A Proximal Approach for Nonsmooth Composite-Constrained Optimization

Wim van Ackooij[1], Yi Chu[2*], Welington de Oliveira[2], Pedro Pérez-Aros[3]

[1]OSIRIS, EDF Lab Paris-Saclay, 7 Boulevard Gaspard Monge, Palaiseau, 91120, France.

[2]CMA – Centre de Mathématiques Appliquées, MINES Paris – PSL, Sophia Antipolis, France.

[3]Centro de Modelamiento Matemático, Universidad de Chile, Santiago, Chile.

*Corresponding author(s). E-mail(s): yi.chu@minesparis.psl.eu;

Abstract

We propose a proximal-type algorithm for nonsmooth and nonconvex optimization problems with composite constraints. The constraint is defined by the composition of a locally upper-$\boldsymbol{C^2}$ outer function with a locally Lipschitz continuous inner mapping. The method is based on an improvement function that balances objective decrease and constraint satisfaction, and on a surrogate model obtained by linearizing the outer function with respect to the composite argument. Each trial point is obtained as a stationary point of a regularized master problem, which may be nonlinear and nonconvex; nevertheless, we assume such a point can be computed efficiently using an off-the-shelf solver. The resulting scheme combines stability-center updates with a sufficient-decrease criterion, while preserving the nonlinear structure of the inner mapping. We prove that every accumulation point of the serious iterates is critical for the original problem and, under a suitable constraint qualification, satisfies a generalized KKT condition. Numerical experiments on chance-constrained optimization problems demonstrate the practical performance and robustness of the proposed method.

Keywords: Nonsmooth nonconvex optimization, Composite optimization, Proximal methods, Improvement functions

1 Introduction

Optimization problems featuring composite functions are prevalent in numerous fields, including energy management, where the operation of cascading reservoirs under uncertainty naturally leads to such structures. These problems pose significant computational challenges, especially when the composite terms appear in the constraints. One reason is that a composition may lose structural properties enjoyed by its individual components: even if the outer function and the inner mapping are convex, their composition need not be convex, and nonsmoothness in either component typically propagates to the composite function. Moreover, approximating each component separately does not necessarily produce a tractable or reliable approximation of the full composition. This motivates algorithmic approaches that exploit the composite structure directly rather than treating the resulting constraint as a black-box nonsmooth function.

Motivated by these applications, we consider the following nonsmooth, nonconvex optimization problem:

$$\min_{x\in X} f(x) \quad \text{s.t.} \quad g_0(G(x)) \leq 0. \tag{P}$$

Here, $f\colon \mathbb{R}^n \to \mathbb{R}$ is a tractable objective function, for instance smooth or convex, $X \subset \mathbb{R}^n$ is a nonempty closed set, and the constraint is given by the composition of a locally Lipschitz continuous inner mapping $G\colon \mathbb{R}^n \to \mathbb{R}^m$ and a locally upper-C^2 outer function $g_0\colon \mathbb{R}^m \to \mathbb{R}$.

This structure arises naturally in stochastic programming. In particular, as detailed in [1] and in Section 4 below, given $p \in (0,1)$ and a random vector ξ with probability measure $\mathbb{P}$, one obtains a standard chance-constrained optimization problem with right-hand-side uncertainty by choosing the outer function as a probability functional; see Equation (15) below.

Another source of such constraints is large-scale nonlinear optimization with an embedded lower-level problem. Let $Y \subset \mathbb{R}^N$ be compact and let $\varphi\colon \mathbb{R}^n \times \mathbb{R}^N \to \mathbb{R}$ be twice continuously differentiable, with φ convex in its second argument. Consider

$$\min_{x\in X,\, y\in Y} f(x) \quad \text{s.t.} \quad \varphi(G(x), y) \leq c,$$

where $c \in \mathbb{R}$ is fixed. Problems of this form appear, for instance, in recourse-constrained stochastic programs [2]. Depending on the dimensions of x and y, such problems may be difficult for off-the-shelf nonlinear optimization solvers, making decomposition attractive. By defining

$$g_0(G(x)) := \min_{y\in Y} \varphi(G(x), y) - c,$$

we recover problem (P). Indeed, since Y is compact and φ is twice continuously differentiable, the resulting value function g_0 is upper-C^2. If Y is convex, respectively mixed-integer, then the problem defining g_0 is convex, respectively mixed-integer, for fixed x.

Composite structures have been extensively studied in optimization, especially when they appear in the objective function. In the classical additive sense, composite optimization refers to objectives consisting of a smooth term and a nonsmooth structured term. This setting underlies proximal-gradient and accelerated gradient methods, as well as proximal and bundle-type methods for convex, weakly convex, stochastic, and hybrid composite objectives [3–5]. A more explicit outer–inner composite structure arises when a nonsmooth outer function is applied to a smooth inner mapping, possibly together with an additional smooth objective term. Such structures have been studied through Gauss–Newton, proximal, and stochastic model-based methods [6–8].

By comparison, explicit composite structures in constraints have received less systematic algorithmic treatment. Nevertheless, such structures arise naturally in probabilistic constraints, where feasibility is expressed through the value of a probability function evaluated at a decision-dependent threshold or set. Joint chance constraints with random right-hand sides provide a particularly relevant example: their convexity and structural properties have been studied under independence assumptions [9] and, more generally, in the presence of dependence through copula representations [10]. In hydro-reservoir management, such constraints arise when uncertain inflows must satisfy reservoir-related safety requirements with a prescribed joint probability [11, 12]. From the numerical point of view, chance-constrained programs via composition with copulæ have been investigated in [13, 14], and via difference-of-convex programming in [15–17]. Related work also includes sensitivity analysis for parametric optimization problems whose constraints admit special composite representations [18], as well as application models involving composite quality constraints in pooling problems [19]. The idea of linearizing one component of a composite expression has also been explored recently in [1], but in a different setting: there, the composite structure appears in the objective function and both the outer function and the inner mapping are assumed to be locally of class ($C^{1,1}$). In contrast, the present work focuses on an explicit composite constraint ($g_0(G(x)) \leq 0$), exploits the two-level structure of the constraint directly, and imposes regularity assumptions separately on the outer function and the inner mapping.

Our work is also closely related to the proximal-type framework of [20], which uses improvement functions, stability-center ideas, and proximal regularization for nonsmooth and nonconvex constrained problems. The latter setting treats the constraint as a general scalar function and constructs a model of the whole improvement function, possibly as the pointwise minimum of finitely many convex models; in some situation, this leads to a strong notion of model criticality based on global minimization of the active models. In contrast, we exploit an explicit composite constraint structure and develop a dedicated approximation model by linearizing the outer function with respect to the inner mapping. This preserves the nonlinear structure, and hence the curvature, of the inner mapping while keeping the resulting master problem amenable to computation, at least to stationarity. Our assumptions are also milder and tailored to composite constraints: regularity is imposed separately on the outer and inner components rather than on a model of the full constraint, allowing the proposed framework to apply to a broader class of composite-constrained problems. Furthermore, unlike [20], we do not assume that X is convex nor that the master program is solved globally.

In order to tackle problem (P), we base our approach on a novel proximal algorithm. The latter iteratively solves a sequence of surrogate problems. The key idea here is to linearize the outer function g_0 within the constraint. This strategy is viable under the crucial assumption that the resulting master problem, regularized with a proximal term, is efficiently solvable. This holds true in many important cases, such as when f and G are smooth (leading to a smooth NLP), or when the problem has a DC (Difference-of-Convex) structure. In that case, after linearization of g_0, the problem has a DC surrogate problem with a DC constraint. This class of nonsmooth and nonconvex optimization problems can be addressed using dedicated solution techniques; see, for instance, [21, Ch. 14].

The main contributions of this paper are:

- A novel algorithmic framework: we introduce a proximal point algorithm that employs an improvement function to handle the composite constraint. By linearizing the outer function g_0, our method transforms the difficult problem (P) into a sequence of tractable surrogate problems. In fact our assumption regarding the solvability of this surrogate problem is a practical one and as a result the class of amenable problems is likely to expand with time as more efficient solvers become available. Our proximal method shares commonalities with bundle methods such as the idea of a stability center and sufficient decrease condition. The produced stability centers can be thought of as the best guess the algorithm has produced of the optimal solution.
- Convergence guarantees: we prove that under mild and standard assumptions, any limit point of the sequence generated by our algorithm is a critical point for problem (P). Furthermore, we establish that under a suitable constraint qualification, these critical points are in fact generalized Karush-Kuhn-Tucker (KKT) points.
- Numerical scalability: The framework is versatile as it accommodates various structures (smooth, convex, DC). It is also readily extended in case f is also of composite structure. Then it would suffice to linearize f_0 in $f(x) = f_0(F(x))$. Its practical effectiveness is confirmed through numerical experiments on challenging chance-constrained problem. The investigated instances come from data of real energy systems.

This paper is organized as follows. Section 2 provides background material on improvement functions and various notions used throughout the work. The algorithm is presented in Section 3 and its convergence analyzed. Section 4 presents the numerical test bed showing the performance and scalability of the suggested algorithm on instances from practice. Finally, Section 5 presents concluding remarks.

Notation and Terminology. We work in the Euclidean space $\mathbb{R}^n$ with inner product $\langle x, y\rangle = x^\top y$ for $x, y \in \mathbb{R}^n$ and norm $\|x\| = \sqrt{\langle x, x\rangle}$. For a real number a, we denote by $[a]_+$ the value $\max\{a, 0\}$. The Fréchet normal cone to a closed set $X \subset \mathbb{R}^n$ at $x \in X$ is denoted by $\widehat{N}_X(x)$ and consists of those $x^* \in \mathbb{R}^n$ satisfying $\langle x^*, x' - x\rangle \leq o(\|x' - x\|)$ for all $x' \in X$. The Mordukhovich (or limiting) normal cone to X at $x \in X$ is $N_X(x) = \{x^* \in \mathbb{R}^n \,|\, \exists v^\nu \to x^*, x^\nu \in X \to x \text{ with } v^\nu \in \widehat{N}_X(x^\nu)\}$. These cones are empty when $x \notin X$. Furthermore, they coincide with the convex normal cone $\{y : \langle y, z - x\rangle \leq 0 \;\text{ for all }\; z \in X\}$ when X is furthermore convex. The convex hull of a set X is $\operatorname{co} X$, its closure is $\operatorname{cl} X$, and the indicator function of X is defined as $\mathbf{i}_X(x) = 0$ if $x \in X$ and $\mathbf{i}_X(x) = +\infty$ otherwise. The domain of a function $\varphi\colon \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is represented by $\mathcal{D}\mathrm{om}(\varphi) = \{x \in \mathbb{R}^n : \varphi(x) < +\infty\}$ and its epigraph is denoted by $\operatorname{epi} \varphi = \{(x, \alpha) \in \mathbb{R}^{n+1} \,|\, \varphi(x) \leq \alpha\}$. Function φ is lower semicontinuous (lsc) if $\operatorname{epi} \varphi$ is a closed subset of $\mathbb{R}^n \times \mathbb{R}$ and it is convex if $\operatorname{epi} \varphi$ is convex. It is proper if $\operatorname{epi} \varphi$ is nonempty and $\varphi(x) > -\infty$ for all $x \in \mathbb{R}^n$. Function φ is said to be *locally Lipschitz continuous* on an a neighbourhood of X if of each $x' \in X$ if there exist a neighbourhood $V_{x'}$ of x' and constant $L_{x'} \geq 0$, such that $|\varphi(x) - \varphi(y)| \leq L_{x'} \|x - y\|$ for all $x, y \in V_{x'}$. The function φ is said to be *Lipschitz continuous* on an a neighbourhood V of X if $L_{x'} = L$ can be taken independent of $x' \in X$, and $V_{x'}$ in the above inequality is replaced with V. (These definitions extend straightforwardly for a mapping $G : \mathbb{R}^n \to \mathbb{R}^m$.) If φ is locally Lipschitz at x, its Clarke subdifferential is defined by $\partial^{\mathrm{C}}\varphi(x) := \operatorname{co}\{\lim_{\iota\to\infty} \nabla\varphi(x_\iota), x_\iota \to x, \varphi \text{ differentiable at } x_\iota\}$. The mapping $\partial^{\mathrm{C}}\varphi$ is locally bounded in the interior of $\mathcal{D}\mathrm{om}(\varphi)$. As a result, the image $\partial^{\mathrm{C}}\varphi(\mathcal{O})$ of every bounded subset $\mathcal{O}$ of the interior of $\mathcal{D}\mathrm{om}(\varphi)$ is bounded. Finally, we say that $\bar{x} \in X$ is a stationary point of the problem $\min_{x\in X} \varphi(x)$ if $0 \in \partial^{\mathrm{C}} f(\bar{x}) + N_X(\bar{x})$.

2 Main Definitions, Assumptions, and Criticality Conditions

Let $U \subset \mathbb{R}^n$ be an open set. A function $\varphi : U \to \mathbb{R}$ is said to be *upper-*C^2 on U if, for every point $\bar{x} \in U$, there exist a neighbourhood V of $\bar{x}$, a compact index set C (endowed with some topology), and a family of functions $\{\varphi_c\}_{c\in C}$ of class C^2 on V such that

$$\varphi(x) = \min_{c\in C} \varphi_c(x) \quad \text{for all } x \in V,$$

and such that the mappings φ_c together with all their first- and second-order partial derivatives vary continuously with respect to $(x, c) \in V \times C$. We say that φ is *upper-*C^2 *around* $\bar{x}$ if this property holds on some neighbourhood of $\bar{x}$.

The next result shows that this notion may be characterized by a nonsmooth analogue of the classical descent lemma. A related statement appears in [22, Prop. 3.2].

Proposition 1 *Let U be an open subset of $\mathbb{R}^n$, and suppose that $\varphi : \mathbb{R}^n \to \mathbb{R}$ is locally Lipschitz on U. Then the following conditions are equivalent:*

i) φ *is upper-*C^2 *on* U*;*

ii) *for every $\bar{x} \in U$, there exist $K_{\bar{x}} \geq 0$ and a neighbourhood V of $\bar{x}$ such that, for each $x \in V$ and each $w \in \partial^{\mathrm{C}}\varphi(x)$,*

$$\varphi(y) \leq \varphi(x) + \langle w, y - x\rangle + \frac{K_{\bar{x}}}{2}\|y - x\|^2, \qquad \text{for all } y \in V. \tag{1}$$

Assumption 1 *Throughout this work, we make the following assumptions:*

A.1 $X \neq \emptyset$ *is closed;*

A.2 $f : \mathbb{R}^n \to \mathbb{R}$ *is a locally Lipschitz continuous function and there are constants $a > 0$, $b \in \mathbb{R}$ such that $f(x) \geq -\frac{a}{2}\|x\|^2 + b$ for all $x \in X$;*

A.3 $G : \mathbb{R}^n \to \mathbb{R}^m$ *is a locally Lipschitz continuous mapping on a neighbourhood of X. We denote the i^{th}-component of G by $g_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \ldots, m$;*

A.4 $g_0 : \mathbb{R}^m \to \mathbb{R}$ *is upper-*C^2 *on a neighbourhood of* $G(X)$*, the image of* G *over* X*;*

A.5 The set X*, function* f*, and mapping* G *have amenable structures for numerical optimization;*

A.6 The optimal value of problem (P) *is finite, that is,*

$$\inf \{f(x) : x \in X \text{ and } g_0(G(x)) \le 0\} \in \mathbb{R}.$$

We emphasize that these assumptions are mild. Item A.2 only requires that f be locally Lipschitz continuous and prox-regular, which, together with item A.1, ensures that the master program (MP) introduced below is well defined. Item A.5 is a practical assumption that guarantees that the master program can be implemented and effectively "solved" by an available optimization solver or method.

The following original result shows that conditions A.3 and A.4 provide a generalized descent condition.

Lemma 2 *Let* $\hat{x} \in X$ *be given and let items A.3 and A.4 of Assumption 1 hold true. Then, there exist a neighbourhood* W *of* $\hat{x}$ *and a constant* $\ell_{\hat{x}} > 0$ *such that, for all* $x, y \in X \cap W$*, and all* $v \in \partial^{\mathrm{C}} g_0(z)$ *with* $z = G(x)$ *we have that*

$$g_0(G(y)) \;\le\; g_0(G(x)) + \langle v, G(y) - G(x)\rangle + \frac{\ell_{\hat{x}}}{2}\|y - x\|^2 .$$

In particular, if G *is Lipschitz on* X *and* g_0 *is* K*-weakly concave on* $G(X)$*, then we may take* $\ell_{\hat{x}} = \ell$ *independently of* $\hat{x} \in X$*, and the above inequality holds for all* $x, y \in X$*.*

Proof Since g_0 is upper-C^2, it follows from Proposition 1 that there exist a neighbourhood V of $G(\hat{x})$ and a constant $K_{\hat{x}} > 0$ such that, for all $z, w \in V$ and each $v \in \partial^{\mathrm{C}} g_0(z)$,

$$g_0(w) \le g_0(z) + \langle v, w - z\rangle + \frac{K_{\hat{x}}}{2}\|w - z\|^2 .$$

Now, since G is continuous at $\hat{x}$, there exists a neighbourhood W of $\hat{x}$ such that $G(W) \subset V$. Shrinking W if necessary, we may also assume (thanks to A.3) that G is $L_{\hat{x}}$-Lipschitz continuous on W for some $L_{\hat{x}} > 0$. Therefore, for all $x, y \in X \cap W$, taking $z = G(x)$ and $w = G(y)$ in the previous inequality yields

$$\begin{aligned} g_0(G(y)) &\le g_0(G(x)) + \langle v, G(y) - G(x)\rangle + \frac{K_{\hat{x}}}{2}\|G(y) - G(x)\|^2 \\ &\le g_0(G(x)) + \langle v, G(y) - G(x)\rangle + \frac{K_{\hat{x}}}{2} L_{\hat{x}}^2 \|y - x\|^2 . \end{aligned}$$

The result follows by setting $\ell_{\hat{x}} := K_{\hat{x}} L_{\hat{x}}^2$.

In particular, if G is L-Lipschitz on X and g_0 is K-weakly concave on $G(X)$, we can take $L_{\hat{x}} = L$ and $K_{\hat{x}} = K$ independently of $\hat{x} \in X$. □

2.1 Improvement function and criticality

In order to tackle the constraints of the problem, we will make use of the so-called improvement function that will weigh satisfaction of constraints and progress in the objective function:

$$H(x; \hat{x}) := \max\{\, f(x) - \tau(\hat{x}),\; g_0(G(x)) \,\}, \qquad \tau(\hat{x}) := f(\hat{x}) + \rho[g_0(G(\hat{x}))]_+, \quad \rho \ge 0.$$

The use of improvement functions for nonsmooth optimization is a well established alternative to the use of penalty functions, see e.g., [21, Chps. 12 and 14]. One can readily show that if $\hat{x}$ is a local solution of problem (P), then $\hat{x}$ also locally solves

$$\min_{x \in X} H(x; \hat{x}).$$

Computing a guaranteed global solution of the above problem is out of reach due to the lack of convexity and smoothness. For this reason, we focus instead on the computation of critical points, that is, points satisfying weaker necessary optimality conditions. A key ingredient of our notion of criticality is the following upper estimate of the subdifferential of the composite function $g_0 \circ G(x) := g_0(G(x))$:

$$\mathcal{G}(x) := \mathrm{cl}\,\mathrm{co}\left\{ \sum_{j=1}^{m} v_j u_j : u_j \in \partial^{\mathrm{C}} g_j(x),\ j = 1, \ldots, m,\ v \in \partial^{\mathrm{C}} g_0(G(x)) \right\}. \tag{2}$$

It follows from the chain rule and calculus rule for the Clarke subdifferential that (e.g., [21, Prop. 3.9])

$$\partial^{\mathrm{C}}(g_0 \circ G)(x) \subset \mathcal{G}(x) \quad \forall x \in \mathbb{R}^n.$$

The inclusion becomes equality, for instance, when g_0 is smooth at $G(x)$, and G is smooth at x.

Definition 1 (Criticality) A point $\hat{x} \in X$ is said to be critical for problem (P) if

$$0 \in \lambda \partial^{\mathrm{C}} f(\hat{x}) + (1-\lambda)\mathcal{G}(\hat{x}) + N_X(\hat{x}) \quad \text{with} \quad \lambda \in \begin{cases} \{1\} & \text{if } \ g_0(G(\hat{x})) < 0 \\ [0,1] & \text{if } \ g_0(G(\hat{x})) = 0 \\ \{0\} & \text{if } \ g_0(G(\hat{x})) > 0. \end{cases}$$

We stress that criticality is a weaker condition than stationarity for the problem of minimizing the improvement function.

Lemma 3 *If $\hat{x} \in X$ is stationary for $\min_{x \in X} H(x; \hat{x})$ then $\hat{x}$ is critical for problem (P).*

Proof Since $\hat{x} \in X$ is stationary for $\min_{x \in X} H(x; \hat{x})$, then $0 \in \partial^{\mathrm{C}} H(\cdot; \hat{x})(\hat{x}) + N_X(\hat{x})$. The calculus rule for the Clarke subdifferential of the maximum function gives (see, e.g., [21, Coro. 3.2])

$$\partial^{\mathrm{C}} H(\cdot; \hat{x})(x) \subset \begin{cases} \partial^{\mathrm{C}} f(x), & \text{if } f(x) - \tau(\hat{x}) > g_0(G(x)), \\ \mathrm{co}\big(\partial^{\mathrm{C}} f(x),\ \partial^{\mathrm{C}}(g_0 \circ G)(x)\big), & \text{if } f(x) - \tau(\hat{x}) = g_0(G(x)), \\ \partial^{\mathrm{C}}(g_0 \circ G)(x), & \text{if } f(x) - \tau(\hat{x}) < g_0(G(x)). \end{cases}$$

Note that $g_0(G(\hat{x})) - [f(\hat{x}) - \tau(\hat{x})] = g_0(G(\hat{x})) + \rho[g_0(G(\hat{x}))]_+$ which equals $g_0(G(\hat{x}))$ if $g_0(G(\hat{x})) \leq 0$. As a result, the above inclusion written with $x = \hat{x}$ boils down to

$$\partial^{\mathrm{C}} H(\cdot; \hat{x})(\hat{x}) \subset \begin{cases} \partial^{\mathrm{C}} f(\hat{x}), & \text{if } g_0(G(\hat{x})) < 0, \\ \mathrm{co}\big(\partial^{\mathrm{C}} f(\hat{x}),\ \partial^{\mathrm{C}}(g_0 \circ G)(\hat{x})\big), & \text{if } g_0(G(\hat{x})) = 0, \\ \partial^{\mathrm{C}}(g_0 \circ G)(\hat{x}), & \text{if } g_0(G(\hat{x})) > 0. \end{cases}$$

As $\partial^{\mathrm{C}}(g_0 \circ G)(\hat{x}) \subset \mathcal{G}(\hat{x})$ we conclude that $\hat{x}$ satisfies Definition 1. □

The above result explains why our definition of criticality encompass potentially infeasible points. Indeed, if $\hat{x} \in X$ is infeasible for problem (P) (i.e., $g_0(G(\hat{x})) > 0$), then $\hat{x}$ nonetheless satisfies the necessary optimality condition $0 \in \mathcal{G}(\hat{x}) + N_X(\hat{x})$ associated with the infeasibility minimization problem $\min_{x \in X} g_0(G(x))$. When $\hat{x}$ is feasible and a constraint qualification holds, our criticality definition boils down to the following generalized KKT condition.

Definition 2 (Generalized KKT condition.) A point $\hat{x} \in X$ is said to satisfy the generalized KKT system of problem (P) if there exists $\hat{\lambda} \geq 0$ such that

$$g_0(G(\hat{x})) \leq 0, \quad \hat{\lambda}\, g_0(G(\hat{x})) = 0, \quad \text{and} \quad 0 \in \partial^{\mathrm{C}} f(\hat{x}) + \hat{\lambda}\, \mathcal{G}(\hat{x}) + N_X(\hat{x}).$$

Lemma 4 *Suppose that $\hat{x} \in X$ is critical for problem (P). Furthermore, assume that either $g_0(G(\hat{x})) < 0$ or $g_0(G(\hat{x})) = 0$ and the following constraint qualification (CQ) holds: $0 \notin \mathcal{G}(\hat{x}) + N_X(\hat{x})$. Then $\hat{x}$ satisfies the generalized KKT condition of Definition 2.*

Proof By definition of criticality, there exists $\lambda \in [0,1]$ such that $0 \in \lambda\partial^{\mathrm{C}} f(\hat{x}) + (1-\lambda)\mathcal{G}(\hat{x}) + N_X(\hat{x})$. If $g_0(G(\hat{x})) < 0$, then $\lambda = 1$ and the desired equations of Definition 2 hold with $\hat{\lambda} = 0$. Suppose now that $g_0(G(\hat{x})) = 0$. The assumed constraint qualification condition yields $\lambda > 0$. Dividing the above inclusion by λ and defining $\hat{\lambda} = (1-\lambda)/\lambda \geq 0$ we get $0 \in \partial^{\mathrm{C}} f(\hat{x}) + \hat{\lambda}\,\mathcal{G}(\hat{x}) + N_X(\hat{x})$ as desired. □

Having established the link between criticality and generalized KKT conditions, the next section proposes an algorithm for computing a critical point of problem (P). Central to our development is the improvement function. However, in the general setting considered here, this function can be somewhat unwieldy. For this reason, we replace it with a suitable model $M_\nu(\cdot)$. The latter is constructed by linearizing the outer function g_0 at a $G(\hat{x}^\nu)$, with $\hat{x}^\nu \in X$ the ν^{th}-*stability center* (a previous iterate generated by our approach):

$$M_\nu(x) := g_0(G(\hat{x}^\nu)) + \langle v^\nu, G(x) - G(\hat{x}^\nu)\rangle, \quad \text{with arbitrary} \quad v^\nu \in \partial^{\mathrm{C}} g_0(G(\hat{x}^\nu)). \tag{3}$$

Remark 1 (Global upper approximation.) Under the assumptions of Lemma 2, let W denote the neighbourhood of $\hat{x}$ given therein. Then, for every $\hat{x}^\nu \in X \cap W$, we have

$$g_0(G(y)) \leq M_\nu(y) + \frac{\ell_{\hat{x}^\nu}}{2}\|y - \hat{x}^\nu\|^2 \quad \text{for all } y \in X \cap W. \tag{4}$$

If G is Lipschitz on X and g_0 is K-weakly concave on $G(X)$, then we can take $\ell_{\hat{x}^\nu} = \ell$ independently of $\hat{x}^\nu \in X$, and $W = X$.

This remark asserts that M_ν yields a global upper approximation of the composite constraint function $g_0 \circ G$ on X, up to a quadratic correction term. Through this linearization we define the surrogate model for the improvement function

$$H_\nu^M(x; \hat{x}^\nu) := \max\{\, f(x) - \tau(\hat{x}^\nu),\ M_\nu(x)\,\}. \tag{5}$$

Thanks to assumptions A.2 and A.5, working with this model is considerably simpler than working directly with the improvement function. Importantly, $H_\nu^M(\cdot; \hat{x}^\nu)$ retains the necessary local information to guarantee criticality, as asserted by Lemma 5 in the next section.

3 A Proximal Method with Improvement Function

We now present our method for computing critical points. Our approach generates a sequence of trial points (iterates) $(x^k)_{k\in\mathbb{N}}$. Adopting the terminology of bundle methods (see, e.g., [21, Chp. 11]), we extract from this sequence a subsequence $(\hat{x}^\nu)_{\nu\in\mathbb{N}}$ of *serious iterates*, namely points at which the improvement function exhibits a substantial decrease. More precisely, starting from $\hat{x}^0 = x^0 \in X$, at every iteration $k \geq 0$ and for $\nu \geq 0$, our method defines the next trial point x^{k+1} as a stationary point of

$$\min_{x\in X}\ H_\nu^M(x; \hat{x}^\nu) + \frac{\mu_k}{2}\|x - \hat{x}^\nu\|^2, \tag{MP}$$

where $\mu_k > 0$ is a given prox-parameter. The trial point x^{k+1} is declared a *serious iterate* whenever the improvement function decreases sufficiently. In practice, this is enforced through the descent test

$$H(x^{k+1}; \hat{x}^\nu) \leq H(\hat{x}^\nu; \hat{x}^\nu) - \frac{\gamma}{2}\left\|x^{k+1} - \hat{x}^\nu\right\|^2,$$

for some parameter $\gamma \in (0,1)$. If this condition is satisfied, we set $\hat{x}^{\nu+1} = x^{k+1}$ and increment ν by 1. Otherwise, both $\hat{x}^{\nu}$ and ν remain unchanged but the prox-parameter μ_k is increased. In either case, k is incremented by 1.

Due to the presence of the quadratic term in the master problem (MP), it is natural to refer to the most recent serious iterate $\hat{x}^{\nu}$ as the *stability center* of the model. Indeed, the generation of new trial points is anchored in a neighbourhood of $\hat{x}^{\nu}$, which represents the best candidate currently available for solving (P). We are now in measure to state the Algorithm 1 for computing a critical point of problem (P).

Algorithm 1 Proximal-like method

1: Input: $x^0 \in X, \mu_{\max} > \mu_0 > 0, \gamma \in (0,1), \beta > 1, \rho \geq 0, \mathtt{Tol} \geq 0$
2: Output: candidate solution $\hat{x}^{\nu}$, $f(\hat{x}^{\nu})$ and $g_0(G(\hat{x}^{\nu}))$
3: Initialize $\hat{x}^0 = x^0$, $k = \nu = 0$
4: **for** $k = 0, 1, \ldots$ **do**
5: Define $H_{\nu}^{M}(\cdot;\hat{x}^{\nu})$ by (5)
6: Compute a stationary point x^{k+1} of (MP) satisfying

$$H_{\nu}^{M}(x^{k+1};\hat{x}^{\nu}) + \frac{\mu_k}{2}\left\|x^{k+1} - \hat{x}^{\nu}\right\|^2 \leq H_{\nu}^{M}(\hat{x}^{\nu};\hat{x}^{\nu}) \tag{6}$$

7: **if** $\left\|x^{k+1} - \hat{x}^{\nu}\right\| \leq \mathtt{Tol}$ **then**
8: **return** $\hat{x}^{\nu}$, $f(\hat{x}^{\nu})$, $g_0(G(\hat{x}^{\nu}))$
9: **end if**
10: **if** $H(x^{k+1};\hat{x}^{\nu}) \leq H(\hat{x}^{\nu};\hat{x}^{\nu}) - \frac{\gamma}{2}\left\|x^{k+1} - \hat{x}^{\nu}\right\|^2$ **then** ▷ Serious Step
11: $\hat{x}^{\nu+1} \leftarrow x^{k+1}$, $\nu \leftarrow \nu + 1$, and choose $\mu_{k+1} \in (0, \mu_{\max})$
12: **else** ▷ Null Step
13: $\mu_{k+1} \leftarrow \beta\,\mu_k$, and go back to step 6
14: **end if**
15: **end for**

Remark 2 (Comments on the algorithm.) Some comments are in order.

(i) The master program (MP) is clearly feasible since $\hat{x}^{\nu} \in X$. Moreover, the optimal value of problem (MP) is finite for large enough μ_k by Assumption A.2. Indeed, it follows from item A.2 that there exists $a > 0$ and $b \in \mathbb{R}$ such that, for all $x \in X$,

$$\begin{aligned}
-\frac{a}{2}\|x\|^2 + b - \tau(\hat{x}^{\nu}) + \frac{\mu_k}{2}\|x - \hat{x}^{\nu}\|^2 &\leq f(x) - \tau(\hat{x}^{\nu}) + \frac{\mu_k}{2}\|x - \hat{x}^{\nu}\|^2 \\
&\leq \max\left\{f(x) - \tau(\hat{x}^{\nu}),\, M_{\nu}(x)\right\} + \frac{\mu_k}{2}\|x - \hat{x}^{\nu}\|^2 \\
&= H_{\nu}^{M}(x;\hat{x}^{\nu}) + \frac{\mu_k}{2}\|x - \hat{x}^{\nu}\|^2.
\end{aligned}$$

If $\mu_k > a > 0$, then the left-hand side above is a coercive function on X. As $H_{\nu}^{M}(\cdot;\hat{x}^{\nu})$ is a continuous function, the master program (MP) has a solution (and thus a stationary point). Observe that x^{k+1} is also well defined under the assumption that X is moreover bounded (in which case we can take $a = 0$).

(ii) It is worth noting that condition (6) is not restrictive. Indeed, since $\hat{x}^{\nu} \in X$ is available, we can initialize the solver with $\hat{x}^{\nu}$ when computing the next iterate. Any mathematically sound solver implementing a descent algorithm will return a stationary point that is at least as good as the initial point.

(iii) From an implementation point of view, we can write (MP) in an epigraphical form:

$$\begin{cases}
\min\limits_{x,r} \; r + \dfrac{\mu_k}{2}\|x - \hat{x}^{\nu}\|^2 & \\
\text{s.t.} \;\; f(x) - r & \leq \tau(\hat{x}^{k}) \\
\qquad \sum_{j=1}^{m} v_j^k g_j(x) - r & \leq \sum_{j=1}^{m} v_j^k g_j(\hat{x}^{\nu}) - g_0(G(\hat{x}^{\nu})) \\
\qquad x \in X,\; r \in \mathbb{R}. &
\end{cases}$$

Observe that if f and g_j, $j = 1, \ldots, m$, are smooth functions, then the master problem is a standard nonlinear optimization problem. When X is convex, f and g_j, $j = 1, \ldots, m$, are DC functions, then this problem can be handled by specialized DC methods [21, Chp. 4 and 14].

(iv) After a null step, only the proximal parameter μ_k is updated to $\mu_{k+1} > \mu_k$. No call to the oracle associated with g_0 is required, which makes the proposed algorithm particularly attractive for problems involving a hard-to-evaluate function g_0. This situation arises, for instance, when $g_0(z)$ is defined as the optimal value of a parametric optimization problem, and described in the Introduction.

3.1 Convergence Analysis

Throughout this section we consider Algorithm 1 with `Tol` $= 0$ applied to problem (P) satisfying Assumption 1. Furthermore, we assume that $\mu_k > 2a \geq 0$, with a given in Assumption A.2 (recall that $a = 0$ if X is bounded). As commented above, having the prox-parameter greater than $2a$ ensures that the master program is well defined.

We start by introducing an additional notation: $k(\nu)$ is the iteration index at which the stability center $\hat{x}^{\nu+1}$ is produced. In other words, $\hat{x}^{\nu+1}$ is a stationary point of $\min_{x\in X} H^M_\nu(x;\hat{x}^\nu) + \frac{\mu_{k(\nu)}}{2}\|x-\hat{x}^\nu\|^2$, i.e.,

$$0 \in \partial^{\mathrm{C}} H^M_\nu(\cdot;\hat{x}^\nu)(\hat{x}^{\nu+1}) + \mu_{k(\nu)}(\hat{x}^{\nu+1}-\hat{x}^\nu) + N_X(\hat{x}^{\nu+1}). \tag{7}$$

The following lemma is central to the asymptotic analysis of Algorithm 1.

Lemma 5 *Let $\mathcal{N} \subset \mathbb{N}$ be an infinite index set and suppose that there exists $\hat{x} \in X$ such that $\lim_{\mathcal{N}\ni\nu\to\infty}\hat{x}^{\nu+1} = \lim_{\mathcal{N}\ni\nu\to\infty}\hat{x}^\nu = \hat{x}$. If $(\mu_{k(\nu)})_{\nu\in\mathcal{N}}$ admits a bounded subsequence, then $\hat{x}$ is a critical point of problem (P).*

If, on the other hand, $x^{k+1} = \hat{x}^\nu$ for some $k \in \mathbb{N}$, then $\hat{x}^\nu$ is critical for problem (P).

Proof Since $\hat{x}^{\nu+1}$ is a stationary point of $\min_{x\in X}\ H^M_\nu(x;\hat{x}^\nu) + \frac{\mu_{k(\nu)}}{2}\|x-\hat{x}^\nu\|^2$, it satisfies (7). It follows from the definition of $H^M_\nu(\cdot;\hat{x}^\nu)$ that

$$\partial^{\mathrm{C}} H^M_\nu(\cdot;\hat{x}^\nu)(\hat{x}^{\nu+1}) \subset \begin{cases} \partial^{\mathrm{C}} f(\hat{x}^{\nu+1}), & \text{if } M_\nu(\hat{x}^{\nu+1}) < f(\hat{x}^{\nu+1}) - \tau(\hat{x}^\nu), \\ \mathrm{co}\big(\partial^{\mathrm{C}} f(\hat{x}^{\nu+1}),\ \sum_{j=1}^m v_j^\nu \partial^{\mathrm{C}} g_j(\hat{x}^{\nu+1})\big), & \text{if } M_\nu(\hat{x}^{\nu+1}) = f(\hat{x}^{\nu+1}) - \tau(\hat{x}^\nu), \\ \sum_{j=1}^m v_j^\nu \partial^{\mathrm{C}} g_j(\hat{x}^{\nu+1}), & \text{if } M_\nu(\hat{x}^{\nu+1}) > f(\hat{x}^{\nu+1}) - \tau(\hat{x}^\nu). \end{cases} \tag{8}$$

Recall that the Clarke subdifferential of a locally Lipschitzian function is convex, closed, and locally bounded. Since $\lim_{\mathcal{N}\ni\nu\to\infty}\hat{x}^{\nu+1} = \lim_{\mathcal{N}\ni\nu\to\infty}\hat{x}^\nu = \hat{x}$ by assumption, we have that

$$\lim_{\nu\to\infty} M_\nu(\hat{x}^{\nu+1}) = \lim_{\nu\to\infty}[g_0(G(\hat{x}^\nu)) + \langle v^\nu,\, G(\hat{x}^{\nu+1}) - G(\hat{x}^\nu)\rangle] = g_0(G(\hat{x}))$$

and $\lim_{\nu\to\infty}\tau(\hat{x}^\nu) = f(\hat{x}) + \rho[g_0(G(\hat{x}))]_+$. Let us distinguish three exclusive cases:

i) Case 1: $g_0(G(\hat{x})) < 0$. Continuity of $g_0 \circ G(\cdot)$ ensures the existence of $\varepsilon > 0$ such that $g_0(G(\hat{x}^{\nu+1})) < -\varepsilon < 0$ for all $\nu \in \mathcal{N}$ large enough. As $M_\nu(\hat{x}^{\nu+1})$ tends to $g_0(G(\hat{x}))$ along $\nu \in \mathcal{N}$, we conclude that $M_\nu(\hat{x}^{\nu+1}) \leq -\varepsilon$ for all $\nu \in \mathcal{N}$ large enough. As a result,

$$M_\nu(\hat{x}^{\nu+1}) - [f(\hat{x}^{\nu+1}) - \tau(\hat{x}^\nu)] = M_\nu(\hat{x}^{\nu+1}) - [f(\hat{x}^{\nu+1}) - f(\hat{x}^\nu)] \leq -\varepsilon - [f(\hat{x}^{\nu+1}) - f(\hat{x}^\nu)].$$

As $\lim_{\mathcal{N}\ni\nu\to\infty}[f(\hat{x}^{\nu+1}) - f(\hat{x}^\nu)] = 0$ by continuity of f, there exists $\nu' \in \mathcal{N}$ such that $-[f(\hat{x}^{\nu+1}) - f(\hat{x}^\nu)] < \varepsilon/2$ for all $\mathcal{N} \ni \nu > \nu'$. Thus,

$$M_\nu(\hat{x}^{\nu+1}) - [f(\hat{x}^{\nu+1}) - \tau(\hat{x}^\nu)] \leq -\varepsilon + \varepsilon/2 = -\varepsilon/2.$$

This shows that

$$M_\nu(\hat{x}^{\nu+1}) < f(\hat{x}^{\nu+1}) - \tau(\hat{x}^\nu)$$

and thus for all $\mathcal{N} \ni \nu > \nu'$,

$$\partial^{\mathrm{C}} H^M_\nu(\cdot;\hat{x}^\nu)(\hat{x}^{\nu+1}) \subset \partial^{\mathrm{C}} f(\hat{x}^{\nu+1}).$$

For such ν, the stationary condition in (7) then boils down to

$$\mu_{k(\nu)}(\hat{x}^{\nu} - \hat{x}^{\nu+1}) \in \partial^{\mathrm{C}} f(\hat{x}^{\nu+1}) + N_X(\hat{x}^{\nu+1}).$$

By assumption, there exists an infinite index set $\mathcal{N}' \subset \mathcal{N}$ such that $(\mu_{k(\nu)})_{\nu\in\mathcal{N}'}$ is bounded. By passing to the limit in the latter inclusion with $\nu \in \mathcal{N}'$ tending to infinity, and recalling that the Clarke subdifferential and the normal cone are outer semicontinuous, we conclude that

$$0 \in \partial^{\mathrm{C}} f(\hat{x}) + N_X(\hat{x}) \quad \text{if} \quad g_0(G(\hat{x})) < 0.$$

ii) Case 2: $g_0(G(\hat{x})) > 0$. An analogous analysis yields $\partial^{\mathrm{C}} H_\nu^M(\cdot;\hat{x}^\nu)(\hat{x}^{\nu+1}) \subset \sum_{j=1}^m v_j^\nu \partial^{\mathrm{C}} g_j(\hat{x}^{\nu+1})$ for all $\nu \in \mathcal{N}$ large enough. Thus, the stationarity condition gives $\mu_{k(\nu)}(\hat{x}^\nu - \hat{x}^{\nu+1}) \in \sum_{j=1}^m v_j^\nu \partial^{\mathrm{C}} g_j(\hat{x}^{\nu+1}) + N_X(\hat{x}^{\nu+1})$. As any cluster point $\hat{v}$ of the bounded sequence $(v^\nu)_{\nu\in\mathcal{N}'}$ belongs to $\partial^{\mathrm{C}} g_0(G(\hat{x})))$, we conclude that in the limit (after passing to a subsequence if necessary)

$$0 \in \sum_{j=1}^m \hat{v}_j \partial^{\mathrm{C}} g_j(\hat{x}) + N_X(\hat{x}) \subset \mathcal{G}(\hat{x}) + N_X(\hat{x}) \quad \text{if} \quad g_0(G(\hat{x})) > 0.$$

iii) Case 3: $g_0(G(\hat{x})) = 0$. In this setting, we can pick up $\lambda_\nu \in [0,1]$ such that $\partial^{\mathrm{C}} H_\nu^M(\cdot;\hat{x}^\nu)(\hat{x}^{\nu+1}) \subset \lambda_\nu \partial^{\mathrm{C}} f(\hat{x}^{\nu+1}) + (1-\lambda_\nu)\sum_{j=1}^m v_j^\nu \partial^{\mathrm{C}} g_j(\hat{x}^{\nu+1})$. Using the same reasoning above, by passing to the limit in the inclusion $\mu_{k(\nu)}(\hat{x}^\nu - \hat{x}^{\nu+1}) \in \lambda_\nu \partial^{\mathrm{C}} f(\hat{x}^{\nu+1}) + (1-\lambda_\nu)\sum_{j=1}^m v_j^\nu \partial^{\mathrm{C}} g_j(\hat{x}^{\nu+1}) + N_X(\hat{x}^{\nu+1})$ with ν tending to infinity along a subsequence of $\mathcal{N}'$ gives, for some $\lambda \in [0,1]$,

$$0 \in \lambda \partial^{\mathrm{C}} f(\hat{x}) + (1-\lambda)\mathcal{G}(\hat{x}) + N_X(\hat{x}) \quad \text{if} \quad g_0(G(\hat{x})) = 0.$$

These three cases show that $\hat{x} \in X$ is a critical point of (P).

If, on the other hand, $x^{k+1} = \hat{x}^\nu$ for some $k \in \mathbb{N}$, then the subdifferential inclusion in (8) reads with $\hat{x}^{\nu+1} = \hat{x}^\nu$ and $M_\nu(\hat{x}^{\nu+1}) = g_0(G(\hat{x}^\nu))$. As in this case $\sum_{j=1}^m v_j^\nu \partial^{\mathrm{C}} g_j(\hat{x}^{\nu+1}) = \sum_{j=1}^m v_j^\nu \partial^{\mathrm{C}} g_j(\hat{x}^\nu) \subset \mathcal{G}(\hat{x}^\nu)$, we can follow the steps of Lemma 3 to conclude $\hat{x}^\nu$ is critical. □

Given this result, we assume for the remainder of this section that Algorithm 1 loops indefinitely.

Proposition 6 (Null steps.) *Suppose that Assumption 1 holds. Then Algorithm 1 applied to problem (P) produces only finitely many consecutive null steps.*

If, instead of A.3 and A.4, we assume that G is Lipschitz on X and that g_0 is K-weakly concave on $G(X)$, and additionally require that the algorithm does not update the prox-parameter μ_k after serious steps, then only finitely many null steps are generated, and the sequence $(\mu_k)_{k\in\mathbb{N}}$ eventually becomes constant.

Proof We proceed with a proof by contradiction. Suppose that the algorithm produces a last stability center $\hat{x}^\nu$ at iteration $\hat{k}$ and only null steps are produced thereafter:

$$\nu \text{ is fixed and } H(\hat{x}^\nu;\hat{x}^\nu) < H(x^{k+1};\hat{x}^\nu) + \tfrac{\gamma}{2}\|x^{k+1} - \hat{x}^\nu\|^2 \text{ for all } k > \hat{k}. \tag{9}$$

Then $\mu_k \to \infty$ by the update rule for the prox-parameter after null steps (recall that $\beta > 1$ in the algorithm). It follows from condition (6) that $H_\nu^M(x^{k+1};\hat{x}^\nu) + \frac{\mu_k}{2}\|x^{k+1} - \hat{x}^\nu\|^2 \le H_\nu^M(\hat{x}^\nu;\hat{x}^\nu) = H(\hat{x}^\nu;\hat{x}^\nu)$, where the equality follows from definition of H_ν^M. Furthermore, it follows from A.2 that there exist $\tilde{a}, \tilde{b} \in \mathbb{R}$ such that $-\frac{\tilde{a}}{2}\|x^{k+1} - \hat{x}^\nu\|^2 + \tilde{b} \le H_\nu^M(x^{k+1};\hat{x}^\nu)$. As a result,

$$\tilde{b} + \frac{\mu_k - \tilde{a}}{2}\left\|x^{k+1} - \hat{x}^\nu\right\|^2 \le H_\nu^M(x^{k+1};\hat{x}^\nu) + \frac{\mu_k}{2}\|x^{k+1} - \hat{x}^\nu\|^2 \le H(\hat{x}^\nu;\hat{x}^\nu) \quad \forall\, k > \hat{k}.$$

As $\hat{x}^\nu$ is fixed and $\mu_k \to \infty$, we conclude that $\lim_{k\to\infty} x^{k+1} = \hat{x}^\nu$.

Now let W be a neighbourhood of $\hat{x}^\nu$ and let $\ell_{\hat{x}^\nu} > 0$ be the constant given by Lemma 2. Since $x^{k+1} \to \hat{x}^\nu$, we may assume, without loss of generality, that $x^k \in W$ for all sufficiently large k. Therefore,

$$\begin{aligned} H(\hat{x}^\nu;\hat{x}^\nu) &\ge H_\nu^M(x^{k+1};\hat{x}^\nu) + \frac{\mu_k}{2}\left\|x^{k+1} - \hat{x}^\nu\right\|^2 \\ &= \max\Big\{f(x^{k+1}) - \tau(\hat{x}^\nu),\ M_\nu(x^{k+1})\Big\} + \frac{\mu_k}{2}\left\|x^{k+1} - \hat{x}^\nu\right\|^2 \end{aligned}$$

$$
\begin{aligned}
&\geq \max\Big\{f(x^{k+1}) - \tau(\hat{x}^\nu),\ g_0(G(x^{k+1})) - \frac{\ell_{\hat{x}^\nu}}{2}\left\|x^{k+1} - \hat{x}^\nu\right\|^2\Big\} + \frac{\mu_k}{2}\left\|x^{k+1} - \hat{x}^\nu\right\|^2 \\
&\geq \max\Big\{f(x^{k+1}) - \tau(\hat{x}^\nu),\ g_0(G(x^{k+1}))\Big\} + \frac{\mu_k - \ell_{\hat{x}^\nu}}{2}\left\|x^{k+1} - \hat{x}^\nu\right\|^2 \\
&= H(x^{k+1}; \hat{x}^\nu) + \frac{\mu_k - \ell_{\hat{x}^\nu}}{2}\left\|x^{k+1} - \hat{x}^\nu\right\|^2, \qquad (10)
\end{aligned}
$$

where the penultimate inequality follows from (4). Combining this inequality with (9), we obtain

$$\frac{\mu_k - \ell_{\hat{x}^\nu}}{2}\left\|x^{k+1} - \hat{x}^\nu\right\|^2 < \frac{\gamma}{2}\left\|x^{k+1} - \hat{x}^\nu\right\|^2.$$

This is impossible for k large enough, since $\mu_k \to +\infty$ and $\hat{x}^\nu$ is fixed by assumption. Hence, only finitely many consecutive null steps can occur.

If we impose that G is Lipschitz and g_0 is K-weakly concave on $G(X)$, then the constant $\ell_{\hat{x}^\nu} \geq 0$ featuring in (4) is independent of $\hat{x}^\nu$ and hold uniformly on X, as commented in Remark 1. Therefore, (10) holds with $\ell_{\hat{x}^\nu} = \ell$ fixed. Thus the same reasoning as above implies only finitely many consecutive null steps can occur.

□

Proposition 7 (Serious steps.) *Suppose that Assumption 1 holds and Algorithm 1 applied to problem (P) loops indefinitely. Then $\nu \to \infty$ and every cluster point (if any) of the sequence of stability centers $(\hat{x}^\nu)_{\nu \in \mathbb{N}}$ is a critical point of problem (P).*

Proof Since the algorithm loops indefinitely, it follows from Proposition 6 that the algorithm produces infinitely many serious steps: $\nu \to \infty$. We consider two cases:

i) There exists i such that $g_0(G(\hat{x}^i)) \leq 0$. In this case, $H(\hat{x}^i; \hat{x}^i) = \max\{f(\hat{x}^i) - (f(\hat{x}^i) + \rho[g_0(G(\hat{x}^i))]_+),\, g_0(G(\hat{x}^i))\} = 0$, and the descent test implies

$$g_0(G(\hat{x}^{i+1})) \leq H(\hat{x}^{i+1}; \hat{x}^i) \leq H(\hat{x}^i; \hat{x}^i) - \frac{\gamma}{2}\left\|\hat{x}^{i+1} - \hat{x}^i\right\|^2 = -\frac{\gamma}{2}\left\|\hat{x}^{i+1} - \hat{x}^i\right\|^2,$$

showing that $g_0(G(\hat{x}^{i+1})) \leq 0$. Furthermore, the definition of the improvement function gives

$$f(\hat{x}^{i+1}) - f(\hat{x}^i) \leq H(\hat{x}^{i+1}; \hat{x}^i) \leq -\frac{\gamma}{2}\left\|\hat{x}^{i+1} - \hat{x}^i\right\|^2,$$

showing that

$$\frac{\gamma}{2}\|\hat{x}^{\iota+1} - \hat{x}^\iota\|^2 \leq f(\hat{x}^\iota) - f(\hat{x}^{\iota+1}) \quad \text{for all} \quad \iota = i, i+1, \ldots.$$

Summing over $\iota = i, \ldots, \nu$ we get

$$\sum_{\iota=i}^{\nu} \frac{\gamma}{2}\|\hat{x}^{\iota+1} - \hat{x}^\iota\|^2 \leq \sum_{\iota=i}^{\nu}[f(\hat{x}^\iota) - f(\hat{x}^{\iota+1})] \leq f(\hat{x}^i) - f(\hat{x}^{\nu+1}) < \infty,$$

where the last inequality follows from the fact that $(\hat{x}^\nu)_{\nu \geq i}$ is feasible to problem (P) and Assumption A.6 holds. This shows that $\lim_{\nu\to\infty}\|\hat{x}^{\nu+1} - \hat{x}^\nu\| = 0$.

ii) $g_0(G(\hat{x}^\nu)) > 0$ for all ν. In this case, $H(\hat{x}^\nu; \hat{x}^\nu) = \max\{f(\hat{x}^\nu) - (f(\hat{x}^\nu) + \rho g_0(G(\hat{x}^\nu))),\, g_0(G(\hat{x}^\nu))\} = g_0(G(\hat{x}^\nu))$. The descent step yields

$$g_0(G(\hat{x}^{\nu+1})) \leq H(\hat{x}^{\nu+1}; \hat{x}^\nu) \leq H(\hat{x}^\nu; \hat{x}^\nu) - \frac{\gamma}{2}\left\|\hat{x}^{\nu+1} - \hat{x}^\nu\right\|^2 = g_0(G(\hat{x}^\nu)) - \frac{\gamma}{2}\left\|\hat{x}^{\nu+1} - \hat{x}^\nu\right\|^2,$$

showing that

$$\sum_{\iota=0}^{\nu} \frac{\gamma}{2}\|\hat{x}^{\iota+1} - \hat{x}^\iota\|^2 \leq \sum_{\iota=0}^{\nu}[g_0(G(\hat{x}^\iota)) - g_0(G(\hat{x}^{\iota+1}))] \leq g_0(G(\hat{x}^0)) - g_0(G(\hat{x}^{\nu+1})) < g_0(G(\hat{x}^0)).$$

Again, we have that $\lim_{\nu\to\infty}\|\hat{x}^{\nu+1} - \hat{x}^\nu\| = 0$.

Let $\hat{x} \in X$ be an arbitrary accumulation point of $(\hat{x}^\nu)_{\nu\in\mathbb{N}}$. Then there exists an infinite index set $\mathcal{N} \subset \mathbb{N}$ such that $\lim_{\mathcal{N}\ni\nu\to\infty} \hat{x}^\nu = \hat{x}$. As the difference $\hat{x}^{\nu+1} - \hat{x}^\nu$ vanishes, we also get that $\lim_{\mathcal{N}\ni\nu\to\infty} \hat{x}^{\nu+1} = \hat{x}$. As specified at the beginning of this section, $k(\nu)$ denotes the iteration index at which the stability center $\hat{x}^{\nu+1}$ is generated. Consequently, $x^{k(\nu)+1}$ necessarily corresponds to a serious iterate. If there exists an infinite index set $\mathcal{N}' \subset \mathcal{N}$ such that $(\mu_{k(\nu)})_{\nu\in\mathcal{N}'}$ is bounded, we can rely on Lemma 5 to conclude that $\hat{x}$ is a critical point of problem (P). Hence, to finalize the proof it only remains to show that such index set exists. This is clearly the case if the algorithm generates only finitely many null steps: we can take $\mathcal{N}' = \mathcal{N}$ as the whole sequence $(\mu_k)_{k\in\mathbb{N}}$ is bounded in this case. In what follows we consider the remaining case in which the algorithm generates infinitely many null steps (interchanged with serious steps). To this end, let us denote the index set

$$\mathcal{N}^{\mathtt{n}} = \{\nu \in \mathcal{N} :\, x^{k(\nu)} \text{ is a null iterate}\} \quad = \quad \{\nu \in \mathcal{N} :\, x^{k(\nu)} \neq \hat{x}^{\nu'}\ \forall\, \nu'\},$$

and consider two alternatives: $\mathcal{N}^{\mathtt{n}}$ is finite or infinite.

If $\mathcal{N}^{\mathtt{n}}$ is finite (or even empty), we conclude that $x^{k(\nu)} = \hat{x}^\nu$ for all $\nu \in \mathcal{N}$ large enough. As a result, the rule for updating μ_k after a serious step ensures that $\mu_{k(\nu)} \le \mu_{\max}$ for all $\nu \in \mathcal{N}$ large enough. Hence, there exists $\mathcal{N}' \subset \mathcal{N}$ such that $(\mu_{k(\nu)})_{\nu\in\mathcal{N}'}$ is bounded. Therefore, it remains to consider the case in which $\mathcal{N}^{\mathtt{n}}$ is infinite. Suppose, by contradiction, that $\mathcal{N}^{\mathtt{n}}$ does not contain an infinite subset $\mathcal{N}'$ such that $(\mu_{k(\nu)})_{\nu\in\mathcal{N}'}$ is bounded. Equivalently,

$$\lim_{\mathcal{N}^{\mathtt{n}}\ni\nu\to\infty} \mu_{k(\nu)} = +\infty.$$

By definition, $x^{k(\nu)}$ is a null iterate for every $\nu \in \mathcal{N}^{\mathtt{n}}$. Moreover, condition (6) gives

$$\begin{aligned} H_\nu^M(x^{k(\nu)}; \hat{x}^\nu) + \frac{\mu_{k(\nu)-1}}{2}\|x^{k(\nu)} - \hat{x}^\nu\|^2 \le H_\nu^M(\hat{x}^\nu; \hat{x}^\nu) &= \max\{\rho[g_0(G(\hat{x}^\nu))]_+, g_0(G(\hat{x}^\nu))\} \\ &\le \max\{\rho, 1\}[g_0(G(\hat{x}^0))]_+ =: b_1, \end{aligned} \tag{11}$$

due to items i) and ii) above. On the other hand, by Assumption A.2, we have

$$\begin{aligned} H_\nu^M(x^{k(\nu)}; \hat{x}^\nu) &\ge -\frac{a}{2}\|x^{k(\nu)}\|^2 + b - \tau(\hat{x}^\nu) \\ &\ge -a\|x^{k(\nu)} - \hat{x}^\nu\|^2 - a\|\hat{x}^\nu\|^2 + b - \tau(\hat{x}^\nu). \end{aligned}$$

Since τ is continuous and $(\hat{x}^\nu)_{\nu\in\mathcal{N}^{\mathtt{n}}}$ is bounded, there exists a constant $b_2 > -\infty$ such that

$$-a\|\hat{x}^\nu\|^2 + b - \tau(\hat{x}^\nu) \ge b_2 \qquad \text{for all } \nu \in \mathcal{N}^{\mathtt{n}}.$$

Combining this bound with (11), we obtain

$$b_2 + \frac{\mu_{k(\nu)-1} - 2a}{2}\|x^{k(\nu)} - \hat{x}^\nu\|^2 \le b_1. \tag{12}$$

Since $\mu_{k(\nu)} \to +\infty$ along $\mathcal{N}^{\mathtt{n}}$, and since $\mu_{k(\nu)-1} = \beta^{-1}\mu_{k(\nu)}$, it follows from (12) that

$$\lim_{\mathcal{N}^{\mathtt{n}}\ni\nu\to\infty} \|x^{k(\nu)} - \hat{x}^\nu\| = 0.$$

Consequently,

$$\lim_{\mathcal{N}^{\mathtt{n}}\ni\nu\to\infty} x^{k(\nu)} = \lim_{\mathcal{N}^{\mathtt{n}}\ni\nu\to\infty} \hat{x}^\nu = \hat{x}.$$

Now, since $x^{k(\nu)}$ is a null iterate for every $\nu \in \mathcal{N}^{\mathtt{n}}$, we have

$$H(x^{k(\nu)}; \hat{x}^\nu) > H(\hat{x}^\nu; \hat{x}^\nu) - \frac{\gamma}{2}\|x^{k(\nu)} - \hat{x}^\nu\|^2.$$

Moreover, condition (6) yields

$$H_\nu^M(x^{k(\nu)}; \hat{x}^\nu) + \frac{\mu_{k(\nu)-1}}{2}\|x^{k(\nu)} - \hat{x}^\nu\|^2 \le H_\nu^M(\hat{x}^\nu; \hat{x}^\nu).$$

Since $H(\hat{x}^\nu; \hat{x}^\nu) = H_\nu^M(\hat{x}^\nu; \hat{x}^\nu)$, combining the two inequalities gives

$$\begin{aligned} H(x^{k(\nu)}; \hat{x}^\nu) > H(\hat{x}^\nu; \hat{x}^\nu) - \frac{\gamma}{2}\|x^{k(\nu)} - \hat{x}^\nu\|^2 &\ge H_\nu^M(x^{k(\nu)}; \hat{x}^\nu) + \frac{\mu_{k(\nu)-1} - \gamma}{2}\|x^{k(\nu)} - \hat{x}^\nu\|^2 \\ &= H_\nu^M(x^{k(\nu)}; \hat{x}^\nu) + \frac{\beta^{-1}\mu_{k(\nu)} - \gamma}{2}\|x^{k(\nu)} - \hat{x}^\nu\|^2. \end{aligned} \tag{13}$$

As in Lemma 2, let W be a neighbourhood of $\hat{x}$. Since

$$\hat{x}^\nu \to \hat{x} \qquad \text{and} \qquad x^{k(\nu)} \to \hat{x},$$

we have $\hat{x}^\nu, x^{k(\nu)} \in X \cap W$ for all $\nu \in \mathcal{N}^{\mathtt{n}}$ sufficiently large. Hence, Lemma 2 as well as Remark 1 apply, and we obtain

$$g_0(G(x^{k(\nu)})) \leq M_\nu(x^{k(\nu)}) + \frac{\ell_{\hat{x}}}{2}\|x^{k(\nu)} - \hat{x}^\nu\|^2$$

for all sufficiently large $\nu \in \mathcal{N}^{\mathtt{n}}$. Therefore,

$$\begin{aligned} H(x^{k(\nu)}; \hat{x}^\nu) &= \max\left\{f(x^{k(\nu)}) - \tau(\hat{x}^\nu), g_0(G(x^{k(\nu)}))\right\} \\ &\leq \max\left\{f(x^{k(\nu)}) - \tau(\hat{x}^\nu), M_\nu(x^{k(\nu)})\right\} + \frac{\ell_{\hat{x}}}{2}\|x^{k(\nu)} - \hat{x}^\nu\|^2 \\ &= H_\nu^M(x^{k(\nu)}; \hat{x}^\nu) + \frac{\ell_{\hat{x}}}{2}\|x^{k(\nu)} - \hat{x}^\nu\|^2. \end{aligned}$$

Combining this estimate with (13), we get

$$H_\nu^M(x^{k(\nu)}; \hat{x}^\nu) + \frac{\ell_{\hat{x}}}{2}\|x^{k(\nu)} - \hat{x}^\nu\|^2 > H_\nu^M(x^{k(\nu)}; \hat{x}^\nu) + \frac{\beta^{-1}\mu_{k(\nu)} - \gamma}{2}\|x^{k(\nu)} - \hat{x}^\nu\|^2.$$

Hence,

$$\beta^{-1}\mu_{k(\nu)} \leq \gamma + \ell_{\hat{x}}$$

for all sufficiently large $\nu \in \mathcal{N}^{\mathtt{n}}$, which contradicts the fact that

$$\lim_{\mathcal{N}^{\mathtt{n}} \ni \nu \to \infty} \mu_{k(\nu)} = +\infty.$$

Consequently, there exists an infinite index set $\mathcal{N}' \subset \mathcal{N}^{\mathtt{n}}$ such that $(\mu_{k(\nu)})_{\nu \in \mathcal{N}'}$ is bounded. Since

$$\lim_{\mathcal{N} \ni \nu \to \infty} \hat{x}^\nu = \hat{x} = \lim_{\mathcal{N} \ni \nu \to \infty} \hat{x}^{\nu+1},$$

we can apply Lemma 5 and conclude that $\hat{x} \in X$ is a critical point of problem (*P*).

□

We are now in a position to state the main convergence theorem for Algorithm 1, which characterizes the asymptotic behavior of the method.

Theorem 8 (Convergence) *Consider Algorithm 1 with* `Tol` $=0$ *applied to problem (P). Suppose that Assumption 1 holds,* $\mu_k > 2a \geq 0$ *for all k, with a given in item A.2. The following holds.*

i) The sequence of iterates $(x^k)_{k \in \mathbb{N}}$ *is well defined;*

ii) If there exists $\iota \in \mathbb{N}$ *such that* $g_0(G(\hat{x}^\iota)) \leq 0$*, then all* $\hat{x}^\nu$*, with* $\nu \geq \iota$*, are feasible for problem (P);*

iii) The algorithm either terminates after finitely many steps, in which case its last stability center $\hat{x}^\nu$ *is a critical point of problem (P), or the algorithm loops indefinitely and produces an infinite sequence* $(\hat{x}^\nu)_{\nu \in \mathbb{N}}$ *of stability centers. All cluster points (if any) of this sequence are critical points of problem (P);*

iv) Let $\hat{x} \in X$ *be a cluster point of* $(\hat{x}^\nu)_{\nu \in \mathbb{N}}$ *and* $\mathcal{G}$ *be the set-valued mapping defined in (2). If either* $g_0(G(\hat{x})) < 0$ *or* $g_0(G(\hat{x})) = 0$ *and the constraint qualification* $0 \notin \mathcal{G}(\hat{x}) + N_X(\hat{x})$ *holds, then* $\hat{x}$ *satisfies the generalized KKT condition of Definition 2.*

Proof Item i) follows from Remark 2(i). Item ii) is a well-known property of the improvement function, which was also derived in the proof of Proposition 7 (see item i) therein). We now proceed to show item iii). Observe that if the algorithm stops at iteration k with `Tol` $=0$, then $x^{k+1} = \hat{x}^\nu$ and this last stability center is, from Lemma 5, a critical point. Otherwise, the algorithm loops indefinitely. Proposition 7 asserts that the algorithm produces an infinite sequence $(\hat{x}^\nu)_{\nu \in \mathbb{N}}$ of stability centers, and every cluster point of this sequence is critical for problem (*P*). Item iv) follows from Lemma 4. □

4 Numerical Illustrations

We consider an industrially motivated cascaded reservoir management problem formulated as a joint chance-constrained program (JCCP). This benchmark provides a suitable test bed for evaluating the proposed algorithm and comparing its performance with established baseline methods.

All benchmark instances are written in the following canonical JCCP form:

$$\begin{aligned} \min_{x \in \mathbb{R}^n_+} \quad & c^\top x \\ \text{s.t.} \quad & Ax \leq b, \\ & A_{\text{eq}} x = b_{\text{eq}}, \\ & \varphi(x) := \mathbb{P}(\xi \leq Hx + h) \geq p, \end{aligned} \tag{14}$$

where A, A_{eq}, and H are deterministic matrices; c, b, b_{eq}, and h are deterministic vectors of compatible dimensions; and ξ is a random vector following an elliptically symmetric distribution with mean μ and covariance matrix Σ. In the implementation, whenever available, the parameters μ and Σ are chosen according to specifications reported in the literature (see [23]) in order to ensure consistency with previous numerical studies.

4.1 Checking the Assumptions

Note that the numerical instances in (14) fit the abstract problem (P). Define

$$X := \left\{ x \in \mathbb{R}^n_+ : Ax \leq b,\ A_{\text{eq}} x = b_{\text{eq}} \right\}, \qquad f(x) := c^\top x.$$

Let F_i denote the marginal distribution function of ξ_i, $i = 1, \dots, m$ and let $h(x) := Hx + h$ for simplicity. By Sklar's representation, there exists a copula C associated with ξ such that

$$F_\xi(y) = \mathbb{P}(\xi \leq y) = C\big(F_1(y_1), \dots, F_m(y_m)\big), \qquad y \in \mathbb{R}^m.$$

Note the copula C need not be available in closed form. However, as shown later in this section, its gradient information can be obtained at a stability center by solving a linear system involving the gradient $\nabla \varphi$ of the joint probability function. Therefore,

$$\mathbb{P}(\xi \leq h(x)) = C\big(F_1(h_1(x)), \dots, F_m(h_m(x))\big).$$

The joint chance constraint can be written as

$$\mathbb{P}(\xi \leq h(x)) \geq p \quad \Longleftrightarrow \quad p - C\big(F_1(h_1(x)), \dots, F_m(h_m(x))\big) \leq 0.$$

Hence (14) is an instance of (P) with

$$G(x) := \big(F_1(h_1(x)), \dots, F_m(h_m(x))\big), \qquad g_0(z) := p - C(z).$$

Indeed,

$$g_0(G(x)) = p - C\big(F_1(h_1(x)), \dots, F_m(h_m(x))\big) = p - \mathbb{P}(\xi \leq h(x)), \tag{15}$$

so that $g_0(G(x)) \leq 0$ is exactly the joint chance constraint in (14).

For the benchmark distributions, we impose the following regularity conditions. The random vector ξ is assumed to be continuous, and elliptically symmetric, with joint distribution function F_ξ, marginal distribution functions F_i, $i = 1, \dots, m$, and associated copula C.

We assume that F_ξ is locally $C^{1,1}$ on a neighbourhood of $\{h(x) : x \in X\}$. Moreover, for each $i = 1, \dots, m$, the marginal distribution function F_i is locally $C^{1,1}$ on a neighbourhood of $\{h_i(x) : x \in X\}$, and its marginal density $f_i = F_i'$ satisfies

$$f_i(t) > 0 \qquad \text{for all } t \in \{h_i(x) : x \in X\}.$$

Note first that these conditions imply local $C^{1,1}$-regularity of the copula on the transformed feasible image. Indeed, by Sklar's representation,

$$C(u) = F_\xi\big(F_1^{-1}(u_1), \dots, F_m^{-1}(u_m)\big), \qquad u \in (0,1)^m,$$

where F_i^{-1} denotes the generalized inverse of F_i. Fix any point

$$\bar{u} = \big(F_1(\bar{y}_1), \dots, F_m(\bar{y}_m)\big) \in G(X), \qquad \bar{y} = h(\bar{x}) \quad \text{for some } \bar{x} \in X.$$

For each $i = 1, \dots, m$, we have $f_i(\bar{y}_i) > 0$. Since F_i is locally $C^{1,1}$, it is in particular C^1, and therefore the classical inverse-function theorem implies that F_i^{-1} is locally C^1 around $F_i(\bar{y}_i)$; see, e.g., [24, Theorem 1A.1]. Moreover, on this local neighbourhood,

$$(F_i^{-1})'(u_i) = \frac{1}{f_i(F_i^{-1}(u_i))}.$$

Since $f_i = F_i'$ is locally Lipschitz and F_i^{-1} is locally C^1, the composition $f_i \circ F_i^{-1}$ is locally Lipschitz. In addition, because $f_i(\bar{y}_i) > 0$ and f_i is continuous, $f_i \circ F_i^{-1}$ is locally bounded away from zero around $F_i(\bar{y}_i)$. Hence the reciprocal

$$u_i \mapsto \frac{1}{f_i(F_i^{-1}(u_i))}$$

is locally Lipschitz. Therefore F_i^{-1} is locally $C^{1,1}$ around $F_i(\bar{y}_i)$. Thus the mapping

$$u \mapsto \big(F_1^{-1}(u_1), \dots, F_m^{-1}(u_m)\big)$$

is locally $C^{1,1}$ around $\bar{u}$. Combining this with the local $C^{1,1}$-regularity of F_ξ, we obtain that C is locally $C^{1,1}$ around $\bar{u}$. Since $\bar{u} \in G(X)$ was arbitrary, C is locally $C^{1,1}$ on a neighbourhood of

$$G(X) = \big\{\big(F_1(h_1(x)), \dots, F_m(h_m(x))\big) : x \in X\big\}.$$

Under these conditions, the benchmark instances satisfy Assumption 1. First, Assumption A.1 holds because $X = \big\{x \in \mathbb{R}^n_+ : Ax \le b,\ A_{\mathrm{eq}}x = b_{\mathrm{eq}}\big\}$ is a closed polyhedron, and the benchmark instances are constructed with $X \neq \emptyset$. Assumption A.2 also holds since $f(x) = c^\top x$ is affine. In particular, f is locally Lipschitz continuous and satisfies the required lower quadratic bound on X. Next, Assumption A.3 holds. Indeed, the map $x \mapsto h(x)$ is affine, and each marginal distribution function F_i is locally $C^{1,1}$ on the relevant marginal range. Therefore $G(x) = \big(F_1(h_1(x)), \dots, F_m(h_m(x))\big)$ is locally $C^{1,1}$, and hence locally Lipschitz, on a neighbourhood of X. Moreover, Assumption A.4 holds because the copula C is locally $C^{1,1}$ on a neighbourhood of $G(X)$. Hence $g_0(z) = p - C(z)$ is locally $C^{1,1}$, and therefore upper-C^2, on a neighbourhood of $G(X)$. Finally, Assumption A.5 follows from the polyhedral structure of X, the affine structure of f, and the smooth composite structure of G and g_0 on the relevant regions. Assumption A.6 is satisfied by construction of the benchmark instances, whose feasible sets are nonempty and whose optimal values are finite.

The benchmark experiments use hydro-valley instances, described in detail in Subsection 4.3, driven by primitive inflow vectors following nondegenerate Gaussian and Student-t distributions. Under the specified covariance or scale matrices, these primitive inflow distributions have locally $C^{1,1}$ distribution functions and marginal distribution functions, with positive marginal densities on the relevant ranges. Consequently, the benchmark instances satisfy the regularity conditions stated above and fit the framework of (P).

4.2 Experimental Setup

Note that by replacing the corresponding g_0 and G in (3), we can use, at each stability center $\hat{x}^\nu$, the following model:

$$M_\nu(x) = p - \varphi(\hat{x}^\nu) - \sum_{j=1}^{m} v_j^\nu \big[F_j(h_j(x)) - F_j(h_j(\hat{x}^\nu))\big], \qquad v^\nu = \nabla C\big(F(h(\hat{x}^\nu))\big). \tag{16}$$

This model keeps the marginal transformations inside the inner map G, rather than linearizing the full probability function directly in the decision variable. Consequently, the approximation retains information on how each individual marginal threshold $h_j(x)$ affects the joint probability through the copula weights v^ν. In this sense, the copula representation provides an informative inner function G, and the model (16) gives a structured approximation of the composite constraint $g_0 \circ G \equiv p - \varphi$. As a result, model (16) can be interpreted as a linearization of marginal correlations, as the copula encodes their dependence structure.

The copula representation is also used to obtain gradient information with respect to the copula C. Note

$$\varphi(x) = C\big(F_1(h_1(x)), \dots, F_m(h_m(x))\big) = C\big(F(h(x))\big).$$

Assume that the gradient $\nabla\varphi(\hat{x}^\nu)$ is available at a stability center $\hat{x}^\nu \in X$. In our implementation, this gradient is supplied by the probability oracle described below. Following the marginal-assessment model of [1], the vector $v^\nu \in [0,1]^m$, which represents the copula-gradient weights at the point $F(h(\hat{x}^\nu))$ is computed as a solution of the linear system, without requiring explicit knowledge of the copula:

$$\sum_{j=1}^{m} v_j f_j\big(h_j(\hat{x}^\nu)\big)\nabla h_j(\hat{x}^\nu) = \nabla\varphi(\hat{x}^\nu),$$

where $f_j = F_j'$ is the density of the j-th marginal distribution. When C is differentiable at $F(h(\hat{x}^\nu))$, by [25, Theorem 1.6.1], above linear system admits at least one solution of the form

$$v^\nu = \nabla C\big(F(h(\hat{x}^\nu))\big) \in [0,1]^m.$$

Under the above setting, in particular, the fact that each marginal distribution function F_i is locally $C^{1,1}$ and copula C is locally $C^{1,1}$, our master problem (MP) can then be written as

$$\min_{x\in X} \max\{\, c^\top x - \tau(\hat{x}^\nu),\ M_\nu(x)\,\} + \frac{\mu_k}{2}\|x - \hat{x}^\nu\|^2,$$

which can be solved to stationarity by a standard NLP solver using the epigraph formulation in the form of Remark 2.

All benchmark experiments for solving problems of the form (14) are conducted using our *Linear Programming with Chance Constraints (LPCC)* toolbox. The toolbox is implemented in the Julia programming language and integrated with the JuMP modeling environment, and will be released as open-source software. It includes the following modules:

1. An oracle estimating $\varphi(x) := \mathbb{P}(\xi \le Hx + h)$ together with a Clarke subgradient estimator. The oracle used to estimate φ and its gradient is based on the spherical–radial decomposition (SRD) technique (see, e.g., [26–28]).

2. Algorithms for solving (14) under different distributional assumptions for ξ.
3. The benchmark instances used in this paper.

For each benchmark instance, we compare: (i) Ipopt, a primal–dual interior-point method; (ii) the extended Level Bundle method; and (iii) our Proximal method (see Algorithm 1). Unless otherwise stated, the oracle precision used to evaluate $\varphi(x)$ and its gradient is set to 2×10^{-5} across all methods. Sampling over the unit sphere in the SRD estimator is performed using quasi–Monte Carlo (QMC) sequences (see [29]), combined with antithetic directions to reduce estimator variance.

For each probability level $p \in \{0.8, 0.85, 0.9, 0.95, 0.99\}$, we report:

1. CPU time (seconds),
2. the number of oracle calls, measured as the number of evaluations of the joint chance constraint and its gradient.

All numerical experiments were conducted on a Dell Precision 3571 workstation equipped with a 12th Gen Intel Core i9-12900H processor (2.50,GHz) and 32 GB of RAM, running a 64-bit operating system on an x64-based architecture.

We also report data and performance profiles (see [30] and [31]) for the number of oracle calls $\#F$ and for CPU time (in seconds) across all benchmark instances for Gaussian and Student-t distributions.

Whenever convergence is achieved, all methods return nearly identical objective values and satisfy the chance constraint within numerical tolerance, unless stated otherwise.

Three remarks regarding the benchmark design and algorithm implementation are in order:

R.1 (Individual chance constraints.) In addition to the joint chance constraint, we impose the following individual chance constraints:

$$\mathbb{P}(\xi_i \leq (Hx + h)_i) \geq p, \qquad i = 1, \ldots, m.$$

Since ξ follows a symmetric elliptical distribution with known mean μ and covariance matrix Σ, each individual chance constraint admits the equivalent deterministic reformulation

$$(Hx)_i \geq \mu_i + \kappa_\xi(p)\,\sigma_i - h_i, \qquad i = 1, \ldots, m,$$

where $\sigma_i = \sqrt{\Sigma_{ii}}$ and $\kappa_\xi(p)$ denotes the p-quantile of the standardized one-dimensional marginal distribution associated with the chosen elliptical family.
Any solution satisfying the original joint chance constraint necessarily satisfies each of the individual constraints above, since the event $\{\xi \leq Hx + h\}$ implies $\{\xi_i \leq (Hx + h)_i\}$ for every component i. Consequently, adding these constraints does not exclude the optimal solution of the original problem. Because the resulting inequalities are linear in x, they can be enforced at negligible computational cost. In practice, they significantly reduce the region explored by the solvers and prevent iterates that are strongly infeasible with respect to the chance constraint, thereby improving numerical stability and computational efficiency.
Accordingly, these individual chance constraints are included together with the polyhedral constraints in (14) when computing an initial point for each solver, and are also imposed in the master problems throughout the algorithms.

R.2 (Applicability of the Level Bundle method.) In our implementation of the Level Bundle method, the joint chance constraint is enforced through the log-transformed function

$$\ell(x) := \log(p) - \log(\varphi(x)) \leq 0, \qquad \varphi(x) = \mathbb{P}(\xi \leq Hx + h).$$

The Level Bundle method relies on convexity of $\ell(\cdot)$ in order to construct valid cutting-plane models. A sufficient condition for this convexity is the log-concavity of $\varphi(\cdot)$, which holds when the distribution of ξ is log-concave (for example, when ξ is Gaussian). If ξ does not admit a log-concave density (e.g., the Student-t distribution), then $\varphi(\cdot)$ may fail to be log-concave and consequently $\ell(\cdot)$ is generally nonconvex. In such situations the assumptions required by the Level Bundle framework are violated, which may lead to unstable behavior in practice. For this reason, the Level Bundle method is excluded from experiments involving non-log-concave distributions.

R.3 (Bonferroni initialization method.) For each instance, we also compare the algorithms using a starting point obtained from a Bonferroni-type approximation of the joint chance constraint. Given an auxiliary probability level $(0, 1) \ni p_{\text{search}} \geq p$ and m marginal constraints, we replace the joint chance constraint by the row-wise sufficient conditions

$$p_i = 1 - \frac{1 - p_{\text{search}}}{m}, \qquad i = 1, \ldots, m,$$

that is,

$$\mathbb{P}(\xi_i \leq (Hx + h)_i) \geq p_i, \qquad i = 1, \ldots, m.$$

This is a conservative approximation of the original joint chance constraint.

In the implementation, this approximation is used only as an initialization device. Starting from an initial value of p_{search}, the procedure solves a sequence of row-wise Bonferroni approximation problems and evaluates the true joint probability $\varphi(x)$ at the resulting candidate points. The auxiliary level p_{search} is adjusted by a coarse search, followed when possible by bisection, in order to find a point whose true joint probability is close to the target reliability level. Specifically, we seek an initial point x_0 satisfying

$$p \leq \varphi(x_0) \leq p + 10^{-3}.$$

If no candidate is found in this target band within the prescribed evaluation budget, the closest feasible candidate found during the Bonferroni search is used instead.

R.4 (Scaling.) In the implementation of the Proximal method, the master problem (MP) is solved using the Julia interface to Ipopt. To improve numerical stability, the two hydro-valley benchmark instances are scaled before being passed to the solver. Since the Isère and Ain instances have different numerical ranges, we use instance-dependent scaling factors.

4.3 Hydro Reservoir Management Problem

The benchmark instances are hydro-valley reservoir management problems, namely the Isère Valley and Ain Valley instances, arising from realistic energy planning applications and widely studied in the context of joint chance-constrained programming; see, e.g., [23].

The model describes a cascaded multi-reservoir system operated over a finite planning horizon. At each time stage, release decisions must simultaneously satisfy electricity production requirements and maintain reservoir storage levels within admissible bounds. Uncertainty enters through stochastic inflows to the reservoirs, which accumulate over time and propagate through the cascading structure.

Let T denote the planning horizon and n the number of reservoirs. The decision vector x collects release decisions and other operational variables across all time stages. Let $\xi' \in \mathbb{R}^{48}$ denote the vector of cumulative stochastic inflows over the horizon. In the benchmark experiments, ξ' is modeled by a nondegenerate continuous elliptically symmetric distribution with location vector $\mu \in \mathbb{R}^{48}$ and positive definite dispersion matrix $\Sigma \in \mathbb{R}^{48\times 48}$. The radial law is chosen to give either the multivariate Gaussian case or the multivariate Student-t case. In the Gaussian case, Σ is the covariance matrix, whereas in the Student-t case, Σ denotes the scale matrix (the covariance matrix up to a multiplicative constant), together with the prescribed degrees of freedom.

As shown in [23], the reservoir level constraints can be expressed as a two-sided probabilistic requirement of the form

$$\mathbb{P}(Ax + a \leq \xi' \leq Ax + b) \geq p,$$

where the matrix A captures the cumulative impact of release decisions on reservoir levels, and a, b incorporate deterministic components such as initial storage levels, deterministic inflows, and admissible storage bounds, with $a_i < b_i$ for all i. For algorithmic convenience, this two-sided constraint is rewritten in an equivalent one-sided joint form. Define

$$B := \begin{bmatrix} I \\ -I \end{bmatrix}, \qquad H := \begin{bmatrix} A \\ -A \end{bmatrix}, \qquad h := \begin{bmatrix} b \\ -a \end{bmatrix}, \qquad \xi := B\xi' = \begin{bmatrix} \xi' \\ -\xi' \end{bmatrix}.$$

Then

$$Ax + a \leq \xi' \leq Ax + b \quad \iff \quad \xi \leq Hx + h,$$

and hence the chance constraint can be written compactly as $\mathbb{P}(\xi \leq Hx + h) \geq p$.

Since ξ is obtained from ξ' by the linear mapping B, the stacked vector remains elliptically symmetric, with location vector and dispersion matrix

$$B\mu = \begin{bmatrix} \mu \\ -\mu \end{bmatrix}, \qquad B\Sigma B^\top = \begin{bmatrix} \Sigma & -\Sigma \\ -\Sigma & \Sigma \end{bmatrix}.$$

The same expression applies to both the Gaussian and Student-t cases, with the interpretation of Σ as covariance in the Gaussian case and as scale matrix in the Student-t case. The stacked representation is degenerate, even when the primitive inflow vector ξ' is nondegenerate, because $B\Sigma B^\top$ is rank deficient.

This degeneracy does not affect the local regularity required by the algorithms. The stacked vector is introduced only to rewrite the two-sided reservoir-level constraint in the canonical one-sided form (14). Although the distribution of the stacked vector is singular in the ambient space, its joint distribution function is locally $C^{1,1}$ on the threshold region relevant to the benchmark instances.

Indeed, evaluating the stacked distribution function near the hydro-valley threshold set is equivalent to evaluating the probability that the primitive inflow vector ξ' lies between the corresponding lower and upper reservoir bounds. Under the specified Gaussian or Student-t inflow laws, the primitive inflow vector has a smooth positive density. Moreover, since the lower and upper reservoir bounds satisfy $a_i < b_i$ for every component, the corresponding intervals remain strictly separated on the region considered by the algorithms. Hence the associated two-sided box probability is locally $C^{1,1}$ with respect to the relevant threshold values.

The marginal regularity assumptions are also satisfied. Each marginal of the stacked vector is either a marginal of ξ' or of $-\xi'$. Therefore, in the Gaussian and Student-t cases, the marginal distribution functions are locally $C^{1,1}$, and their densities are positive on the relevant threshold ranges. Together with the preceding Sklar-based argument, this implies that the copula representation induced by the stacked vector is locally $C^{1,1}$ on the transformed image, even though the stacked distribution is singular globally.

With this representation, the hydro-valley reservoir management problem can be expressed as a joint chance-constrained problem of the form (14), with $x \in \mathbb{R}^{672}$ and stacked random vector $\xi \in \mathbb{R}^{96}$.

Here, the deterministic constraints $Ax \leq b$ encode operational and capacity limits, while the joint chance constraint ensures that reservoir levels remain within admissible bounds at all time–reservoir combinations with probability at least p.

The hydro benchmark is reasonably large and structurally coupled. In particular, the dimension of the random vector equals the number of time–reservoir combinations, and evaluating the joint probability involves high-dimensional probability estimation. This makes the problem especially suitable for assessing the scalability and robustness of algorithms for joint chance-constrained optimization.

4.3.1 Isère Hydro Benchmark Results

We report benchmark results for the Isère hydro-valley instance under both Gaussian and Student-t distributions in Table 1–4.

Table 1 Isère hydro benchmark (Gaussian distribution), without applying the Bonferroni procedure to obtain a good starting point. Each entry is reported as Time in seconds (oracle calls). Entries marked with $*$ indicate that the method did not attain an objective value within relative tolerance 10^{-4} of the best feasible objective value for that instance.

Algorithm	$p = 0.8$	$p = 0.85$	$p = 0.9$	$p = 0.95$	$p = 0.99$
Proximal	383.83 (34)	411.22 (35)	308.50 (24)	255.70 (19)	343.51 (25)
Level Bundle	366.33 (42)	331.78 (38)	307.99 (35)	284.08 (33)	247.02 (29)
IPOPT	871.09 (52)	590.41 (35)	674.98 (40)	849.59 (49)	1261.48 (72)$*$

Table 2 Isère hydro benchmark (Gaussian distribution), with the Bonferroni procedure applied to obtain a good starting point. Each entry is reported as Time in seconds (oracle calls). Entries marked with $*$ indicate that the method did not attain an objective value within relative tolerance 10^{-4} of the best feasible objective value for that instance. The average number of oracle calls required to construct the Bonferroni initial point is 11 for all methods, and is included in the table.

Algorithm	$p = 0.8$	$p = 0.85$	$p = 0.9$	$p = 0.95$	$p = 0.99$
Proximal	443.10 (39)	442.24 (39)	340.27 (29)	263.25 (23)	366.07 (26)
Level Bundle	432.56 (50)	442.60 (46)	380.24 (43)	349.54 (41)	245.16 (29)
IPOPT	1265.18 (76)$*$	770.64 (48)	2095.61 (117)$*$	1998.40 (112)$*$	1388.54 (73)$*$

Table 3 Isère hydro benchmark (Student-t distribution), without applying the Bonferroni procedure to obtain a good starting point. Each entry is reported as Time in seconds (oracle calls). Entries marked with $*$ indicate that the method did not attain an objective value within relative tolerance 10^{-4} of the best feasible objective value for that instance.

Algorithm	$p = 0.8$	$p = 0.85$	$p = 0.9$	$p = 0.95$
Proximal	472.63 (37)	376.49 (28)	297.64 (23)	277.91 (22)
IPOPT	1890.50 (97)$*$	1862.93 (95)	997.37 (53)	1992.11 (104)$*$

Table 4 Isère hydro benchmark (Student-t distribution), with the Bonferroni procedure applied to obtain a good starting point. Each entry is reported as Time in seconds (oracle calls). Entries marked with $*$ indicate that the method did not attain an objective value within relative tolerance 10^{-4} of the best feasible objective value for that instance. The average number of oracle calls required to construct the Bonferroni initial point is 9.3 for all methods.

Algorithm	$p = 0.8$	$p = 0.85$	$p = 0.9$	$p = 0.95$
Proximal	310.32 (27)	284.12 (26)	371.33 (30)	382.70 (30)
IPOPT	924.35 (53)	1898.51 (113)$*$	915.00 (57)	1322.21 (88)$*$

Under the Gaussian distribution, both Proximal and Level Bundle substantially outperform Ipopt on the Isère instance. Without Bonferroni initialization, Level Bundle is usually the fastest method, while Proximal is competitive in runtime and consistently uses fewer oracle calls. With Bonferroni initialization, the two methods become closer: Proximal is faster for three of the five reliability levels and remains more economical in oracle calls, whereas Level Bundle is faster at $p = 0.8$ and $p = 0.99$. In contrast, Ipopt is generally slower and frequently fails to attain the prescribed objective tolerance.

For the Student-t distribution, where Level Bundle is not applicable, Proximal consistently outperforms Ipopt in both runtime and oracle calls, with and without Bonferroni initialization. Overall, the Isère results show that Level Bundle is highly competitive in the Gaussian case, while Proximal is more robust across distributions and more efficient in terms of oracle evaluations.

4.3.2 Ain Hydro Benchmark Results

We next report benchmark results for the Ain hydro-valley instance under both Gaussian and Student-t distributions in Table 5–8.

Table 5 Ain hydro benchmark (Gaussian distribution), without applying the Bonferroni procedure to obtain a good starting point. Each entry is reported as Time in seconds (oracle calls). Entries marked with $*$ indicate that the method did not attain an objective value within relative tolerance 10^{-4} of the best feasible objective value for that instance.

Algorithm	$p = 0.8$	$p = 0.85$	$p = 0.9$	$p = 0.95$	$p = 0.99$
Proximal	403.13 (27)	296.05 (23)	278.87 (19)	312.49 (19)	327.26 (24)
Level Bundle	491.83 (53)	471.28 (18)$*$	630.41 (36)	538.57 (49)	452.62 (46)
IPOPT	2024.57 (94)$*$	2387.95 (105)$*$	1823.07 (86)$*$	1360.02 (63)	1845.07 (69)$*$

Table 6 Ain hydro benchmark (Gaussian distribution), with the Bonferroni procedure applied to obtain a good starting point. Each entry is reported as Time in seconds (oracle calls). Entries marked with $*$ indicate that the method did not attain an objective value within relative tolerance 10^{-4} of the best feasible objective value for that instance. The average number of oracle calls required to construct the Bonferroni initial point is 9.4 for all methods.

Algorithm	$p = 0.8$	$p = 0.85$	$p = 0.9$	$p = 0.95$	$p = 0.99$
Proximal	279.66 (26)	226.58 (21)	235.41 (21)	257.82 (22)	348.71 (25)
Level Bundle	575.60 (61)	537.29 (26)$*$	817.00 (57)	633.51 (57)	429.13 (46)
IPOPT	1355.93 (87)	2035.11 (134)$*$	890.47 (57)	740.94 (52)	933.07 (63)

Table 7 Ain hydro benchmark (Student-t distribution), without applying the Bonferroni procedure to obtain a good starting point. Each entry is reported as Time in seconds (oracle calls). Entries marked with $*$ indicate that the method did not attain an objective value within relative tolerance 10^{-4} of the best feasible objective value for that instance.

Algorithm	$p = 0.8$	$p = 0.85$	$p = 0.9$	$p = 0.95$
Proximal	241.53 (19)	353.87 (21)	493.50 (21)	739.88 (31)
IPOPT	1817.90 (114)$*$	2028.34 (125)$*$	1868.81 (122)$*$	969.03 (55)

Table 8 Ain hydro benchmark (Student-t distribution), with the Bonferroni procedure applied to obtain a good starting point. Each entry is reported as Time in seconds (oracle calls). Entries marked with $*$ indicate that the method did not attain an objective value within relative tolerance 10^{-4} of the best feasible objective value for that instance. The average number of oracle calls required to construct the Bonferroni initial point is 9.3 for all methods.

Algorithm	$p = 0.8$	$p = 0.85$	$p = 0.9$	$p = 0.95$
Proximal	583.40 (30)	620.13 (27)	541.87 (24)	573.29 (30)
IPOPT	922.22 (66)	820.88 (59)	576.90 (43)	745.15 (49)

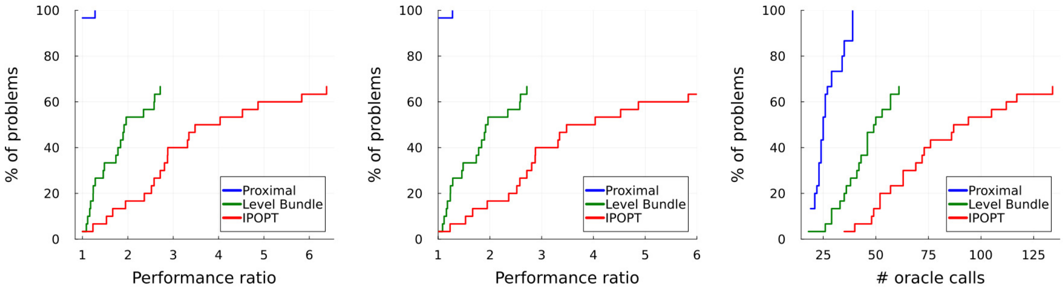

Fig. 1 Gaussian case. Performance profile (full scale), performance profile (zoomed), and data profile with respect to the number of oracle calls ($\#F$).

Across all reported Ain instances, the Proximal method is the fastest method among the reported algorithms. In the Gaussian case, it outperforms both Level Bundle and IPOPT at every tested reliability level, with and without Bonferroni initialization. Level Bundle remains competitive for some values of p, but is consistently slower than Proximal, while IPOPT is substantially more expensive and often fails to attain the prescribed objective tolerance. Bonferroni initialization further strengthens the performance of the Proximal method in the Gaussian case, reducing both runtime and oracle calls.

For the Student-t distribution, where Level Bundle is not applicable, Proximal also consistently outperforms IPOPT in runtime and oracle calls. Although Bonferroni initialization does not uniformly reduce its runtime in this case, Proximal remains the best reported method across all tested reliability levels. Overall, the Ain results identify Proximal as the most robust and efficient method for this benchmark.

We conclude by reporting aggregated performance results across all Gaussian benchmarks. Aggregated results for the t-Student benchmarks are not included, as the comparison in that setting is limited to IPOPT, and our algorithm. As evidenced by the previous tables, our approach significantly outperforms IPOPT,.

We report in Figures 1 and 2 the performance profile (full scale), the zoomed performance profile, and the data profile with respect to both the number of oracle calls ($\#F$) and CPU time, respectively.

4.4 Summary

The aggregate performance and data profiles confirm that the Proximal method is the most robust method overall on the hydro-valley benchmarks. Its advantage should become particularly more dominant as the dimension of the random vector increases, since probability-oracle evaluations become more expensive in this regime. The Proximal method requires solving a more involved master problem than the other reported methods, due to its non-linear property. But the numerical results show that this additional per-iteration cost is well balanced by the reduced number of oracle calls. In particular, the cost of solving the master problems does not dominate the total runtime to the extent that it offsets the oracle savings.

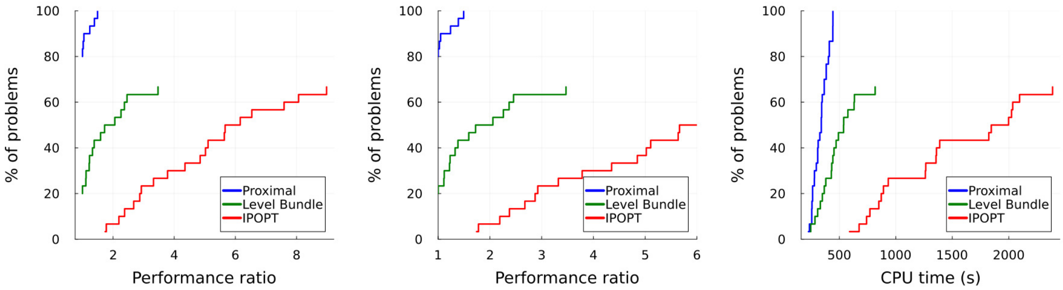

Fig. 2 Gaussian case. Performance profile (full scale), performance profile (zoomed), and data profile with respect to CPU time (seconds).

As a result, Proximal achieves the best aggregate CPU-time profile while remaining the most economical method in terms of oracle evaluations.

In the Gaussian case, Level Bundle remains a strong competitor and clearly outperforms IPOPT, but it is less economical in oracle calls and is restricted to settings where its distributional assumptions are satisfied. For the Student-t instances, where Level Bundle is not applicable, Proximal clearly outperforms IPOPT in both runtime and oracle calls. Overall, the profiles indicate that Proximal provides the best balance between master- problem complexity and oracle-evaluation efficiency across the tested hydro-valley benchmarks.

5 Conclusion

We have proposed a proximal-type method for nonsmooth and nonconvex optimization problems with an explicit composite constraint of the form $g_0(G(x)) \leq 0$. The method exploits the outer–inner structure of the constraint by linearizing the outer function with respect to the composite argument while preserving the structure of the inner mapping. This leads to a sequence of regularized master problems based on an improvement function that balances objective decrease and constraint satisfaction. Under mild assumptions on the objective, the feasible set, and the two components of the composite constraint, we proved that every accumulation point of the serious iterates is critical for the original problem. Moreover, under a suitable constraint qualification, feasible critical points satisfy a generalized KKT condition. The numerical experiments on joint chance-constrained hydro-valley benchmarks show that the proposed Proximal method is a robust and efficient alternative to standard nonlinear programming solvers. In particular, it consistently improves over IPOPT in the Student-t cases and performs competitively in the Gaussian cases, where the Level Bundle method is also applicable. The aggregate performance and data profiles further indicate that the Proximal method is especially economical in oracle calls, making it attractive when probability evaluations are expensive. These results support the practical value of exploiting composite constraint structure directly rather than treating the constraint as a black-box nonsmooth function.

Statements and Declarations

Funding. Y. Chu and W. de Oliveira acknowledge financial support from PGMO (Programme Gaspard Monge pour l'Optimisation) through the project "Linear Chance-Constrained Programming: A Computational Tool".

Competing interests. W. de Oliveira is a member of the Editorial Board of *Computational Optimization and Applications*. The authors declare no other relevant financial or non-financial interests.

Data availability. The benchmark instances used in the numerical experiments were generated computationally from the hydro-reservoir model described in [23], using MATLAB generation code provided by coauthor Wim van Ackooij. The generated benchmark instances and the numerical results supporting the findings of this study are available from the corresponding author upon reasonable request.

Code availability. The numerical algorithms and experiments were implemented by the authors in Julia. A reproducibility version of the code used to obtain the results reported in this article is available from the corresponding author upon reasonable request.

Author contributions. Y. Chu prepared the first draft of the manuscript and carried out all benchmark implementations and numerical experiments. Y. Chu, W. de Oliveira, and P. Pérez-Aros contributed to the theoretical analysis. W. van Ackooij edited all sections after the first round of revisions. All authors read and approved the final manuscript.

References

[1] van Ackooij, W., Chiu, C., de Oliveira, W., Pérez-Aros, P.: Lipschitz Gradient Guarantees for Probability Functions and a New Algorithm for Probability Maximization. Preprint available at Optimization Online (2026). https://optimization-online.org/?p=34983

[2] Higle, J.L., Sen, S.: Statistical approximations for recourse constrained stochastic programs. Annals of Operations Research **56**(1), 157–175 (1995) https://doi.org/10.1007/BF02031705

[3] Nesterov, Y.: Gradient methods for minimizing composite functions. Mathematical programming **140**(1), 125–161 (2013)

[4] Liang, J., Monteiro, R.D.: A unified analysis of a class of proximal bundle methods for solving hybrid convex composite optimization problems. Mathematics of Operations Research **49**(2), 832–855 (2024)

[5] Liang, J., Monteiro, R.D., Zhang, H.: Proximal bundle methods for hybrid weakly convex composite optimization problems. arXiv preprint arXiv:2303.14896 (2023)

[6] Burke, J.V., Ferris, M.C.: A gauss–newton method for convex composite optimization. Mathematical Programming **71**(2), 179–194 (1995)

[7] Lewis, A.S., Wright, S.J.: A proximal method for composite minimization. Mathematical Programming **158**(1), 501–546 (2016)

[8] Davis, D., Drusvyatskiy, D.: Stochastic model-based minimization of weakly convex functions. SIAM Journal on Optimization **29**(1), 207–239 (2019)

[9] Henrion, R., Strugarek, C.: Convexity of chance constraints with independent random variables. Computational Optimization and Applications **41**(2), 263–276 (2008)

[10] Henrion, R., Strugarek, C.: Convexity of chance constraints with dependent random variables: The use of copulae. In: Bertocchi, M., Consigli, G., Dempster, M.A.H. (eds.) Stochastic Optimization Methods in Finance and Energy, pp. 427–439. Springer, New York (2011). https://doi.org/10.1007/978-1-4419-9586-5_17

[11] Andrieu, L., Henrion, R., Römisch, W.: A model for dynamic chance constraints in hydro power reservoir management. European journal of operational research **207**(2), 579–589 (2010)

[12] van Ackooij, W., Henrion, R., Möller, A., Zorgati, R.: Joint chance constrained programming for hydro reservoir management. Optimization and Engineering **15**(2), 509–531 (2014)

[13] van Ackooij, W., de Oliveira, W.: Convexity and optimization with copulae structured probabilistic

constraints. Optimization **65**(7), 1349–1376 (2016) https://doi.org/10.1080/02331934.2016.1179302

[14] Butyn, E., Karas, E.W., de Oliveira, W.: A derivative-free trust-region algorithm with copula-based models for probability maximization problems. European J. Oper. Res. **298**(1), 59–75 (2022) https://doi.org/10.1016/j.ejor.2021.09.040

[15] van Ackooij, W., Demassey, S., Javal, P., Morais, H., de Oliveira, W., Swaminathan, B.: A bundle method for nonsmooth dc programming with application to chance-constrained problems. Computational Optimization and Applications **78**(2), 451–490 (2021) https://doi.org/10.1007/s10589-020-00241-8

[16] Cui, Y., He, Z., Pang, J.-S.: Nonconvex robust programming via value-function optimization. Computational Optimization and Applications **78**(2), 411–450 (2021) https://doi.org/10.1007/s10589-020-00245-4

[17] Syrtseva, K., de Oliveira, W., Demassey, S., Morais, H., Javal, P., Swaminathan, B.: Difference-of-convex approach to chance-constrained optimal power flow modelling the dso power modulation lever for distribution networks. Sustainable Energy, Grids and Networks **35**, 101168 (2023) https://doi.org/10.1016/j.segan.2023.101168

[18] Levy, A.B., Mordukhovich, B.S.: Coderivatives in parametric optimization. Mathematical programming **99**(2), 311–327 (2004)

[19] Hellemo, L., Tomasgard, A.: A generalized global optimization formulation of the pooling problem with processing facilities and composite quality constraints. Top **24**(2), 409–444 (2016)

[20] Sempere, G.M., Oliveira, W., Royset, J.O.: A proximal-type method for nonsmooth and nonconvex constrained minimization problems. Journal of Optimization Theory and Applications **204**(3), 54 (2025)

[21] van Ackooij, W.S., de Oliveira, W.L.: Methods of Nonsmooth Optimization in Stochastic Programming: From Conceptual Algorithms to Real-World Applications. International Series in Operations Research & Management Science, vol. 363. Springer, Cham (2025). https://doi.org/10.1007/978-3-031-84837-7

[22] Aragón-Artacho, F.J., Campoy, R., Pérez-Aros, P., Torregrosa-Belén, D.: Nonmonotone subgradient methods based on a local descent lemma (2026). https://arxiv.org/abs/2510.19341

[23] van Ackooij, W., de Oliveira, W.: Level bundle methods for constrained convex optimization with various oracles. Computational Optimization and Applications **57**(3), 555–597 (2014)

[24] Dontchev, A.L., Rockafellar, R.T.: Implicit Functions and Solution Mappings. Springer Monographs in Mathematics. Springer, New York (2009). https://doi.org/10.1007/978-0-387-87821-8

[25] Durante, F., Sempi, C.: Principles of Copula Theory. CRC Press, Boca Raton, FL (2016)

[26] van Ackooij, W., Malick, J.: Eventual convexity of probability constraints with elliptical distributions. Mathematical Programming **175**(1), 1–27 (2019)

[27] van Ackooij, W., Henrion, R.: Gradient formulae for nonlinear probabilistic constraints with gaussian and gaussian-like distributions. SIAM Journal on Optimization **24**(4), 1864–1889 (2014)

[28] van Ackooij, W., Aleksovska, I., Munoz-Zuniga, M.: (sub-) differentiability of probability functions with elliptical distributions. Set-Valued and Variational Analysis **26**(4), 887–910 (2018)

[29] Niederreiter, H.: Random Number Generation and Quasi-Monte-Carlo Methods. Society for Industrial and Applied Mathematics, Philadelphia, PA. (1992)

[30] Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Mathematical programming **91**(2), 201–213 (2002)

[31] Moré, J.J., Wild, S.M.: Benchmarking derivative-free optimization algorithms. SIAM Journal on Optimization **20**(1), 172–191 (2009)