View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by INRIA a CCSD electronic archive server

Chance constrained problems: a bilevel convex optimization perspective

Yassine Laguel, Jérôme Malick, Wim Ackooij

To cite this version:

Yassine Laguel, Jérôme Malick, Wim Ackooij. Chance constrained problems: a bilevel convex optimization perspective. 2021. hal-03174589

HAL Id: hal-03174589

https://hal.archives-ouvertes.fr/hal-03174589

Preprint submitted on 19 Mar 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Chance constrained problems: a bilevel convex optimization perspective

Yassine Laguel[1] Jérôme Malick[1] Wim Van Ackooij[2]
[1]Univ. Grenoble Alpes, CNRS, LJK, 38000 Grenoble, France
[2]EDF R&D, Saclay, France

Abstract

Chance constraints are a valuable tool for the design of safe decisions in uncertain environments; they are used to model satisfaction of a constraint with a target probability. However, because of possible non-convexity and non-smoothness, optimizing over a chance constrained set is challenging. In this paper, we establish an exact reformulation of chance constrained problems as a bilevel problems with convex lower-levels. We then derive a tractable penalty approach, where the penalized objective is a difference-of-convex function that we minimize with a suitable bundle algorithm. We release an easy-to-use open-source python toolbox implementing the approach, with a special emphasis on fast computational subroutines.

Keywords Stochastic programming · Chance constraints · Bi-level optimization · DC programming

1 Introduction

Chance constraints appear as a versatile way to model the exposure to uncertainty in optimization. Introduced in [4], they have been used in many applications, such as in energy management [24, 30], in telecommunications [19] or for reinforcement learning [5], to name of few of them. We refer to the seminal paper [23], the book chapter [8] for introduction to the theory and to the recent article [29] for a discussion covering recent developments.

In this paper, we consider a general chance-constrained optimization problem of the following form. For a fixed safety probability level $p \in [0, 1)$, we write:

$$\begin{aligned} &\min_{x \in \mathcal{X}} \quad f(x) \\ &\text{s.t.} \quad \mathbb{P}[g(x, \xi) \leq 0] \geq p, \end{aligned} \tag{1}$$

where $f\colon \mathbb{R}^{\mathrm{d}} \to \mathbb{R}$ and $g\colon \mathbb{R}^{\mathrm{d}} \times \mathbb{R}^{\mathrm{m}} \to \mathbb{R}$ are two given functions, ξ is a random vector valued in $\mathbb{R}^{\mathrm{m}}$ and $\mathcal{X} \subset \mathbb{R}^{\mathrm{d}}$ is a (deterministic) closed constraint set.

We consider the case of underlying convexity: We assume that f and g are convex (with respect to x). For our practical developments, we also assume that we have first-order oracles for f and g and that the $\mathcal{X}$ is a box constraint on the decision variable x. Even with underlying convexity, modeling uncertainty may make the chance constraint feasible set non-convex (see e.g. [11] for discussion on possible convexity when p is close to 1). Though solving (non-convex) chance-constrained problems is difficult, several computational methods have been proposed, regardless of any considerations of convexity and smoothness, and under various assumptions on uncertainty. Let us mention: sample average approximation [17, 22], scenario approximation [3], convex approximation [20], or p-efficient points [9]; see e.g. [29] for an overview.

In this paper, we propose an original approach for solving chance-constrained optimization problems. First, we present an exact reformulation of (nonconvex) chance-constrained problems as (convex) bilevel optimization problems. This reformulation is simple and natural, involving superquantiles (also called conditional vale-at-risk), a risk measure studied by T. Rockafellar and his co-authors; see e.g., the tutorial [25]. Second, exploiting this bilevel reformulation, we propose a general algorithm for solving chance-constrained problems, and we release an open-source python toolbox implementing it. In the case where we make no assumption on the underlying uncertainty and have only samples of ξ, we propose and analyse a double penalization method, leading to an unconstrained single level DC (Difference of Convex) programs. Our approach enables to deal with a fairly large sample of data-points in comparison with state-of-the-art methods based on mixed-integer reformulations, e.g. [1]. Thus our work mixes a variety of techniques coming from different subdomains of optimization: penalization, error bounds, DC programming, bundle algorithm, Nesterov's smoothing; relevant references are given along the discussion.

This paper is structured as follows. In Section 2, we leverage the known link with (super)quantiles and chance-constraint to establish a novel bilevel reformulation of general chance constrained problems. In Section 3, we propose and analyse a penalty approach revealing the underlying DC structure. In Section 4, we discuss implementation of this approach in our publicly available toolbox. In section 5, we provide illustrative numerical experiments, as a proof of concept, showing the interest of the method. Technical details on secondary theoretical points and on implementation issues are postponed to appendices.

2 Chance constrained problems seen as bilevel problems

In this section, we derive the reformulation of a chance constraint as a bilevel program wherein both the upper and lower level problems, when taken individually, are convex. We first recall in Section 2.1 useful definitions. Our terminology and notations closely follow those of [25].

2.1 Basics: cumulative distributions functions, quantiles, and superquantiles

In what follows, we consider a probability space and integrable real random variables. Given a random variable X, its *cumulative distribution function*, denoted by $F_{\mathrm{X}} : \mathbb{R} \to [0, 1]$, is defined as:

$$F_{\mathrm{X}}(t) := \mathbb{P}[\mathrm{X} \leq t] \quad \forall t \in \mathbb{R}\,. \tag{2}$$

The cumulative distribution function is known to be both non-decreasing and right-continuous. Its jumps occur exactly at the atoms of X, that is the values $t \in \mathbb{R}$ at which $\mathbb{P}[\mathrm{X} = t] > 0$. These properties enable one to define the *quantile function* $p \mapsto Q_p(\mathrm{X})$ as the following generalized inverse:

$$Q_p(\mathrm{X}) = \inf\{t \in \mathbb{R} : \; F_{\mathrm{X}}(t) \geq p\}, \quad \forall p \in [0, 1). \tag{3}$$

If X is assumed to belong to $\mathcal{L}^1$, we can additionally define for any $p \in [0, 1)$ its *p-superquantile*, $\bar{Q}_p(\mathrm{X})$ as follows:

$$\bar{Q}_p(\mathrm{X}) = \frac{1}{1-p} \int_{p'=p}^{1} Q_{p'}(\mathrm{X}) dp'. \tag{4}$$

For a given random variable X, as a consequence of [26, Th. 2], one can recover from the cumulative distribution function F_{X} both the p-quantile and the p-superquantile functions as functions of p and reciprocally, knowing either the p-quantile or the p-superquantile for all $p \in [0, 1]$ suffices to recover F_{X}.

From a statistical viewpoint, these three notions are also equally consistent [25, Th. 4] in the sense that convergence in distribution for a sequence of random variables $(\mathrm{X}_n)_{n\geq 0}$ is equivalent to the pointwise convergence of the two sequences of functions $p \mapsto Q_p(\mathrm{X}_n)$, $p \mapsto \bar{Q}_p(\mathrm{X}_n)$. This result is particularly relevant when the distributions are observed through data sampling. We can use the empirical cumulative distribution functions, quantiles and super-quantiles all while upholding asymptotic convergence as the sample size grows.

From an optimization point of view though, these three objects are very different. In contrast with the others, the superquantile has several good properties (including convexity [2, 10, 28]), useful with respect to numerical computation and optimization. In our developments, we use the following key result [27, Th. 1] linking quantiles and superquantile through a one-dimensional problem.

Lemma 1. *For an integrable random variable* X *and a probability level* p*, the superquantile* $\bar{Q}_p(\mathrm{X}))$ *and quantile* $Q_p(\mathrm{X})$ *are respectively the optimal value and the optimal solution of the convex one-dimensional problem*

$$\inf_{\eta\in\mathbb{R}} \ \eta + \frac{1}{1-p}\,\mathbb{E}[\max(\mathrm{X}-\eta, 0)]. \tag{5}$$

2.2 Reformulation as a bilevel problems

By definition, the chance constraint in (1) involves the cumulative distribution function: we have for any fixed $x \in \mathbb{R}^{\mathrm{d}}$, $\mathbb{P}[g(x,\xi) \leq 0] \geq p \Leftrightarrow F_{g(x,\xi)}(0) \geq p$. Following the discussion of the previous section, we easily rewrite this constraint using quantiles, as formalized in the next lemma.

Lemma 2. *For any* $x \in \mathbb{R}^{\mathrm{d}}$ *and* $p \in [0,1)$*, we have:*

$$\mathbb{P}[g(x,\xi) \leq 0] \geq p \iff Q_p(g(x,\xi)) \leq 0.$$

Proof. By definition of the quantile and continuity on the right of the cumulative distribution function, we always have $p \leq \mathbb{P}[g(x,\xi) \leq Q_p(g(x,\xi))]$. Thus, since cumulative distribution functions are increasing, if $Q_p(g(x,\xi)) \leq 0$, then $\mathbb{P}[g(x,\xi) \leq Q_p(g(x,\xi))] \leq \mathbb{P}[g(x,\xi) \leq 0]$ which implies that $\mathbb{P}[g(x,\xi) \leq 0] \geq p$. Conversely, since $Q_p(g(x,\xi))$ is the infimum of $\{t \in \mathbb{R} \ : \mathbb{P}[X \leq t] \geq p\}$, if $Q_p(g(x,\xi)) > 0$, then necessarily we have $\mathbb{P}[g(x,\xi) \leq 0] < p$. □

Together with (5), we obtain from the previous easy lemma a bilevel formulation of the general chance-constrained problem (7). The idea is simple: introducing an auxiliary variable $\eta \in \mathbb{R}^{\mathrm{d}}$ to recast the potentially non-convex chance constraint of (1) as two constraints, a simple bound constraint and a difficult optimality constraint, forming a lower subproblem. Introducing the lower objective function $G : \mathcal{X} \times \mathbb{R} \to \mathbb{R}$

$$G(x,s) = s + \frac{1}{1-p}\,\mathbb{E}[\max(g(x,\xi)-s, 0)], \tag{6}$$

we have the following exact reformulation of chance-constrained problems.

Theorem 3. *Problem* (1) *is equivalent to the bilevel problem:*

$$\begin{cases} \min_{x\in\mathcal{X},\eta\in\mathbb{R}} & f(x) \\ \textit{s.t.} & \eta \leq 0 \\ & \eta \in S(x) = \arg\min_{s\in\mathbb{R}} G(x,s). \end{cases} \tag{7}$$

More precisely, if $x^\star$ *is an optimal solution of* (1)*, then* $(x^\star, Q_p(g(x^\star,\xi)))$ *is an optimal solution of the above bilevel problem, and conversely.*

Proof. It is clear with Lemma 2 that problem (1) is equivalent to

$$\begin{cases} \min\limits_{x\in\mathcal{X},\eta\in\mathbb{R}} & f(x) \\ \text{s.t.} & \eta \leq 0 \\ & \eta = Q_p(g(x,\xi)) \end{cases} . \tag{8}$$

By Lemma 1, $Q_p(g(x,\xi)) \in S(x)$ for any $x \in \mathbb{R}^d$. Hence, any solution (x,η) of (8) is feasible for (7). Conversely, any solution $(x^\star,\eta^\star)$ of (7) satisfies: $Q_p(g(x^\star,\xi)) \leq \eta^\star \leq 0$ which implies that $(x^\star, Q_p(g(x^\star,\xi))$ is a feasible point of (8). Since both problems have the same objective, they are equivalent. □

The first constraint $\eta \leq 0$ is an easy one-dimensional bound constraint which does not involve the decision variable x. The second constraint, which constitutes the lower level problem is more difficult; when this constraint is satisfied, η is exactly the p-quantile of $g(x,\xi)$. We readily see the joint convexity of the objective function of the lower level problem in (7) with respect to s and x.

This bilevel reformulation is nice, natural and seemingly new; we believe that it opens the door to new approaches for solving chance-constrained problems. In the next section, we propose such an approach based on the reformulation.

3 A double penalization scheme for chance constrained problems

In this section, we explore one possibility offered by the bilevel formulation of chance-constrained problems, presented in the previous section. We propose a (double) penalization approach for solving the bilevel optimization problem, with a different treatment of the two constraints: a basic penalization of the easy constraint together with an exact penalization of the hard constraint formalized as the lower problem. We first derive in Section 3.1, some growth properties of the lower problem. We show then in Section 3.2 to what extent these properties help to provide an exact penalization of the "hard" constraint. We finally present the double penalty scheme in section 3.3.

From the bilevel problem (7), we derive the two following penalized problems, associated with two penalization parameters $\mu, \lambda > 0$ and

$$(P_\mu) \quad \begin{cases} \min\limits_{(x,\eta)\in\mathcal{X}\times\mathbb{R}} & f(x) + \mu \max(\eta, 0) \\ \text{s.t.} & \eta \in \arg\min_{s\in\mathbb{R}} G(x,s) \end{cases} \tag{9}$$

and

$$(P_{\lambda,\mu}) \quad \min_{(x,\eta)\in\mathcal{X}\times\mathbb{R}} f(x) + \lambda\left(G(x,\eta) - \min_{s\in\mathbb{R}} G(x,s)\right) + \mu\max(\eta,0). \tag{10}$$

We consider a general data-driven situation where the uncertainty ξ is just known through a sample (or, said alternatively, follows an equiprobable discrete distribution over $n \in \mathbb{N}$ arbitrary values): we assume that there exists $\xi_1, \xi_2, \ldots, \xi_n \in \mathbb{R}^{\mathrm{m}}$ such that $\mathbb{P}[\xi = \xi_i] = \frac{1}{n}$ for all $i \in \{1, \ldots, n\}$. The set $\mathcal{I}_n$ defined as

$$\mathcal{I}_n = \left\{ \frac{i}{n}, \quad i \in \{0, ..., n-1\} \right\} \tag{11}$$

plays a special role in our developments. In particular, we use the distance to $\mathcal{I}_n$, denoted by $d_{\mathcal{I}_n}(p)$, to define a key quantity appearing in the variational results of this section: we introduce

$$\delta = \begin{cases} \dfrac{1}{n(1-p)} & \text{if } p \in \mathcal{I}_n \\ \dfrac{d_{\mathcal{I}_n}(p)}{(1-p)} & \text{otherwise,} \end{cases} \tag{12}$$

which depends implicitly on the number of samples n and the fixed safety parameter p.

3.1 Analysis of the Value function

In view of the forthcoming exact penalization, we study here the value function $h : \mathcal{X} \times \mathbb{R} \to \mathbb{R}$ defined, from G in (6), as

$$h(x, \eta) = G(x, \eta) - \min_{s \in \mathbb{R}} G(x, s). \tag{13}$$

The next result relates h to $d_{S(x)}(\cdot)$, the distance function to $S(x)$, the solution set of the lower level problems in (7). This is our main technical result, on which next propositions are based.

Theorem 4. *Let $p \in [0, 1)$ be fixed but arbitrary. The function h defined in (13) satisfies for any $(x, \eta) \in \mathcal{X} \times \mathbb{R}$*

$$h(x, \eta) \geq \delta \; d_{S(x)}(\eta) \qquad \textit{with } \delta \textit{ defined by (12).}$$

Proof. Let us fix $x \in \mathcal{X}$ and denote by q_p the p-quantile of $g(x, \xi)$. We first note that by the arguments in the proof of Lemma 2, we have

$$p \leq \mathbb{P}[g(x, \xi) \leq q_p],$$

with equality holding in the left inequality, if and only if p belongs to $\mathcal{I}_n$.

For any fixed but arbitrary $\eta \in \mathbb{R}$, we have the following identity:

$$\begin{aligned} h(x, \eta) &= \eta + \frac{1}{1-p}\mathbb{E}[\max(g(x, \xi) - \eta, 0)] - \left(q_p + \frac{1}{1-p}\mathbb{E}[\max(g(x, \xi) - q_p, 0)]\right) \\ &= (\eta - q_p) + \frac{1}{1-p}\mathbb{E}\left[\max(g(x, \xi), \eta) - \eta - (\max(g(x, \xi), q_p) - q_p)\right] \\ &= (\eta - q_p)(1 - \frac{1}{1-p}) + \frac{1}{1-p}\mathbb{E}\left[\max(g(x, \xi), \eta) - \max(g(x, \xi), q_p)\right]. \end{aligned}$$

Now, by employing a case distinction on the location of η with respect to q_p, we will derive the desired inequalities. Let us first consider that $\eta > q_p$, then we have:

$$\begin{aligned} h(x, \eta) &= (\eta - q_p)(1 - \frac{1}{1-p}) + \frac{1}{1-p}\mathbb{E}\left[(\eta - g(x, \xi))\, \mathbb{1}_{q_p < g(x,\xi) \leq \eta} + (\eta - q_p)\, \mathbb{1}_{g(x,\xi) \leq q_p}\right] \\ &= (\eta - q_p)(1 - \frac{1}{1-p} + \frac{1}{1-p}\,\mathbb{P}[g(x, \xi) \leq q_p]) + \frac{1}{1-p}\,\mathbb{E}[(\eta - g(x, \xi))\, \mathbb{1}_{q_p < g(x,\xi) \leq \eta}] \end{aligned}$$

which finally gives:

$$h(x, \eta) = \frac{(\eta - q_p)}{1-p}\left(\mathbb{P}[g(x, \xi) \leq q_p] - p + \mathbb{E}\left[\frac{\eta - g(x, \xi)}{\eta - q}\, \mathbb{1}_{q_p < g(x,\xi) \leq \eta}\right]\right). \tag{14}$$

Now if $\mathbb{P}[g(x,\xi) \leq q_p] > p$, this implying that $p \notin \mathcal{I}_n$, then by non-negativity of the expectation term above, we have:

$$h(x,\eta) \geq (\eta - q_p)\frac{1}{1-p}\left(\mathbb{P}[g(x,\xi) \leq q_p] - p\right) \geq \frac{d_{\mathcal{I}_n}(p)}{1-p}\, d_{S(x)}(\eta).$$

Here we use that clearly $|\eta - q_p| \geq d_{S(x)}(\eta)$, since $q_p \in S(x)$ as already recalled. Furthermore $p \leq \mathbb{P}[g(x,\xi) \leq q_p] < p + \frac{1}{n}$ by Lemma 2 and by definition. We also observe that $\mathbb{P}[g(x,\xi) \leq q_p] \in \mathcal{I}_n$, so that altogether we have:

$$\begin{aligned} 0 &\leq (\mathbb{P}[g(x,\xi) \leq q_p] - p) \\ (\mathbb{P}[g(x,\xi) \leq q_p] - p) &\geq d_{\mathcal{I}_n}(p) \\ d_{\mathcal{I}_n}(p) &\leq \tfrac{1}{n}\ . \end{aligned}$$

If to the contrary $p \in \mathcal{I}_n$ which implies that $\mathbb{P}[g(x,\xi) \leq q_p] = p$, then $h(x,\eta) = \frac{1}{1-p}\,\mathbb{E}[(\eta - g(x,\xi))\,\mathbb{1}_{q_p < g(x,\xi) \leq \eta}]$. We let q_p^+ be the successor quantile, i.e.,

$$q_p^+ = \inf\{t \geq \mathbb{R} :\ \mathbb{P}[g(x,\xi) \leq t] > p\}.$$

Since $p \leq \frac{n-1}{n}$, it follows that $q_p^+ < \infty$ and $q_p^+ > q_p$. Now if $\eta \in (q_p, q_p^+)$, we have $h(x,\eta) = 0$. If $\eta \geq q_p^+$, then

$$\begin{aligned} h(x,\eta) &= \frac{1}{1-p}\,\mathbb{E}\left[(\eta - g(x,\xi))\,\mathbb{1}_{q_p^+ \leq g(x,\xi) \leq \eta}\right] \\ &\geq (\eta - q_p^+)\frac{\mathbb{P}[g(x,\xi) = q_p^+]}{1-p} \geq \frac{1}{n(1-p)} d_{S(x)}(\eta) \\ &\geq \frac{d_{\mathcal{I}_n}(p)}{(1-p)} d_{S(x)}(\eta), \end{aligned}$$

where the last inequality results from our earlier estimates.

The second case to consider involves the situation $\eta < q_p$. Here, we have:

$$\begin{aligned} h(x,\eta) &= (\eta - q_p)(1 - \frac{1}{1-p}) + \frac{1}{1-p}\,\mathbb{E}\left[(g(x,\xi) - q_p)\,\mathbb{1}_{\eta < g(x,\xi) \leq q_p} + (\eta - q_p)\,\mathbb{1}_{g(x,\xi) \leq \eta}\right] \\ &= (\eta - q_p)\left(1 - \frac{1}{1-p} + \frac{1}{1-p}\,\mathbb{P}[g(x,\xi) \leq \eta]\right) + \frac{1}{1-p}\,\mathbb{E}\left[(g(x,\xi) - q_p)\,\mathbb{1}_{\eta < g(x,\xi) \leq q_p}\right] \end{aligned}$$

which leads us to

$$h(x,\eta) = \frac{(q_p - \eta)}{1-p}\left(p - \mathbb{P}[g(x,\xi) \leq \eta] - \mathbb{E}\left[\frac{q_p - g(x,\xi)}{q_p - \eta}\mathbb{1}_{\eta < g(x,\xi) \leq q_p}\right]\right). \tag{15}$$

Now if $\mathbb{P}[g(x,\xi) \leq q_p] > p$, this implies that $p \notin \mathcal{I}_n$, then let us define the antecessor quantile q_p^- as

$$q_p^- = \max\left\{\sup\{t \geq \mathbb{R} :\ \mathbb{P}[g(x,\xi) \leq t] < p\}, \min\{g(x,\xi_i)\}_{i=1}^n - 1\right\}.$$

We can first observe that since $p \notin \mathcal{I}_n$, we can entail $p > 0$, hence $q_p^- > -\infty$ is well defined. For any $\eta \in (q_p^-, q_p)$, it follows that $\mathbb{E}\left[\frac{q_p - g(x,\xi)}{q_p - \eta}\,\mathbb{1}_{\eta < g(x,\xi) \leq q_p}\right] = 0$. We may thus consider that $\eta \leq q_p^-$, in which

case we have:

$$
\begin{aligned}
h(x,\eta) &= (q_p-\eta)\frac{1}{1-p}\left(p-\mathbb{P}[g(x,\xi)\le\eta]-\mathbb{E}\left[\frac{q_p-g(x,\xi)}{q_p-\eta}\,\mathbb{1}_{\eta<g(x,\xi)\le q_p^-}\right]\right)\\
&\ge (q_p-\eta)\frac{1}{1-p}\left(p-\mathbb{P}[g(x,\xi)\le\eta]-\mathbb{E}\left[\mathbb{1}_{\eta<g(x,\xi)\le q_p^-}\right]\right)\\
&\ge (q_p-\eta)\frac{1}{1-p}\left(p-\mathbb{P}[g(x,\xi)\le q_p^-]\right)\\
&\ge \frac{1}{n(1-p)}\,d_{S(x)}(\eta)\ge\frac{d_{\mathcal{I}_n}(p)}{1-p}\,d_{S(x)}(\eta),
\end{aligned}
$$

where we have used that $\frac{q_p-g(x,\xi)}{q_p-\eta}\le 1$ on $\mathbb{1}_{\eta<g(x,\xi)\le q_p}$.
If to the contrary, $p\in\mathcal{I}_n$, which implies $\mathbb{P}[g(x,\xi)\le q_p]=p$, recalling the identity $p-\mathbb{P}[g(x,\xi)\le\eta]=\mathbb{E}[\mathbb{1}_{\eta<g(x,\xi)<q_p}]+\mathbb{P}[g(x,\xi)=q_p]$, we obtain:

$$
\begin{aligned}
h(x,\eta) &= (q_p-\eta)\frac{1}{1-p}\left(\mathbb{E}\left[1-\frac{(q_p-g(x,\xi))}{q_p-\eta}\,\mathbb{1}_{\eta<g(x,\xi)<q_p}\right]+\mathbb{P}[g(x,\xi)=q_p]\right)\\
&\ge (q_p-\eta)\frac{\mathbb{P}[g(x,\xi)=q_p]}{1-p}\\
&\ge \frac{1}{n}\frac{1}{1-p}d_{S(x)}(\eta)\ge\frac{d_{\mathcal{I}_n}(p)}{1-p}\,d_{S(x)}(\eta),
\end{aligned}
$$

where we have used that $\mathbb{P}[g(x,\xi)=q_p]=\frac{1}{n}$. The last case $\eta=q_p$, gives by construction $\eta\in S(x)$, i.e., $d_{S(x)}(\eta)=0$ and clearly $h(x,\eta)=0$ so that the desired inequality holds. □

Following the terminology of [33], this theorem shows that h is a uniform parametric error bound. We note that the quality of this bound is altered by the number n of data points considered. This drawback actually passes to the limit in the sense that $(x,\eta)\mapsto h_x(\eta)$ fails to be a uniform parametric error bound when ξ follows a continuous distribution; this is an interesting but secondary result that we prove in Appendix A.1.

3.2 An exact penalization for the hard constraint

We show here that $(P_{\lambda,\mu})$ is an exact penalization of (P_μ), when λ is large enough. The proof of this result follows usual rationale (see e.g., [6, Prop. 2.4.3]); the main technicality is the sharp growth of h established in Theorem 4.

Proposition 5. *Let $\mu>0$ be given and assume that there is a solution to (P_μ) defined in (9). Then for any $\lambda>\mu/\delta$ with δ defined in (12), the solution set of (P_μ) coincides with the one of $(P_{\lambda,\mu})$ defined in (10).*

Proof. Take $\mu>0$, define $\lambda_\mu=\mu/\delta$, and take $\lambda>\lambda_\mu$ arbitrary but fixed. Let us first take a solution $(x^\star,\eta^\star)\in\mathcal{X}\times\mathbb{R}$ of (P_μ) and show by contradiction that it is also a solution of $(P_{\lambda,\mu})$. Indeed, to the contrary, assume there exists some $\varepsilon>0$ and $(x',\eta')\in\mathcal{X}\times\mathbb{R}$ such that:

$$
f(x')+\mu\max(0,\eta')+\lambda h_{x'}(\eta')\le f(x^\star)+\mu\max(0,\eta^\star)+\lambda\,h_{x^\star}(\eta^\star)-\varepsilon.
$$

Let then $\eta'_p \in S(x')$ be such that : $|\eta'_p - \eta'| \leq d_{S(x')}(\eta') + \frac{\varepsilon}{2\mu}$. Then the point (x', η'_p) is a feasible for P_μ (recall $\eta'_p \in S(x')$) and since $\eta \mapsto \mu \max(0, \eta)$ is μ-Lipschitz, we first have

$$\begin{aligned} f(x') + \mu \max(0, \eta'_p) &\leq f(x') + \mu \max(\eta', 0) + \mu |\eta'_p - \eta'| \\ &\leq f(x') + \mu \max(\eta', 0) + \mu \left(d_{S(x')}(\eta') + \frac{\varepsilon}{2\mu} \right). \end{aligned}$$

Using Theorem 4, we then have

$$\begin{aligned} f(x') + \mu \max(0, \eta'_p) &\leq f(x') + \mu \max(\eta', 0) + \mu \left(\frac{1}{\delta} h(x', \eta') + \frac{\varepsilon}{2\mu} \right) \\ &\leq f(x') + \mu \max(\eta', 0) + \lambda_\mu \, h(x', \eta') + \frac{\varepsilon}{2} \\ &\leq f(x^\star) + \mu \max(\eta^\star, 0) - \frac{\varepsilon}{2} \end{aligned}$$

which gives the contradiction. Hence any solution of (P_μ) is also a solution to problem $(P_{\lambda,\mu})$.

Let now $(\bar{x}, \bar{\eta})$ be a solution of $(P_{\lambda,\mu})$ and let us show that it is actually a solution for P_μ. Let again $(x^\star, \eta^\star)$ be an arbitrary solution of (P_μ). We first note that that a result of optimality of $(\bar{x}, \bar{\eta})$ for $(P_{\lambda,\mu})$, we have:

$$f(\bar{x}) + \mu \max(0, \bar{\eta}) + \lambda \underbrace{h(\bar{x}, \bar{\eta})}_{\geq 0} \leq f(x^\star) + \mu \max(0, \eta^\star) + \lambda \underbrace{h(x^\star, \eta^\star)}_{=0},$$

which by positivity of the function h and feasibility for (P_μ), i.e., $h(x^\star, \eta^\star) = 0$ of $(x^\star, \eta^\star)$ yields:

$$f(\bar{x}) + \mu \max(0, \bar{\eta}) \leq f(x^\star) + \mu \max(0, \eta^\star).$$

It remains to show that $(\bar{x}, \bar{\eta})$ is a feasible point for (P_μ). By the first point, $(x^\star, \eta^\star)$ is both a solution of $(P_{\lambda,\mu})$ and $(P_{\frac{\lambda+\lambda_\mu}{2},\mu})$. Hence, we have:

$$\begin{aligned} f(\bar{x}) + \mu \max(0, \bar{\eta}) + \lambda h(\bar{x}, \bar{\eta}) &\leq f(x^\star) + \mu \max(0, \eta^\star) \\ &= f(x^\star) + \mu \max(0, \eta^\star) + \frac{\lambda + \lambda_\mu}{2} h(x^\star, \eta^\star) \\ &\leq f(\bar{x}) + \mu \max(0, \bar{\eta}) + \frac{\lambda + \lambda_\mu}{2} h(\bar{x}, \bar{\eta}) \end{aligned}$$

But since $\lambda > \lambda_\mu$ we necessarily have: $h(\bar{x}, \bar{\eta}) = 0$ which implies by the properties of the value function that $(\bar{x}, \bar{\eta})$ is a feasible point for (P_μ). □

3.3 Double penalization scheme

From the previous results, we get that solving the sequence of penalized problems gives approximations of the solution of the initial problem. We formalize this in the next proposition suited for our context of double penalization. The proof of this result follows standard arguments; see e.g. [18, Ch. 13.1].

Proposition 6. *Assume that Problem (7) has a non-empty feasible set. Let $(\mu_k)_{k\geq 0}$ be an increasing sequence such that $\mu_k \nearrow \infty$, and $(\lambda_k)_{k\geq 0}$ be taken such that $\lambda_k > \frac{\mu_k}{\delta}$ with δ as defined in (12). If, for all k, there exists a solution of (P_{λ_k,μ_k}) (denoted by (x_k, η_k)), then any cluster point of the sequence (x_k, η_k) is an optimal solution of (1).*

Proof. The fact that (x_k, η_k) is an optimal solution of (P_{λ_k,μ_k}) implies that

$$\begin{aligned} f(x_k) + \mu_k \max(0, \eta_k) + \lambda_k h(x_k, \eta_k) \\ \leq f(x_{k+1}) + \mu_k \max(0, \eta_{k+1}) + \lambda_k h(x_{k+1}, \eta_{k+1}) \end{aligned} \tag{16}$$

Similarly for (x_{k+1}, η_{k+1}), we get

$$\begin{aligned} f(x_{k+1}) + \mu_{k+1} \max(0, \eta_{k+1}) + \lambda_{k+1} h(x_{k+1}, \eta_{k+1}) \\ \leq f(x_k) + \mu_{k+1} \max(0, \eta_k) + \lambda_{k+1} h(x_k, \eta_k). \end{aligned}$$

By Proposition 5, η_k (resp. η_{k+1}) is feasible for (P_{μ_k}) (resp. $(P_{\mu_{k+1}})$); in other words, we have $h(x_k, \eta_k) = h(x_{k+1}, \eta_{k+1}) = 0$. Hence summing up these two inequalities yields

$$\max(\eta_k, 0) \geq \max(\eta_{k+1}, 0).$$

Using this last inequality with (16) gives:

$$f(x_k) - f(x_{k+1}) \leq \mu_k \left(\max(\eta_{k+1}, 0) - \max(\eta_k, 0)\right) \leq 0,$$

and as a consequence the sequence $\{f(x_k)\}_{k\geq 0}$ increases. Let (x', η') be an arbitrary feasible solution for (P). By definition of the sequence (x_k, η_k), for any $k \in \mathbb{N}$, we have:

$$f(x_k) \leq f(x_k) + \mu_k \max(\eta_k, 0) \leq f(x') + \mu_k \max(\eta', 0) \leq f(x'). \tag{17}$$

Therefore for any cluster point $(\bar{x}, \bar{\eta})$ of the sequence $\{(x_k, \eta_k)\}_{k\geq 0}$, we have $f(\bar{x}) \leq f(x')$. In order to show that $(\bar{x}, \bar{\eta})$ is a solution of (7), it remains to show its feasibility. With the right hand side inequality of (17), we obtain

$$\max(\eta_k, 0) \leq \frac{f(x') - f(x_k)}{\mu_k} \leq \frac{f(x') - f(x_0)}{\mu_k} \xrightarrow[k\to\infty]{} 0,$$

so that we may deduce that, $\bar{\eta} \leq 0$. Moreover, continuity of h ensures that $h(\bar{x}, \bar{\eta}) = 0$ which completes the proof. □

In words, cluster points of a sequence of solutions obtained as μ grows to $+\infty$ are feasible solutions of the initial chance-constrained problem. In practice though, we have observed that taking a fixed μ is enough for reaching good approximations of the solution with increasing λ's; see in particular the numerical experiments of Section 5. In the next section, we discuss further the practical implementation of the conceptual double penalization scheme.

4 Double penalization in practice

In this section, we propose a practical version of the double penalization scheme for solving chance-constrained optimization problems. First, we present in Section 4.1 how to tackle the inner penalized problem $(P_{\lambda,\mu})$ by leveraging its difference-of-convex (DC) structure. Then we quickly describe, in Section 4.2, the python toolbox that we release, implementing this bundle algorithm and efficient oracles within the double penalization method.

4.1 Solving penalized problems by a bundle algorithm

We discuss here an algorithm for solving $(P_{\lambda,\mu})$ by revealing the DC structure of the objective function. Notice indeed that, introducing the two convex functions

$$\varphi_1(x,\eta) = f(x) + \lambda G(x,\eta) + \mu \max(\eta, 0) \quad \text{and} \quad \varphi_2(x,\eta) = \lambda \min_{s\in\mathbb{R}} G(x,s)$$

we can write $(P_{\lambda,\mu})$ as the DC problem

$$\min_{(x,\eta)\in\mathcal{X}\times\mathbb{R}} \varphi(x,\eta) = \varphi_1(x,\eta) - \varphi_2(x,\eta). \tag{18}$$

We then propose to solve this problem by the bundle algorithm of [7], which showed to be a method of choice for DC problems. This bundle algorithm interacts with first-order oracles for φ_1 and φ_2; in our situation, there exist computational procedures to compute subgradients of φ_1 and φ_2 from output of oracles of f and g, as formalized in the next proposition. The proof of this proposition is deferred to Appendix B. Note that at the price of more heavy expressions, we could derive the whole subdifferential.

Proposition 7. *Let $(x,\eta) \in \mathcal{X}\times\mathbb{R}$ be fixed. Let s_f be a subgradient of f at x and $s_{g_1},\dots,s_{g_n}$ be respective subgradients of $g(\cdot,\xi_1),\dots,g(\cdot,\xi_n)$ at x. For a given $t\in\mathbb{R}$, denote by $I_{>t}$ the set of indices such that $g(x,\xi_i) > t$ and by $I_{=t}$ the set of indices such that $g(x,\xi_i) = t$. Let finally $\alpha = \frac{\mathbb{P}[g(x,\xi)\le Q_p(g(x,\xi)]-p}{\#(I_{=Q_p(g(x,\xi))})}$. Then, s_{φ_1} and s_{φ_2} defined as:*

$$s_{\varphi_1} = \left(s_f + \frac{\lambda}{n(1-p)} \sum_{i\in I_{>\eta}}^{n} s_{g_i}\ ,\ 1 + \mu \mathbb{1}_{\eta>0} - \lambda \frac{\#(I_{>\eta})}{n(1-p)} \right)$$

$$s_{\varphi_2} = \left(\frac{\lambda}{n(1-p)} \left(\sum_{i\in I_{>Q_p(g(x,\xi))}} s_{g_i} + \alpha \sum_{i\in I_{=Q_p(g(x,\xi))}} s_{g_i} \right)\ ,\ 0 \right)$$

are respectively subgradients of φ_1 and φ_2 at (x,η).

Notice now that the convergence result for the bundle algorithm [7, Th. 1] guarantees convergence towards a point $\bar{u} = (\bar{x},\bar{\eta})$ satisfying

$$\partial\varphi_2(\bar{u}) \cap \partial\varphi_1(\bar{u}) \neq \emptyset, \tag{19}$$

which is a weak notion of criticality. Thus, we propose to furthermore replace φ_2 in (18) by a smooth approximation of it, denoted by $\widetilde{\varphi}_2$. The reason is that the bundle method minimizing $\widetilde{\varphi} = \varphi_1 - \widetilde{\varphi}_2$ then reaches a Clarke-stationary point: indeed, (19) reads $\nabla\widetilde{\varphi}_2(\bar{u}) \subset \partial\varphi_1(\bar{u})$, which gives $0 \in \partial\varphi(\bar{u}) = \partial\varphi_1(\bar{u}) + \nabla\widetilde{\varphi}_2(\bar{u})$, i.e., that $\bar{u}$ is Clarke-stationary (for the smoothed problem). To smooth φ_2, we use the efficient smoothing procedure of [15] for superquantile-based functions (implementing the Nesterov's smoothing technique [21]). More precisely, [15, Prop. 2.2] reads as follows.

Proposition 8. *Assume that g is differentiable. For a smoothing parameter $\rho > 0$, the function*

$$\widetilde{\varphi}_2(x,\eta) = \lambda \sup_{\substack{0\le q_i\le \frac{1}{n(1-p)} \\ q_1+\dots+q_n=1}} \sum_{i=1}^{n} \left\{ q_i\, g(x,\xi_i) - \frac{\rho}{2}(q_i - \tfrac{1}{n})^2 \right\} \tag{20}$$

is a global approximation of φ_2, such that $\widetilde{\varphi}_2(x,\eta) \leq \varphi_2(x,\eta) \leq \widetilde{\varphi}_2(x,\eta) + \frac{\lambda\rho}{2}$ for all $(x,\eta) \in \mathbb{R}^{d+1}$. Moreover, the function is differentiable and its gradient writes, with $S = (s_{g_i})_{1\leq q\leq n}$ the Jacobian of $x \mapsto (g(x,\xi_1),\ldots,g(x,\xi_n))$, as

$$\nabla\widetilde{\varphi}_2(x,\eta) = (\lambda\, S\, \tilde{q}\ ,\ 0)$$

where $\tilde{q}$ is the (unique) optimal solution of (20).

Note that the computation of $\widetilde{q}$ can be performed with fast computational procedures, as proposed in [15].

4.2 A python toolbox for chance constrained optimization

We release `TACO`, an open-source python toolbox for solving chance constrained optimization problems (1). The toolbox implements the penalization approach outlined in section 3 together with the bundle method [7] for the inner penalized subproblems. `TACO` routines rely on just-in-time compilation supported by Numba [16]. The routines are optimized to provide fast performances on reasonably large datasets. Documentation is available at:

https://yassine-laguel.github.io/taco

We provide here basic information on `TACO`; for further information, we refer to section B in appendix and the online documentation.

The python class `Problem` wraps up all information about the problem to be solved. This class possesses an attribute `data` which contains the values of ξ and is formatted as a `numpy` array in 64-bit float precision. The class also implements two methods giving first-order oracles: `objective_func` and `objective_grad` for the objective function f, and `constraint_func` and `constraint_grad` for the constraint function g.

Let us take a simple quadratic problem in $\mathbb{R}^2$ to illustrate the instantiation of a problem. We consider

$$\begin{aligned} &\min_{x\in\mathbb{R}^2} \quad \|x-a\|^2 && a = [1.0, 2.0]^\top \\ &\text{s.t.} \quad \mathbb{P}[x^\top\xi \leq 0] \geq 0.9, && \text{with 1000 samples of } \xi \sim \mathcal{N}(0,1). \end{aligned}$$

The instance of `Problem` is in this case:

```
import numpy as np
class Problem:
   def __init__(self, dim=2, sample_size=1000):
      self.data = np.random.normal(size=(sample_size, dim),
         dtype=np.float64)
      self.a = np.array([1.0, 2.0], dtypte=np.float64)
   def objective_fun(self,x):
      return np.dot(x-self.a,x-self.a)
   def objective_grad(self, x):
      return x
   def contraint_func(self, x, z):
      return np.dot(x,z)
   def constraint_grad(self, x, z)
      return z
problem = Problem()
```

TACO handles the optimization process with a python class named Optimizer. Given an instance of Problem and hyper-parameters provided by the user, the class Optimizer runs an implementation of the bundle method of [7] on the penalized problem (10). The toolbox gives the option to update the penalization parameters μ, λ along the running process to escape possible stationary points for the DC objective that are non-feasible for the chance constraint.

```
from taco import Optimizer
problem = Problem()
optimizer = Optimizer(problem, p=0.9, starting_point=np.zeros(2,
    dtype=np.float64), pen1=1.0, pen2=10.0)
sol = optimizer.run()
```

Customizable parameters are stored in a python dictionary, called params, and designed as an attribute of the class Optimizer. The main parameters to tune are: the safety level of probability p, the starting penalization parameters $\mu =$ pen1 and $\lambda =$ pen2, the starting point of the algorithm and the starting value for the proximal parameter of the bundle method. Others parameters are filled with default values when instantiating an Optimizer; for instance:

```
custom_options = {
    'p': 0.9,
    'pen1': 1.0,
    'pen2': 10.0,
    'bund_mu_start': 50.0,
    'bund_max_size_bundle_set': 30,
}
custom_optimizer = Optimizer(problem, params=custom_options)
```

Some important parameters (such as the safety probability level, or the starting penalization parameters) may also be given directly to the constructor of the class Optimizer, when instantiating the object; as in the first example.

5 Numerical illustrations

We illustrate our double penalisation approach implemented in the toolbox TACO on two problems: a 2-dimensional quadratic problem with a non-convex chance constraint (in Section 5.1), and a family of problems with explicit solutions (in Section 5.2). These proof-of-concept experiments are not meant to be extensive but to show that our approach is viable. These experiments are reproducible: the experimental framework is available on the toolbox's website.

5.1 Visualization of convergence on a 2-d problem

We consider a two-dimensional toy quadratic problem in order to track the convergence of the iterates on the sublevel sets. We take [31, Ex. 4.1] which considers an instance of problem (1) with

$$\begin{aligned} f(x) &= \frac{1}{2}(x-a)^\top Q(x-a) \quad \text{with } a = \begin{pmatrix} 2. \\ 2. \end{pmatrix}, Q = \begin{pmatrix} 5.5 & 4.5 \\ 4.5 & 5.5 \end{pmatrix} \\ g(x,z) &= z^\top W(x) z + w^\top z \quad \text{with } W(x) = \begin{pmatrix} x_1^2 + 0.5 & 0. \\ 0. & |x_2 - 1|^3 + 1 \end{pmatrix} \\ \xi &\sim \mathcal{N}(\mu, \Sigma) \qquad 10^4 \text{ samplings with } \mu = \begin{pmatrix} 1. \\ 1. \end{pmatrix}, \Sigma = \begin{pmatrix} 20. & 0. \\ 0. & 20. \end{pmatrix}. \end{aligned} \tag{21}$$

For this example, [31] shows that the chance constraint is convex for large enough probability levels, but here we take a low probability level $p = 0.008$ to have a non-convex chance-constraint. We can see this on Figure 1, ploting the level sets of the objective function and the constraint function: the chance-constrained region for $p = 0.008$ is delimited by a black dashed line; the optimal value of this problem is located at the star.

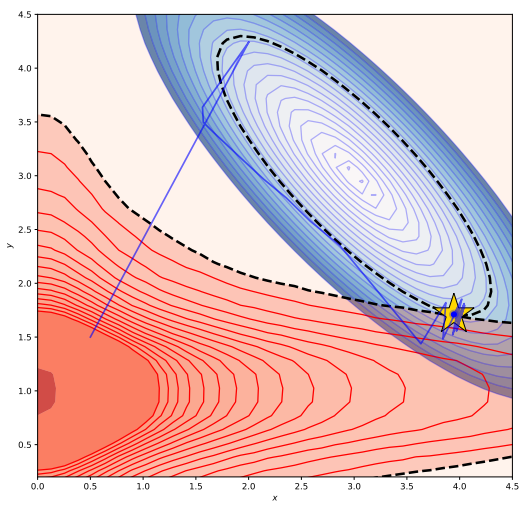

Figure 1: Trajectory of the iterates (in blue) on the plot of the level sets of the chance-constraint and the objective for the 2d problem with data (21).

We apply our double penalization method to solve this problem, with the setting described in Appendix B.3 and available on the `TACO` website. We plot on the sublevel sets of Figure 1 the path (in deep blue) taken by the sequence of iterates starting from the point $[0.5, 1.5]$ moving towards the solution. We observe that the sequence of iterates, after a exploration of the functions landscape, gets rapidly close to the optimal solution. At the end of the convergence, we also see a zigzag behaviour around the frontier of the chance constraint. This can be explained by the penalization term which is activated asymptotically whenever the sequence gets out of the chance constraint.

5.2 Experiments on a family of problems

We consider the family of d-dimensional norm problems of [13, section 5.1]. For a given dimension d, the problem writes as an instance of (1) with

$$f(x) = -\|x\|_1 \qquad \text{and} \qquad g(x, Z) = \max_{i \in \{1,\dots,10\}} \sum_{j=1}^{d} Z_{i,j}^2 x_j^2 - 100 \tag{22}$$

and ξ is random matrix of dimensions $10 \times d$ statisfying for all i, j, $\xi_{i,j} \sim \mathcal{N}(0, 1)$. The interest of this family of problems is that they have explicit solutions: for given d, the optimal value is

$$f^\star = -\frac{10\,d}{\sqrt{F_{\chi_d^2}^{(-1)}(p^{\frac{1}{10}})}}$$

where $F_{\chi_d^2}$ is χ^2 cumulative distribution with d degrees of freedom. We consider four instances of this problems with dimension d from 2 to 200 and the safety probability threshold p set to 0.8. We consider the case of the rich information on uncertainty: ξ is sampled 10000 times. In this case, a direct approach consisting in solving the standard mixed-integer quadratic reformulations (see e.g. [1]) with efficient MINLP solvers (we used `Juniper` [14]) does not provide reasonable solutions; see basic information in Appendix B.3.

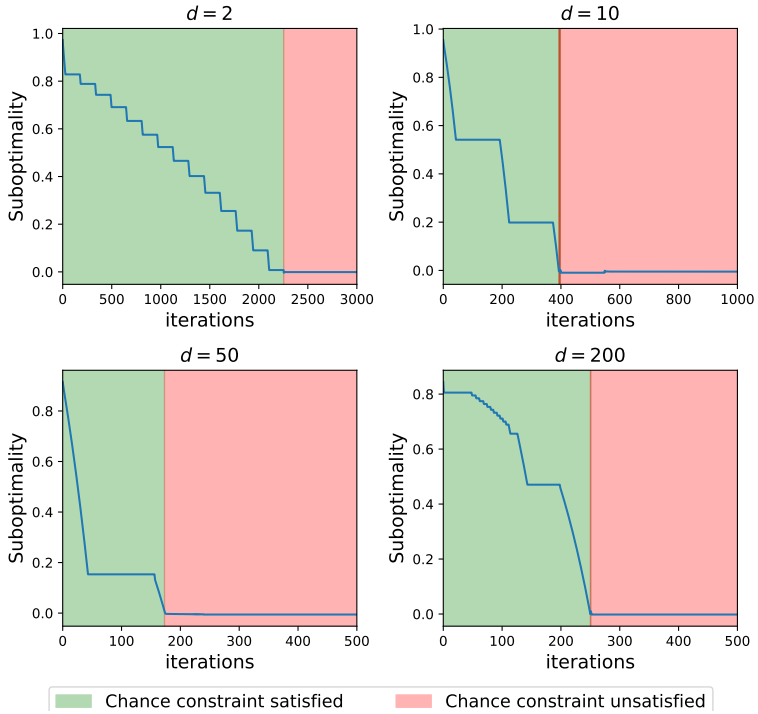

Figure 2: Convergence of the algorithm on four norm problems (22) with $d = 2, 10.50, 200$.

We solve these instances with our double penalization approach, parameterized as described in Appendix B.3 (see the `TACO` website for the material and settings provided in order to reproduce experiments). Figure 2

plots the relative suboptimality

$$(f(x_k) - f^\star)/|f^\star|$$

along iterations. The green (resp. red) regions represent iterates that, respectively, satisfy (resp. do not satisfy) the chance constraint.

In the four instances, we take a first iterate well inside the feasible region. We observe an initial decrease of the objective function down to optimal value. Then the chance constraint starts to be violated only when this threshold is reached, and the last part of convergence deals with local improvement of precision and feasibility.

Table 1 reports the final suboptimality and satisfaction of the probabilistic constraint. The probability constraint is evaluated for 100 sampled points from of the total $N = 10000$ points. We give the resulting probability; the standard deviation is 0.004 for the four instances.

Dimension	Suboptimality	$\mathbb{P}[g(x,\xi) \leq 0]$
$d = 2$	8.9×10^{-4}	0.799
$d = 10$	5.0×10^{-3}	0.787
$d = 50$	5.6×10^{-3}	0.769
$d = 200$	1.8×10^{-3}	0.781

Table 1: Final suboptimality and feasibility for (22) (where $p = 0.8$).

We observe that the algorithm reaches an accuracy of order of 10^{-3}. Regarding satisfaction of the constraint $\mathbb{P}[g(x,\xi) \leq 0] \geq 0.8$, it is achieved to a 10^{-4} precision for $d = 2$ but it slightly degrades as the dimension grows.

A Proofs of complementary results

A.1 Uniform bound at the limit

We show here that the uniform error bound derived in Section 3.1 vanishes at the limiting case of continuous distributions. We assume that, for a fixed $x \in \mathbb{R}^{\mathrm{d}}$, the random variable $g(x,\xi)$ has a continuous density $f_{x,\xi} : \mathbb{R} \to \mathbb{R}$ denoted by $f_{x,\xi}$: we have, for all $a \leq b$,

$$\mathbb{P}[a \leq g(x,\xi) \leq b] = \int_a^b f_{x,\xi}(t)\,\mathrm{d}\,t.$$

Proposition 9. *Fix $x \in \mathbb{R}^d$ and denote by q_p the p-quantile of the distribution followed by the random variable $g(x,\xi)$. If $g(x,\xi)$ has a continuous density, then the value function $\eta \mapsto h(x,\eta)$ defined in (13) is differentiable at $\eta = q_p$ (with $h'(x,q_p) = 0$).*

Proof. We first note that the existence of a density ensures the continuity of the cumulative distribution function of $g(x,\xi)$, which in turns implies $\mathbb{P}[g(x,\xi) \leq q_p] = p$. Let us now come back to expressions

established in the proof of Theorem 4. From (14), we have, for $\eta > q_p$,

$$\begin{aligned} h(x,\eta) &= (\eta - q_p)\frac{1}{1-p}\left(\mathbb{P}[g(x,\xi) \le q_p] - p + \mathbb{E}\left[\frac{\eta - g(x,\xi)}{\eta - q_p}\mathbb{1}_{q_p < g(x,\xi) \le \eta}\right]\right) \\ &= \frac{1}{1-p}\mathbb{E}\left[(\eta - g(x,\xi))\mathbb{1}_{q_p < g(x,\xi) \le \eta}\right] \;=\; \frac{1}{1-p}\int_{q_p}^{\eta}(\eta - t) f_{x,\xi}(t)\,\mathrm{d}\,t \\ &= \frac{1}{1-p}\left(\eta\int_{q_p}^{\eta} f_{x,\xi}(t)\,\mathrm{d}\,t - \int_{q_p}^{\eta} t f_{x,\xi}(t)\,\mathrm{d}\,t\right). \end{aligned}$$

By continuity of the above integrands, we can use the fundamental theorem of calculus to get that $h(x,\cdot)$ admits a right derivative at $\eta = q_p$ such that

$$\begin{aligned} h'_+(x,\eta) &= \lim_{\substack{\eta \to q_p \\ \eta > q_p}} \frac{h(x,\eta) - h(x,q_p)}{\eta - q_p} \\ &= \lim_{\substack{\eta \to q_p \\ \eta > q_p}} \frac{1}{1-p}\left(\eta\frac{\int_{q_p}^{\eta} f_{x,\xi}(t)\,\mathrm{d}\,t}{\eta - q_p} - \frac{\int_{q_p}^{\eta} t f_{x,\xi}(t)\,\mathrm{d}\,t}{\eta - q_p}\right) \\ &= \lim_{\substack{\eta \to q_p \\ \eta > q_p}} \frac{1}{1-p}\left(\eta f_{x,\xi}(q_p) - q_p f_{x,\xi}(q_p)\right) \;=\; 0\,. \end{aligned}$$

For the case $\eta < q_p$, we have from (15), together with $\mathbb{P}[g(x,\xi) = q_p] = 0$:

$$\begin{aligned} h(x,\eta) &= (q_p - \eta)\frac{1}{1-p}\left(\mathbb{E}\left[1 - \frac{(q_p - g(x,\xi))}{q_p - \eta}\mathbb{1}_{\eta < g(x,\xi) < q_p}\right] + \mathbb{P}[g(x,\xi) = q_p]\right) \\ &= \frac{1}{1-p}\left((\eta - q_p)\int_{\eta}^{q_p} f_{x,\xi}(t)\,\mathrm{d}\,t - \int_{\eta}^{q_p}(q_p - t) f_{x,\xi}(t))\,\mathrm{d}\,t\right). \end{aligned}$$

Using again to the fundamental theorem of calculus, we get that $h(x,\cdot)$ admits a left derivative at $\eta = q_p$ with:

$$\begin{aligned} h'_-(x,\eta) &= \lim_{\substack{\eta \to q_p \\ \eta < q_p}} \frac{h(x,\eta) - h(x,q_p)}{\eta - q_p} \\ &= \lim_{\substack{\eta \to q_p \\ \eta < q_p}} \frac{1}{1-p}\left((\eta - q_p)\frac{\int_{\eta}^{q_p} f_{x,\xi}(t)\,\mathrm{d}\,t}{\eta - q_p} - \frac{\int_{\eta}^{q_p}(q_p - t) f_{x,\xi}(t)\,\mathrm{d}\,t}{\eta - q_p}\right) \;=\; 0\,. \end{aligned}$$

We can conclude that $h(x,\cdot)$ is differentiable at q_p with zero as derivative. □ □

A.2 Proof of the subgradient explicit expressions

We provide here a direct proof of the subgradient expressions of Proposition 7. Let $(x,\eta) \in \mathcal{X} \times \mathbb{R}$ be fixed, and consider first the case of φ_1. For $i \in \{1,\dots,n\}$, by successive applications of Theorems 4.1.1 and 4.4.2 from [12, Chap. D] to the functions

$$\varphi_1^{(i)} : (x,\eta) \mapsto \frac{1}{n}\left[f(x) + \mu\max(\eta,0) + \lambda\left(\eta + \frac{1}{1-p}\max(g(x,\xi_i) - \eta, 0)\right)\right]$$

we get for any $i \in \{1, \dots, n\}$

$$\frac{1}{n} s_f + \frac{\lambda}{n(1-p)} \mathbb{1}_{g(x,\xi_i)>\eta} s_g \in \partial_x \varphi_1^i(x, \eta)$$
$$\frac{\mu}{n} \mathbb{1}_{\eta>0} + \frac{\lambda}{n} - \frac{\lambda}{n(1-p)} \mathbb{1}_{g(x,\xi_i)>\eta} \in \partial_\eta \varphi_1^i(x, \eta).$$

Since $\varphi_1 = \sum_{i=1}^n \varphi_1^{(i)}$, we thus have

$$\left(s_f + \frac{\lambda}{n(1-p)} \sum_{i \in I_{>\eta}} s_{g_i} \ , \ \mu \mathbb{1}_{\eta>0} + \lambda - \lambda \frac{\#(I_{>\eta})}{n(1-p)} \right) \in \partial \varphi_1(x, \eta)$$

For φ_2 we need first the whole subdifferential of the function G, which, using above mentioned properties, writes

$$\partial G(x, \eta) = \Bigg\{ \Bigg(\frac{1}{1-p} \sum_{i=1}^n \frac{s_{g_i}}{n} (\mathbb{1}_{g(x,\xi_i)>\eta} + \beta_i \mathbb{1}_{g(x,\xi_i)=\eta}),$$
$$1 - \frac{1}{1-p} \sum_{i=1}^n \frac{1}{n} (\mathbb{1}_{g(x,\xi_i)>\eta} + \beta_i \mathbb{1}_{g(x,\xi_i)=\eta}) \Bigg), \ \beta_i \in [0,1], \quad \forall i \in \{1, \dots, n\} \Bigg\}.$$

By taking $\beta_i = \alpha$ (for all $i \in \{1, \dots, n\}$) with the specific α given in the statement, we can zero the second term in the above expression. Now since $\varphi_2(x, \eta) = \lambda \min_{s \in \mathbb{R}} G(x, s)$ with $\arg\min_{s \in \mathbb{R}} G(x, s) = Q_p(g(x, \xi))$, we apply Corollary 4.5.3 of [12, Chap. D] to obtain a subgradient of φ_2:

$$s_{\varphi_2} = \left(\frac{\lambda}{n(1-p)} \left(\sum_{i \in I_{>Q_p(g(x,\xi)}} s_{g_i} + \alpha \sum_{i \in I_{=Q_p(g(x,\xi)}} s_{g_i} \right), 0 \right)$$

which completes the proof.

B Implementation details on TACO

B.1 Further customization

TACO relies on a set of hyperparameters to be provided by the user and specified

in a single dictionnary passed as an argument of the class `Optimizer`. There are two families of parameters to be specified. First, the parameters concerning the oracles φ_1 and φ_2. These are the starting penalization parameters λ and μ, the multiplicative factors to increment them along the penalization process, and the smoothing parameter of $\tilde{\varphi}_2$.

The second family of parameters concerns the bundle method. It gathers the proximal parameters of the bundle method, the precision targeted, the starting point of the algorithm, the maximal size of the bundle information, and parameters related used when restarting the bundle method (see more in the following section).

Overall the most important parameters to specify are the starting penalization parameters μ and λ with respective keys `'pen1'` and `'pen2'` and the starting proximal parameter of the bundle algorithm. In the

toolbox, we provide the set of parameters used in our numerical experiments. In addition of the final solution, it is possible to log the iterates, function values and time values, by calling the method with the option `logs=True`. The `verbose=True` option also allows the user to observe in real time the progression of the algorithm along the iterations.

Finally we underline that `TACO` subroutines rely on just-in-time compilation supported by `Numba`, which consistently improves the running time. Further improvements can be achieved when the instance considered can be cast as a `Numba jitclass`. The parameter `'numba'` in the input dictionnary of the associated `Optimizer` object should then be set to `True`.

B.2 On the bundle algorithm

Here are some information on our implementation of the bundle algorithm of [7] to tackle the double penalized problem $(P_{\lambda,\mu})$ written as a DC problem. We discuss the parameters used at various steps of the procedure. We refer to [7] for more details.

- **Overall run**: The starting point, the maximum number of iterations as well as the precision tolerance for termination may be set by the user.
- **Subproblems**: Each iteration of the bundle algorithm requires solving a quadratic subproblem (see [7, Eq. (9)]), for which we use the solver `cvxopt` [32] by simplicity.
- **Stabilization center**: Whenever the solution of a subproblem satisfies a sufficient decrease in terms a function value, it is considered as a new stability center. The condition to qualify sufficient decrease is given in [7, Eq. (12)]. It involves a constant κ which may be tuned by the user.
- **Proximal parameters**: The initial value of the proximal parameter involved in quadratic subproblems can be set by the user. The user can also specify upper and lower acceptance bounds for it. After each iteration, the prox-parameter is updated: it is increased by a constant factor in case of serious step, and decreased otherwise. Both factors can be tuned by the user.
- **Bundle information**: The bundle of cutting-planes is augmented after each null step with new linearization, and emptied after each serious step. We fix a maximum size for the bundle: above this parameter, the bundle is emptied and proximal parameter is restarted to a specified restarting value. When the bundle is emptied, we have the chance of a specific improvement: if the stability center is feasible in the chance-constraint, we replace the coordinate playing the role of η by the p-quantile of $g(x, \xi)$, thus decrease the objective function.
- **Termination Criteria**: We use a simple stopping criteria: we stop when the euclidean distance between the current iterate and the current stability center falls below a certain threshold specified by the user.

B.3 Experimental settings

Setting of Section 5.1. For this 2d problem, we use the starting point $x = (0.5, 1.5)$ (and $\eta = 0.01$) well-inside the chance-constraint. The initial penalization parameters μ and λ are respectively initialized to 400 and 600. The initial proximal parameter is fixed to 38.0 with lower and upper acceptance bounds set to 10^{-3} and 10^{3}. Increasing and decreasing factors for this parameter are fixed to 1.05 and 0.95. The classification rule parameter is set to 10^{-4}. The maximal size of the information bundle is set to 20 and the threshold of the termination criteria is set to 10^{-7}.

Setting for Section 5.2. For any fixed dimension d comprised in $\{2, 10, 50, 200\}$, the algorithm is run from the starting point $(0.1, \ldots, 0.1) \in \mathbb{R}^{d+1}$. The starting penalization parameter μ, constant for the 4 instances, is set to $\mu = 10.0$. We tuned the second penalization parameter λ along problems: we observed that $\lambda = \{1.75, 1.5, 1.5, 2.0\}$ give good performances for the considered problems.

The starting proximal parameters is fixed to 60.0 with lower and upper acceptance bounds set to 10^{-4} and 10^5 respectively. Increasing and decreasing factors for the proximal parameter are fixed to 1.01 and 0.99. The classification rule parameter is set to 10^{-4}. The maximal size of the information bundle is set to 300.

Limitations of MINLP approach. Mixed-integer reformulation approaches (see e.g. [1]) are often considered as the state-of-the-art to solve chance constrained optimization problems by sample average approximation. Applying directly such a reformulation to Problem (22) in Section 5.2 leads to the equivalent mixed integer quadratic program:

$$\begin{aligned} \min_{x \in \mathbb{R}^d,\, z \in \{0,1\}^N} \quad & -\sum_{i=1}^{d} x_i \\ \text{s.t.} \quad & \sum_{k=1}^{d} (\xi_i)_{j,k}^2 x_k^2 - 100 \leq M\ z_i, \quad \forall i \in [\![1, N]\!],\ \forall j [\![1, 10]\!] \\ & \sum_{i=1}^{N} z_i \leq pN, \ \ x \geq 0. \end{aligned}$$

where M is a large "big-M" constant. In our setting, such formulation involves $10 \times N = 100000$ quadratic constraint involving binary variables. We were not able to solve the resulting mixed-integer problem in reasonable times using the MINLP solver `Juniper` [14] (that is based on Ipopt and JuMP). This shows that a direct application of reformulation techniques combined with reliable software failed on this problem in contrast with our approach.

References

[1] S. Ahmed and A. Shapiro. Solving chance-constrained stochastic programs via sampling and integer programming. In *State-of-the-art decision-making tools in the information-intensive age*, pages 261–269. Informs, 2008.

[2] A. Ben-Tal and M. Teboulle. An old-new concept of convex risk measures: The optimized certainty equivalent. *Mathematical Finance*, 17(3):449–476, 2007.

[3] G. C. Calafiore and M. C. Campi. The scenario approach to robust control design. *IEEE Transactions on Automatic Control*, 51(5):742–753, 2006.

[4] A. Charnes and W. W. Cooper. Chance-constrained programming. *Management science*, 6(1):73–79, 1959.

[5] Y. Chow, M. Ghavamzadeh, L. Janson, and M. Pavone. Risk-constrained reinforcement learning with percentile risk criteria. *The Journal of Machine Learning Research*, 18(1):6070–6120, 2017.

[6] F. H. Clarke. *Optimization and nonsmooth analysis*, volume 5. Siam, 1990.

[7] W. de Oliveira. Proximal bundle methods for nonsmooth dc programming. *Journal of Global Optimization*, 2019.

[8] D. Dentcheva. Optimisation models with probabilistic constraints. In A. Shapiro, D. Dentcheva, and A. Ruszczyński, editors, *Lectures on Stochastic Programming. Modeling and Theory*, volume 9 of *MPS-SIAM series on optimization*. SIAM, 2009.

[9] D. Dentcheva, A. Prékopa, and A. Ruszczynski. Concavity and efficient points of discrete distributions in probabilistic programming. *Mathematical Programming*, 89(1), 2000.

[10] H. Föllmer and A. Schied. Convex measures of risk and trading constraints. *Finance and stochastics*, 6 (4):429–447, 2002.

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

[12] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex analysis and minimization algorithms I: Fundamentals*, volume 305. Springer science & business media, 2013.

[13] L. J. Hong, Y. Yang, and L. Zhang. Sequential convex approximations to joint chance constrained programs: A monte carlo approach. *Operations Research*, 59(3), 2011.

[14] O. Kröger, C. Coffrin, H. Hijazi, and H. Nagarajan. Juniper: An open-source nonlinear branch-and-bound solver in julia. In *Integration of Constraint Programming, Artificial Intelligence, and Operations Research*. Springer International Publishing, 2018. ISBN 978-3-319-93031-2.

[15] Y. Laguel, J. Malick, and Z. Harchaoui. First-order optimization for superquantile-based supervised learning. In *2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2020.

[16] S. K. Lam, A. Pitrou, and S. Seibert. Numba: A llvm-based python jit compiler. In *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, LLVM '15, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450340052.

[17] J. Luedtke and S. Ahmed. A sample approximation approach for optimization with probabilistic constraints. *SIAM Journal on Optimization*, 19:674–699, 2008.

[18] D. G. Luenberger and Y. Ye. *Linear and nonlinear programming*, volume 2. Springer, 1984.

[19] E. Medova. Chance-constrained stochastic programming forintegrated services network management. *Annals of Operations Research*, 81:213–230, 1998.

[20] A. Nemirovski and A. Shapiro. Convex approximations of chance constrained programs. *SIAM Journal on Optimization*, 17(4):969–996, 2006.

[21] Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical programming*, 103(1): 127–152, 2005.

[22] B. K. Pagnoncelli, S. Ahmed, and A. Shapiro. Sample average approximation method for chance constrained programming: theory and applications. *Journal of optimization theory and applications*, 142(2):399–416, 2009.

[23] A. Prékopa. *Stochastic Programming*. Kluwer, Dordrecht, 1995. doi: 10.1007/978-94-017-3087-7.

[24] A. Prékopa and T. Szántai. Flood control reservoir system design using stochastic programming. In *Mathematical programming in use*, pages 138–151. Springer, 1978.

[25] R. T. Rockafellar and J. O. Royset. Superquantiles and their applications to risk, random variables, and regression. In *Theory Driven by Influential Applications*. INFORMS, 2013.

[26] R. T. Rockafellar and J. O. Royset. Random variables, monotone relations, and convex analysis. *Mathematical Programming*, 148(1-2):297–331, 2014.

[27] R. T. Rockafellar and S. Uryasev. Optimization of conditional value-at-risk. *Journal of risk*, 2:21–42, 2000.

[28] A. Ruszczyński and A. Shapiro. Optimization of convex risk functions. *Mathematics of operations research*, 31(3):433–452, 2006.

[29] W. van Ackooij. A discussion of probability functions and constraints from a variational perspective. *Set-Valued and Variational Analysis (online)*, 28:585–609, 2020.

[30] W. van Ackooij, R. Henrion, A. Möller, and R. Zorgati. Joint chance constrained programming for hydro reservoir management. *Optimization and Engineering*, 15, 2014.

[31] W. Van Ackooij, Y. Laguel, J. Malick, and G. Matiussi-Ramalho. On the convexity of level-sets of probability functions. *Submitted*, 2020.

[32] L. Vandenberghe. The cvxopt linear and quadratic cone program solvers. *Online: http://cvxopt.org/documentation/coneprog. pdf*, 2010.

[33] J. J. Ye, D. Zhu, and Q. J. Zhu. Exact penalization and necessary optimality conditions for generalized bilevel programming problems. *SIAM Journal on optimization*, 7(2):481–507, 1997.