

# SOLVING CHANCE-CONSTRAINED PROBLEMS VIA A SMOOTH SAMPLE-BASED NONLINEAR APPROXIMATION\*

ALEJANDRA PEÑA-ORDIERES<sup>†</sup>, JAMES R. LUEDTKE<sup>‡</sup>, AND ANDREAS WÄCHTER<sup>†</sup>

**Abstract.** We introduce a new method for solving nonlinear continuous optimization problems with chance constraints. Our method is based on a reformulation of the probabilistic constraint as a quantile function. The quantile function is approximated via a differentiable sample average approximation. We provide theoretical statistical guarantees of the approximation, and illustrate empirically that the reformulation can be directly used by standard nonlinear optimization solvers in the case of single chance constraints. Furthermore, we propose an  $S\ell_1$ QP-type trust-region method to solve instances with joint chance constraints. We demonstrate the performance of the method on several problems, and show that it scales well with the sample size and that the smoothing can be used to counteract the bias in the chance constraint approximation induced by the sample approximation.

**Key words.** chance constraints, nonlinear optimization, quantile function, sample average approximation, smoothing, sequential quadratic programming, trust region

**AMS subject classifications.** 90C15, 90C30, 90C55

**1. Introduction.** Consider the following minimization problem subject to a chance constraint

$$(1.1a) \quad \min_{x \in X} f(x)$$

$$(1.1b) \quad \text{s.t. } \mathbb{P}(c_j(x, \xi) \leq 0, j \in \{1, \dots, m\}) \geq 1 - \alpha,$$

where  $X \subseteq \mathbb{R}^n$  represents a closed deterministic feasible region,  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  represents the objective function to be minimized,  $c : \mathbb{R}^n \times \mathbb{R}^s \rightarrow \mathbb{R}^m$  is a vector-valued function,  $\xi$  is a random vector with support  $\Xi \subseteq \mathbb{R}^s$ , and  $\alpha \in (0, 1)$  is a risk parameter given a priori. This kind of problem is known in the literature as a joint chance-constrained problem (JCCP), in which all  $m$  constraints must be satisfied simultaneously with probability at least  $1 - \alpha$ . A special case is when  $m = 1$  and only one constraint needs to be satisfied with probability  $1 - \alpha$ ; this is called a single chance-constrained problem (SCCP).

Chance-constrained optimization problems were introduced in [8] and have been extensively studied ever since; see e.g., [28]. These problems can be found in a plethora of applications such as water management [20], optimization of chemical processes [14, 15], and optimal power flow [4, 23], to mention a few.

This formulation gives rise to several difficulties. First, the structural properties of  $c(x, \xi)$  may not be passed to the constraint  $\mathbb{P}(c_j(x, \xi) \leq 0, j \in \{1, \dots, m\}) \geq 1 - \alpha$ . For example, even if  $c_1(x, \xi), \dots, c_m(x, \xi)$  are all linear in  $x$ , the probabilistic constraint may not define a convex feasible region. Second, the distribution of  $\xi$  may not be known. And finally, even if the distribution of  $\xi$  is known, there might not be a tractable analytical function to describe the constraint.

---

\*Submitted to the editors DATE.

<sup>†</sup>Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL ([alejandra.pena@u.northwestern.edu](mailto:alejandra.pena@u.northwestern.edu), [andreas.waechter@northwestern.edu](mailto:andreas.waechter@northwestern.edu)). These authors were funded in part by National Science Foundation grant DMS-1522747.

<sup>‡</sup>Department of Industrial and Systems Engineering, University of Wisconsin-Madison, Madison, WI ([jim.luedtke@wisc.edu](mailto:jim.luedtke@wisc.edu)). This author was funded by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research (ASCR) under Contract DE-AC02-06CH11347.

We focus on problems in which  $f$  and  $c$  are continuous and differentiable, with respect to  $x$ , and where  $\xi$  is such that the random variable defined by  $c(x, \xi)$  has a continuous cumulative density function (cdf) for all  $x \in X$ . Other than that, we make no assumptions on the type of constraint  $c(x, \xi)$ ; we do not limit ourselves to linear or convex functions.

**1.1. Literature Review.** In this section we review several methods for solving chance-constrained problems. The first couple of methods make assumptions on the distribution of  $\xi$ , or at least require that some knowledge of the distribution  $\xi$  is available, whereas the latter methods make no such assumptions. Our idea falls into the latter category, and does not impose restrictions on the form of the constraints or on the distribution, except that it is a continuous distribution.

Some methods assume a certain distribution over  $\xi$  such that, for the constraint of interest, one can get a closed-form analytic approximation to the probability function. For example, in [4], if  $\xi$  has an elliptical symmetric distribution (e.g., multivariate normal random variable) and if  $c(x, \xi) = \xi^T x - b$ , then the problem can be expressed as a second-order cone program, and can be solved in a tractable and efficient manner using standard solvers. Other cases, e.g. [33], assume  $\xi$  has a multivariate Gaussian distribution, and provide a way to directly compute the cdf and its derivatives.

There are several approaches that avoid making assumptions on  $\xi$  by sampling to approximate the probabilistic constraint. We divide such techniques into those that use an integer reformulation, and those that are structured as nonlinear programming problems (NLP).

In the first category, the probabilistic constraint is replaced by the empirical distribution function and formulated as a mixed-integer nonlinear program (MINLP), e.g. [21]. This method takes realizations of the random variable  $\xi$  and requires that at least  $(1 - \alpha)\%$  of the realized constraints are satisfied. This idea has several advantages; it finds the global optimum of the approximation, and as the sample size increases, the solution converges to the true solution of (1.1). However, the complexity of the problem increases with the sample size because the number of binary variables increases. This issue has been circumvented to some extent for certain types of constraints  $c(x, \xi)$ , e.g. linear constraints [22], but this procedure cannot be applied to general constraints.

To avoid the need for solving MINLPs, a conservative and more tractable approximation can be obtained by using the scenario-based approach, e.g. [5, 6, 25]. These methods seek to satisfy all of the constraints in a sample of pre-specified size to ensure that the solution is feasible with high confidence. The reformulation proposed in [5] allows one to obtain solutions very easily since the structure of the original constraints  $c(x, \xi)$  is preserved. For example, if  $f$  is convex,  $X$  defines a convex set and  $c(x, \xi_i)$  define convex functions on  $x$  for all  $i = \{1, \dots, N\}$  (where  $N$  is the size of the sample), then the problem that arises is convex. However, the limitation of this method is that one may obtain highly conservative solutions and that it does not offer direct control on the probability level of the chance constraint.

Another tractable conservative approximation can be obtained by using conditional value-at-risk, which is based on using a conservative convex approximation of the indicator function [24]. This approximation typically finds feasible but sub-optimal solutions. To avoid overly conservative solutions, some authors propose a difference-of-convex functions approximation to the indicator function [18], while some others propose outer or inner approximations to the indicator function [7, 13]. The main drawback of the last two methods mentioned is that the constraint function

can become very “flat” in the regions of interest. That is, when the probability of the constraint satisfaction evaluated at a given  $x$  is close to one or close to zero, the information obtained from the gradient may not provide a sufficiently accurate approximation of the constraints or may even be erroneous. We discuss this issue further when we motivate our proposed reformulation of (1.1) in section 2.

**1.2. Our Approach.** In this paper we propose an NLP reformulation of the problem based on sample average approximation (SAA), in which we rewrite the probability constraint as a quantile constraint. We first argue that the latter formulation is more suitable for gradient-based optimization methods than the former. Then, we present a way to approximate the quantile constraint using samples in a way that provides a smooth constraint. We find in our numerical experiments that the smoothing helps control the variance of the solutions obtained from different samples. The approximation we present uses a parameter  $\epsilon$  that depends on the scaling of the problem, which might be difficult to select; thus, we also discuss an algorithm that finds an appropriate value of  $\epsilon$ .

The proposed approach has the following advantages. First, we observe empirically that the solutions we obtain are very often “more robust” compared to solutions obtained from solving the traditional (non-smoothed) SAA problem via mixed-integer programming (MIP) formulations that use the empirical cdf. More specifically, when we compare the solutions obtained from our method to those returned by the MIP, we consistently find that our method attains a better objective function value and has less variability across different samples, while still maintaining out-of-sample feasibility. We attribute this to the fact that our approach uses information from the entire sample, whereas the empirical cdf only takes into account whether enough constraints are satisfied. These numerical experiments suggest that the solutions we attain are able to “generalize”, i.e., the solutions obtain better performance on out-of-sample scenarios. Second, our method is scalable; we find that the time to solve an instance with our approach grows modestly with the sample size. Third, our formulation does not assume any type of structure from the constraint functions save for continuity and differentiability with respect to  $x$ ; therefore, we are able to find local solutions to non-linear and non-convex problems. Finally, for SCCPs, a standard NLP solver can be used to find the solution of the problem for the appropriate choice of the parameters involved in the formulation, and for JCCPs we are able to derive an  $S\ell_1QP$  algorithm that has proven empirically to give fast convergence to the solution after only a few iterations.

The remainder of this paper is organized as follows. In section 2 we present the proposed reformulation of the problem. In section 3 we study the convergence and feasibility of the solutions to the approximate problem to those of the true problem (1.1). In section 4 we discuss single chance-constrained problems, and in section 5 we extend this approach to solve joint chance-constrained problems. In section 6 we present numerical results. Finally, we make some concluding remarks in section 7.

**2. Reformulation and approximation of the CCP.** The quantile of a random variable  $Y$  at the level  $1 - \alpha \in (0, 1)$  is defined as

$$Q^{1-\alpha}(Y) = \inf\{y \in \mathbb{R} \mid \mathbb{P}(Y \leq y) \geq 1 - \alpha\}.$$

Therefore,  $\mathbb{P}(C(x, \xi) \leq 0) \geq 1 - \alpha$  is equivalent to  $Q^{1-\alpha}(C(x, \xi)) \leq 0$ , where  $C : \mathbb{R}^n \times \mathbb{R}^s \rightarrow \mathbb{R}$  is a real valued function,  $x \in \mathbb{R}^n$ , and  $\xi$  is a random vector taking values in  $\mathbb{R}^s$ .

Notice that  $Q^{1-\alpha}(C(x, \xi))$  is only well defined if  $C(x, \xi)$  is a scalar random variable, i.e., when  $C$  is a scalar-valued function. However, in the case when  $c$  is a vector-valued function, letting  $C(x, \xi) = \max_{j=1, \dots, m} c_j(x, \xi)$ , we obtain the following reformulation of problem (1.1),

$$\begin{aligned} \min_{x \in X} \quad & f(x), \\ \text{s.t.} \quad & Q^{1-\alpha}(C(x, \xi)) \leq 0. \end{aligned}$$

The advantage of rewriting the constraint as the quantile is that this reduces the “flatness” encountered when the probability function  $\mathbb{P}(C(x, \xi) \leq 0)$  is used as the constraint, see Figure 1 as an example. Now the feasible region is measured in the image of  $C(x, \xi)$  and not in the bounded space  $[0, 1]$ , which is the image of the probability function.

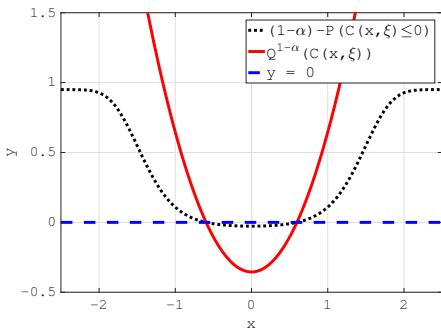


Fig. 1: Comparison of  $Q^{1-\alpha}(C(x, \xi))$  and  $(1 - \alpha) - P(C(x, \xi) \leq 0)$  where  $C(x, \xi) = x^2 - 2 + \xi$  and  $\xi \sim N(0, 1)$ .

Figure 1 also shows how the feasible regions coincide for both formulations. Yet, although the two formulations are equivalent, solvers that depend on gradient information are likely to perform better when the quantile function is used, as we will see in the numerical experiments. This can be attributed to the fact that the linearization of the quantile function provides an approximation of the constraint that is reasonably accurate for a wider range than the linearization of the probabilistic constraint. The consequence of “flatness” is that a non-specialized gradient-based solver that starts at a point  $x$  where  $\mathbb{P}(C(x, \xi) \leq 0) \approx 1$  takes a big step to decrease the objective, disregarding the constraint function. Thus, the solver is forced to reject many trial steps, which can lead to an increased number of iterations.

To illustrate this claim, we compared the numerical performance of the nonlinear programming solver `Knitro` for each of the two formulations for the constraints  $C(x, \xi) = x^2 - 2 + \xi$ . We used `Knitro` to maximize  $x$  using: (1) the probabilistic constraint  $\mathbb{P}(C(x, \xi) \leq 0) = \Phi(2 - x^2) \geq 1 - \alpha$ , and (2) the quantile constraint  $Q^{1-\alpha}(C(x, \xi)) = x^2 - 2 + \Phi^{-1}(1 - \alpha) \leq 0$ ; see Figure 1. We chose  $x_0 = 3$  as starting point. Table 1 illustrates the performance in terms of iteration count using the two types of constraints. Each column shows the number of iterations using different bounds on  $x$ . The results in the table suggest that the quantile function is more robust since the number of iterations is independent of the bounds and the solver always returned feasible and optimal solutions.

Motivated by the aforementioned arguments, we propose to work with the quantile formulation. However, as with the probabilistic formulation, an explicit expression to compute the quantile is usually not available or can be intractable to compute. To

Bounds	$[-1, 1]$	$[-10, 10]$	$[-100, 100]$	$(-\infty, \infty)$
$\mathbb{P}(C(x, \xi) \leq 0) \geq 1 - \alpha$	6	36	14	5*
$Q(C(x, \xi)) \leq 0$	6	6	6	6

Table 1: Iterations performed by `Knitro` when maximizing  $x$  using two different formulations to describe the feasible region and different bounds on  $x$ . \*: `Knitro` gives the following error message: “Convergence to an infeasible point. Problem appears to be locally infeasible.”

address this issue, we introduce a sample average approximation of the quantile in the following section.

**2.1. Smooth approximation of the quantile function.** A quantile estimator can be obtained by inverting an estimator of the cdf when the distribution is continuous. Let  $\{\xi_1, \dots, \xi_N\}$  be an independent and identically distributed (i.i.d.) sample and consider the empirical cdf of  $\{C(x, \xi_1), C(x, \xi_2), \dots, C(x, \xi_N)\}$  at a given point  $x$ ,

$$\tilde{F}^N(t; x) = \frac{1}{N} \sum_{i=1}^N \mathbb{1}(C(x, \xi_i) \leq t).$$

An estimate of the  $(1 - \alpha)$ -quantile (and hence  $Q^{1-\alpha}(C(x, \xi))$ ) is obtained from a value  $t$  such that  $\tilde{F}^N(t; x) \approx 1 - \alpha$ . The  $(1 - \alpha)$ -empirical quantile at  $x$  is defined as

$$\tilde{Q}^{1-\alpha}(C^N(x)) = \inf \left\{ y \mid \frac{1}{N} \sum_{i=1}^N \mathbb{1}(C(x, \xi_i) \leq y) \geq 1 - \alpha \right\} = C_{[M]}(x),$$

for  $M = \lceil (1 - \alpha)N \rceil$ . Here,  $\mathbb{1}$  is the indicator function and  $C_{[j]}$  denotes the  $j$ th smallest observation of the values  $\{C(x, \xi_1), \dots, C(x, \xi_N)\}$  for a fixed  $x$ .

There are two main drawbacks to using the empirical quantile as an approximation. First, the empirical approximation is not differentiable. The “active scenario” (the scenario  $\xi_i$  such that  $C_{[M]}(x) = C(x, \xi_i)$ ) changes for different values of  $x$ . Hence, even if the constraint  $C(x, \xi)$  is smooth for a fixed value of  $\xi$ , this does not hold for  $\tilde{Q}^{1-\alpha}(C^N(x))$ . Second, the aforementioned feature also introduces artificial local minima to the problem, as can be seen in [Figure 2](#). For a particular sample  $\{\xi_1, \dots, \xi_N\}$ , this figure depicts the feasible region

$$\{x \in \mathbb{R}^2 \mid Q^{1-\alpha}(C(x, \xi)) \leq 0\},$$

and its approximation based on the empirical quantile function; where  $C(x, \xi) = \xi^T x - 1$ ,  $\xi \sim N(\mu, \Sigma)$ ,  $\mu = \mathbf{0} \in \mathbb{R}^2$ , and  $\Sigma$  is the  $2 \times 2$  identity matrix.

Notice that even if the true feasible region is convex and has a smooth boundary (dotted curve), its approximation (solid curve) is non-smooth and introduces a lot of “inward kinks”, which break the convexity of the feasible region. These “inward kinks” can potentially cause a local optimization algorithm to converge to inferior local minima more easily. Furthermore, as the number of scenarios increases, the number of non-smooth points increases, leading to more sub-optimal local minima.

In order to avoid the aforementioned problems, a smooth approximation of the quantile is obtained by estimating the cdf with a smooth function  $\hat{F}$ , and solving the

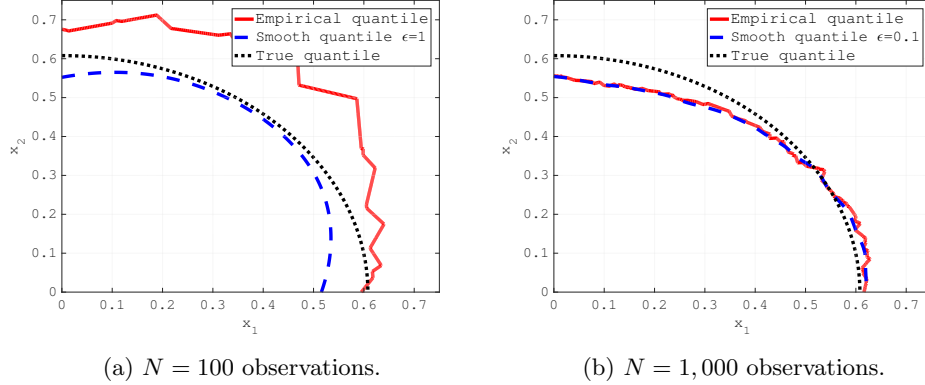


Fig. 2: Different quantile approximations ( $\tilde{Q}^{1-\alpha}(C^N(x))$  as a solid red line, and  $Q_\epsilon^{1-\alpha}(C^N(x))$  as a dashed blue line) vs. true quantile  $Q^{1-\alpha}(C(x, \xi))$ .

equation  $\hat{F}(t; x) = 1 - \alpha$ , where  $t$  is the approximation of  $Q^{1-\alpha}(C(x, \xi))$ . Following an approach similar to [13, 31], we approximate  $F$  at a given point  $x$  as

$$(2.1) \quad F_\epsilon^N(t; x) = \frac{1}{N} \sum_{i=1}^N \Gamma_\epsilon(C(x, \xi_i) - t),$$

where  $\epsilon > 0$  is a parameter of the approximation,

$$(2.2) \quad \Gamma_\epsilon(y) = \begin{cases} 1, & y \leq -\epsilon \\ \gamma_\epsilon(y), & -\epsilon < y < \epsilon \\ 0, & y \geq \epsilon \end{cases}$$

and  $\gamma_\epsilon : [-\epsilon, \epsilon] \rightarrow [0, 1]$  is a symmetric and strictly decreasing function such that it makes  $\Gamma_\epsilon$  differentiable. With this choice of  $\gamma_\epsilon$ ,  $F_\epsilon^N(t; x)$  is a differentiable approximation of the empirical cdf,  $\tilde{F}^N(t; x)$ .

Assuming that  $C(x, \xi)$  is a continuous random variable for a fixed value of  $x$ , we can approximate the  $(1 - \alpha)$ -quantile as the inverse of  $F_\epsilon^N$  at the  $1 - \alpha$  level. Thus, we define our  $\epsilon$ -approximation to the  $(1 - \alpha)$ -quantile at  $x$  as the value  $Q_\epsilon^{1-\alpha}$  such that the following equation holds

$$(2.3) \quad \sum_{i=1}^N \Gamma_\epsilon([C^N(x)]_i - Q_\epsilon^{1-\alpha}) = (1 - \alpha)N,$$

where  $C^N(x) \in \mathbb{R}^N$  is a vector with elements  $[C^N(x)]_i = C(x, \xi_i)$ . However, the value  $Q_\epsilon^{1-\alpha}$  is not unique if the equation  $F_\epsilon^N(t; x) = 1 - \alpha$  has more than one solution. In order to guarantee the existence of a unique solution for all possible choices of  $\epsilon > 0$ , we need to ensure that  $[F_\epsilon^N]'(t; x) \neq 0$  when evaluated at the quantile of interest. The following proposition gives conditions under which  $Q_\epsilon^{1-\alpha}$  is uniquely defined.

**PROPOSITION 2.1.** *Let  $\gamma_\epsilon : [-\epsilon, \epsilon] \rightarrow [0, 1]$  be a strictly decreasing function such that  $\Gamma_\epsilon$  is differentiable, and let  $(1 - \alpha)N \notin \mathbb{Z}$ . Then, equation (2.3) has a unique solution.*

*Proof.* Let  $\psi(Q) = \sum_{i=1}^N \Gamma_\epsilon(C(x, \xi_i) - Q) - (1 - \alpha)N$ . We have that  $\psi'(Q) = -\sum_{i=1}^N \Gamma'_\epsilon(C(x, \xi_i) - Q)$ . Given that  $\gamma_\epsilon$  is strictly decreasing and  $\Gamma'_\epsilon(y) = 0$  for all  $y \geq \epsilon$  and  $y \leq -\epsilon$ , we conclude that  $\psi'(Q) \geq 0$  for all  $Q$ , which means that  $\psi$  is an increasing function. If we prove that  $\psi'(Q) \neq 0$  for all  $Q$  such that  $\psi(Q) = 0$ , then we show that there is a unique solution of equation (2.3).

Assume  $\psi'(Q) = 0$ , then  $Q \leq C(x, \xi_i) - \epsilon$  or  $Q \geq C(x, \xi_i) + \epsilon$  for all scenarios  $i \in \{1, \dots, N\}$ . This means that  $\sum_{i=1}^N \Gamma_\epsilon(C(x, \xi_i) - Q) \in \mathbb{N}$ . Because  $(1 - \alpha)N \notin \mathbb{N}$ , it means that  $\psi(Q) \neq 0$ , and therefore any  $Q$  such that  $\psi'(Q) = 0$  cannot be a solution.  $\square$

*Remark 2.2.* The assumption  $(1 - \alpha)N \notin \mathbb{Z}$  of Proposition 2.1 is mild because if it is not satisfied one can either increase  $N$ , perturb  $\alpha$ , or add a constant  $b \in (0, 1)$  to the left hand side of (2.3). In fact, if a constant is added to the left-hand side, the impact will disappear as  $N \rightarrow \infty$ .

Proposition 2.1 shows that (2.3) defines a function  $Q_\epsilon^{1-\alpha}$  that maps a vector  $C^N(x) \in \mathbb{R}^N$  to the root of (2.3). In order to find solutions for (1.1), we focus on solving the following problem

$$(2.4) \quad \begin{aligned} \min_{x \in X} \quad & f(x) \\ \text{s.t.} \quad & Q_\epsilon^{1-\alpha}(C^N(x)) \leq 0. \end{aligned}$$

The above formulation is a sample average approximation of the original problem (1.1). Convergence and feasibility of this approximation are analyzed in section 3.

One could have approximated constraint (1.1b) with  $F_\epsilon^N(0; x) \geq 1 - \alpha$ , where  $F_\epsilon^N$  is given by (2.1). However, as mentioned at the beginning of this section and as shown in the numerical results (section 6.2.2), working with the probabilistic constraint, or its estimate, instead of the quantile leads to more iterations or failure of the optimization algorithm. Therefore, in contrast with previous works that express the chance constraint using variations of (2.1), we only use  $F_\epsilon^N$  to obtain an approximation of the quantile and work directly with the quantile function.

While other options are possible, our choice of the smoothed indicator function,  $\Gamma_\epsilon$ , is motivated by a kernel estimation of the cdf. In [1], the authors present the following estimator

$$\hat{F}_\epsilon^N(t; x) = \int_{-\infty}^t \frac{1}{N\epsilon} \sum_{i=1}^N \varphi\left(\frac{u - C(x, \xi_i)}{\epsilon}\right) du = \frac{1}{N} \sum_{i=1}^N \Phi\left(\frac{t - C(x, \xi_i)}{\epsilon}\right),$$

where  $\varphi(\cdot)$  is a real-valued non-negative function, known as kernel function, that integrates to one over all of its domain (features that make it a probability density function) and is symmetric around 0. The function  $\Phi(\cdot)$  is the cdf of  $\varphi(\cdot)$ .

If  $\varphi(\cdot)$  has a finite support  $[-1, 1]$ , then  $\Phi\left(\frac{t - c(x, \xi)}{\epsilon}\right)$  has the structure of the  $\Gamma_\epsilon$  function in (2.2). Furthermore, according to [2], if  $\epsilon \rightarrow 0$  as  $N \rightarrow \infty$ , it can be proven that asymptotically

$$(2.5) \quad \mathbb{E} \left[ \left( \hat{F}_\epsilon^N(t; x) - F(t; x) \right)^2 \right] \sim \beta_1 \frac{1}{N} - \beta_2 \frac{\epsilon}{N} + \beta_3 \epsilon^4,$$

where  $\beta_i, i = 1, \dots, 3$ , are non-negative constants that depend on  $t$  and  $x$ . Azzalini in [2] also shows that the properties of  $\hat{F}_\epsilon^N$  extend to the quantile estimator, i.e., to the root of  $\hat{F}_\epsilon^N(t; x) = 1 - \alpha$ . In general, these findings indicate that large values of

$\epsilon$  help reduce the variance of the estimator but increase its bias, while small values behave exactly opposite.

These observations regarding the behavior of the quantile approximation coincide with the findings of our numerical experiments. The solutions returned by the solver for large values of  $\epsilon$  are generally feasible but sub-optimal due to the bias introduced by  $\epsilon$ . Large values of  $\epsilon$  usually generate a conservative approximation of the quantile function; however, the solutions have less variance. We also find that solving the approximation with larger values of  $\epsilon$  tends to require fewer iterations. Thus, there is a trade-off in the choice of  $\epsilon$ ; small values of  $\epsilon$  reduce bias, whereas larger values of  $\epsilon$  reduce the variance and the computational effort. In [section 6.1](#), we suggest a method for selecting an appropriate value of  $\epsilon$  for an instance.

For the experiments in [section 6](#) we use the quartic kernel [\[30, p. 353\]](#) to define function  $\Gamma_\epsilon$ . This gives the following  $\gamma_\epsilon$ :

$$(2.6) \quad \gamma_\epsilon(y) = \frac{15}{16} \left( -\frac{1}{5} \left( \frac{y}{\epsilon} \right)^5 + \frac{2}{3} \left( \frac{y}{\epsilon} \right)^3 - \left( \frac{y}{\epsilon} \right) + \frac{8}{15} \right),$$

which makes  $\Gamma_\epsilon(\cdot)$  a twice continuously differentiable function.

*Note 2.3.* For simplicity of notation, we assume in the following sections that the quantile of interest is always the  $(1 - \alpha)$ -quantile and drop the superscript of the quantile approximation. Therefore, the  $(1 - \alpha)$ -quantile  $\epsilon$ -approximation is denoted just as  $Q_\epsilon$  for now onward.

**3. Convergence and feasibility of the approximation.** In this section we study how the solutions obtained by our method approximate those of the real problem [\(1.1\)](#). Throughout the rest of the paper we make the following assumption.

*Assumption 3.1.* For each  $x \in X$ , the random variable  $C(x, \xi)$  has a continuous distribution.

Notice that, if  $C(x, \xi)$  is not a continuous random variable, we can make [Assumption 3.1](#) hold by adding a continuous random variable  $\bar{\xi}$  to  $C(x, \xi)$  with expectation zero and small variance. This could be treated as a fixed approximation, or one could solve a sequence of approximations in which the variance of  $\bar{\xi}$  is decreased to zero.

Let the feasible region of the original problem [\(1.1\)](#) be defined as

$$X_\alpha = \{x \in X \mid F(0; x) \geq 1 - \alpha\},$$

where  $F(t; x) = \mathbb{P}(C(x, \xi) \leq t)$ . Moreover, define  $F_\epsilon(t; x)$ , for  $\epsilon > 0$ , as

$$(3.1) \quad F_\epsilon(t; x) = \int_{\mathbb{R}^s} \Gamma_\epsilon(C(x, \xi) - t) d\mathbb{P}_\xi,$$

where  $\Gamma_\epsilon$  is the smooth approximation of the indicator function given by [\(2.2\)](#).

Now, let  $\{\xi_i\}_{i=1}^N$  be a random sample of the variable  $\xi$  of size  $N$ . As in [section 2.1](#), we approximate  $F_\epsilon$  with  $F_\epsilon^N$ , and denote the feasible region defined by the approximation as follows

$$(3.2) \quad X_{\epsilon, \delta}^{N, t} = \{x \in X \mid F_\epsilon^N(-t; x) \geq 1 - \delta\}, \quad \delta \in [0, 1], \quad t \in \mathbb{R}.$$

Notice that if  $\delta = \alpha$  and  $t = 0$ , then  $X_{\epsilon, \alpha}^{N, 0}$  corresponds to the feasible region of [\(2.4\)](#). We consider this approximation more generally with  $\delta \leq \alpha$  and  $t \geq 0$  as we will show how the parameters  $t$  and  $\delta$  in [\(3.2\)](#) can be chosen to increase the likelihood



that the solution to the approximate model is feasible to the true model. First, notice that if  $t = \epsilon$ , then  $F_\epsilon^N(-t; x) \leq \tilde{F}^N(0; x)$ . This implies that a point  $x \in X_{\epsilon, \alpha}^{N, \epsilon}$  satisfies  $\tilde{F}^N(0; x) \geq 1 - \alpha$ , so the solutions returned by our method are feasible for the empirical approximation of the feasible region. Moreover, the  $\delta$  parameter can be used, as in [21], to ensure that the solutions obtained are feasible with high probability.

We start the analysis of (2.4) by showing asymptotic convergence of the optimal value of the approximation to the optimal value of the true problem. In particular, we analyze convergence of the approximation with respect to both the smoothing parameter and the sample size, whereas previous results, to the best of our knowledge, only consider convergence with respect to the smoothing parameter ([13, 18, 19]) or the sample size ([21, 27]) separately. Then, we show complementary analysis of the approximation problem that indicates that, with appropriate choice of parameters, any feasible solution to the approximation problem will be feasible with high probability for sufficiently large sample size.

**3.1. Asymptotic convergence.** In the following theorems we state results analogous to Proposition 2.1 and 2.2 in [27], thus the proofs we present are similar to those in that paper. We make the following assumptions.

*Assumption 3.2.*  $C(x, \xi)$  is a Carathéodory function, i.e.,  $C(x, \cdot)$  is measurable for every  $x \in X$  and  $C(\cdot, \xi)$  continuous for a.e.  $\xi \in \Xi$ .

*Assumption 3.3.* There exists an optimal solution of (1.1),  $\bar{x}$ , such that for any  $\delta > 0$  there is  $x \in X$  such that  $\|x - \bar{x}\| \leq \delta$  and  $F(0; x) > 1 - \alpha$ .

This assumption implies that there exists a sequence  $\{x_k\}_{k=1}^\infty \subseteq X$  that converges to an optimal solution  $\bar{x}$  such that  $F(0; x_k) > 1 - \alpha$  for all  $k \in \mathbb{N}$ .

**THEOREM 3.4.** *Suppose that Assumptions 3.1 and 3.2 hold. Then,  $F_\epsilon^N(0; x)$  converges to  $F(0; x)$  uniformly on any compact set  $U \subseteq X$  w.p.1, i.e.,*

$$\sup_{x \in U} |F_\epsilon^N(0; x) - F(0; x)| \rightarrow 0, \quad \text{w.p.1, as } N \rightarrow \infty \text{ and } \epsilon \rightarrow 0.$$

*Proof.* We prove uniform convergence of  $F_\epsilon^N(0; x)$  to  $F(0; x)$  by first showing uniform convergence of  $F_\epsilon^N(0; x)$  to  $F_\epsilon(0; x)$  over  $x \in U$  and  $\epsilon \in [0, 1]$  as  $N \rightarrow \infty$ , and then uniform convergence of  $F_\epsilon(0; x)$  to  $F(0; x)$  over  $x \in U$  as  $\epsilon \rightarrow 0$ .

To prove uniform convergence of  $F_\epsilon^N(0; x)$  to  $F_\epsilon(0; x)$  in the compact set  $U \subseteq X$  for all  $\epsilon \in [0, 1]$ , we start by showing that  $\Gamma_\epsilon(C(x, \xi))$  is continuous w.p.1 for all  $x \in U$  and  $\epsilon \in [0, 1]$ . Notice that  $\Gamma_\epsilon(C(x, \xi))$  is continuous w.p.1 for all  $x \in U$  and  $\epsilon > 0$ . This follows because  $\Gamma_\epsilon(\cdot)$  is continuous and  $C(x, \xi)$  is a Carathéodory function. It remains to prove that  $\Gamma_\epsilon(C(x, \xi))$  is continuous in  $x$  and  $\epsilon$  at  $\epsilon = 0$  w.p.1. Let  $x$  and  $\xi$  be such that  $C(x, \xi)$  is continuous and strictly positive. Then, there exists  $\epsilon_0 > 0$  and  $\delta_0 > 0$  such that  $C(x, \xi) > \epsilon_0$  and  $C(y, \xi) > \epsilon_0$  for all  $y \in \mathcal{B}_{\delta_0}(x)$  (where  $\mathcal{B}_{\delta_0}(x) = \{y \mid \|x - y\| \leq \delta_0\}$ ). Thus, for all  $\rho > 0$ ,  $|\Gamma_0(C(x, \xi)) - \Gamma_\epsilon(C(y, \xi))| = 0 < \rho$  for all  $0 \leq \epsilon \leq \epsilon_0$  and  $y \in \mathcal{B}_{\delta_0}(x)$ . This shows that  $\Gamma_0(C(x, \xi))$  is continuous when  $C(x, \xi) > 0$ . An analogous proof can be derived if  $C(x, \xi)$  is strictly negative. Finally, because  $C(x, \xi) \neq 0$  w.p.1 for all  $x \in X$  from Assumption 3.1, we conclude that  $\Gamma_\epsilon(C(x, \xi))$  is continuous w.p.1 for all  $x \in U$  and  $\epsilon \in [0, 1]$ .

Uniform convergence of  $F_\epsilon^N(0; x)$  to  $F_\epsilon(0; x)$  over  $x \in U$  and  $\epsilon \in [0, 1]$  is then a result of applying Theorem 7.53 in [31]. This is because function  $\Gamma_\epsilon(C(x, \xi))$  is continuous w.p.1 for all  $x \in U$  and  $\epsilon \in [0, 1]$ ,  $|\Gamma_\epsilon(C(x, \xi))| \leq 1$  for all  $x \in X$ , and the sample is i.i.d.

Uniform convergence of  $F_\epsilon(0; x)$  to  $F(0; x)$  in the compact set  $U \subseteq X$  as  $\epsilon \rightarrow 0$  is a result of applying Theorem 4.91 in [31]. This comes from Assumptions 3.1 and 3.2.

Combining these two results, for every  $\delta > 0$ , we can find  $\epsilon_\delta > 0$  and  $N_\delta \in \mathbb{N}$  such that  $|F_\epsilon(0; x) - F(0; x)| < \delta/2$  and  $|F_\epsilon^N(0; x) - F_\epsilon(0; x)| < \delta/2$ , w.p.1, for all  $x \in U$ ,  $\epsilon \leq \min\{1, \epsilon_\delta\}$ , and  $N \geq N_\delta$ , which ensures that

$$\sup_{x \in U} |F_\epsilon^N(0; x) - F(0; x)| < \delta.$$

We can therefore conclude that  $F_\epsilon^N(0; x)$  converges to  $F(0; x)$  uniformly in the compact set  $U$ .  $\square$

We denote the sets of optimal solutions for problems (1.1) and (2.4) as  $S$  and  $S_\epsilon^N$ , respectively. We also denote  $v^*$  and  $v_\epsilon^N$  the optimal values of (1.1) and (2.4), respectively. We show next that, under some assumptions,  $v_\epsilon^N$  and  $S_\epsilon^N$  converge w.p.1 to their counterparts of the true problem as  $N$  increases and  $\epsilon$  decreases.

**THEOREM 3.5.** *Suppose that  $X$  is compact, the function  $f$  is continuous, and Assumptions 3.1 to 3.3 hold. Then,  $v_\epsilon^N \rightarrow v^*$  and  $\mathbb{D}(S_\epsilon^N, S) \rightarrow 0$  w.p.1 as  $N \rightarrow \infty$  and  $\epsilon \rightarrow 0$ .*

*Proof.* Notice that for any given  $N$  and  $\epsilon$ , a point  $x$  in the feasible set of (2.4) satisfies  $F_\epsilon^N(0; x) \geq 1 - \alpha$ . Thus, we can show convergence of the feasible set of (2.4) to the feasible set of (1.1) by analyzing the function  $F_\epsilon^N$ .

Now, by Assumption 3.3, the set  $S$  is nonempty and there is  $x \in X$  such that  $F(0; x) > 1 - \alpha$ . Because  $F_\epsilon^N(0; x)$  converges to  $F(0; x)$ , by Theorem 3.4, there exists  $\epsilon_0$  small enough and  $N_0$  large enough such that  $F_{\epsilon_0}^{N_0}(0; x) \geq 1 - \alpha$  w.p.1. Because  $F_{\epsilon_0}^{N_0}$  is continuous in  $x$  and  $X$  is compact, the feasible set of the approximation problem (2.4) is compact as well, and hence  $S_\epsilon^N$  is nonempty w.p.1 for all  $N \geq N_0$  and  $\epsilon \leq \epsilon_0$ .

Let  $\{N_k\}_{k=1}^\infty \geq N_0$  and  $\{\epsilon_k\}_{k=1}^\infty \leq \epsilon_0$  be two sequences such that  $N_k \rightarrow \infty$  and  $\epsilon_k \rightarrow 0$ . Let  $\hat{x}_k \in S_{\epsilon_k}^{N_k}$ , i.e.,  $\hat{x}_k \in X$ ,  $F_{\epsilon_k}^{N_k}(0; \hat{x}_k) \geq 1 - \alpha$  and  $v_{\epsilon_k}^{N_k} = f(\hat{x}_k)$ . Let  $\hat{x} \in X$  be any limit point of  $\{\hat{x}_k\}_{k=1}^\infty$  and  $\{\hat{x}_l\}_{l=1}^\infty$  be a subsequence converging to  $\hat{x}$ . Because  $F_{\epsilon_l}^{N_l}(0; x)$  is continuous and converges uniformly to  $F(0; x)$  on  $X$  w.p.1 by Theorem 3.4, we have that  $F(0; \hat{x}) = \lim_{l \rightarrow \infty} F_{\epsilon_l}^{N_l}(0; \hat{x}_l)$ , w.p.1. Hence,  $F(0; \hat{x}) \geq 1 - \alpha$  and  $\hat{x}$  is feasible for the true problem, and therefore  $f(\hat{x}) \geq v^*$ . Also  $f(\hat{x}_l) \rightarrow f(\hat{x})$  w.p.1 which means that  $\lim_{l \rightarrow \infty} v_{\epsilon_l}^{N_l} \geq v^*$  w.p.1. Since this is true for any limit point of  $\{\hat{x}_k\}_{k=1}^\infty$  in the compact set  $X$ , we have

$$(3.3) \quad \liminf_{k \rightarrow \infty} v_{\epsilon_k}^{N_k} \geq v^*, \quad \text{w.p.1.}$$

Now, by Assumption 3.3, there exists an optimal solution  $\bar{x}$  and a sequence  $\{x_l\}_{l=1}^\infty$  converging to  $\bar{x}$  with  $F(0; x_l) > 1 - \alpha$ . Since  $F_{\epsilon_k}^{N_k}(0; x_l)$  converges to  $F(0; x_l)$  w.p.1, there exist  $K(l)$  such that  $F_{\epsilon_k}^{N_k}(0; x_l) \geq 1 - \alpha$  for every  $k \geq K(l)$  and every  $l$ , w.p.1. Without loss of generality we can assume that  $K(l) < K(l+1)$  for every  $l$  and define the sequence  $\{\bar{x}_k\}_{k=K(1)}^\infty$  by setting  $\bar{x}_k = x_l$  for all  $k$  and  $l$  with  $K(l) \leq k < K(l+1)$ . We then have  $F_{\epsilon_k}^{N_k}(0; \bar{x}_k) \geq 1 - \alpha$ , which implies  $v_{\epsilon_k}^{N_k} \leq f(\bar{x}_k)$  for all  $k \geq K(1)$ . Since  $f$  is continuous, we have that

$$(3.4) \quad \limsup_{k \rightarrow \infty} v_{\epsilon_k}^{N_k} \leq f(\bar{x}) = v^*, \quad \text{w.p.1.}$$

It follows from (3.3) and (3.4) that  $v_{\epsilon_k}^{N_k} \rightarrow v^*$  w.p.1. To show that  $\mathbb{D}(S_\epsilon^N, S) \rightarrow 0$  w.p.1. we defer to Theorem 5.3 in [31] which can easily be adapted to our context.  $\square$

**3.2. Finite sample feasibility of approximation problem solutions.** We now consider conditions under which an optimal solution for the approximation problem (2.4), if it exists, is feasible for the true problem (1.1). These results generalize those presented in Section 2.2 in [21], which considers the empirical cdf instead of a smooth approximation; therefore, some of our proofs follow from the ones presented there.

We will make use of Hoeffding's inequality.

**THEOREM 3.6** (Hoeffding's Inequality [17]).

Let  $Y_1, \dots, Y_N$  be independent random variables, with  $\mathbb{P}(Y_i \in [a_i, b_i]) = 1$ , where  $a_i \leq b_i$  for  $i = 1, \dots, N$ . Then if  $t > 0$ ,

$$\mathbb{P} \left\{ \sum_{i=1}^N (Y_i - \mathbb{E}[Y_i]) \geq tN \right\} \leq \exp \left\{ -\frac{2N^2 t^2}{\sum_{i=1}^N (b_i - a_i)^2} \right\}.$$

*Assumption 3.7.* Let the scalars  $t \in \mathbb{R}$  and  $\delta \in [0, \alpha]$  be chosen such that  $M := \inf_{x \in X} M_x > 0$ , where  $M_x = F(0; x) - F_\epsilon(-t; x) + (\alpha - \delta)$ .

The above assumption can always be satisfied if we pick  $t = \epsilon$  and  $\delta < \alpha$ . This claim can easily be observed from the definition of  $F_\epsilon(-t; x)$  in (3.1), which implies that  $F_\epsilon(-\epsilon; x) \leq F(0; x)$  for all  $x \in X$ .

**THEOREM 3.8** (Probabilistic feasibility guarantee).

Let  $x \in X$  be such that  $x \notin X_\alpha$ , and suppose that *Assumption 3.7* holds. Then

$$\mathbb{P} (F_\epsilon^N(-t; x) \geq 1 - \delta) \leq \exp \{-2NM_x^2\}.$$

*Proof.* Let  $Y_i = \Gamma_\epsilon(C(x, \xi_i) + t)$  for all  $i = 1, \dots, N$ , then  $\mathbb{P}(Y_i \in [0, 1]) = 1$  and  $\mathbb{E}[Y_i] = F_\epsilon(-t; x)$ . Given that  $x$  is not a feasible solution for the original problem,  $F(0; x) < 1 - \alpha$ . We therefore have

$$\begin{aligned} \mathbb{P} (F_\epsilon^N(-t; x) \geq 1 - \delta) &= \mathbb{P} (F_\epsilon^N(-t; x) - F_\epsilon(-t; x) \geq 1 - \alpha + \alpha - \delta - F_\epsilon(-t; x)) \\ &\leq \mathbb{P} (F_\epsilon^N(-t; x) - F_\epsilon(-t; x) > F(0; x) - F_\epsilon(-t; x) + (\alpha - \delta)) \\ &= \mathbb{P} (F_\epsilon^N(-t; x) - F_\epsilon(-t; x) > M_x) \\ &= \mathbb{P} \left( \sum_{i=1}^N (Y_i - \mathbb{E}[Y_i]) \geq M_x N \right) \\ &\leq \exp \left\{ -\frac{2N^2 M_x^2}{\sum_{i=1}^N (1 - 0)^2} \right\} = \exp \{-2NM_x^2\}, \end{aligned}$$

where the last inequality is obtained from Hoeffding's Inequality.  $\square$

**Theorem 3.8** shows that if we shift the approximation of the cdf appropriately and/or decrease the risk level from  $\alpha$  to  $\delta \leq \alpha$ , then a point  $x$  that is feasible for the original problem (1.1) is also in  $X_{\epsilon, \delta}^{N, t}$  with a probability that increases exponentially with the size of the sample  $N$ . Furthermore, under certain conditions, we can set  $t = 0$  and  $\delta = \alpha$  and still get the results stated above. This can be seen in the following corollary.

**COROLLARY 3.9.** Let  $x \in X$  be such that  $Y_x = C(x, \xi)$  is a continuous random variable with a strictly decreasing probability density function (pdf)  $h_x(y)$  in the interval  $(-\epsilon, \epsilon)$ . If  $F(0; x) < 1 - \alpha$ , then

$$\mathbb{P} (F_\epsilon^N(0; x) \geq 1 - \alpha) \leq \exp \{-2N\beta_x^2\},$$

where  $\beta_x = F(0; x) - F_\epsilon(0; x) > 0$ .

*Proof.* This follows from [Theorem 3.8](#) for  $t = 0$ ,  $\delta = \alpha$ , and noticing that  $\beta_x = M_x = F(0; x) - F_\epsilon(0; x) > 0$  because

$$\begin{aligned}
F(0; x) - F_\epsilon(0; x) &= \int_{\mathbb{R}^s} \mathbb{1}(C(x, \xi)) d\mathbb{P}_\xi - \int_{\mathbb{R}^s} \Gamma_\epsilon(C(x, \xi)) d\mathbb{P}_\xi \\
&= \int_{-\infty}^{\infty} \mathbb{1}(y) h_x(y) dy - \int_{-\infty}^{\infty} \Gamma_\epsilon(y) h_x(y) dy \\
&= \int_{-\epsilon}^0 (1 - \Gamma_\epsilon(y)) h_x(y) dy - \int_0^\epsilon \Gamma_\epsilon(y) h_x(y) dy \\
&= \int_{-\epsilon}^0 \Gamma_\epsilon(-y) h_x(y) dy - \int_0^\epsilon \Gamma_\epsilon(y) h_x(y) dy \\
&= \int_0^\epsilon \Gamma_\epsilon(y) h_x(-y) dy - \int_0^\epsilon \Gamma_\epsilon(y) h_x(y) dy \\
&= \int_0^\epsilon \Gamma_\epsilon(y) [h_x(-y) - h_x(y)] dy > 0.
\end{aligned}$$

The second equality is the result of defining  $\Gamma_\epsilon$  as the integral of a symmetric pdf, which implies that  $1 - \Gamma_\epsilon(y) = \Gamma_\epsilon(-y)$ . The last inequality is due to the fact that the pdf of  $Y_x$  is strictly decreasing in the interval  $(-\epsilon, \epsilon)$ , which implies that  $h_x(-y) > h_x(y)$  for  $y \in (0, \epsilon)$ , and due to  $\Gamma_\epsilon(y) = \gamma_\epsilon(y) > 0$  for  $y \in [0, \epsilon)$ .  $\square$

One of the properties that follows from [Corollary 3.9](#) is that, for  $\epsilon > 0$ , our approximation is asymptotically conservative. In other words, for a fixed  $\epsilon > 0$  and under a monotonicity assumption, a point  $x$  that is feasible for the original problem [\(1.1\)](#) is also feasible for the approximation problem [\(2.4\)](#) with a probability that increases exponentially with the sample size  $N$ . This result holds for the vast majority of the common continuous distributions such as exponential, normal and Weibull, to mention a few, when  $\alpha$  is small.

**COROLLARY 3.10.** *Assume that  $\Gamma_\epsilon$  is defined as in [\(2.2\)](#), where  $\gamma_\epsilon(\cdot)$  is given by [\(2.6\)](#). Let  $x \in X$  be such that  $Y_x = C(x, \xi)$  is a continuous random variable with pdf  $h_x(y)$ . If there exists  $m_1, m_2 > 0$  such that  $h_x(y) \geq -m_1 y + h_x(0)$  for all  $y \in (-\epsilon, 0]$ , and  $h_x(y) \leq -m_2 y + h_x(0)$  for all  $y \in [0, \epsilon)$ . Then, we have that*

$$\beta_x \geq \frac{(m_1 + m_2)\epsilon^2}{28},$$

and thus, for all  $x \notin X_\alpha$ ,

$$\mathbb{P}(F_\epsilon^N(0; x) \geq 1 - \alpha) \leq \exp\{-N(m_1 + m_2)^2 \epsilon^4 / 14\}.$$

*Proof.* From the definition of  $\beta_x = F(0; x) - F_\epsilon(0; x)$ , we get

$$\begin{aligned}
F(0; x) - F_\epsilon(0; x) &= \int_0^\epsilon \Gamma_\epsilon(y) [h_x(-y) - h_x(y)] dy \\
&\geq \int_0^\epsilon \Gamma_\epsilon(y) [(m_1 + m_2)y] dy \\
&= \frac{(m_1 + m_2)\epsilon^2}{28},
\end{aligned}$$

where  $\int_0^\epsilon \Gamma_\epsilon(y) y dy = \int_0^\epsilon \gamma_\epsilon(y) y dy = \epsilon^2 / 28$ .  $\square$

We now consider conditions under which the entire feasible region of the approximation problem is a subset of the feasible region of problem (1.1). For these results to hold we need to make additional assumptions. We begin with a preliminary result that holds under the assumption that the feasible region  $X$  is finite.

**THEOREM 3.11.** *Suppose that  $X$  is finite, and that [Assumption 3.7](#) holds. Then*

$$\mathbb{P}\left(X_{\epsilon,\delta}^{N,t} \subseteq X_\alpha\right) \geq 1 - |X \setminus X_\alpha| \exp\{-2NM^2\}.$$

*Proof.* Using the result from [Theorem 3.8](#), we get the following

$$\begin{aligned} \mathbb{P}\left(X_{\epsilon,\delta}^{N,t} \not\subseteq X_\alpha\right) &= \mathbb{P}\left(\exists x \in X_{\epsilon,\delta}^{N,t} \text{ such that } x \notin X_\alpha\right) \\ &\leq \sum_{x \in X \setminus X_\alpha} \mathbb{P}\left(x \in X_{\epsilon,\delta}^{N,t}\right) \leq \sum_{x \in X \setminus X_\alpha} \exp(-2NM_x^2) \\ &\leq |X \setminus X_\alpha| \exp\{-2NM^2\}. \quad \square \end{aligned}$$

We now extend this result to the case in which  $X$  is not finite. For this result, we make the following Lipschitz continuity assumption on  $C$ .

**Assumption 3.12.** There exists  $L > 0$  such that

$$|C(x, \xi) - C(y, \xi)| \leq L\|x - y\|_\infty, \quad \forall x, y \in X, \text{ w.p.1.}$$

This assumption was made in [21]. The proof stated below follows the same ideas as Theorem 10 in [21].

**THEOREM 3.13.** *Assume that  $X$  is bounded with diameter  $D$  and [Assumption 3.12](#) holds. Also, suppose that [Assumption 3.7](#) holds with  $t > 0$  and  $\delta > 0$ . Then, for any  $\beta \in (0, \delta]$ , the following is true:*

$$\mathbb{P}\left(X_{\epsilon,(\delta-\beta)}^{N,2t} \subseteq X_\alpha\right) \geq 1 - \lceil 1/\beta \rceil \lceil 2LD/t \rceil^n \exp\{-2NM^2\}.$$

*Proof.* Let  $J = \lceil 1/\beta \rceil$ . For  $j = 1, \dots, J-1$ , define

$$X_j = \left\{x \in X \mid \frac{j-1}{J} \leq F(0; x) < \frac{j}{J}\right\},$$

and let  $X_J = \{x \in X \mid (J-1)/J \leq F(0; x) \leq 1\}$ . Following the arguments used to prove Theorem 10 in [21], we have that for each  $j$  there exists a finite set  $Z_j \subseteq X_j$  such that  $|Z_j| \leq \lceil 2LD/t \rceil^n$  and that for all  $x \in X_j$  there exists  $z \in Z_j$  such that  $\|x - z\|_\infty \leq t/L$ .

Now define  $Z = \cup_{j=1}^J Z_j$ , and notice that  $|Z| \leq \lceil 1/\beta \rceil \lceil 2LD/t \rceil^n$ . Next, define  $Z_{\alpha-\beta} = \{x \in Z \mid F(0; x) \geq 1 - \alpha + \beta\}$  and

$$Z_{\delta-\beta}^{N,t} = \{x \in Z \mid F_\epsilon^N(-t; x) \geq 1 - \delta + \beta\}.$$

Since  $Z$  is finite, we can apply [Theorem 3.11](#) to obtain

$$(3.5) \quad \mathbb{P}\left(Z_{\delta-\beta}^{N,t} \subseteq Z_{\alpha-\beta}\right) \geq 1 - \lceil 1/\beta \rceil \lceil 2LD/t \rceil^n \exp\{-2NM^2\}.$$

Consider now any  $x \in X_{\epsilon,(\delta-\beta)}^{N,2t}$ . Let  $j \in \{1, \dots, J\}$  be such that  $x \in X_j$ . By definition of  $Z_j$ , there exists  $z \in Z_j$  such that  $\|x - z\|_\infty \leq t/L$ . By [Assumption 3.12](#),

we know that  $|C(x, \xi_i) - C(z, \xi_i)| \leq t$  w.p.1. Therefore,  $C(z, \xi_i) + t \leq C(x, \xi_i) + 2t$ , which implies that  $\Gamma_\epsilon(C(z, \xi_i) + t) \geq \Gamma_\epsilon(C(x, \xi_i) + 2t)$  w.p.1, and thus  $F_\epsilon^N(-t; z) \geq F_\epsilon^N(-2t; x) \geq 1 - \delta + \beta$  w.p.1. We conclude that if  $x \in X_{\epsilon, (\delta - \beta)}^{N, 2t}$ , then there exists  $z \in Z_{\delta - \beta}^{N, t}$  w.p.1. Furthermore, by the definition of  $X_j$  and because  $Z_j \subseteq X_j$ , we have  $|F(0; x) - F(0; z)| \leq \beta$ . If  $Z_{\delta - \beta}^{N, t} \subseteq Z_{\alpha - \beta}$  then  $F(0; z) \geq 1 - \alpha + \beta$ , which implies that  $F(0; x) \geq F(0; z) - \beta \geq 1 - \alpha$ , and the result follows.  $\square$

**4. Single Chance-Constrained Problems.** A single chance constrained problem is one in which  $m = 1$ , which means that  $C(x, \xi) = c(x, \xi)$ . Defining  $C^N(x) = c^N(x) = [c(x, \xi_1), \dots, c(x, \xi_N)]^T$ , problem (2.4) then becomes

$$(4.1a) \quad \min_{x \in X} f(x)$$

$$(4.1b) \quad \text{s.t. } q(x) = Q_\epsilon(c^N(x)) \leq 0.$$

The advantage of the single-chance constrained case is that the probabilistic constraint can be formulated in such a way that it results in a single smooth constraint (assuming  $c$  is smooth in  $x$ ) that can be handled by standard nonlinear optimization solvers and can be added to any nonlinear optimization problem. Furthermore, for a given  $x$  and  $\epsilon > 0$ , we can use the implicit function theorem to obtain the gradient of  $q$  (and provide it to the solver) as follows

$$(4.2) \quad \nabla q(x) = \frac{\sum_{i=1}^N \Gamma'_\epsilon(c(x, \xi_i) - Q_\epsilon(c^N(x))) \nabla c(x, \xi_i)}{\sum_{j=1}^N \Gamma'_\epsilon(c(x, \xi_j) - Q_\epsilon(c^N(x)))},$$

which is well defined as shown in Proposition 2.1.

In fact, the following proposition shows that if we define  $\gamma_\epsilon$  as (2.6), then function  $Q_\epsilon(\cdot)$  has Lipschitz continuous gradients. Using the chain rule, this property can be extended to  $q(x)$  as long as  $c(x, \xi)$  has Lipschitz continuous gradients for all instances  $\xi_i$ .

**PROPOSITION 4.1.** *If  $\gamma_\epsilon$  is defined as (2.6) and  $(1 - \alpha)N \notin \mathbb{Z}$ , then the function  $Q_\epsilon : \mathbb{R}^N \rightarrow \mathbb{R}$  has bounded Lipschitz continuous gradients.*

*Proof.* Let  $z \in \mathbb{R}^N$ . It follows from Proposition 2.1 and the implicit function theorem that

$$[\nabla Q_\epsilon(z)]_i = \frac{\Gamma'_\epsilon(z_i - Q_\epsilon(z))}{\sum_{j=1}^N \Gamma'_\epsilon(z_j - Q_\epsilon(z))}.$$

We begin by showing that  $\sum_{j=1}^N \Gamma'_\epsilon(z_j - Q_\epsilon(z))$  is bounded away from zero. We prove this claim by contradiction. Assume that there exists  $z \in \mathbb{R}^N$  such that  $|z_j - Q_\epsilon(z)| \geq \epsilon$  for all  $j \in \{1, \dots, N\}$ . Then, from (2.2),  $\Gamma_\epsilon(z_j - Q_\epsilon(z)) \in \mathbb{N}$  for all  $j$ . This implies that  $\sum_{j=1}^N \Gamma_\epsilon(z_j - Q_\epsilon(z)) \in \mathbb{N}$ , which means that  $\sum_{j=1}^N \Gamma_\epsilon(z_j - Q_\epsilon(z)) \neq (1 - \alpha)N$ . Therefore, there must exist  $i \in \{1, \dots, N\}$  and  $0 < \epsilon_0 < \epsilon$  such that  $|z_i - Q_\epsilon(z)| \leq \epsilon_0$  and  $\sum_{j=1}^N \Gamma'_\epsilon(z_j - Q_\epsilon(z)) \leq \Gamma'_\epsilon(z_i - Q_\epsilon(z)) \leq \gamma'_\epsilon(\epsilon_0) < 0$ , for all  $z \in \mathbb{R}^N$ .

The above result shows that  $0 \leq [\nabla Q_\epsilon(z)]_i \leq 1$  for all  $i$ , which means that  $\nabla Q_\epsilon(z)$  is bounded. Furthermore, because  $\Gamma'_\epsilon$  exists and is bounded, then  $\Gamma'_\epsilon$  is Lipschitz continuous. Given that the denominator is bounded away from zero, we can conclude that  $Q_\epsilon(z)$  has Lipschitz continuous gradients.  $\square$

If we also assume that  $f$  and  $c$  are twice continuously differentiable, then we can provide the Hessian of constraint (4.1b) to the solver as well. The Hessian is then computed as

$$(4.3) \quad \nabla^2 q(x) = \sum_{i=1}^N [\nabla Q_\epsilon(c^N(x))]_i [\nabla^2 c(x, \xi_i)] + \nabla c^N(x) [\nabla^2 Q_\epsilon(c^N(x))] [\nabla c^N(x)]^T,$$

where  $[\nabla Q_\epsilon(\cdot)]_i$  is the  $i$ th entry of the gradient of  $Q_\epsilon(\cdot)$ , and  $\nabla^2 Q_\epsilon(\cdot)$  is the Hessian of the smooth approximation of the quantile with respect to each scenario, i.e.,  $\nabla^2 Q_\epsilon(\cdot) \in \mathbb{R}^{N \times N}$  is such that

$$\begin{aligned} [\nabla^2 Q_\epsilon(z)]_{ii} &= \frac{w_i^2 W' - 2w_i w'_i W + w'_i W^2}{W^3}, \\ [\nabla^2 Q_\epsilon(z)]_{ij} &= \frac{w_i w_j W' - W(w_i w'_j + w'_i w_j)}{W^3}, \quad i \neq j, \end{aligned}$$

where  $w_\ell = \Gamma'_\epsilon(z_\ell - Q(z))$ ,  $w'_\ell = \Gamma''_\epsilon(z_\ell - Q(z))$ ,  $W = \sum_{\ell=1}^N w_\ell$ , and  $W' = \sum_{\ell=1}^N w'_\ell$ .

In section 6.2 we discuss the performance of this approximation. We give two examples in which the nonlinear constraint  $q(x)$  and its derivatives are given directly to `Knitro`.

**5. Joint Chance-Constrained Problems.** In this section, we consider problems where  $c(x, \xi)$  is a vector-valued function from  $\mathbb{R}^n \times \mathbb{R}^s$  to  $\mathbb{R}^m$  for  $m > 1$ , and therefore  $C(x, \xi) = \max_{j=1, \dots, m} \{c_j(x, \xi)\}$  is not smooth even when  $c_j(x, \xi)$  are smooth for  $j = 1, \dots, m$ .

Assuming that there exists a deterministic differentiable function,  $g : \mathbb{R}^n \rightarrow \mathbb{R}^p$ , such that the deterministic feasible set  $X$  can be written as  $X = \{x \in \mathbb{R}^n \mid g(x) \leq 0\}$ , we can rewrite the approximation of (2.4) as

$$\begin{aligned} (5.1a) \quad & \min_{x \in \mathbb{R}^n} f(x) \\ (5.1b) \quad & \text{s.t. } g(x) \leq 0, \\ (5.1c) \quad & Q_\epsilon(C^N(x)) \leq 0. \end{aligned}$$

The discussion in this section can be extended to solve problems with deterministic equality constraints in the definition of the set  $X$ , but we consider only inequality constraints for a succinct exposition.

Notice that, because  $C(x, \xi)$  is no longer smooth, the constraint (5.1c) is not differentiable. To obtain a smooth reformulation of the quantile constraint (5.1c), we introduce auxiliary variables  $z_i$  that correspond to  $\max_j \{c_j(x, \xi_i)\}$  and consider

$$\begin{aligned} (5.2a) \quad & \min_{x \in \mathbb{R}^n, z \in \mathbb{R}^N} f(x) \\ (5.2b) \quad & \text{s.t. } g(x) \leq 0, \\ (5.2c) \quad & c(x, \xi_i) \leq z_i \mathbf{1}_m, \quad \forall i = \{1, \dots, N\} \\ (5.2d) \quad & Q_\epsilon(z) \leq 0, \end{aligned}$$

where  $\mathbf{1}_m$  is a vector of ones of size  $m$ .

It is clear that a local minimum of (5.1) is also a local minimum of (5.2) if we define  $z_i^* = [C^N(x^*)]_i$ . However, a local minimum of (5.2) is not necessarily a local

minimum of (5.1) if  $z_i^* \neq [C^N(x^*)]_i$ . Thus, the downside of formulating the JCCP as the SCCP in (5.2) is that a standard NLP solver might get stuck at a spurious local minimum. We therefore introduce a tailored algorithm that is designed to solve JCCPs.

**5.1. Exact-penalty formulation.** In order to solve problem (5.1), we write an equivalent unconstrained optimization problem in which the constraints are handled by an  $\ell_1$ -penalty term. Our approach is motivated by the  $S\ell_1$ QP method for regular NLPs [26]. We define the penalty function

$$(5.3) \quad \phi_\pi(x) = f(x) + \pi \left( \left\| [g(x)]^+ \right\|_1 + [Q_\epsilon(C^N(x))]^+ \right),$$

where  $\pi > 0$  is a penalty parameter and  $[x]_i^+ = \max\{0, x_i\}$ . If  $x^*$  is a minimizer of  $\phi_\pi$  for  $\pi > 0$  and  $\left\| [g(x)]^+ \right\|_1 + [Q_\epsilon(C^N(x))]^+ = 0$ , then  $x^*$  solves (5.1) [16, p. 299]. The following theorem, stated in [10, Theorem 2.1], considers the converse situation and shows that there exists  $\pi > 0$  such that we can use the minimization of (5.3) to obtain a solution for (5.1).

**THEOREM 5.1.** *Suppose  $f$ ,  $g$ ,  $c$ , and  $Q_\epsilon$  are locally Lipschitz on  $\mathbb{R}^n$  and let  $x^*$  be a local minimizer of (5.1). Moreover, suppose that the Mangasarian-Fromowitz condition holds for problem (5.2) at  $x^*$  and  $z^* = C^N(x^*)$ . Then, there exists  $\pi^* > 0$  and  $\delta > 0$  such that for all  $x \in \mathbb{B}_\delta(x^*)$ ,*

$$\phi_\pi(x) \geq \phi_\pi(x^*) \quad \text{for all } \pi \geq \pi^*.$$

To minimize function (5.3), we propose a trust-region method that generates steps from the minimization of a piecewise quadratic model of  $\phi_\pi$  in every iteration. By making sure that the model approximates  $\phi_\pi$  to first-order, we can invoke existing convergence results that show that the trust-region method converges to a stationary point of  $\phi_\pi$ . A piecewise quadratic model with such characteristics is defined in the following proposition.

**PROPOSITION 5.2.** *Let  $f$  and  $g$  be differentiable with Lipschitz continuous gradients. Also, given a sample  $\{\xi_1, \dots, \xi_N\}$ , let  $c(x, \xi_i)$  be differentiable with respect to  $x$  for all  $i \in \{1, \dots, N\}$  and assume there exists  $L > 0$  and  $L' > 0$  such that*

$$(5.4) \quad |c(x, \xi_i) - c(y, \xi_i)| \leq L\|x - y\|, \quad \forall x, y \in X, \quad \forall i \in \{1, \dots, N\},$$

$$(5.5) \quad |\nabla c(x, \xi_i) - \nabla c(y, \xi_i)| \leq L'\|x - y\|, \quad \forall x, y \in X, \quad \forall i \in \{1, \dots, N\}.$$

*Given a point  $x \in \mathbb{R}^n$  and a direction  $d \in \mathbb{R}^n$ , consider the function*

$$(5.6) \quad m(x, H; d) = f(x) + \nabla f(x)^T d + \frac{1}{2} d^T H d + \pi \left( \left\| [g(x) + \nabla g(x)^T d]^+ \right\|_1 + [\tilde{Q}_{\epsilon, C}(x; d)]^+ \right),$$

*where  $H \in \mathbb{R}^{n \times n}$  is a symmetric matrix, and*

$$\begin{aligned} \tilde{Q}_{\epsilon, C}(x; d) &= \tilde{Q}_\epsilon(C^N(x); \tilde{C}^N(x; d) - C^N(x)), \\ \tilde{Q}_\epsilon(z; p) &= Q_\epsilon(z) + \nabla Q_\epsilon(z)^T p, \\ [\tilde{C}^N]_i(x; d) &= \max_j \{c_j(x, \xi_i) + \nabla c_j(x, \xi_i)^T d\}. \end{aligned}$$



Then,  $m(x, H; d)$  is a first-order model of  $\phi_\pi(x + d)$  in the sense that

$$|\phi_\pi(x + d) - m(x, H; d)| = \mathcal{O}(\|d\|^2),$$

for all  $d, x \in \mathbb{R}^n$  and all  $H$  with  $\|H\| \leq L_H$ , for fixed  $L_H$ .

*Proof.* We only prove that  $|Q_\epsilon(C^N(x + d)) - \tilde{Q}_{\epsilon, C}(x; d)| = \mathcal{O}(\|d\|^2)$  since the rest follows from the Lipschitz continuity of  $\nabla f$  and  $\nabla g$ , and boundedness of  $H$ .

For brevity in the proof, we will use the following notation  $C_y = C^N(y)$  and  $\tilde{C}_{(x; d)} = \tilde{C}^N(x; d)$ . We have that,

$$\begin{aligned} \left| Q_\epsilon(C_{x+d}) - \tilde{Q}_{\epsilon, C}(x; d) \right| &= \left| Q_\epsilon(C_{x+d}) - \tilde{Q}_\epsilon(C_x; \tilde{C}_{(x; d)} - C_x) \right| \\ &\leq \left| Q_\epsilon(C_{x+d}) - \tilde{Q}_\epsilon(C_x; C_{x+d} - C_x) \right| + \left| \tilde{Q}_\epsilon(C_x; C_{x+d} - C_x) - \tilde{Q}_\epsilon(C_x; \tilde{C}_{(x; d)} - C_x) \right|. \end{aligned}$$

Analyzing the first term, we get

$$\begin{aligned} \left| Q_\epsilon(C_{x+d}) - \tilde{Q}_\epsilon(C_x; C_{x+d} - C_x) \right| &= \left| Q_\epsilon(C_{x+d}) - Q_\epsilon(C_x) - \nabla Q_\epsilon(C_x)^T [C_{x+d} - C_x] \right| \\ &= \mathcal{O}(\|C_{x+d} - C_x\|^2) = \mathcal{O}(\|d\|^2), \end{aligned}$$

where the second and last equalities come from [Proposition 4.1](#) and [\(5.4\)](#).

For the second term we have

$$\begin{aligned} \left| \tilde{Q}_\epsilon(C_x; C_{x+d} - C_x) - \tilde{Q}_\epsilon(C_x; \tilde{C}_{(x; d)} - C_x) \right| &= \left| Q_\epsilon(C_x) + \nabla Q_\epsilon(C_x)^T [C_{x+d} - C_x] - Q_\epsilon(C_x) - \nabla Q_\epsilon(C_x)^T [\tilde{C}_{(x; d)} - C_x] \right| \\ &= \left| \nabla Q_\epsilon(C_x)^T [C_{x+d} - \tilde{C}_{(x; d)}] \right| \leq \|\nabla Q_\epsilon(C_x)\| \|C_{x+d} - \tilde{C}_{(x; d)}\|. \end{aligned}$$

Because  $\nabla Q_\epsilon$  by [Proposition 4.1](#) is bounded, it only remains to show that  $\|C_{x+d} - \tilde{C}_{(x; d)}\| = \mathcal{O}(\|d\|^2)$ .

To prove this, first notice that

$$c_j(x, \xi_i) + \nabla c_j(x, \xi_i)^T d \leq [\tilde{C}_{(x; d)}]_i, \quad \forall j \in \{1, \dots, m\}, i \in \{1, \dots, N\},$$

and therefore

$$\begin{aligned} [C_{x+d}]_i - [\tilde{C}_{(x; d)}]_i &= c_{j_1^i}(x + d, \xi_i) - [\tilde{C}_{(x; d)}]_i \\ (5.7) \quad &\leq c_{j_1^i}(x + d, \xi_i) - c_{j_1^i}(x, \xi_i) - \nabla c_{j_1^i}(x, \xi_i)^T d = \mathcal{O}(\|d\|^2), \end{aligned}$$

where  $j_1^i = \operatorname{argmax}_j \{c_j(x + d, \xi_i)\}$ . Also,

$$(5.8) \quad [\tilde{C}_{(x; d)}]_i - [C_{x+d}]_i \leq c_{j_2^i}(x, \xi_i) + \nabla c_{j_2^i}(x, \xi_i)^T d - c_{j_2^i}(x + d, \xi_i) = \mathcal{O}(\|d\|^2),$$

where  $j_2^i = \operatorname{argmax}_j \{c_j(x, \xi_i) + \nabla c_j(x, \xi_i)^T d\}$ . It follows from [\(5.7\)](#) and [\(5.8\)](#) that  $\|C_{x+d} - \tilde{C}_{(x; d)}\| = \mathcal{O}(\|d\|^2)$ .  $\square$

Now that model [\(5.6\)](#) has been established, we proceed to give an outline of the algorithm used to minimize  $\phi_\pi$ .

**5.2. Trust-Region Algorithm.** In this section we present an  $S\ell_1$ QP-type trust-region algorithm to minimize (5.3). This algorithm seeks to obtain a first-order stationary point of  $\phi_\pi$  by solving a sequence of quadratic programs (QP). At each iteration  $k$ , we compute a step  $d^k$  by minimizing the model  $m(x^k, H^k; d)$  within a radius  $\Delta_k$  for a given  $H_k$ . This problem is equivalent to

$$(5.9) \quad \begin{aligned} \min \quad & f(x^k) + \nabla f(x^k)^T d + \frac{1}{2} d^T H^k d + \pi [t^T \mathbf{1}_p + w] \\ \text{s.t.} \quad & \nabla g(x^k)^T d + g(x^k) \leq t, \\ & \nabla c(x^k, \xi_i)^T d + c(x^k, \xi_i) \leq z_i \mathbf{1}_m, \quad i = \{1, \dots, N\}, \\ & \nabla Q_\epsilon(C^N(x^k))^T (z - C^N(x^k)) + Q_\epsilon(C^N(x^k)) \leq w, \\ & t, w \geq 0, \|d\|_\infty \leq \Delta^k. \end{aligned}$$

We describe the choice of  $H^k$  in section 5.3.

A step  $d^k$ , obtained from (5.9), is accepted if it results in sufficient decrease of  $\phi_\pi$ , i.e., the step  $d^k$  is taken only if the value  $\phi_\pi(x^k + d^k)$  is sufficiently smaller than  $\phi_\pi(x^k)$ . We then proceed with iteration  $k+1$ , where now  $x^{k+1} = x^k + d^k$ . Algorithm 5.1 states the full algorithm.

The following theorem follows from [9, Theorem 11.2.5]. It states that any limit point  $x^*$  of the sequence  $\{x^k\}$  generated by Algorithm 5.1 is a stationary point of function  $\phi_\pi$ . Thus, we conclude that  $x^*$  minimizes (5.1) if  $x^*$  is a local minimizer of  $\phi_\pi$ ,  $g(x^*) \leq 0$ , and  $Q_\epsilon(C^N(x^*)) \leq 0$ .

**THEOREM 5.3.** *Suppose that Algorithm 5.1 does not terminate at a stationary point of  $\phi_\pi$  in step 3, and that  $x^*$  is a limit point of the sequence  $\{x^k\}$  generated by Algorithm 5.1. Let the assumptions from Proposition 5.2 hold and assume that, at each iteration, problem (5.9) is solved exactly. Then  $x^*$  is a first-order critical point of  $\phi_\pi$ .*

The condition that problem (5.9) is solved exactly can be relaxed as long as some Cauchy decrease condition holds (refer to [9, Theorem 11.2.5]).

**5.3. Choice of Hessian.** In this section we propose a choice of the Hessian  $H^k$  in (5.9) that aims to attain fast local convergence. The key observation is a close relationship between the non-smooth formulation (5.1) and the smooth optimization problem

$$(5.10) \quad \begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & g(x) \leq 0, \\ & \bar{q}(x) := Q_\epsilon(\bar{C}^N(x)) \leq 0, \end{aligned}$$

where  $\bar{C}^N(x)$  is defined below in Lemma 5.4.

Recall that, if  $x^*$  is a local minimizer of (5.1), then  $x^*$  and  $z^* = C^N(x^*)$  is a KKT point of (5.2) under some constraint qualification (see Theorem 5.1). The following lemma shows that  $x^*$  is also a KKT point of (5.10).

**LEMMA 5.4.** *Let  $(x^*, z^*, \nu^*, \mu^*, \lambda^*)$  be a KKT point of (5.2) with  $z^* = C^N(x^*)$ , where  $\nu^*$ ,  $\mu^*$  and  $\lambda^*$  are the multipliers associated to constraints (5.2b), (5.2c) and (5.2d), respectively. For each  $i \in \{1, \dots, N\}$ , set*

$$(5.11) \quad [\bar{\mu}_i]_j = \frac{[\mu_i^*]_j}{\lambda^* [\nabla Q_\epsilon(C^N(x^*))]_i}, \quad \forall j \in \{1, \dots, m\},$$

---

**Algorithm 5.1**  $Sl_1QP$  trust-region algorithm to solve JCCP
 

---

**Inputs:**  $\pi > 0$  (penalty parameter);  $\hat{\Delta} > 0$ ,  $\Delta_0 \in (0, \hat{\Delta})$ ,  $\eta \in (0, 1)$ ,  $\tau_1 \in (0, 1)$ , and  $\tau_2 > 1$  such that  $1/\tau_2 \leq \tau_1$  (trust region parameters);  $x_0$  (initial point)

- 1: **for**  $k = 0, 1, 2, \dots$  **do**
  - 2: Set  $C(x^k, \xi_i) = \max_{j=1, \dots, m} \{c_j(x^k, \xi_i)\}$  for all scenarios, compute  $Q_\epsilon(C^N(x^k))$  and  $\nabla Q_\epsilon(C^N(x^k))$ , and choose  $H^k$ .
  - 3: Obtain  $d^k$  by solving (5.9) (**if**  $d^k = 0$  **stop**, stationary point reached).
  - 4: Compute the ratio  $\rho^k = \frac{\phi_\pi(x^k) - \phi_\pi(x^k + d^k)}{m(x^k, H^k; 0) - m(x^k, H^k; d^k)}$ .
  - 5: **if**  $\rho^k < \eta$  **then**
  - 6:      $\Delta^{k+1} = \tau_1 \min\{\Delta^k, \|d^k\|_\infty\}$ ;  $x^{k+1} = x^k$
  - 7: **else if**  $\rho^k \geq \eta$  and  $\|d^k\|_\infty = \Delta^k$  **then**
  - 8:      $\Delta^{k+1} = \min\{\tau_2 \Delta^k, \hat{\Delta}\}$ ;  $x^{k+1} = x^k + d^k$
  - 9: **else**
  - 10:      $\Delta^{k+1} = \Delta^k$ ;  $x^{k+1} = x^k + d^k$
  - 11: **end if**
  - 12:  $k = k + 1$
  - 13: **end for**
- 

if  $\lambda^* \neq 0$  and  $[\nabla Q_\epsilon(C^N(x^*))]_i \neq 0$ . Else, select one  $j$  such that  $c_j(x^*, \xi_i) = C(x^*, \xi_i)$  and define  $[\bar{\mu}_i]_j = 1$  and  $[\bar{\mu}_i]_\ell = 0$  if  $\ell \neq j$ . Also, define the following function

$$(5.12) \quad [\bar{C}^N(x)]_i = \bar{\mu}_i^T c(x, \xi_i), \quad \forall i \in \{1, \dots, N\}.$$

Then,  $(x^*, \nu^*, \lambda^*)$  is a KKT point of (5.10).

*Proof.* Because  $(x^*, z^*, \nu^*, \mu^*, \lambda^*)$  is a KKT point of (5.2), the following conditions hold

$$(5.13a) \quad \nabla f(x^*) + \nabla g(x^*)\nu^* + \sum_{i=1}^N \nabla c(x^*, \xi_i)\mu_i^* = 0,$$

$$(5.13b) \quad \lambda^* [\nabla Q_\epsilon(z^*)]_i - \mu_i^{*T} \mathbf{1}_m = 0, \quad \forall i \in \{1, \dots, N\},$$

$$(5.13c) \quad c(x^*, \xi_i) - z_i^* \mathbf{1}_m \leq 0, \quad \forall i \in \{1, \dots, N\},$$

$$(5.13d) \quad Q_\epsilon(z^*) \leq 0,$$

$$(5.13e) \quad \mu_i^T (c(x^*, \xi_i) - z_i^* \mathbf{1}_m) = 0, \quad \forall i \in \{1, \dots, N\},$$

$$(5.13f) \quad \lambda^* Q_\epsilon(z^*) = 0, \quad \forall i \in \{1, \dots, N\},$$

$$(5.13g) \quad \lambda^* \geq 0, \quad \mu_i^* \geq 0, \quad \forall i \in \{1, \dots, N\}.$$

Notice that the KKT conditions of (5.2) associated with  $g$  have been left out because they trivially extend to the KKT conditions of (5.10).

If  $\lambda^* \neq 0$  and  $[\nabla Q_\epsilon(C^N(x^*))]_i \neq 0$ , then

$$(5.14) \quad [\bar{C}^N(x^*)]_i \stackrel{(5.12)}{=} \bar{\mu}_i^T c(x^*, \xi_i) \stackrel{(5.11)}{=} \sum_{j=1}^m \left( \frac{[\mu_i^*]_j}{\lambda^* [\nabla Q_\epsilon(C^N(x^*))]_i} \right) c_j(x^*, \xi_i)$$

$$\stackrel{(5.13e)}{=} z_i^* \sum_{j=1}^m \left( \frac{[\mu_i^*]_j}{\lambda^* [\nabla Q_\epsilon(z^*)]_i} \right) \stackrel{(5.13b)}{=} z_i^*.$$

If  $\lambda^* = 0$  or  $[\nabla Q_\epsilon(C^N(x^*))]_i = 0$ , it follows from the definition of  $\bar{\mu}_i$ , that  $[\bar{C}^N(x^*)]_i = \bar{\mu}_i^T c(x^*, \xi_i) = z_i^*$ . Thus, feasibility,  $Q_\epsilon(\bar{C}^N(x^*)) = Q_\epsilon(z^*) \leq 0$ , and complementarity,  $\lambda^* Q_\epsilon(z^*) = \lambda^* Q_\epsilon(\bar{C}^N(x^*)) = 0$ , hold.

Now, note that the gradient of the Lagrangian of (5.10) is given by

$$(5.15) \quad \nabla \bar{\mathcal{L}}(x, \nu, \lambda) = \nabla f(x) + \nabla g(x)\nu + \lambda \sum_{i=1}^N [\nabla Q_\epsilon(\bar{C}^N(x))]_i \nabla c(x, \xi_i) \bar{\mu}_i.$$

Then,

$$\begin{aligned} \nabla \bar{\mathcal{L}}(x^*, \nu^*, \lambda^*) &\stackrel{(5.14)}{=} \nabla f(x^*) + \nabla g(x^*)\nu^* + \lambda^* \sum_{i=1}^N [\nabla Q_\epsilon(\bar{C}^N(x^*))]_i \nabla c(x^*, \xi_i) \bar{\mu}_i^* \\ &\stackrel{(5.11)}{=} \nabla f(x^*) + \nabla g(x^*)\nu^* + \sum_{i=1}^N \nabla c(x^*, \xi_i) \mu_i^* \stackrel{(5.13a)}{=} 0. \end{aligned}$$

Thus,  $(x^*, \nu^*, \lambda^*)$  is a KKT point of (5.10).  $\square$

Lemma 5.4 shows that we can use the norm of (5.15) in the termination criterion for our algorithm. Also, from complementary slackness and feasibility, we have that most of the entries of the vector  $\mu_i^*$  are zero. The reason is that only the  $j$ th entries of  $\mu_i^*$  such that  $c_j(x^*, \xi_i) = C(x^*, \xi_i)$  can be nonzero. This means that the positive entries of  $\mu_i^*$  indicate which entries of vector  $c(x^*, \xi_i)$  correspond to the maximum  $C(x^*, \xi_i)$ . Therefore,  $\bar{C}^N(\cdot)$  can be interpreted as a smooth function that coincides with the maximum  $C^N(\cdot)$  at the solution  $x^*$ .

Suppose we apply the basic SQP algorithm [26] to (5.10) and compute steps  $d^k$  from the solution of the QP

$$(5.16) \quad \begin{aligned} \min \quad & f(x^k) + \nabla f(x^k)^T d + \frac{1}{2} d^T H^k d \\ \text{s.t.} \quad & \nabla g(x^k)^T d + g(x^k) \leq 0, \\ & \nabla \bar{q}(x^k)^T d + \bar{q}(x^k) \leq 0 \end{aligned}$$

to update the iterate  $x^{k+1} = x^k + d^k$ . If  $H^k$  is chosen as the Hessian of the Lagrangian function  $\bar{\mathcal{L}}$  of (5.10) at the current iterate  $(x^k, \nu^k, \lambda^k)$ , i.e.,

$$(5.17) \quad \begin{aligned} H^k = & \nabla^2 f(x^k) + \sum_{j=1}^p \nu_j^k \nabla^2 g_j(x^k) + \lambda^k \sum_{i=1}^N [\nabla Q_\epsilon(\bar{C}^N(x^k))]_i \left[ \sum_{j=1}^m \bar{\mu}_{ij} \nabla^2 c_j(x^k, \xi_i) \right] \\ & + \lambda^k \nabla \bar{C}^N(x^k) \left[ \nabla^2 Q_\epsilon(\bar{C}^N(x^k)) \right] \left[ \nabla \bar{C}^N(x^k) \right]^T, \end{aligned}$$

where  $[\nabla \bar{C}^N(x^k)]_i = \nabla c(x, \xi_i) \bar{\mu}_i$  for all  $i \in \{1, \dots, N\}$ , then this algorithm exhibits quadratic local convergence, under standard regularity assumptions.

Our goal is to mimic this behavior in Algorithm 5.1. Using arguments similar to those in the proof of Lemma 5.4 one can show that the steps generated by the QP (5.9) converge to the SQP steps from (5.16) up to first order as  $x^k \rightarrow x^*$ , under suitable regularity assumptions. This suggests to choose  $H^k$  defined by (5.17) for the step computation (5.9) in Algorithm 5.1, with the hope that the steps used in Algorithm 5.1 approach the SQP steps for (5.10) and result in fast local convergence.

However, the definition of (5.10) requires the knowledge of the optimal solution in (5.11), which is not available. Instead, we approximate the optimal solution with the current iterate and define

$$[\bar{\mu}_i^k]_j = \frac{[\mu_i^k]_j}{\lambda^k [\nabla Q_\epsilon(C^N(x^k))]_i}, \quad \forall j \in \{1, \dots, m\}, \quad i \in \{1, \dots, N\}.$$

Then, the Hessian matrix in the QP (5.9) is chosen as follows:

$$(5.18) \quad H^k = \nabla^2 f(x^k) + \sum_{j=1}^p \nu_j^k \nabla^2 g_j(x^k) + \lambda^k \sum_{i=1}^N [\nabla Q_\epsilon(C^N(x^k))]_i \left[ \sum_{j=1}^m \bar{\mu}_{i_j}^k \nabla^2 c_j(x^k, \xi_i) \right] + \lambda^k \nabla \bar{C}^N(x^k) [\nabla^2 Q_\epsilon(C^N(x^k))] [\nabla \bar{C}^N(x^k)]^T,$$

where  $[\nabla \bar{C}^N(x^k)]_{\cdot, i} = \nabla c(x, \xi_i) \bar{\mu}_i^k$  for all  $i \in \{1, \dots, N\}$ . Note the subtle difference with (5.17): The argument at which derivatives of  $Q_\epsilon$  are evaluated is based on the true maximum values given by the non-smooth function  $C^N(x^k)$  instead of the smoothed version  $\bar{C}^N(x^k)$ . With this choice of  $H^k$ , we observed superlinear local convergence of Algorithm 5.1 in our experiments.

**6. Numerical results.** In this section we report results demonstrating the performance of our proposed approach on two single chance-constrained (SCC) examples and one joint chance-constrained (JCC) example. The goal is to observe the performance of our method from different perspectives such as: solution quality, speed, and features related to the smoothing parameter  $\epsilon$ .

In all the experiments we use the  $\Gamma_\epsilon$  function defined from  $\gamma_\epsilon$  in (2.6), which defines a twice continuously differentiable function. This allows us to incorporate first- and second-order derivatives into the algorithms. All computations were executed on Ubuntu 16.04 with 256GB RAM and a CPU with two Intel Xeon processors with 10 cores each, running at 3.10GHz. The algorithm and all experiments were implemented in Matlab R2015b.

In these experiments, because our choices of  $\alpha$  and  $N$  are such that  $(1-\alpha)N \in \mathbb{Z}$ , we opt to add a constant  $b = 1/2$  to the left-hand side of (2.3) in our implementation (refer to Remark 2.2).

**6.1. Choosing the smoothing parameter.** We introduced an approximation of problem (1.1) that does not need explicit information about the distribution of the random variables, their mean or their variance, and that only requires a sample to compute the constraint approximation. However, the quality of the solutions obtained from the approximation depends on the choice of the parameter  $\epsilon$ . As established in section 3,  $\epsilon$  is related to the feasibility of the solutions. Moreover, in our experiments we observed that large values of  $\epsilon$  are associated with fewer iterations and less variability in the solutions but at the same time lead to conservativeness, while small values work in the opposite manner. Therefore, a good choice of  $\epsilon$  is of vital importance to the solution quality.

For a fixed sample, we propose a binary search algorithm to find a value of  $\epsilon$  which yields a solution that is feasible to the chance constraint, but not too conservative. We begin by setting  $\ell = 0$ , selecting an initial  $\epsilon_0$ , and setting  $\epsilon_{LB} = 0$  and  $\epsilon_{UB} = \infty$ . Then, we solve problem (5.1) for  $\epsilon_\ell$  and estimate the true probability  $p_\ell = \mathbb{P}(C(x_\ell^*, \xi) \leq 0)$ , where  $x_\ell^*$  is the optimal solution of problem (5.1) using the parameter  $\epsilon_\ell$ . If  $p_\ell > 1-\alpha$ ,

then  $\epsilon_{\ell+1} = (\epsilon_\ell + \epsilon_{LB})/2$  and  $\epsilon_{UB} = \epsilon_\ell$ ; if  $p_\ell < 1 - \alpha$ , then  $\epsilon_{\ell+1} = (\epsilon_{UB} + \epsilon_\ell)/2$ , or  $\epsilon_{\ell+1} = 2\epsilon_0$  if  $\epsilon_{UB} = \infty$ , and  $\epsilon_{LB} = \epsilon_\ell$ . We now solve the problem again for  $\epsilon_{\ell+1}$  and compute  $p_{\ell+1}$ . We continue in this fashion until the estimated probability,  $p_\ell$ , is within a threshold of the target probability  $1 - \alpha$ .

One way to pick  $\epsilon_0$  is as follows. First, compute  $x_0^*$  as the optimal solution of the “robust” formulation of (1.1),

$$\min_{x \in X} f(x) \quad \text{s.t.} \quad C(x, \xi_i) \leq 0 \quad \forall i \in \{1, \dots, N\}.$$

Then, evaluate the standard deviation of the sample  $\{C(x_0^*, \xi_1), \dots, C(x_0^*, \xi_N)\}$ , and set  $\epsilon_0$  to be twice the standard deviation. This way of selecting  $\epsilon_0$  usually renders conservative solutions; however, because we have observed that the optimization algorithm converges more quickly for large values of  $\epsilon$ , we prefer to start with a large value of  $\epsilon$  that is decreased afterwards. This choice of  $\epsilon_0$  is motivated by the attempt to minimize the mean square error in (2.5), for a fixed  $x$ . In [1], the authors show that the best choice of  $\epsilon$  is proportional to its standard deviation if everything is assumed to be normal. Thus, we define  $\epsilon_0$  as a multiple of the standard deviation evaluated at  $x_0^*$ .

To estimate the true probability of constraint violation, we obtain an out-of-sample approximation using a sample with a large number of scenarios  $N'$  and compute the quantity  $\sum_{i=1}^{N'} \mathbb{1}(C(x, \xi_i) \leq 0)/N'$  as an approximation of  $\mathbb{P}(C(x, \xi) \leq 0)$ .

To speed up the binary search, we use the solution  $x_{\epsilon_k}^*$ , with its multipliers  $\lambda_{\epsilon_k}^*$  and  $\bar{\mu}_{\epsilon_k}^*$ , to initialize the algorithm for  $\epsilon_{k+1}$ , i.e.,  $x_0 = x_{\epsilon_k}^*$ ,  $\lambda_0 = \lambda_{\epsilon_k}^*$ , and  $\bar{\mu}_0 = \bar{\mu}_{\epsilon_k}^*$  to solve problem (5.2) for  $\epsilon_{k+1}$ .

For the experiments in the following section, we terminate the binary search if  $|\mathbb{P}(C(x_\ell^*, \xi) \leq 0) - (1 - \alpha)| \leq 10^{-4}$  or the number of bisections exceeds 10.

**6.2. Single Chance Constraints.** In this section, we present the outcome of two experiments using a nonlinear solver to solve our reformulation for a SCCP. Both examples are solved by `Knitro 10.1.2`. To compute  $Q_\epsilon$ , we use the `Matlab` function `fzero` to find the root of the nonlinear function (2.3) from the initial point  $C_{[M]}(x)$ . We also provide `Knitro` both the gradient and Hessian of  $Q_\epsilon$  as given in (4.2) and (4.3).

**6.2.1. A non-convex example.** We first investigate the performance of the NLP reformulation for the non-convex problem

$$\min_{x,y} y \quad \text{s.t.} \quad \mathbb{P}(c(x) \leq y) \geq 0.95,$$

where  $c(x) = 0.25x^4 - 1/3x^3 - x^2 + 0.2x - 19.5 + \xi_1x + \xi_2$ , and  $\xi_1 \sim N(0, 3)$  and  $\xi_2 \sim N(0, 144)$  are independent random variables. This problem is similar to the one presented in [11].

In this experiment we use a sample of size  $N = 1,000$  and two different smoothing parameters:  $\epsilon = 1$  and  $\epsilon = 0.1$ . We used 10 different feasible initial points for each  $\epsilon$ , given by  $x_0 = (t, 2.5)$ , for  $t$  equally spaced in the interval  $[-1.5, 2.5]$ . In Figure 3 we observe the results of all the different runs. In this figure, the solid line represents the boundary of the empirical feasible region, while the dotted line represents the boundary of the smooth approximation to the feasible region. All the points above the solid and dotted line are feasible for the empirical and smooth approximations, respectively. We see that the smaller value of  $\epsilon$  translates into a smooth curve that follows the empirical quantile more closely, while the larger value renders a “less noisy” function.

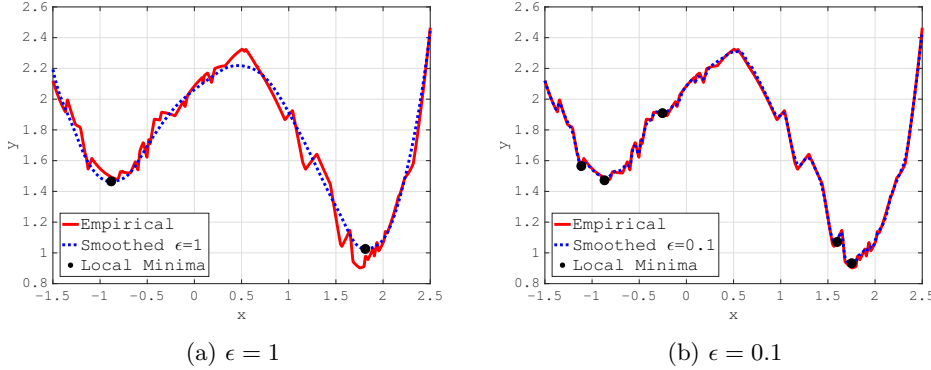


Fig. 3: Local minima found starting from 10 different points for two values of  $\epsilon$ .

Figure 3 shows that our approach is designed to find local minima instead of global minima, and that the quality of the solution not only depends on the starting point but on the choice of  $\epsilon$ . This can be seen from the points in Figure 3. Notice that the true problem seems to have only two local minimizers, one around  $x = 1.8$  and another around  $x = -0.8$ . However, when  $\epsilon = 0.1$ , the solver gets attracted to spurious local minima for three different starting points.

It is also interesting to notice that the algorithm takes longer to find a solution the smaller  $\epsilon$  is. For  $\epsilon = 1$ , the number of iterations ranged between 6 and 10 with an average of 8.2; however, for  $\epsilon = 0.1$ , the number of iterations ranges between 7 and 14 with an average of 11.6. This exemplifies some of the claims made in section 2.1 and emphasizes the importance of choosing  $\epsilon$  appropriately.

**6.2.2. Reinsurance portfolio example.** We consider a Value-at-Risk (VaR) minimization problem for reinsurance portfolios. The decision variable  $x_j$ , for  $j = 1, \dots, n$ , represents the fraction of the risk contract  $j$  that the reinsurer is willing to take. We would like to choose the optimal fraction of the contracts such that the portfolio premium satisfies a pre-established minimum value and that the 0.95-VaR is minimized. Each contract is subject to a stochastic loss. This data is simulated to represent risk and returns of businesses. The characteristics of these loss matrices include non-negativity, sparsity, positive skewness, and high kurtosis (this problem was constructed as in [12]).

Notice that the 0.95-VaR is defined the same way as the 0.95-quantile, so we can therefore use  $Q^{0.95}$  as the objective function. Thus, the problem can be modeled as

$$\min_{x \in [0,1]^n} Q^{0.95}(L^T x) \quad \text{s.t. } c^T x \geq c^*,$$

where the random variable  $L_j$  represents the losses of contract  $j$ ,  $c_j$  are the market premiums of the risk contracts, and  $c^* = 0.1 \sum_{j=1}^n c_j$  is the minimum level of portfolio premium. The chance-constrained reformulation of the problem is

$$\min_{x \in [0,1]^n, z \in \mathbb{R}} z \quad \text{s.t. } \mathbb{P}(L^T x \leq z) \geq 1 - \alpha, \quad c^T x \geq c^*.$$

For this example, we considered 25 contracts ( $n = 25$ ) for different sample sizes  $N$ , and we ran 10 replications for each  $N$ .

In [Table 2](#), we present the statistics of the `Knitro` runs using formulation (2.4), the quantile-constrained NLP approximation of the problem. The times reported include all NLP solves executed during the bisection search to tune the smoothing parameter  $\epsilon$  described in [section 6.1](#). We see that the solution time grows modestly with respect to the sample size, and that the number of iterations are roughly the same even when the sample size increases. The last column displays the average value of the tuned smoothing parameter. Consistent with our theoretical analysis in [section 3](#) and the results in [2], we observe that the value of  $\epsilon$  that yields a solution that is “just feasible” decreases as the sample size increases.

N	Avg. time (sec)	Max. time (sec)	Avg. iter (per $\epsilon$ )	Avg. $\epsilon$
200	1.1301	1.8285	13.236	1628.3
500	1.4797	2.6307	14.250	1135.6
2000	10.293	13.643	13.479	700.68
5000	49.943	113.91	14.073	554.47

Table 2: Number of iterations and CPU time needed by `Knitro` to return a solution.

We compare our approach to the empirical probability formulation  $\tilde{F}^N$ . This formulation is solved via a mixed-integer linear programming (MIP) formulation [29], using `Gurobi` as the MIP solver with a time limit of 10 minutes. For the solutions obtained by either method, we computed an out-of-sample approximation of the true quantile,  $Q^{0.95}(L^T x)$ , via the empirical 0.95-quantile using a sample of  $N' = 100,000$  observations. The results of the experiments are shown in [Figure 4](#). We see that, for  $N = 200$  and  $N = 5,000$ , the variance of the solutions obtained by the NLP solver is considerably smaller than the variance of the solutions obtained using the MIP formulation. The spread observed on the quality of the solutions from the MIP for  $N = 5,000$  may be due to the 10 minute limit, since the incumbent returned by the solver is not proved to be optimal for the sample approximation for this sample size. We also observe that the solutions obtained by the NLP formulation are consistently better in all cases and are improving as the sample size grows.

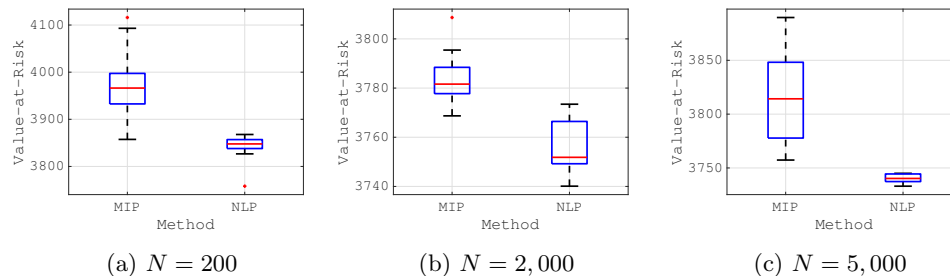


Fig. 4: Comparison between an MILP approach and our NLP method.

We now compare the performance of the NLP solver when constraint (4.1b) is substituted by  $F_\epsilon(0; x) = \frac{1}{N} \sum_{i=1}^N \Gamma_\epsilon(c(x, \xi_i)) \geq 1 - \alpha$ . We use this example to evaluate the advantage of the quantile formulation of the constraint compared to the



probability formulation. We did these experiments under the same settings as above. The solutions that the solver found with the probability formulation are the same as for the quantile formulation (within numerical accuracy), except for two cases for  $N = 2,000$  and one for  $N = 5,000$  in which `Knitro`'s iterates diverge. The number of iterations and the time taken by the solver is considerably larger compared to the quantile formulation as can be seen in Figure 5. Furthermore, while the quantile formulation requires roughly the same number of iterations for different number of scenarios, the iteration count for the probabilistic constraint grows with the sample size. Therefore, although finding the root of (2.3) is more expensive than using  $F_\epsilon(0; x)$  directly, working with the quantile is preferable in terms of the performance of the solver.

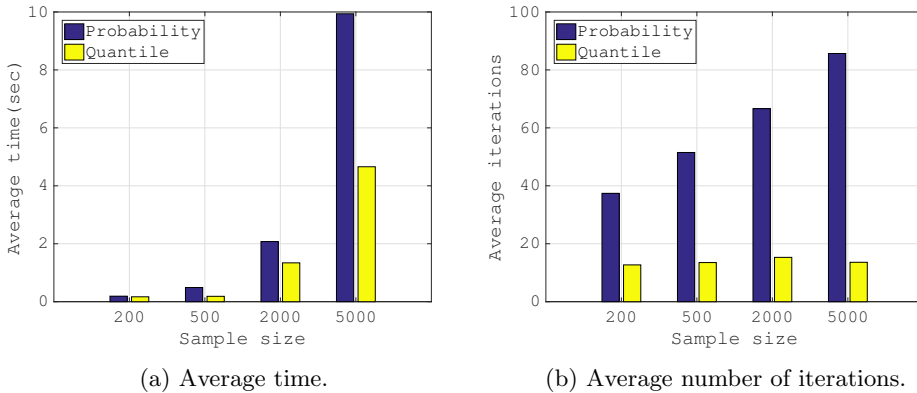


Fig. 5: Performance comparison between the probability and quantile approximations.

**6.3. Joint Chance Constraints.** The last example we consider is a continuous probabilistic multi-dimensional knapsack problem

$$(6.1) \quad \begin{aligned} & \max_{x \in [0,1]} \sum_{j=1}^n p_j x_j \\ & \text{s.t. } \mathbb{P} \left( \sum_{j=1}^n W_{ij} x_j \leq w_i, i \in \{1, \dots, m\} \right) \geq 0.95. \end{aligned}$$

The problem consists of a set of  $n$  items, each with an associated profit  $p_j$ , and different “weights” for each item (for example, both volume and weight, where the volume and weight of each item are not necessarily related). Here,  $W_{ij}$  represents the random  $i$ th “weight” of item  $j$ , and  $w_i$  is the maximum “weight” limit for constraint  $i$  ( $w_i$  is a deterministic quantity). The decision variable  $x_j \in [0, 1]$  represents the fraction of an item that is to be included. The goal is to select the optimal fraction each item in order to maximize the profit, while satisfying all the “weight” constraints simultaneously at least 95% of the time. The number of items considered for this experiment was  $n = 20$  with  $m = 10$  constraints. The randomness comes from the weights  $W_{ij}$ , whose distribution is described next.

We implemented [Algorithm 5.1](#) in `Matlab`, using `CPLEX 12.6.3` to solve the QP (5.9). We selected the following parameters for the algorithm:  $\pi = 10$ ,  $\hat{\Delta} = 10^6$ ,  $\Delta_0 = 1$ ,  $\eta = 10^{-8}$ ,  $\tau_1 = 1/2$  and  $\tau_2 = 2$ . We defined  $H^k$  as (5.18), and terminate [Algorithm 5.1](#) when  $\|\nabla\tilde{\mathcal{L}}(x_k, \nu, \lambda)\|_\infty \leq 10^{-6}$ ,  $g(x_k) \leq 10^{-6}\mathbf{1}_m$ , and  $Q_\epsilon(C^N(x_k)) < 10^{-6}$ .

We generate data for (6.1) according to the knapsack instance *mk-20-10* from [32], which is based on an available deterministic knapsack instance in [3]. Each item  $j$  is available or not according to a Bernoulli distribution. An item that is not available has weight zero in all rows. As a result, the item weights in different rows are not independent. If an item is available, then its weight in each row is normally distributed with mean equal to its weight in the deterministic instance and standard deviation equal 0.1 times the mean. For each sample size  $N$ , we ran 10 replications.

The results of our proposed algorithm are presented in [Table 3](#) under the column heading ‘NLP’. The computation times include the total time for the binary search algorithm to find a tuned value of  $\epsilon$ . For the solutions obtained, we computed an out-of-sample approximation of  $\mathbb{P}(C(x, \xi))$  using the empirical cdf with  $N' = 10^6$  scenarios. First, we can see that the time it takes the algorithm to obtain a solution grows modestly with the sample size  $N$ . Second, the best solution obtained from solving the problem with 10 different samples of size 100 has an objective value nearly identical to the best solution obtained using samples of size 1,000. This indicates that solving the approximation with multiple small samples may be more effective at obtaining good solutions than solving the approximation with a single large sample. Finally, the last row shows that the average value of the tuned smoothing parameter decreases with increased sample size, similar to our observation in [Table 2](#).

We also compared our solutions to the solutions obtained by solving the empirical probability formulation  $\tilde{F}^N$  via an MIP formulation. In order to obtain comparable solutions between the MIP and the NLP formulations, we tuned the risk level  $\alpha$  of the MIP formulation to ensure that the solutions obtained were all feasible. This is similar to tuning the  $\epsilon$  parameter in the NLP formulation. We limited the overall tuning time for the  $\alpha$  parameter in the MIP to ten minutes. For each  $\alpha$  trial of the MIP, we gave `Gurobi` 0.5, 0.5, 1, 5, and 10 minutes for sample size 100, 500, 1,000, 5,000, and 10,000, respectively. These time limits were chosen to give the MIP algorithm significantly more time than the NLP approach. The results are given in [Table 3](#) under the column heading ‘MIP’. It can be seen that the MIP solutions have generally lower optimal values and larger variability, the same behavior observed in [section 6.2.2](#). On the other hand, solutions obtained from [Algorithm 5.1](#) consistently present larger objective function values, less variability, and are solved well under 3 minutes for most sample sizes (except  $N = 10,000$ ). It is possible that giving the MIP algorithm more time could lead to better solutions, but these results clearly indicate that our NLP approach is more effective at finding good solutions quickly. Furthermore, the results show that the binary search for  $\epsilon$  can attain the desired risk-level more accurately than changing the  $\alpha$  parameter, given the time limit.

**7. Summary.** In this paper we developed a technique to solve nonlinear problems with probabilistic constraints. We first presented an approximation for single chance constraints with the following characteristics: (1) it can be represented as a single smooth constraint, (2) it does not require information about the distribution of the random variables or information about the moments, and is solely based on sampling, (3) it does not restrict the constraints to be convex, (4) it can be handled by standard continuous nonlinear programming techniques and solvers, and (5) it avoids

Sample size	100		500		1,000	
Method	NLP	MIP	NLP	MIP	NLP	MIP
Min. prob	0.9499	0.9445	0.9499	0.9503	0.9499	0.9502
Avg. prob	0.9504	0.9591	0.9503	0.9520	0.9511	0.9514
Max. prob	0.9521	0.9723	0.9526	0.9549	0.9591	0.9535
Min. obj	5816.6	5771.3	5834.4	5830.3	5827.5	5831.3
Avg. obj	5828.9	5806.0	5839.0	5833.2	5840.2	5836.6
Max. obj	5842.6	5836.1	5842.6	5839.2	5842.7	5841.0
Avg. time (s)	8.7	90	29.3	168	37.4	372
Max. time (s)	24.6	120	49.8	180	80.5	420
Avg. $\epsilon$	17	-	9.8	-	7.5	-

Sample size	5,000		10,000	
Method	NLP	MIP	NLP	MIP
Min. prob	0.9499	0.9514	0.9499	0.9506
Avg. prob	0.9500	0.9562	0.9500	0.9606
Max. prob	0.9501	0.9617	0.9503	0.9733
Min. obj	5840.8	5818.1	5841.2	5798.0
Avg. obj	5842.3	5828.7	5842.4	5819.4
Max. obj	5842.9	5839.9	5842.9	5833.8
Avg. time (s)	88.265	330	324.39	600
Max. time (s)	163.96	600	582.53	600
Avg. $\epsilon$	5.8	-	4.8	-

Table 3: Statistics for the multi-knapsack problem.

the combinatorial complexity of a branch and bound search, and scales well with the number of variables and samples.

We also presented an extension of the SCC formulation that is able to handle joint chance constraints. We constructed a specialized trust-region solver for JCCPs utilizing a Hessian matrix that has proven empirically to give fast local convergence.

## REFERENCES

- [1] R. ALEMANY, C. BOLANCÉ, AND M. GUILLÉN, *A nonparametric approach to calculating value-at-risk*, Insurance: Mathematics and Economics, 52 (2013), pp. 255–262.
- [2] A. AZZALINI, *A note on the estimation of a distribution function and quantiles by a kernel method*, Biometrika, 68 (1981), pp. 326–328.
- [3] J. E. BEASLEY, *Or-library: Distributing test problems by electronic mail*, Journal of the Operational Research Society, 41 (1990), pp. 1069–1072.
- [4] D. BIENSTOCK, M. CHERTKOV, AND S. HARNETT, *Chance-constrained optimal power flow: Risk-aware network control under uncertainty*, SIAM Review, 56 (2014), pp. 461–495.
- [5] G. CALAFIORE AND M. C. CAMPI, *Uncertain convex programs: randomized solutions and confidence levels*, Mathematical Programming, 102 (2005), pp. 25–46.
- [6] M. C. CAMPI, S. GARATTI, AND M. PRANDINI, *The scenario approach for systems and control design*, Annual Reviews in Control, 33 (2009), pp. 149–157.
- [7] Y. CAO AND V. ZAVALA, *A sigmoidal approximation for chance-constrained nonlinear programs*, 2017, [http://www.optimization-online.org/DB\\_FILE/2017/10/6236.pdf](http://www.optimization-online.org/DB_FILE/2017/10/6236.pdf).
- [8] A. CHARNES, W. W. COOPER, AND G. H. SYMONDS, *Cost horizons and certainty equivalents: an approach to stochastic programming of heating oil*, Management Science, 4 (1958), pp. 235–263.

- [9] A. R. CONN, N. I. GOULD, AND P. L. TOINT, *Trust region methods*, vol. 1, SIAM, 2000.
- [10] F. E. CURTIS AND M. L. OVERTON, *A sequential quadratic programming algorithm for non-convex, nonsmooth constrained optimization*, SIAM Journal on Optimization, 22 (2012), pp. 474–500.
- [11] F. E. CURTIS, A. WÄCHTER, AND V. M. ZAVALA, *A sequential algorithm for solving nonlinear optimization problems with chance constraints*, SIAM Journal on Optimization, 28 (2018), pp. 930–958.
- [12] M. FENG, A. WÄCHTER, AND J. STAUM, *Practical algorithms for value-at-risk portfolio optimization problems*, Quantitative Finance Letters, 3 (2015), pp. 1–9.
- [13] A. GELETU, A. HOFFMANN, M. KLOPPEL, AND P. LI, *An inner-outer approximation approach to chance constrained optimization*, SIAM Journal on Optimization, 27 (2017), pp. 1834–1857.
- [14] R. HENRION, P. LI, A. MÖLLER, M. C. STEINBACH, M. WENDT, AND G. WOZNY, *Stochastic optimization for operating chemical processes under uncertainty*, in Online Optimization of Large Scale Systems, Springer, 2001, pp. 457–478.
- [15] R. HENRION AND A. MÖLLER, *Optimization of a continuous distillation process under random inflow rate*, Computers & Mathematics with Applications, 45 (2003), pp. 247–262.
- [16] J.-B. HIRIART-URRUTY AND C. LEMARECHAL, *Convex Analysis and Minimization Algorithms I: Fundamentals*, vol. 305, Springer Science & Business Media, 1996.
- [17] W. HOEFFDING, *Probability inequalities for sums of bounded random variables*, Journal of the American Statistical Association, 58 (1963), pp. 13–30.
- [18] L. J. HONG, Y. YANG, AND L. ZHANG, *Sequential convex approximations to joint chance constrained programs: A monte carlo approach*, Operations Research, 59 (2011), pp. 617–630.
- [19] Z. HU, L. J. HONG, AND L. ZHANG, *A smooth monte carlo approach to joint chance-constrained programs*, IIE Transactions, 45 (2013), pp. 716–735.
- [20] A. LODI, E. MALAGUTI, G. NANNICINI, AND D. THOMOPULOS, *Nonlinear chance-constrained problems with applications to hydro scheduling*, tech. report, IBM Research Report RC25594 (WAT1602-046), 2016.
- [21] J. LUEDTKE AND S. AHMED, *A sample approximation approach for optimization with probabilistic constraints*, SIAM Journal on Optimization, 19 (2008), pp. 674–699.
- [22] J. LUEDTKE, S. AHMED, AND G. L. NEMHAUSER, *An integer programming approach for linear programs with probabilistic constraints*, Mathematical Programming, 122 (2010), pp. 247–272.
- [23] D. K. MOLZAHN AND L. A. ROALD, *Towards an ac optimal power flow algorithm with robust feasibility guarantees*, in 2018 Power Systems Computation Conference (PSCC), IEEE, 2018, pp. 1–7.
- [24] A. NEMIROVSKI AND A. SHAPIRO, *Convex approximations of chance constrained programs*, SIAM Journal on Optimization, 17 (2006), pp. 969–996.
- [25] A. NEMIROVSKI AND A. SHAPIRO, *Scenario approximations of chance constraints*, in Probabilistic and Randomized Methods for Design Under Uncertainty, Springer, 2006, pp. 3–47.
- [26] J. NOCEDAL AND S. J. WRIGHT, *Numerical optimization*, Springer, 2006.
- [27] B. PAGNONCELLI, S. AHMED, AND A. SHAPIRO, *Sample average approximation method for chance constrained programming: theory and applications*, Journal of Optimization Theory and Applications, 142 (2009), pp. 399–416.
- [28] A. PRÉKOPA, *Probabilistic programming*, Handbooks in Operations Research and Management Science, 10 (2003), pp. 267–351.
- [29] A. RUSZCZYŃSKI, *Probabilistic programming with discrete distributions and precedence constrained knapsack polyhedra*, Mathematical Programming, 93 (2002), pp. 195–215.
- [30] D. W. SCOTT, R. A. TAPIA, AND J. R. THOMPSON, *Kernel density estimation revisited*, Nonlinear Analysis: Theory, Methods & Applications, 1 (1977), pp. 339–372.
- [31] A. SHAPIRO, D. DENTCHEVA, AND A. RUSZCZYŃSKI, *Lectures on stochastic programming: Modeling and theory*, SIAM, second ed., 2014.
- [32] Y. SONG, J. R. LUEDTKE, AND S. KÜÇÜKYAVUZ, *Chance-constrained binary packing problems*, INFORMS Journal on Computing, 26 (2014), pp. 735–747.
- [33] W. VAN ACKOOIJ AND R. HENRION, *Gradient formulae for nonlinear probabilistic constraints with gaussian and gaussian-like distributions*, SIAM Journal on Optimization, 24 (2014), pp. 1864–1889.