k+1} - x^k\|^2\}$ ▷ Step 2: Descent test
  if $f(x^{k+1}) + P_X(x^{k+1}) \le f(x^{\ell_k}) + P_X(x^{\ell_k}) - \frac{\kappa}{2} v_k$ **then**
    Set $\ell_{k+1} \leftarrow k+1$, $x^{\ell_{k+1}} \leftarrow x^{k+1}$, and $\mu^{k+1} \leftarrow \mu^k$ ▷ Serious step
    Choose $r^{\ell_{k+1}} \in [r_{\min}, r_{\max}]$ and compute $g^{\ell_{k+1}} \in \partial^{\mathsf{C}} w(x^{\ell_{k+1}})$
  else
    Define $\ell_{k+1} \leftarrow \ell_k$ and compute ▷ Null step

$$\nu^k \leftarrow 2 \max\left\{ \frac{w(x^{\ell_k}) + \langle g^{\ell_k}, x^{k+1} - x^{\ell_k} \rangle - w(x^{k+1})}{\|x^{k+1} - x^{\ell_k}\|^2},\; 0 \right\} \tag{11}$$

    If $\nu^k \ge \mu^k - 2\kappa$, set $\mu^{k+1} \leftarrow \nu^k + 1$; otherwise $\mu^{k+1} \leftarrow \mu^k$
  end if
end for

Step 1 consists of a PDA iteration applied to the maximal monotone operator

$$\tilde{T}^{\ell_k}(x) := \partial[c + \delta_X](x) - g^{\ell_k} + \mu^k[x - x^{\ell_k}], \tag{12}$$

which is an approximation of the nonmonotone one $T(x) = \partial[c + \delta_X](x) - \partial^{\mathsf{C}} w(x)$ given in (6). Indeed, the point $\hat{x}^k$ satisfies $0 \in T^k(\hat{x}^k)$, with $T^k(x) = \tilde{T}^{\ell_k}(x) - y^k + r^{\ell_k}[x - x^k]$. The points x^{k+1} and y^{k+1} are exactly as in (8), with $e = 0$. Furthermore, as $\hat{x}^k - x^{k+1} = \hat{x}^k - \texttt{proj}_S(\hat{x}^k) = \texttt{proj}_{S^\perp}(\hat{x}^k)$ we get that $y^{k+1} \in S^\perp$ provided $y^k \in S^\perp$. Since $y^1 \in S^\perp$, we conclude from the definition of $\hat{x}^k$ and y^{k+1} that

$$y^k \in \partial[c + \delta_X](\hat{x}^k) - g^{\ell_k} + \mu^k[\hat{x}^k - x^{\ell_k}] + r^{\ell_k}[\hat{x}^k - x^k] \quad \text{and} \quad y^k \in S^\perp \quad \text{for all } k. \tag{13}$$

Note, however, a crucial difference with (8): the operator $\tilde{T}^{\ell_k}$ changes along iterations in contrast with the progressive decoupling algorithm that keeps the same elicitable operator U.

In the examples of Section 2.2, the feasible set is the Cartesian product $X = X_i \times \cdots \times X_N$, and the function c has the following additive structure $c(x) = \sum_{i=1}^N c_i(x_i)$. As a result, the convex subproblem (10) splits into N independent and smaller subproblems:

$$\hat{x}_i^k = \operatorname{argmin}_{x_i \in X_i} \; c_i(x_i) - \langle g_i^{\ell_k} + y_i^k, x_i \rangle + \frac{\mu^k}{2}\|x_i - x_i^{\ell_k}\|^2 + \tfrac{r^{\ell_k}}{2}\|x_i - x_i^k\|^2, \quad i = 1, \ldots, N.$$

Hence, the task of computing $\hat{x}^k$ in Step 1 is embarrassingly parallel.

Observe that Step 1 gives $\hat{x}^k \in X$ and $x^{k+1} \in S$. If condition (9) holds, then it follows that $x^{k+1} \in S \cap X$ and the algorithm produces a sequence of feasible points $\{x^k\}$ to problem (1). In

this favorable case, the penalty function P_X is needless. However, if (9) does not hold, then P_X plays an important role in measuring the algorithm's progress and classifying iterates as serious (incumbent) or null ones.

The dichotomy between serious and null iterates is borrowed from bundle methods [25, Ch. 10]. In such a class of methods, serious iterates are those improving the objective function and are kept (momentary or not) as the method's stability center, that is, the best known solution candidate so far. This is exactly the same case here, as the serious iterate x^{ℓ_k} enters as a stability center in (10). Concerning null steps, they are useful in bundle methods to improve a model that approximates the objective function. This is not the case for Algorithm 1, but note that if $\hat{x}^k = \hat{x}$ and $\mu^k = \mu$ remain fixed forever, then the algorithm boils down to the PDA applied to the operator given in (12) and, as a consequence, will eventually solve the convex problem $\min_{x\in X\cap S}\ c(x) - [w(\hat{x}) + \langle \hat{g}, x - \hat{x}\rangle + \frac{\mu}{2}\|x-\hat{x}\|^2$ (that can be understood as the evaluation of a proximal mapping, just like bundle methods do).

After a null step, the proximal parameter is increased if $\nu^k \geq \mu^k - 2\kappa$. This rule, originally from [2], has the goal of estimating the parameter μ_w from assumption ii). Indeed, μ^k is increased to $\nu^k + 1$ ($\geq \mu^k - 2\kappa + 1 > \mu^k$) whenever the inequality (3) is compromised: note that (11) gives

$$w(x^{k+1}) \geq w(x^{\ell_k}) + \langle g^{\ell_k}, x^{k+1} - x^{\ell_k}\rangle - \frac{\nu^k}{2}\|x^{k+1} - x^{\ell_k}\|^2,$$

which mimics (3) with $y = x^{k+1}$ and $x = x^k$. The inequality $\nu > \mu^k$ indicates that the current prox-parameter μ^k is a poor estimation for μ_w, and thus the algorithm chooses $\mu^{k+1} > \mu^k$. However, in our convergence analysis, it is mandatory that sequence $\{\mu^k\}$ becomes eventually constant if the algorithm stops producing serious iterates. In other words, the prox-parameter is permitted to increase only finitely many times in the event of an infinite sequence of consecutive null steps. This is why the algorithm employs the rule at line 13. This claim is formally stated in the first lemma below.

3.2 Convergence Analysis

We start the convergence analysis of Algorithm 1 (`CwC-PDA`) with the following result from [2] concerning the prox-parameter sequence. For the reader's convenience, we give its proof in the Appendix.

Lemma 3.1 (Lemma 5.1 from [2]) *The value $\mu_{\max} := \sup_{k\in\mathbb{N}} \mu^k$ is finite. Furthermore, if the algorithm produces an infinite sequence of null steps after a last serious step, then the prox-parameter becomes eventually constant.*

Proposition 3.1 (Finitely many serious steps) *Assume* (5a) *and suppose that after a certain iteration, no more serious steps are performed: $\ell_k = \ell$ is fixed. Then*

a) $\lim_{k\to\infty} x^{k+1} = \tilde{x}$ *and* $\lim_{k\to\infty} v_k = \|\tilde{x} - x^\ell\|$, *with $\tilde{x}$ the unique solution to the convex problem*

$$\min_{x\in X\cap S}\ c(x) - [w(x^\ell) + \langle g^\ell, x - x^\ell\rangle] + \frac{\mu'}{2}\|x - x^\ell\|^2, \tag{14}$$

and $\mu' > 0$ is such that $\mu^k = \mu'$ for all k large enough;

b) If $P_X(\cdot)$ is an exact penalty for (14), *that is, $\tilde{x}$ also solves the (partially) penalized problem*

$$\min_{x\in S}\ c(x) + P_X(x) - [w(x^\ell) + \langle g^\ell, x - x^\ell\rangle] + \frac{\mu'}{2}\|x - x^\ell\|^2, \tag{15}$$

then $\tilde{x}$ equals x^ℓ and is a critical point to (1).

Proof We highlight that the updating rule for μ^k in Algorithm 1 ensures that the sequence $\{\mu^k\}_{k>\ell}$ is non-decreasing and becomes constant at a certain value $\mu' \in (0, \mu_{\max}]$ after finitely many steps $k' > \ell$ as a consequence of Lemma 3.1. More precisely, the updating rule at line 13 of Algorithm 1 ensures that

$$\mu^k = \mu' \quad \text{and} \quad \nu^k + 2\kappa < \mu' \quad \text{for all } k > k'. \tag{16}$$

Hence, from iteration k' on, Algorithm 1 behaves as the PDA applied to the convex problem (14). Theorem 2.1 under condition (5a) ensures that: (i) the whole sequence $\{x^k\}$ converges to the point $\tilde{x}$ solving (14); (ii) the whole sequence $\{y^k\}$ converges to a point $\tilde{y} \in S^\perp$; (iii) the pair $(\tilde{x}, \tilde{y}) \in S \times S^\perp$ satisfies $\tilde{y} \in \partial[c + \delta_X](\tilde{x}) - g^\ell + \mu'(\tilde{x} - x^\ell)$. Properties (i)-(iii) and definition of v_k prove item a).

Concerning item b), we claim that $\tilde{x} = x^\ell$ under the stated additional assumption, and thus x^ℓ satisfies, due to (iii) above, the criticality condition (6). To show that, let us assume the opposite, i.e., $\tilde{x} \neq x^\ell$, and arrive to a contradiction. In this case, first observe from (i) and (ii) that for every $\epsilon > 0$ small enough, there exists $k'' \geq k'$ such that $\|x^{k+1} - x^\ell\|^2 \geq \varepsilon \geq \|y^{k+1} - y^k\|^2$ and $\epsilon \geq \|x^{k+1} - x^k\|^2$ for all $k \geq k''$. As a result, the measure v_k becomes $v_k = \|x^{k+1} - x^\ell\|^2$ for all $k \geq k''$, and the inequality

$$f(x^{k+1}) + P_X(x^{k+1}) > f(x^\ell) + P_X(x^\ell) - \frac{\kappa}{2}\|x^{k+1} - x^\ell\|^2 \quad \text{for all } k \geq k'' \tag{17}$$

follows from the assumption that only null steps are produced after $k \geq \ell$. Next, observe that for $k > k''$ $(\geq k')$,

$$w(x^\ell) + \langle g^\ell, x^{k+1} - x^\ell \rangle \leq w(x^{k+1}) + \frac{\nu^k}{2}\|x^{k+1} - x^\ell\|^2 < w(x^{k+1}) + \frac{\mu' - 2\kappa}{2}\|x^{k+1} - x^\ell\|^2.$$

By passing to the limit as k goes to infinity and recalling continuity of w we get

$$-[w(x^\ell) + \langle g^\ell, \tilde{x} - x^\ell \rangle] \geq -w(\tilde{x}) + \frac{2\kappa - \mu'}{2}\|\tilde{x} - x^\ell\|^2.$$

Furthermore, as $P_X(\cdot)$ is an exact penalty function for (14), we get that $\tilde{x}$ solving (14) also solves (15) [26, Prop. 1.5.1] . Since x^ℓ is feasible to the latter problem, the above inequality yields

$$\begin{aligned} f(x^\ell) + P_X(x^\ell) = c(x^\ell) - w(x^\ell) + P_X(x^\ell) &\geq c(\tilde{x}) - [w(x^\ell) + \langle g^\ell, \tilde{x} - x^\ell \rangle] + \frac{\mu'}{2}\|\tilde{x} - x^\ell\|^2 + P_X(\tilde{x}) \\ &\geq c(\tilde{x}) - w(\tilde{x}) + \frac{2\kappa}{2}\|\tilde{x} - x^\ell\|^2 + P_X(\tilde{x}) \\ &= f(\tilde{x}) + P_X(\tilde{x}) + \frac{2\kappa}{2}\|\tilde{x} - x^\ell\|^2. \end{aligned}$$

Thus, in view of (17) and the fact that $P_X(\tilde{x}) = 0$ because $\tilde{x}$ solves (14),

$$f(\tilde{x}) \leq f(x^\ell) + P_X(x^\ell) - \frac{2\kappa}{2}\|\tilde{x} - x^\ell\|^2 < f(x^{k+1}) + P_X(x^{k+1}) + \frac{\kappa}{2}\|x^{k+1} - x^\ell\|^2 - \frac{2\kappa}{2}\|\tilde{x} - x^\ell\|^2.$$

By passing to the limit as k goes to infinity we get $f(\tilde{x}) \leq f(\tilde{x}) + 0 - \frac{\kappa}{2}\|\tilde{x} - x^\ell\|^2$, contradicting thus our assumption that $\tilde{x} \neq x^\ell$. Hence $\tilde{x} = x^\ell$ solves (14), $v_k \to 0$ and the proof is complete. □

Proposition 3.2 (Infinitely many serious steps) *Consider problem* (1) *and assume that the level set* $\mathcal{L}_f(x^1) := \{x \in X \cap S : f(x) \leq f(x^1)\}$ *is bounded. Suppose that Algorithm 1 produces infinitely many serious steps, i.e.,* $\lim_{k\to\infty} \ell_k = \infty$. *Then all cluster points of the sequence* $\{x^{\ell_k}\}_k$ *are critical points to problem* (1). *Furthermore, let* $\mathcal{K}$ *be the index set of a converging subsequence of* $\{x^{\ell_k}\}_k$. *Then* $\lim_{\mathcal{K}\ni k\to\infty} v_k = 0$.

Proof Let $\iota_k := \ell_{k+1} - 1$. With this notation the descent test at serious steps reads as

$$f(x^{\ell_{k+1}}) + P_X(x^{\ell_{k+1}}) \leq f(x^{\ell_k}) + P_X(x^{\ell_k}) - \frac{\kappa}{2} v_{\iota_k},$$

with $v_{\iota_k} = \max\{\|x^{\ell_{k+1}} - x^{\ell_k}\|^2, \|y^{\ell_{k+1}} - y^{\iota_k}\|^2, \|x^{\ell_{k+1}} - x^{\iota_k}\|^2\}$. Monotonicity of $\{f(x^{\ell_k}) + P_X(x^{\ell_k})\}_k$ and the assumption that the level set is bounded yield

$$0 \leq \frac{\kappa}{2} \sum_{k=1}^{\infty} v_{\iota_k} \leq \sum_{\ell=1}^{\infty} [f(x^{\ell_k}) + P_X(x^{\ell_k}) - f(x^{\ell_{k+1}}) - P_X(x^{\ell_{k+1}})] = f(x^1) - \lim_{\ell \to \infty} [f(x^{\ell_{k+1}}) + P_X(x^{\ell_{k+1}})] < \infty,$$

showing that $\lim_{k\to\infty} v_{\iota_k} = 0$. As a result,

$$0 = \lim_{k\to\infty} \|x^{\ell_{k+1}} - x^{\ell_k}\| = \lim_{k\to\infty} \|y^{\ell_{k+1}} - y^{\iota_k}\| = \lim_{k\to\infty} \|x^{\ell_{k+1}} - x^{\iota_k}\|, \tag{18}$$

and thus, from Step 1,

$$\lim_{k\to\infty} (r^{\ell_k})^2 \|\hat{x}^{\iota_k} - x^{\ell_{k+1}}\|^2 = \lim_{k\to\infty} \|y^{\ell_{k+1}} - y^{\iota_k}\|^2 = 0.$$

Boundedness of $\{r^{\ell_k}\}_k$ gives that $\lim_{k\to\infty} \|\hat{x}^{\iota_k} - x^{\ell_{k+1}}\| = 0$. As $\{x^{\ell_k}\}_k \subset \mathcal{L}_f(x^1)$, we conclude that this sequence is bounded and has at least one cluster point $\bar{x}$: there exists an index set $\mathcal{K} \subset \{1, 2, \ldots\}$ such that $\lim_{\mathcal{K}\ni k\to\infty} x^{\ell_k} = \bar{x} \in S$. As $0 = \lim_{k\to\infty} \|x^{\ell_{k+1}} - x^{\ell_k}\|$, then $\lim_{\mathcal{K}\ni k\to\infty} x^{\ell_{k+1}} = \bar{x}$. Moreover, $\lim_{\mathcal{K}\ni k\to\infty} \|\hat{x}^{\iota_k} - \bar{x}\| = \lim_{\mathcal{K}\ni k\to\infty} \|(\hat{x}^{\iota_k} - x^{\ell_{k+1}}) + (x^{\ell_{k+1}} - \bar{x})\| = 0$ due to the above limits. All in all,

$$\bar{x} = \lim_{\mathcal{K}\ni k\to\infty} x^{\ell_k} = \lim_{\mathcal{K}\ni k\to\infty} x^{\ell_{k+1}} = \lim_{\mathcal{K}\ni k\to\infty} \hat{x}^{\iota_k} = \lim_{\mathcal{K}\ni k\to\infty} x^{\iota_k},$$

and $\lim_{\mathcal{K}\ni k\to\infty} v_{\iota_k} = \lim_{\mathcal{K}\ni k\to\infty} v_{\ell_{k+1}-1} = \lim_{\mathcal{K}\ni k\to\infty} v_k = 0$. We now rewrite (13) with $k = \ell_k$ and use the notation ι_k to get

$$g^{\ell_k} - r^{\ell_k}(\hat{x}^{\iota_k} - x^{\iota_k}) - \mu^{\iota_k}(\hat{x}^{\iota_k} - x^{\ell_k}) \in \partial[c + \delta_X](\hat{x}^{\iota_k}) - y^{\iota_k}.$$

As $y^k \in S^\perp$ for all k, we get that $-y^k \in S^\perp$ and thus

$$g^{\ell_k} - r^{\ell_k}(\hat{x}^{\iota_k} - x^{\iota_k}) - \mu^{\iota_k}(\hat{x}^{\iota_k} - x^{\ell_k}) \in \partial[c + \delta_X](\hat{x}^{\iota_k}) + S^\perp.$$

Since the generalized subddifferential is outer-semicontinuous and locally bounded [21, Props. 2.1.2(a) and 2.1.5(b)], we find $\mathcal{K}' \subset \mathcal{K}$ such that $\{g^{\ell_k}\}_{k\in\mathcal{K}'}$ (with $g^{\ell_k} \in \partial^{\mathsf{C}} w(x^{\ell_k})$) converges to an element $\bar{g} \in \partial^{\mathsf{C}} w(\bar{x})$. As $\{\mu^k\}$ is bounded (c.f. Lemma 3.1), by passing to the limit with $k \in \mathcal{K}'$ going to infinity at the above inclusion (recalling (18) and outer semi-continuity of $\partial[c + \delta_X](\hat{x}^{\iota_k})$) we obtain $\bar{g} \in \partial[c + \delta_X](\bar{x}) + S^\perp$, which implies that $\bar{x} \in X$ is a critical point (c.f. (6)). □

Theorem 3.1 *Consider Algorithm 1 applied to problem* (1)*. Under the assumptions i)-iii) on problem* (1)*, suppose further that the level set* $\mathcal{L}_f(x^1) := \{x \in X \cap S : f(x) \leq f(x^1)\}$ *is bounded. Then the following holds:*

a) If only finitely many serious steps are produced, then $\lim_{k\to\infty} x^{k+1} = \tilde{x}$ *and* $\lim_{k\to\infty} v_k = \|\tilde{x} - x^\ell\|$*, with* $\tilde{x}$ *the unique solution to the convex problem* (14)*, with* x^ℓ *the last serious iterate and* $\mu_k = \mu'$ *for all* k *large enough. If, in addition,* $P_X(\cdot)$ *is an exact penalty for* (14)*, then* $\tilde{x}$ *equals* x^ℓ *and is a critical point to* (1)*.*

b) If the algorithm produces infinitely many serious steps, then all cluster points of the sequence $\{x^{\ell_k}\}_k$ *are critical points to problem* (1)*. Furthermore,* $\liminf_k v_k = 0$*.*

Proof Item a) follows from Proposition 3.1 and item b) from Proposition 3.2. □

Corollary 3.1 *If* (9) *holds, then no penalty function is necessary. The results of Theorem 3.1 can be summarized as follows: if the function has bounded level set, then all cluster points of the sequence* $\{x^{\ell_k}\}_k$ *are critical points to problem* (1)*.*

3.3 Additional Comments and Specialization to the DC Setting

A possible stopping test for algorithm 1 is $v_k \leq \texttt{Tol}$, with $\texttt{Tol} \geq 0$ a given tolerance. Theorem 3.1 ensures that, provided $P_X(\cdot)$ is an exact penalty function, $\liminf_k v_k = 0$. It remains to know whether $v_k = 0$ implies that x^{ℓ_k} is critical point.

Proposition 3.3 *Suppose that $v_k = 0$ at a certain iteration of Algorithm 1. Then x^{ℓ_k} is a critical point to problem* (1).

Proof Observe that $v_k = 0$ implies $x^{k+1} = x^k = x^{\ell_k}$ and $y^{k+1} = y^k$. The latter equality implies $\hat{x}^k = x^{k+1} (= x^k)$. Inclusion (13) becomes $y^k \in \partial[c + \delta_X](x^{\ell_k}) - g^{\ell_k}$, with $g^{\ell_k} \in \partial^{\mathtt{C}} w(x^{\ell_k})$, i.e., $x^k \in S$ and $y^k \in S^\perp$ solve the linkage problem (6). □

The sequence $\{v_k\}$ can also be used to inquire about the correctness of the penalty function $P_X(\cdot)$. Indeed, if the algorithm stops performing serious steps, v_k is bounded away from 0 but $v_k >> \|x^{k+1} - x^k\| \approx 0$ and $v_k >> \|y^{k+1} - y^k\| \approx 0$, we can infer from Proposition 3.1 that $\{x^k\}$ is converging to the solution $\tilde{x}$ of subproblem (14), and $\tilde{x} \neq x^\ell$. Given Proposition 3.1 item b), this can only happen if P_X is not an exact penalty function. Thus, if this behavior happens, we can update the penalty function (normally by increasing its penalty parameter) and continue with the algorithm's iterative process.

It turns out that for problems having a DC structure, the quadratic term $\frac{\mu^k}{2}\|x - x^{\ell_k}\|^2$ is needless. Therefore, Algorithm 1 can be simplified by taking $\mu^k = 0$ for all k and removing the rule for updating μ^k. In the following analysis we assume (9) for the sake of simplicity.

Theorem 3.2 (Simplified variant: specialization to the DC setting) *In addition to the assumptions in Theorem 3.1, suppose that* (9) *holds and w is strongly convex with modulus $\gamma > 0$. Furthermore, consider the following simplified variant of Algorithm 1 with $\kappa \in (0, \gamma)$, $\mu^k = 0$ for all k, the penalty function $P_X(\cdot)$ omitted, and equation* (11) *and line 13 suppressed. Then all cluster points of the sequence $\{x^{\ell_k}\}_k$ are critical points to problem* (1).

Proof *Infinitely many serious steps.* For the case in which the algorithm produces infinitely many serious steps, the result follows directly from the proof of Proposition 3.2 by taking therein $\mu^k = 0$ for all k. No additional analysis is necessary.

Finitely many serious steps. Let k' the last serious iteration. Then, for all $k > k'$, $\ell_k = \ell$ is fixed and this variant of Algorithm 1 behaves as the PDA applied to the convex problem (recall that $\mu^k = 0$)

$$\min_{x \in X \cap S} \; c(x) - [w(x^\ell) + \langle g^\ell, x - x^\ell \rangle].$$

Theorem 2.1 under condition (5a) ensures that: (i) the whole sequence $\{x^k\}$ converges to the point $\tilde{x}$ solving this subproblem; (ii) the whole sequence $\{y^k\}$ converges to a point $\tilde{y} \in S^\perp$; (iii) the pair $(\tilde{x}, \tilde{y}) \in S \times S^\perp$ satisfies $\tilde{y} \in \partial[c + \delta_X](\tilde{x}) - g^\ell$. These properties ensure that there exists $k'' > k'$ such that $v_k = \|x^k - x^\ell\|$ for all $k \geq k''$. Furthermore, strong convexity of w implies that

$$c(\tilde{x}) - w(\tilde{x}) + \frac{\gamma}{2}\|\tilde{x} - x^\ell\|^2 \leq c(\tilde{x}) - [w(x^\ell) + \langle g^\ell, \tilde{x} - x^\ell \rangle] \leq c(x^\ell) - w(x^\ell),$$

i.e., $f(\tilde{x}) \leq f(x^\ell) - \frac{\gamma}{2}\|\tilde{x} - x^\ell\|^2$. As $\kappa \in (0, \gamma)$, we conclude that $f(\tilde{x}) < f(x^\ell) - \frac{\kappa}{2}\|\tilde{x} - x^\ell\|^2$. If $\tilde{x} \neq x^\ell$, continuity of f and the fact that $v_k = \|x^k - x^\ell\| \to \|\tilde{x} - x^\ell\|$ would give a new serious step, contradicting the assumption that only null steps are produced after k'. Thus $\tilde{x} = x^\ell$ and optimality of x^ℓ to the above subproblem coincides with criticality to problem (1). □

4 Numerical Experiments

We consider two classes of stochastic optimization problems: two-stage stochastic standard quadratic optimization and two-stage stochastic programming with decision-dependent probability. Our `Matlab` codes are freely available at `https://www.oliveira.mat.br/solvers`.

4.1 Two-Stage Stochastic Standard Quadratic optimization.

This subsection considers a special family of nonconvex stochastic programs (2), with $T = 2$ and quadratic objective. The problem, which finds applications in minimal variance portfolio investments, social network problems, and others [27], is denoted by two-stage stochastic standard quadratic optimization problem and is formulated as:

$$\begin{cases} \min\limits_{z,u} \; z^\top A z + \sum_{i=1}^N p_i [2 z^\top B(\xi^i)^\top u_i + u_i^\top C(\xi^i) u_i] \\ \text{s.t.} \;\; e^\top z + e^\top u_i = 1, \;\; i = 1, \ldots, N \\ \qquad z \geq 0, \; u_i \geq 0, \qquad i = 1, \ldots, N, \end{cases}$$

with $A \in \mathbb{R}^{n_1 \times n_1}$, $B(\xi) \in \mathbb{R}^{n_2 \times n_1}$, and $C(\xi) \in \mathbb{R}^{n_2 \times n_2}$ given (non positive semi-definite) matrices. Observe that he problem's size rapidly increases with the number of scenarios N. To decompose this problem by scenario, we replicate the z vector N times, denote the linear subspace by

$$S := \Big\{ (z_i, u_i) \in \mathbb{R}^{(n_1+n_2)N}, \; i = 1, \ldots, N \; : \; z_1 = \cdots = z_N \Big\}, \tag{19}$$

and rewrite the problem as

$$\begin{cases} \min\limits_{(z,u) \in S} \; \sum_{i=1}^N p_i [z_i^\top A z_i + 2 z_i^\top B(\xi^i)^\top u_i + u_i^\top C(\xi^i) u_i] \\ \text{s.t.} \qquad e^\top z_i + e^\top u_i = 1, \;\; i = 1, \ldots, N \\ \qquad\quad (z_i, u_i) \geq 0, \qquad i = 1, \ldots, N. \end{cases} \tag{20}$$

Without further assumptions, convexity is not elicitable (because the problem is nonconvex even for $N = 1$). However, the above problem fits our general structure (1) with $c \equiv 0$ and w the objective function in (20) multiplied by minus one: w has Lipschitz gradients, and is thus weakly convex. In this case, the solution of the convex subproblem (10) is nothing but the projection of N vectors of dimension $n_1 + n_2$ onto the simplex. These projections can be computed in parallel by efficient specialized algorithms.

We have randomly generated data by the following rule: matrices A and $C(\xi)$ are randomly and uniformly generated in the interval $[0, 1]$, and matrices $B(\xi)$ are uniformly generated in the interval $[-1, 0]$ (in this way, both vectors z_i and u_i composing a solution of (20) are nonzero vectors). None of these matrices needs to be positive semi-definite. We have considered 100 instances of problem (20), obtained by varying $n_1, n_2 \in \{5, 10, 15, 20, 30\}$ and $N \in \{1000, 2000, 3000, 4000\}$, and the following solvers:

- `CwC-PDA` - Convex-weakly convex Progressive Decoupling Algorithm. This is Algorithm 1 coded in `Matlab` with the following choice of parameters: $r^1 = 5 \cdot 10^{-2}$ and $r^k \in [10^{-5}, 10^3]$ increased by 10% after three consecutive null steps, and decreased by the same amount after three consecutive serious steps. We also set $\mu^1 = 2 \cdot 10^{-5} + 10^{-8}$, $\kappa = 10^{-5}$, and $\texttt{MaxIter} = 6 \times 10^3$. We have employed the stopping test $v_k \leq \texttt{Tol}$, with $\texttt{Tol} = 5 \times 10^{-5}$.
- `IPOPT` - Interior Point Optimizer. This solver was called from `Matlab` through the interface available at `https://github.com/ebertolazzi/mexIPOPT`. We applied `IPOPT` twice, with tolerances 10^{-5} and 10^{-6}. Figure 3 employs the functional value and CPU time of `IPOPT` with $\texttt{Tol} = 10^{-6}$ as references.

Since problem (20) does not satisfy condition (9), we have equipped `CwC-PDA` with the penalty function $P(x) = 500 \sum_{i=1}^N p_i |e^\top z_i + e^\top u_i - 1|$. With this choice, the critical points computed by the solver satisfy $\sum_{i=1}^N p_i |\bar{e}^\top \bar{z}_i + e^\top \bar{u}_i - 1| \leq 10^{-7}$ in all problem instances (recall that, being an interior point method, `IPOPT` computes feasible points). Figure 3 summarizes the obtained results.

Concerning the function values, the figure on the left shows that the median (red line) is lower for the values computed by `CwC-PDA`, indicating that the critical points computed by the solver are of better quality than those computed by `IPOPT`. With respect to CPU time (the figure on the right), it is clear that `CwC-PDA` outperformed `IPOPT` in this class of problems. Indeed, `CwC-PDA`

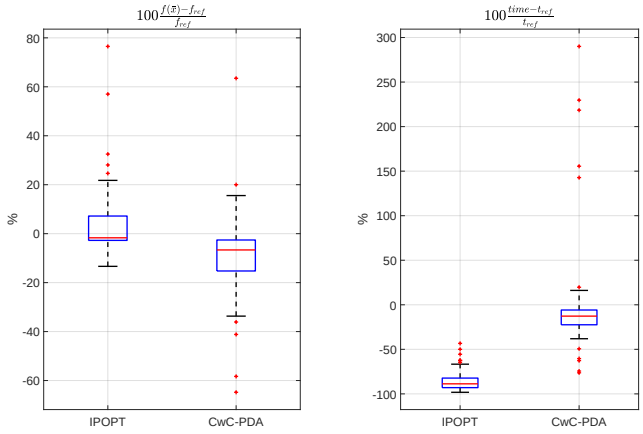

Fig. 3 Comparison between `CwC-PDA` and `IPOPT` (`Tol` $= 10^{-5}$) on 100 instances of two-stage stochastic standard quadratic problem. Functional value and CPU time of `IPOPT` with `Tol` $= 10^{-6}$ are taken as references.

computed critical points for the 100 test problems in less than one hour, while `IPOPT` (`Tol` $= 10^{-6}$) required more than eight hours in a computer with the following configuration: Intel Core i7, CPU @ 2.70GHz, 32 GB RAM, 64-Bit Window 10, and Matlab 2021b.

4.2 Two-Stage Stochastic Programming with Decision-Dependent Probability

This subsection presents numerical results for a nonconvex two-stage problem of the form

$$\begin{cases} \min\limits_{x,y} \tilde{c}^\top x + \sum\limits_{i=1}^{N} p_i(x)[q(\xi^i)^\top y(\xi^i)] \\ \text{s.t.} \;\; x \in X, \; y(\xi^i) \in Y, \qquad\qquad i = 1, \ldots, N \\ \qquad T(\xi^i)x + W(\xi^i)y(\xi^i) = h(\xi^i), \; i = 1, \ldots, N. \end{cases}$$

The difficulty here is that the scenario probability depends on the decision variable x: we assume that $p(x) = \sum_{j \in J} \tilde{p}^j x_j \in \mathbb{R}^N$, with $\tilde{p}^j$ vectors on the $(N+1)$-simplex. The problem fits our general structure (1) with $c \equiv 0$ and w the objective function above multiplied by minus one: w has Lipschitz gradients, and is thus weakly convex. We consider the power system planning application from [19], with data and solver `DPME` available at `https://github.com/lhyoung99/Decomposition_NonvexSP`:

- `DPME` - Partial Moreau envelope method. This is the solver (in `Matlab`) provided by the authors of [19], with default parameters;
- `CwC-PDA` - Convex-weakly Convex Progressive Decoupling Algorithm. This is Algorithm 1 with the following choice of parameters: $r^1 = 5 \cdot 10^{-4}$ and $r^k \in [10^{-5}, 10^3]$ increased by 10% after three consecutive null steps, and decreased by the same amount after three consecutive serious steps. We also set $\mu^1 = 2 \cdot 10^{-5} + 10^{-8}$, $\kappa = 10^{-5}$, and let the algorithm stop after 10 (variant `CwC-PDA-10`) and 20 (variant `CwC-PDA-20`) iterations.

As in [19], the solvers are initialized with the same initial point. Figures 4 and 5 report some results obtained by both solvers on 100 instances of the problem with $N = 5000$ and $N = 10000$ scenarios, respectively. Once the solvers provide an approximate critical point $\bar{x}$, the function value is computed as

$$f(\bar{x}) = \tilde{c}^\top \bar{x} + \sum_{i=1}^{N} p_i(\bar{x}) \left\{ \min_{y \in Y} \; q(\xi^i)^\top y \quad \text{s.t.} \quad W(\xi^i)y = h(\xi^i) - T(\xi^i)\bar{x} \right\}.$$

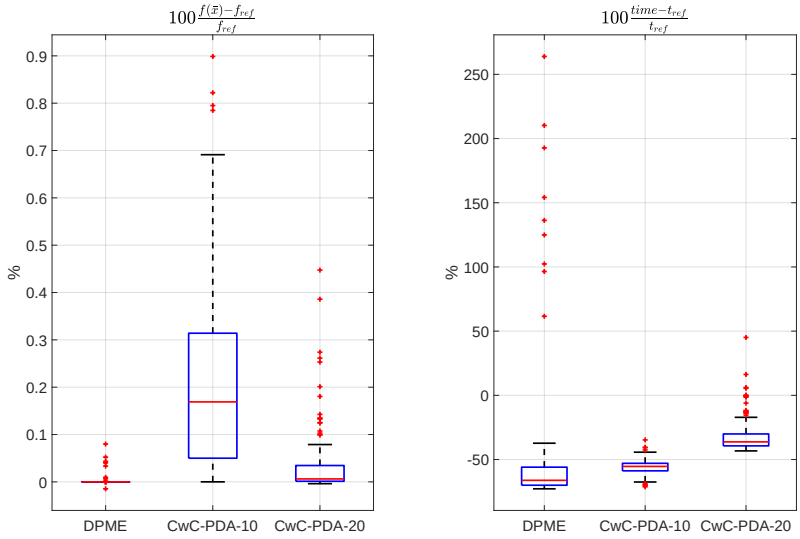

Fig. 4 Comparison between `CwC-PDA` and `DPME` on 100 instances of the two-stage stochastic problem with $N = 5000$ scenarios.

In Figures 4 and 5, t_{ref} stands for the CPU time required by `IPOPT` to compute the functional value f_{ref}.

In terms of functional value, Figure 4 shows that `IPOPT` and `DPME` have similar accuracy. The median of relative errors is 0.1689% for `CwC-PDA-10`, and 0.006% for `CwC-PDA-20`. As for infeasibility, the median is is $7.7 \cdot 10^{-5}$ for `CwC-PDA-10` and $1.1 \cdot 10^{-4}$ for `CwC-PDA-20`. The figure also shows that, compared to `IPOPT`, the solvers `DPME`, `CwC-PDA-10` and `CwC-PDA-20` reduce CPU time, being `DPME` the fastest one. Total CPU time in seconds required by the four solvers to solve the 100 instances are given in Table 1.

	CPU time in seconds			
Number of scenarios	`IPOPT`	`DPME`	`CwC-PDA-10`	`CwC-PDA-20`
5000	4567	2521	2024	3181
10000	8693	4605	3374	5636

Table 1 Total CPU time to solve 100 instances of the considered two-stage stochastic problem with decision-dependent probability.

For the case $N = 5000$, we have observed that, on average, `DPME` satisfies its stopping test in only twelve iterations. On the other hand, as expected from a splitting method, our approach quickly computes a reasonable approximate solution but takes long to improve its quality (compare variants `CwC-PDA-10` and `CwC-PDA-20`).

A similar analysis carries over Figure 5 that considers the case with $N = 10\,000$ scenarios. The median of relative errors is 0.304% for `CwC-PDA-10` and 0.022% for `CwC-PDA-20`. The median of CPU time reduction is 67% for `DPME`, 64% for `CwC-PDA-10`, and 35% for `CwC-PDA-20`. In our numerical experiments, almost all iterates of our approach were declared serious iterates.

We stress that for these test problems, both solvers require solving N quadratic programs (QPs) per iteration. However, if the problem had convex nonlinear second-stage subproblems, the conclusion on CPU time could be different: `DPME` would require solving N convex nonlinear programs per iteration, while `CwC-PDA` the same N QPs. This different computation burden per iteration can be counter-balanced by the fact that being a splitting method, `CwC-PDA` is likely to require more iterations to achieve the same level of quality than `DPME` does. We recall that if the problem had $T > 2$ stages or nonconvex subproblems, `DPME` would not apply. This fact contrasts

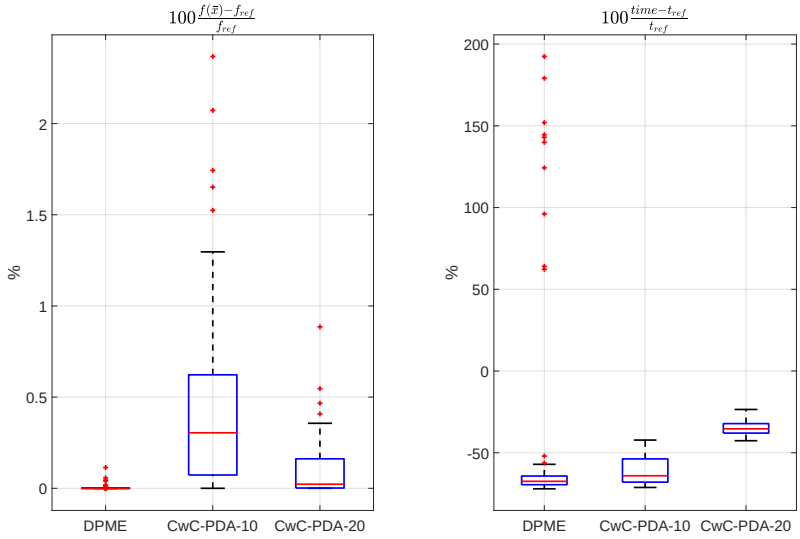

Fig. 5 Comparison between `CwC-PDA` and `DPME` on 100 instances of the two-stage stochastic problem with $N = 10000$ scenarios.

with `CwC-PDA` that is applicable in the nonlinear multistage setting as long as the objective function has the general CwC structure.

5 Conclusions

This work introduces a new variant of the progressive decoupling algorithm of [1] for a broad class of nonconvex and nonsmooth optimization problems consisting of minimizing the difference of convex and weakly convex functions over a linear subspace. Computing a critical point for problems of this class amounts to solving a linkage problem whose operator fails to be monotone. We combine linearization, penalization, and PDA to address the challenge. Penalization is restricted to the descent test and does not enter the optimization subproblems. Furthermore, penalization is dismissed in several applications of the consensus type, where the linear subspace appears as a mere modeling artifice. For the particular case of DC programming, the given algorithm can be significantly simplified (c.f. Theorem 3.2).

We provided convergence analysis (to critical points) and illustrated the approach's numerical performance on two nonconvex two-stage stochastic programs. Studying the method's convergence rate is left for future research.

Acknowledgements The first author acknowledges financial support from the Gaspard-Monge program for Optimization and Operations Research (PGMO) project "SOLEM - Scalable Optimization for Learning and Energy Management".
The datasets generated during and/or analysed during the current study are available from the corresponding author's web-page: `www.oliveira.mat.br/solvers`.

References

1. R. Tyrrell Rockafellar. Progressive decoupling of linkages in optimization and variational inequalities with elicitable convexity or monotonicity. *Set-Valued and Variational Analysis*, 27(4):863–893, oct 2019.
2. K. Syrtseva, W. de Oliveira, S. Demassey, and W. van Ackooij. Minimizing the difference of convex and weakly convex functions via bundle method. *To appear in Pacific Journal of Optimization*, 2024, doi: 10.61208/pjo-2024-004.
3. Roland Glowinski and Patrick Le Tallec. *Augmented Lagrangian Methods for the Solution of Variational Problems*, pages 45–121. 1987.

4. Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.
5. Jonathan E. Spingarn. Partial inverse of a monotone operator. *Applied Mathematics & Optimization*, 10(1):247–265, jun 1983.
6. R. T. Rockafellar and Roger J.-B. Wets. Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research*, 16(1):119–147, February 1991.
7. Welington de Oliveira. Risk-averse stochastic programming and distributionally robust optimization via operator splitting. *Set-Valued and Variational Analysis*, 29(4):861–891, Dec 2021.
8. Jim Douglas and H. H. Rachford. On the numerical solution of heat conduction problems in two and three space variables. *Transactions of the American Mathematical Society*, 82(2):421–439, 1956.
9. Jonathan Eckstein and Dimitri P. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1–3):293–318, April 1992.
10. Guoyin Li and Ting Kei Pong. Global convergence of splitting methods for nonconvex composite optimization. *SIAM Journal on Optimization*, 25(4):2434–2460, January 2015.
11. Guoyin Li and Ting Kei Pong. Douglas–Rachford splitting for nonconvex optimization with application to nonconvex feasibility problems. *Mathematical Programming*, 159(1–2):371–401, November 2016.
12. Andreas Themelis and Panagiotis Patrinos. Douglas–Rachford splitting and admm for nonconvex optimization: Tight convergence results. *SIAM Journal on Optimization*, 30(1):149–181, January 2020.
13. Felipe Atenas. Convergence rate of nonconvex Douglas-Rachford splitting via merit functions, with applications to weakly convex constrained optimization. Technical report, ArXiv 2303.16394, 2023.
14. Yu Yang, Xiaohong Guan, Qing-Shan Jia, Liang Yu, Bolun Xu, and Costas J. Spanos. A survey of ADMM variants for distributed optimization: Problems, algorithms and features. Technical Report 2208.03700, 2022. Availabel at `https://arxiv.org/abs/2208.03700`.
15. P. Hartman. On functions representable as a difference of convex functions. *Pacific Journal of Mathematics*, 9(3):167–198, 1959.
16. Hoai An Le Thi and Tao Pham Dinh. Dc programming and dca: thirty years of developments. *Mathematical Programming*, 169(1):5–68, January 2018.
17. Welington de Oliveira. The ABC of DC programming. *Set-Valued and Variational Analysis*, 28(4):679–706, 2020.
18. Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczyński. *Lectures on Stochastic Programming: Modeling and Theory*. MPS-SIAM Series on Optimization. SIAM - Society for Industrial and Applied Mathematics and Mathematical Programming Society, Philadelphia, 2009.
19. Hanyang Li and Ying Cui. A decomposition algorithm for two-stage stochastic programs with nonconvex recourse. Technical report, ArXiv:2204.01269, 2022.
20. J.-S. Pang Y. Cui. *Modern Nonconvex Nondifferentiable Optimization*. SIAM, 2022.
21. F.H. Clarke. *Optimisation and Nonsmooth Analysis*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, 1990.
22. Jean-Philippe Vial. Strong and weak convexity of sets and functions. *Mathematics of Operations Research*, 8(2):231–259, 1983.
23. R.T. Rockafelar. Favorable classes of lipschitz continuous functions in subgradient optimization. In *Progress in Nondifferentiable Optimization*, IIASA Collaborative Proceedings Series, International Institute of Applied Systems Analysis, Laxenburg, Austria, pages 125–144, 1982.
24. R.T. Rockafellar. *Convex Analysis*. Princeton University Press, 1st edition, 1970.
25. J.F. Bonnans, J.C. Gilbert, C. Lemaréchal, and C. Sagastizábal. *Numerical Optimization: Theoretical and Practical Aspects*. Springer-Verlag, 2nd edition, 2006.
26. Dimitri P. Bertsekas. *Convex Optimization Algorithms*. Number 1st. Athena Scientific, 2015.
27. Immanuel M. Bomze, Markus Gabl, Francesca Maggioni, and Georg Ch. Pflug. Two-stage stochastic standard quadratic optimization. *European Journal of Operational Research*, 299(1):21–34, 2022.

Proof of Lemma 3.1. Let $\bar{\mu} := \max\{\mu_w, \mu^1\} > 0$, with μ_w the convexification parameter of the weakly convex function w, and μ^1 given to the algorithm at initialization. Then, by taking $y := x^{k+1}$ and $x := x^{\ell_k}$ in (3) it follows that

$$2\frac{w(x^{\ell_k}) + \langle g^{\ell_k}, x^{k+1} - x^{\ell_k}\rangle - w(x^{k+1})}{\|x^{k+1} - x^{\ell_k}\|^2} \leq \bar{\mu} \quad \text{for all } k \text{ with } x^{k+1} \neq x^{\ell_k}.$$

As a result, $\nu^k \leq \bar{\mu}$ for all k. Note that the prox-parameter is only increased after a null step such that $\nu^k \geq \mu^k - 2\kappa$. In this case, the rule employed in Step 4 of the algorithm sets $\mu^{k+1} = \nu^k + 1$, which gives $\mu^{k+1} = \nu^k + 1 \leq \bar{\mu} + 1$. Since the algorithm keeps the prox-parameter unchanged after a serious step or null step such that $\nu^k < \mu^k - 2\kappa$, we conclude that $\mu_{\max} := \sup_{k\in\mathbb{N}} \mu^k \leq \bar{\mu} + 1$ is finite. Finally, note that the prox-parameter is sharply increased after a null step such that $\nu^k \geq \mu^k - 2\kappa$: $\mu^{k+1} = \nu^k + 1 \geq \mu^k - 2\kappa + 1 > \mu^k$ because $\kappa \in (0, \frac{1}{2})$. As a result, if the algorithm produces an infinite sequence of null steps after a last serious step, then the inequality $\nu^k < \mu^k - 2\kappa$ will be satisfied for all k large enough and the prox-parameter will become constant (otherwise μ^k would increase indefinitely, which contradicts that $\mu_{\max}$ is finite). □