DC Optimal Power Flow with Joint Chance Constraints

Alejandra Peña-Ordieres, Daniel K. Molzahn, Line A. Roald, and Andreas Wächter

***Abstract*—Managing uncertainty and variability in power injections has become a major concern for power system operators due to the increasing levels of fluctuating renewable energy connected to the grid. This work addresses this uncertainty via a joint chance-constrained formulation of the DC optimal power flow (OPF) problem, which satisfies *all* the constraints *jointly* with a pre-determined probability. The few existing approaches for solving joint chance-constrained OPF problems are typically either computationally intractable for large-scale problems or give overly conservative solutions that satisfy the constraints far more often than required, resulting in excessively costly operation. This paper proposes an algorithm for solving joint chance-constrained DC OPF problems by adopting an Sℓ_1QP-type trust-region algorithm. This algorithm uses a sample-based approach that avoids making strong assumptions on the distribution of the uncertainties, scales favorably to large problems, and can be tuned to obtain less conservative results. We illustrate the performance of our method using several IEEE test cases. The results demonstrate the proposed algorithm's advantages in computational times and limited conservativeness of the solutions relative to other joint chance-constrained DC OPF algorithms.**

***Index Terms*—joint chance constraints, nonlinear optimization, optimal power flow, sample average approximation**

I. Introduction

OPTIMAL power flow (OPF) is a fundamental problem in power systems operations that is used for real-time operations, markets, long-term planning, and many other applications. In its classical form, OPF determines the minimum cost generation dispatch that satisfies the demand for power while adhering to network constraints and engineering limits.

Growing quantities of renewable energy are increasing the variability and uncertainty inherent to power system operations. Many new methods account for and mitigate this uncertainty and variability [1], including two- and multi-stage stochastic programming [2]–[4], robust and worst-case optimization [5]–[8], and chance constraints [9]–[14]. These methods attempt to ensure secure and economical operations despite power injection uncertainty. Defining "security" is an important modelling question that dictates the formulation and solution algorithm. For example, robust optimization defines "secure" as ensuring feasibility for all realizations within a pre-specified uncertainty set, while chance-constrained optimization seeks to satisfy the constraints with a high probability $1-\alpha$, where α is a specified acceptable violation probability.

We propose a formulation and solution algorithm to solve OPF problems with *joint chance constraints* (JCC), which require that *all* engineering limits, including both generation and line flow constraints, are satisfied simultaneously with probability $1-\alpha$. This contrasts with formulations based on single chance constraints (SCC), which split the line flow and generation limits into separate chance constraints (for each line and generator) with individual risk levels, $1-\alpha_j$, for each of those constraints. Allocation of risk to individual components is more straightforward in problems with SCCs, while JCCs give much stronger guarantees on overall system security. Generally, SCCs are much easier to solve [15]. For example, linear SCCs with elliptical symmetric uncertainty distributions can be expressed as a second-order cone program that can be efficiently solved [16, Lemma 2.2].

Most chance-constrained OPF formulations have considered SCCs (e.g., [9], [11], [12]), while a limited number have attempted to solve JCC formulations [10]. In [14], a JCC problem is solved by decomposing the JCC into single chance constraints, which is challenging due to the difficulty in selecting the risk level for each individual constraint. Usually, the Boole or Bonferroni inequality is used to approximate the JCC. Reference [17] observes that even if the individual risk levels are selected optimally, the solution obtained from the single chance-constrained formulation can be suboptimal. Some efforts have been made to reduce the conservativeness of using Boole's inequality (e.g., [14]), and it has been observed that the SCC formulation leads to a low joint violation probability due to the structure of the OPF problem [18]. However, in general, the SCC formulation has the following drawbacks: (1) enforcing the chance constraints individually does not give strong guarantees on the feasibility probability of the entire system, and (2) solutions that are adapted to guarantee joint feasibility can be overly conservative and costly.

The most common methods for directly solving JCCs are based on scenario approximation (SA) (e.g., [19]–[21]), which has been applied to the OPF problem in, e.g., [10], and mixed-integer programming (MIP) (e.g., [22]). Both the SA and the MIP methods provide guarantees on the quality of the solution and are sample-based approximations, meaning that they do not make assumptions on the uncertainty distributions. However, solutions from SA are often highly conservative with much lower violation probabilities than what would be acceptable and, consequently, more costly ([10], [23], [24]). While MIP methods converge to the desired solution with increasing sample size, the complexity of the algorithm also increases, which can result in intractability.

This paper's main contribution is a joint chance-constrained formulation and algorithm to solve the OPF problem. The formulation is based on a sample average approximation (SAA) which gives rise to a non-linear programming (NLP) problem. The algorithm is an adaptation of the JCC algorithm presented in [25], with modifications that address the equalities from the power flow equations and improve computational

performance. The NLP approach has the following advantages: *(1) Sample-based:* Similar to the SA and MIP, we use a sample-based approach that does not rely on distributional assumptions. *(2) Scalable:* The method is scalable to large systems with many uncertain power injections where SA may be impractical, due to the need for a very large sample size, and MIP methods may be numerically intractable, due to the introduction of binary variables. *(3) Tunable:* The chance constraint approximation presented in [25] includes tuning parameters that impact the conservativeness of the solution. Carefully choosing those parameters allows for accurate tuning of the solution to satisfy the prescribed probability. Hence, the proposed method does not render an excessively conservative feasible region, which is an advantage over the SA method.

The remainder of this paper is organized as follows. Section II describes the JCC OPF formulation. Section III proposes a smooth sample-based approximation of the probabilistic constraint of the OPF formulation. Section IV presents our solution algorithm. Section V discusses the tuning parameters. Section VI numerically demonstrates our method, benchmarked against SA. Section VII concludes the paper.

II. Joint Chance-Constrained Optimal Power Flow

We aim to minimize the expected generation cost while satisfying all engineering limits with a high probability via a *joint chance-constrained* OPF problem (JCC-OPF). The user expresses the acceptable risk as the joint violation probability, i.e., the probability that any of the constraints are violated. This section formulates the JCC-OPF. This formulation is closely related to those previously presented in [10]–[12], but differs in the handling of the forecasted operating point.

1) Notation: Consider a power system where the sets of buses, lines, and generators are denoted by $\mathcal{B}$, $\mathcal{L}$, and $\mathcal{G}$, respectively. To simplify notation, we assume that there is one generator with active power generation $g(\omega)$ and one uncertain load $d(\omega)$, where ω is a random variable, at every bus. Then, $g(\omega), d(\omega) \in \mathbb{R}^{|\mathcal{B}|}$. If a bus i does not have a generator or load, we set $g_i(\omega) = 0$ or $d_i(\omega) = 0$, respectively, whereas multiple loads or generators are handled through summation. We use the linearized DC approximation of the active power flows, which assumes that: (1) all voltage magnitudes are 1 per unit, (2) neighboring buses have small angle differences, and (3) the system is lossless.

2) Uncertain loads: All uncertain loads can be represented as $d(\omega) = d+\omega$, where ω is a random variable with zero mean; this can be interpreted as the sum of the forecasted value d and its fluctuation ω. Due to the nature of the renewable energy uncertainty, we model ω as a continuous random variable.

3) Generators: We model the active power generation $g(\omega)$ using an affine control policy, resembling the actions of the automatic generation control (AGC) [10]. Each generator adjusts its output to satisfy a fraction of the total load imbalance,

$$g_i(\omega) = g_i - \beta_i \Omega, \quad \forall i \in \mathcal{G}, \tag{1}$$

where $\Omega = \sum_{i\in\mathcal{B}} \omega_i$ and β_i is the so-called participation factor of generator i. Our formulation's optimization variables include the generation outputs g and the participation factors β.

4) Power Balance: With the lossless system representation, maintaining power balance is equivalent to ensuring that the total power generation equals the total demand,

$$\sum_{i\in\mathcal{G}} g(\omega) + \sum_{j\in\mathcal{B}} d(\omega) = 0, \quad \forall \omega. \tag{2}$$

By substituting the expressions for $g(\omega)$ and $d(\omega)$ from above, we observe that (2) is equivalent to enforcing

$$\sum_{i\in\mathcal{G}} g_i + \sum_{j\in\mathcal{B}} d_j = 0 \quad \text{and} \quad \sum_{i\in\mathcal{G}} \beta_i = 1.$$

Here, the first equation guarantees power balance without fluctuations $\omega = 0$, while the second equation ensures system balance during fluctuations $\omega \neq 0$.

5) Power flows: We denote the line connecting buses i and j as $ij \in \mathcal{L}$. The power flow on the line ij, $f_{ij}(\omega)$, is a linear function of the power injections $p(\omega) = g(\omega) - d(\omega)$:

$$f_{ij}(\omega) = \Phi_{(\cdot,ij)} p(\omega).$$

The matrix Φ denotes the power transfer distribution factors (PTDFs) [26], with $\Phi_{(\cdot,ij)}$ referring to the row of Φ corresponding to the line ij.

6) Cost function: The generators have a quadratic cost function in terms of active power generation:

$$\mathrm{Cost}(x) = \frac{1}{2} x^T M x + v^T x + k_0, \tag{3}$$

where M is a diagonal matrix with non-negative entries. We minimize the expected generation cost $\mathbb{E}\left[\mathrm{Cost}(g(\omega))\right]$. Substituting (1) and taking the expectation in (3), we obtain

$$\begin{aligned}
&\mathbb{E}\left[\mathrm{Cost}(g(\omega))\right] \\
&\quad = \mathbb{E}\left[\frac{1}{2}(g-\beta\Omega)^T M (g-\beta\Omega) + v^T(g-\beta\Omega) + k_0\right] \\
&\quad = \mathrm{Cost}(g) + \mathbb{E}[\Omega]\left(g^T M\beta - v^T\beta\right) + \frac{1}{2}\mathbb{E}[\Omega^2]\beta^T M\beta \\
&\quad = \mathrm{Cost}(g) + \frac{1}{2}\mathbb{V}(\Omega)\beta^T M\beta,
\end{aligned}$$

7) JCC-OPF: With these modelling considerations, we formulate the JCC-OPF as

$$\min_{g,\beta} \quad \mathrm{Cost}(g) + \frac{1}{2}\mathbb{V}(\Omega)\beta^T M\beta \tag{4a}$$

$$\text{s.t.} \quad \sum_{i\in\mathcal{G}} g_i - \sum_{i\in\mathcal{B}} d_i = 0, \tag{4b}$$

$$\sum_{i\in\mathcal{G}} \beta_i = 1, \tag{4c}$$

$$\mathbb{P}\left(\begin{array}{ll} f_{ij}^{\mathrm{LB}} \le \Phi p(\omega) \le f_{ij}^{\mathrm{UB}}, & \forall ij \in \mathcal{L} \\ g_i^{\mathrm{LB}} \le g_i - \beta_i\Omega \le g_i^{\mathrm{UB}}, & \forall i \in \mathcal{G} \end{array}\right) \ge 1-\alpha. \tag{4d}$$

The objective (4a) minimizes the expected cost. The deterministic constraints (4b), (4c) ensure power balance. The joint chance constraint (4d) enforces bounds on the line flows and generator outputs, $f_{ij}^{\mathrm{LB}}, f_{ij}^{\mathrm{UB}}$ and $g_i^{\mathrm{LB}}, g_i^{\mathrm{UB}}$, with probability $1-\alpha$. Here, α represents the acceptable violation probability.

III. Representation of Chance Constraints

Constraint (4d) can be seen as a conventional nonlinear inequality $\vartheta(g,\beta) \geq 1-\alpha$. Problems with smooth nonlinear inequalities can be efficiently solved if one can compute the values of the inequalities as well as their gradients. In [25], the authors propose a sample-based method to obtain a smooth approximation of chance constraints that can be used to efficiently solve chance-constrained problems. This section summarizes the method from [25] and discusses modifications needed to address (4). We refer to one realization of ω as a "scenario" and a set of scenarios as a "sample".

We begin by defining the $(1-\alpha)$-quantile of a random variable Y, denoted by $Q^{1-\alpha}(Y)$:

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

From the above definition, we know that $\mathbb{P}(Y \leq 0) \geq 1-\alpha$ is equivalent to $Q^{1-\alpha}(Y) \leq 0$, where Y is a random variable taking values in $\mathbb{R}$. This definition can be extended to random variables $Y \in \mathbb{R}^m$, $m > 1$, if we let $\widehat{Y} = \max_{j=1,\ldots,m}\{Y_j\}$ and consider $Q^{1-\alpha}(\widehat{Y})$ instead.

This equivalence is applied to (4) by rewriting the probabilistic constraint (4d) as $\mathbb{P}(c(g,\beta;\omega) \leq 0) \geq 1-\alpha$, where

$$c(g,\beta;\omega) = \begin{pmatrix} \Phi\hat{p}(\omega) - f^{\mathrm{UB}} \\ f^{\mathrm{LB}} - \Phi\hat{p}(\omega) \\ g - \Omega\beta - g^{\mathrm{UB}} \\ g^{\mathrm{LB}} - g + \Omega\beta \end{pmatrix}.$$

Then, $c(g,\beta;\omega) \in \mathbb{R}^m$, where $m = 2|\mathcal{L}| + 2|\mathcal{G}|$. Let $C(g,\beta;\omega) = \max\limits_{j=1,\ldots,m}\{c_j(g,\beta;\omega)\}$. Reformulating (4) yields

$$\begin{aligned} \min_{g,\beta} \quad & \mathrm{Cost}(g) + \frac{1}{2}\,\mathbb{V}(\Omega)\beta^T M\beta \\ \text{s.t.} \quad & Q^{1-\alpha}\left(C(g,\beta;\omega)\right) \leq 0, \\ & \text{Eqns. (4b), (4c).} \end{aligned} \tag{5}$$

For continuous random variables, the $(1-\alpha)$-quantile is obtained by inverting the cumulative density function (cdf) at the $(1-\alpha)$-level. Thus, whenever $C(g,\beta;\omega)$ defines a continuous random variable for any fixed value of (g,β), a smooth approximation of the quantile can be obtained from a smooth approximation of the cdf.

One way to approximate the cdf is to consider a sample $\{\omega_1,\ldots,\omega_N\}$ of the random variable ω. The empirical probability that the random variable $C(g,\beta;\omega)$ takes a value less than or equal to t (i.e., the empirical cdf evaluated at t) is

$$F^N(t;g,\beta) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(C(g,\beta;\omega_i) \leq t), \tag{6}$$

where $\mathbb{1}$ is the indicator function, i.e., $\mathbb{1}(A)$ takes the value of 1 if A occurs or zero otherwise. Note that F^N is non-smooth since the indicator function is not continuous at zero. For $t = 0$, F^N is equivalent to the SAA approximation used in MIP approaches (see [22]).

To obtain a smooth approximation of the cdf at the point (g,α), we follow an approach similar to [27], [28] by defining

$$F^N_\epsilon(t;g,\beta) = \frac{1}{N}\sum_{i=1}^{N} \Gamma_\epsilon(C(g,\beta;\omega_i) - t), \tag{7}$$

where $\epsilon > 0$ is a parameter of the following smooth approximation of the indicator function

$$\Gamma_\epsilon(y) = \begin{cases} 1, & y \leq -\epsilon \\ \gamma_\epsilon(y), & -\epsilon < y < \epsilon \\ 0, & y \geq \epsilon \end{cases} \tag{8}$$

and $\gamma_\epsilon : [-\epsilon,\epsilon] \to [0,1]$ is a symmetric and strictly decreasing function such that Γ_ϵ is continuously differentiable. With this choice of γ_ϵ, $F^N_\epsilon(t;g,\beta)$ is a differentiable approximation of the empirical cdf, $F^N(t;g,\beta)$ (see Fig. 1). We use the following γ_ϵ function based on the quartic kernel [29], which makes (8) twice continuously differentiable:

$$\gamma_\epsilon(y) = \frac{15}{16}\left(-\frac{1}{5}\left(\frac{y}{\epsilon}\right)^5 + \frac{2}{3}\left(\frac{y}{\epsilon}\right)^3 - \left(\frac{y}{\epsilon}\right) + \frac{8}{15}\right). \tag{9}$$

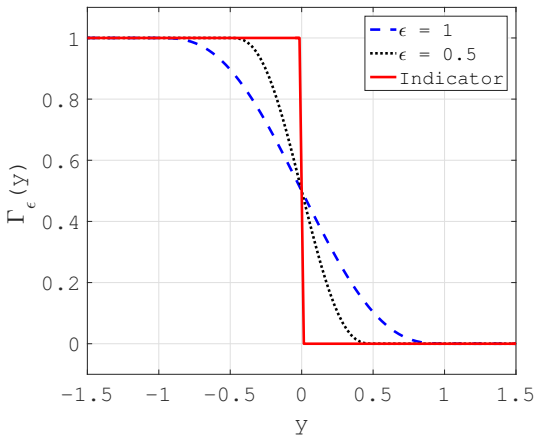

Fig. 1: Function $\Gamma_\epsilon(y-t)$

For a fixed g and β, the approximation of the $(1-\alpha)$-quantile can be computed as the inverse of F^N_ϵ at $1-\alpha$. The inverse can be obtained from the value Q_ϵ such that

$$\sum_{i=1}^{N} \Gamma_\epsilon\left(C(g,\alpha;\omega_i) - Q_\epsilon\right) = N(1-\alpha). \tag{10}$$

Reference [25] shows that Q_ϵ is an approximation to the $(1-\alpha)$-quantile at the point (g,α). Reference [25] also shows that the value Q_ϵ is unique, under mild conditions, and that it defines a function that maps the vector $C^N(g,\beta) = [C(g,\beta;\omega_1),\ldots,C(g,\beta;\omega_N)] \in \mathbb{R}^N$ to the root of (10). We denote such function as $Q_\epsilon(C^N(g,\beta))$.

Hence, we propose the following approximation to (4):

$$\min_{g,\beta} \quad \mathrm{Cost}(g) + \frac{1}{2}\,\mathbb{V}(\Omega)\beta^T M\beta \tag{11a}$$

$$\text{s.t.} \quad \sum_{i\in\mathcal{G}} g_i - \sum_{i\in\mathcal{B}} d_i = 0, \tag{11b}$$

$$\sum_{i\in\mathcal{G}} \beta_i = 1, \tag{11c}$$

$$Q_\epsilon\left(C^N(g,\beta)\right) \leq 0. \tag{11d}$$

Reference [25] discusses convergence and feasibility of this approximation to the solutions of the original problem (4) with increasing sample size. Section V details the choice of ϵ and t.

Since $C(g,\beta;\omega)$ is the maximum of the linear constraints given by the vector $c(g,\beta;\omega)$, (11d) is not smooth, resulting in algorithmic challenges. Section IV discusses how to solve (11).

IV. Solution Algorithm

Adopting from the approach in [25], this section proposes an algorithm for solving (11). To avoid having a non-smooth constraint, namely (11d), we first reformulate (11) to an equivalent unconstrained optimization problem in which the constraints are added to the objective function via terms that penalize infeasible solutions. To address the resulting non-smooth objective, we then propose an iterative algorithm that approximates the non-smooth unconstrained problem with a smooth constrained problem at each step of the algorithm. Finally, to improve tractability, we propose two modifications of this smooth constrained problem which make standard solvers more quickly compute the updates at each iteration.

A. ℓ_1-penalty function

Let $\pi > 0$ be a penalty parameter and $[x]^+ = \max\{0, x\}$. We propose an ℓ_1-penalty function in order to solve (11):

$$\phi_\pi(g,\beta) = \text{Cost}(g) + \frac{1}{2}\,\mathbb{V}(\Omega)\beta^T M\beta + \pi\|V(d,\beta)\|_1 \quad (12)$$

where

$$V(g,\beta) = \left(\sum_{i\in\mathcal{G}} g_i - \sum_{i\in\mathcal{B}} d_i, \sum_{i\in\mathcal{G}}\beta_i - 1, Q_\epsilon(C^N(g,\beta)\right)$$

is the vector of constraint violations. As shown in [25], $\phi_\pi(g,\beta)$ is an exact penalty function, meaning that if (g^*,β^*) is a local minimizer of ϕ_π for $\pi > 0$ and is feasible for problem (11), then (g^*,β^*) solves (11) [30, p. 299]. Moreover, Theorem 2.1 in [31] shows that there exists $\pi > 0$ such that minimization of (12) yields a solution for (11).

B. Minimizing the ℓ_1-penalty function

To minimize (12), we propose an Sℓ_1QP-type trust-region algorithm which solves a sequence of quadratic programs (QP). At each iteration k, ϕ_π is approximated with a piecewise quadratic function that depends on the values of g and β at the current iteration. The trust region determines a region of the search space around the current iterate where the quadratic model provides a good approximation of the objective ϕ_π.

The piecewise quadratic model is

$$\begin{aligned} m(g,\beta,H;\delta) &= \text{Cost}(g) + \frac{1}{2}\,\mathbb{V}(\Omega)\beta^T M\beta + \nabla\,\text{Cost}(g)^T\delta_g \\ &+ \mathbb{V}(\Omega)\beta^T M\delta_\beta + \frac{1}{2}\delta^T H\delta + \pi\left(\left|\sum_{i\in\mathcal{G}}(g_i+\delta_{g_i}) - \sum_{i\in\mathcal{B}} d_i\right|\right. \\ &\left. + \left|\sum_{i\in\mathcal{G}}(\beta_i+\delta_{\beta_i}) - 1\right| + \left[\widetilde{Q}_{\epsilon,C}(g,\beta;\delta)\right]^+\right), \end{aligned} \quad (13)$$

where $\delta = [\delta_g;\delta_\beta] \in \mathbb{R}^{2|\mathcal{G}|}$, $H \in \mathbb{R}^{2|\mathcal{G}|\times 2|\mathcal{G}|}$ is a symmetric matrix, and

$$\begin{aligned} \widetilde{Q}_{\epsilon,C}(g,\beta;\delta) &= \widetilde{Q}_\epsilon(C^N(g,\beta);\widetilde{C}^N(g,\beta;\delta) - C^N(g,\beta)), \\ \widetilde{Q}_\epsilon(z;p) &= Q_\epsilon(z) + \nabla Q_\epsilon(z)^T p, \\ [\widetilde{C}^N]_i(g,\beta;\delta) &= \max_j\{c_j(g,\beta;\omega_i) + \nabla c_j(g,\beta;\omega_i)^T\delta\}. \end{aligned}$$

Each iteration k finds a descent step, δ^k, for ϕ_π by minimizing the model $m(g^k,\beta^k,H^k;\delta)$ within a radius Δ^k for a given H^k. Minimizing $m(g^k,\beta^k,H^k;\delta)$ is challenging due to the non-smoothness introduced by the absolute values and the max operators. Hence, (13) is rewritten as a smooth constrained QP by introducing slack variables u, v, and w:

$$\begin{aligned} \min_{\delta,z,u,v,w}\ & \text{Cost}(g^k) + \frac{1}{2}\,\mathbb{V}(\Omega)(\beta^k)^T M\beta^k + \nabla\,\text{Cost}(g^k)^T\delta_g \\ & + \mathbb{V}(\Omega)(\beta^k)^T M\delta_\beta + \frac{1}{2}\delta^T H^k\delta + \pi\left[(u+v)^T\mathbf{1}_2 + w\right] \end{aligned} \quad (14a)$$

$$\text{s.t.} \sum_{i\in\mathcal{G}}(g_i^k+\delta_{g_i}) - \sum_{i\in\mathcal{B}} d_i = u_1 - v_1 \quad (14b)$$

$$\sum_{i\in\mathcal{G}}(\beta_i^k+\delta_{\beta_i}) = 1 + u_2 - v_2 \quad (14c)$$

$$c(g^k,\beta^k;\omega_i) + \nabla c(g^k,\beta^k;\omega_i)^T\delta \le z_i\mathbf{1}_m,\ \forall i\in[N] \quad (14d)$$

$$\begin{aligned} &\nabla Q_\epsilon(C^N(g^k,\beta^k))^T(z - C^N(g^k,\beta^k)) \\ &\qquad + Q_\epsilon(C^N(g^k,\beta^k)) \le w, \end{aligned} \quad (14e)$$

$$t,u,w \ge 0, \quad \|\delta\|_\infty \le \Delta^k, \quad (14f)$$

where $\mathbf{1}_n$ is the length-n vector of ones and $[N] = \{1,\dots,N\}$. The slack variables u, v, and w ensure feasibility of (14b), (14c), and (14e). The z variable in (14d) represents the maximum of the linerization of c, i.e., $z_i = [\widetilde{C}^N]_i(g,\beta;\delta)$. Thus, (14) is indeed equivalent to minimizing $m(g^k,\beta^k,H^k;\delta)$ with the addition of the trust-region constraint (14f).

A step δ^k obtained from solving (14) is accepted if it results in sufficient decrease of ϕ_π, i.e., we move in the δ^k direction only if the value $\phi_\pi(g^k+\delta_g,\beta^k+\delta_\beta)$ is sufficiently smaller than $\phi_\pi(g^k,\delta^k)$. If the step is accepted, we update $g^{k+1} = g^k+\delta_g$, $\beta^{k+1} = \beta^k+\delta_\beta$ and choose $\Delta^{k+1} \ge \Delta^k$; otherwise, the iterates are not accepted and we choose $\Delta^{k+1} < \Delta^k$.

For fast local convergence, H^k is chosen as

$$\begin{aligned} H^k &= H_{\mathbb{E}} \\ &+ \lambda^k\nabla\overline{C}^N(g^k,\beta^k)\left[\nabla^2 Q_\epsilon(C^N(g^k,\beta^k))\right]\left[\nabla\overline{C}^N(g^k,\beta^k)\right]^T, \end{aligned} \quad (15)$$

where $H_{\mathbb{E}}$ is the Hessian of the expected cost given by

$$\begin{aligned} H_{\mathbb{E}} &= \begin{pmatrix} M & 0 \\ 0 & \mathbb{V}(\Omega)M \end{pmatrix}, \\ [\nabla\overline{C}^N(g^k,\beta^k)]_{\cdot i} &= \nabla c(g^k,\beta^k;\omega_i)\bar{\mu}_i^k,\ \forall i\in[N], \\ [\bar{\mu}_i^k]_j &= \frac{[\mu_i^k]_j}{\lambda^k\left[\nabla Q_\epsilon(C^N(g^k,\beta^k))\right]_i},\ \forall j\in[m],\ \forall i\in[N], \end{aligned}$$

λ^k and μ_i^k are is the multipliers corresponding to (14e) and (14d), respectively, from the previous iteration. If $\lambda^k = 0$ or $\left[\nabla Q_\epsilon(C^N(g^k,\beta^k))\right]_i = 0$, select one j such that $c_j((g^k,\beta^k;\omega_i)) = C(g^k,\beta^k;\omega_i)$ and define $[\bar{\mu}_i^k]_j = 1$ and $[\bar{\mu}_i^k]_\ell = 0$ if $\ell \ne j$.

Lastly, as the stopping criterion of the algorithm, we focus on the norm of

$$\begin{aligned} \nabla\bar{\mathcal{L}}(g^k,\beta^k,\nu^k,\lambda^k) &= \nabla\,\text{Cost}_{\mathbb{E}}(g^k,\beta^k) + \nu_1^k e_{\mathcal{G}} + \nu_2^k e_\beta \\ &+ \lambda^k\sum_{i=1}^N\left[\nabla Q_\epsilon(C^N(g^k,\beta^k))\right]_i\nabla c(g^k,\beta^k;\omega_i)\bar{\mu}_i^k, \end{aligned} \quad (16)$$

where $\bar{\mathcal{L}}$ is an appropriately chosen Lagrangian function and $\nabla \operatorname{Cost}_{\mathbb{E}}(\cdot)$ represents the gradient of the expected cost,

$$\nabla \operatorname{Cost}_{\mathbb{E}}(g,\beta) = \begin{pmatrix} Mg + v \\ \mathbb{V}(\Omega) M\beta \end{pmatrix};$$

ν_1 and ν_2 are the multipliers associated with (14b) and (14c), respectively; λ and $\bar{\mu}$ are defined as before; and $e_{\mathcal{I}} \in \mathbb{R}^{2|\mathcal{G}|}$ is a vector such that $[e_{\mathcal{I}}]_i = 1$ if $i \in \mathcal{I}$ and 0 otherwise. The function $\bar{\mathcal{L}}$ approximates the Lagrangian of a smooth optimization problem whose KKT points coincide with KKT points of (11) (see (5.10) in [25]). Thus, if $\nabla \bar{\mathcal{L}}(g^*, \beta^*, \nu^*, \lambda^*) = 0$, then the point $(g^*, \beta^*, \nu^*, \lambda^*)$ is returned as a stationary point of (12). If constraints (11b)–(11d) are satisfied by (g^*, β^*), we conclude that (g^*, β^*) is a stationary point of (11).

C. Improving the computation time

Since Q_ϵ is a non-convex function, the associated Hessian matrix H^k in (15) is not necessarily positive semi-definite. Thus, (14) need not be convex. In general, the solution of non-convex QPs is more challenging. Our experiments show that the solution times using the non-convex QP solver in `CPLEX` are generally very large and that they increase with the number of buses. To improve tractability, we replace H^k by the positive definite approximation described below.

First, notice that if $\nabla^2 Q_\epsilon(C^N(g^k, \beta^k))$ is positive semi-definite, then (15) is positive semi-definite. Thus, we replace $\nabla^2 Q_\epsilon(C^N(g^k, \beta^k))$ by a positive semi-definite approximation $\widehat{Q}^k$ in which all negative eigenvalues are replaced by zero [32, Section 3.4]. Let $A\Lambda A^T$ denote the spectral decomposition of $\nabla^2 Q_\epsilon(C^N(g^k, \beta^k))$. We define $\widehat{Q}^k$ as

$$\widehat{Q}^k = A\left(\Lambda + \operatorname{diag}(\tau_i)\right) A^T,$$

where

$$\tau_i = \begin{cases} 0, & \lambda_i \geq 0 \\ \lambda_i, & \lambda_i < 0 \end{cases}$$

The ith eigenvalue of $\nabla^2 Q_\epsilon(C^N(g^k, \beta^k))$ is denoted as λ_i. The semi-definite approximation of (15) is

$$\widehat{H}^k = H_{\mathbb{E}} + \lambda^k \nabla \overline{C}^N(g^k, \beta^k) \widehat{Q}^k \left[\nabla \overline{C}^N(g^k, \beta^k)\right]^T. \quad (17)$$

This modification ensures that $\widehat{H}^k$ is positive semi-definite at every iteration. Substituting $\widehat{H}^k$ in (14) makes the optimization problem convex and hence easier to solve. Furthermore, according to Theorem 5.3 in [25], any choice of H^k that is symmetric and bounded results in Algorithm 1 converging to a stationary point of ϕ_π. Hence, while the number of iterates that the algorithm performs might increase, the approximation H^k proposed in (17) does not affect the convergence of the algorithm to a stationary point. The computation time of the extra iterates is offset by the time saved at each iterate by solving a convex QP instead.

To further improve tractability, each QP iteration k only considers a subset of the constraints (14d). We use a lazy constraint generation technique that iteratively adds violated constraints to problem (14) until all are satisfied.

Algorithm 1 describes our proposed approach for solving (11) for a given smoothing parameter $\epsilon > 0$. Here, $\hat{\Delta} > 0$, $\Delta_0 \in (0, \hat{\Delta})$, $\eta \in (0,1)$, $\tau_1 \in (0,1)$, and $\tau_2 > 1$ are standard trust-region parameters (see [32, Chapter 4]). The tolerances $\kappa_1 > 0$ and $\kappa_2 > 0$ represent the numerical accuracy for which we consider the problem optimal and feasible, respectively.

Algorithm 1 Sℓ_1QP trust-region algorithm for CC DC-OPF

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); $\kappa_1 > 0$ and $\kappa_2 > 0$ (optimality and feasibility tolerance); $(g^0, \beta^0, \nu^0, \lambda^0, \bar{\mu}^0)$ (initial point and multipliers); set $k \leftarrow 0$

Let $\mathcal{J} = \{(j,i) \mid c_j(g^0, \beta^0; \omega_i) > -\kappa_2\}$.
while $\|\nabla \bar{\mathcal{L}}(g^k, \beta^k, \nu^k, \lambda^k)\|_\infty > \kappa_1$ **or** $\|V(g^k, \beta^k)\|_\infty > \kappa_1$ **do**
  Set $C^N(g^k, \beta^k; \omega_i) = \max_{j=1,\ldots,m}\{c_j(g^k, \beta^k; \omega_i)\}$ for all scenarios, compute $Q_\epsilon(C^N(g^k, \beta^k))$, $\nabla Q_\epsilon(C^N(g^k, \beta^k))$, and $\widehat{H}^k$.
  Obtain $(\delta, \nu, \mu, \lambda)$ by solving (14) with the constraints of the type (14d) given in $\mathcal{J}$ (**if** $\delta^k = 0$ **stop**, stationary point of ϕ_π reached).
  while There exists (j,i) such that $c_j(g^k + \delta_g, \beta^k + \delta_\beta; \omega_i) > -\kappa_2$ **do**
    $\mathcal{J} = \mathcal{J} \cup \{(j,i) \mid c_j(g^k + \delta_g, \beta^k + \delta_\beta; \omega_i) > -\kappa_2\}$
    Resolve (14) with the constraints of the type (14d) given in $\mathcal{J}$.
  end while
  Compute the ratio $\rho^k = \frac{\phi_\pi(g^k, \beta^k) - \phi_\pi(g^k + \delta_g, \beta^k + \delta_\beta)}{m(g^k, \beta^k, \widehat{H}^k; 0) - m(g^k, \beta^k, \widehat{H}^k; \delta^k)}$
  if $\rho^k < \eta$ **then**
    $\Delta^{k+1} = \tau_1 \min\{\Delta^k, \|d^k\|_\infty\}$
    $g^{k+1} = g^k$; $\beta^{k+1} = \beta^k$
    $\nu^{k+1} = \nu^k$; $\lambda^{k+1} = \lambda^k$; $\bar{\mu}^{k+1} = \bar{\mu}^k$
  else
    $g^{k+1} = g^k + \delta_g$; $\beta^{k+1} = \beta^k + \delta_\beta$
    Set $\bar{\mu}_i = \frac{\mu_i}{\lambda[\nabla Q_\epsilon(C^N(g^k, \beta^k))]_i}$, for all $i = [N]$.
    $\nu^{k+1} = \nu$; $\lambda^{k+1} = \lambda$; $\bar{\mu}^{k+1} = \bar{\mu}$
    if $\rho^k \geq \eta$ **and** $\|d^k\|_\infty = \Delta^k$ **then**
      $\Delta^{k+1} = \min\{\tau_2 \Delta^k, \hat{\Delta}\}$
    else
      $\Delta^{k+1} = \Delta^k$
    end if
  end if
  $k = k + 1$
end while

Return: $(g^k, \beta^k, \nu^k, \lambda^k, \bar{\mu}^k)$, optimal solution and multipliers.

V. Selecting the Smooth-Quantile Parameters

The smooth approximation of the quantile Q_ϵ is motivated by a kernel estimation of the cdf. The properties of the kernel approximation of the cdf can be extended to those of the quantile [33]. These properties indicate that large values of ϵ help to reduce the variance among the estimators obtained from different samples, but can lead to biased estimators that are either consistently infeasible or consistently conservative. We have also empirically observed that increasing the value of ϵ decreases the existence of spurious "non-convexities" in the feasible region. These "non-convexities" are not inherent to the problem but introduced by the sample, and they may

cause local optimization algorithms (such as ours) to converge to local minima with worse objective values. Given the two arguments above, using larger values of ϵ is advantageous as long as we are aware of the bias from a large choice of ϵ.

To counteract the bias of the solution, we propose to relax or strengthen the quantile constraint (11d) by adjusting the right-hand side by $t \in \mathbb{R}$ as follows:

$$Q_\epsilon(C^N(g,\beta)) \leq t. \tag{18}$$

If $t < 0$, (18) is stronger than (11d); if $t > 0$, (11d) is relaxed.

A. Choosing the value of ϵ

To select the smoothing parameter ϵ, we first solve problem (11) for 10 different samples, with $N = 100$ scenarios per sample, and perform the binary search algorithm over ϵ, similar to Algorithm 2 [25]. We then pick the maximum value of ϵ from the samples that achieves the target probability and denote this value as $\hat{\epsilon}$. Based on our empirical experience with $N = 100$, this choice of ϵ seems to minimize the existence of spurious local minima without being too conservative.

A result from [33] provides an approximation of ϵ for different sample sizes. Reference [33] proves that, asymptotically, the optimal choice of ϵ that minimizes the mean square error between the true quantile and the approximated quantile is $\mathcal{O}(N^{-1/3})$. This can be used to estimate appropriate smoothing parameter values for sample sizes other than 100. Thus, for a given sample size N, we consider $\epsilon = \frac{(100^{1/3})\hat{\epsilon}}{N^{1/3}}$.

B. Binary search to determine t

After choosing the smoothing parameter ϵ based on N, we tune the parameter t using the binary search in Algorithm 2 for each separate sample. The parameter $\tau > 0$ determines the maximum difference allowed between the probability attained by the solution (g^*, β^*) and the target probability $1 - \alpha$.

Algorithm 2 Binary search for the right-hand side t

Inputs: $t_0 = 0$; $t_{LB} = -\infty$; $t_{UB} = \infty$; $\tau_t, \kappa_t > 0$; $\ell = 0$
  repeat
    Obtain (g^*, β^*) by solving (11) with t_ℓ.
    Approximate $p_\ell \approx \mathbb{P}(C(g^*, \beta^*; \omega) \leq 0)$.
    if $p_\ell > 1 - \alpha$ **then**
      $t_{\text{UB}} = t_\ell$
      if $t_{\text{LB}} = -\infty$ **then** $t_{\ell+1} = t_\ell - \kappa_t$ **end if**
      if $t_{\text{LB}} > -\infty$ **then** $t_{\ell+1} = (t_{\text{LB}} + t_\ell)/2$ **end if**
    else
      $t_{\text{LB}} = t_\ell$
      if $t_{\text{UB}} = \infty$ **then** $t_{\ell+1} = t_\ell + \kappa_t$ **end if**
      if $t_{\text{UB}} < \infty$ **then** $t_{\ell+1} = (t_{\text{UB}} + t_\ell)/2$ **end if**
    end if
  until $|p_\ell - (1 - \alpha)| \leq \tau_t$
Return: t_ℓ, (g^*, β^*).

When implementing Algorithm 2, we use the optimal solution and multipliers obtained when solving for the right-hand side t^ℓ as the initial points and multipliers for solving the problem with $t^{\ell+1}$. Once the binary search terminates, we select the solution from the ℓth right-hand side iteration that is feasible and has the best objective value. This ensures that we select the best of all the considered values for t.

VI. Case Study

This section demonstrates our method (denoted as the "NLP approach") using variants of the IEEE 14-, 57-, and 118-bus systems from `pglib-opf` [34]. We compare our method against the scenario-approach [19], [21], [35] and the deterministic problem with $\omega = 0$; we refer to the solution of the deterministic problem as the "nominal solution". The obtained solutions are said to be good if they are: (1) consistent over different samples, (2) feasible for the true problem (evaluated with an out-of-sample test), and (3) low cost.

All computations were executed on `Ubuntu 16.04` with 256GB RAM and two `Intel Xeon` processors each with ten 3.10GHz cores. The algorithm is implemented in `Matlab R2015b`, using `CPLEX 12.6.3` to solve the QP in (14). We use the parameters $\pi = 100$, $\hat{\Delta} = 10^6$, $\Delta_0 = 1$, $\eta = 10^{-8}$, $\tau_1 = 1/2$, $\tau_2 = 2$, $\kappa_1 = 10^{-6}$, $\kappa_2 = 0.1$, $\tau_t = 10^{-4}$ and $\kappa_t = 0.01$.

To initiate the search for the right-hand side t, i.e., when $t^0 = 0$, we choose the initial points and multipliers for Algorithm 1 as follows: g^0 as the optimal solution of (4) for $\omega = 0$, $\beta^0 = 1/|\mathcal{G}|$, $\lambda^0 = 0$, and $\bar{\mu}^0$ as described in Section IV using $\lambda^0 = 0$. For subsequent values of t^ℓ, we initiate g^0, β^0, λ^0, and $\bar{\mu}^0$ using the optimal solutions returned from solving the problem with the previous right-hand side, $t^{\ell-1}$.

A. Uncertainty modeling

The experiments in this section are based on normally distributed and independent loads, i.e., $\omega \sim N(\vec{0}, \Sigma)$, where Σ is a diagonal matrix. The diagonal entries of Σ are ζd, where ζ is a constant and d is the vector of forecasted demands. We let $\zeta = 0.1$ for cases 14 and 57; for case 118, we consider $\zeta = 0.05$ since the problem is infeasible for larger values of the value ζ. In case 118, we also consider $\zeta = 0.01$ to compare the quality of the NLP solutions for different levels of variability. We aim to satisfy the probabilistic constraint at least 95% of the time, i.e., $\alpha = 0.05$. For all solutions obtained in this section, out-of-sample approximations of $\mathbb{P}(C(g^*, \beta^*; \omega) \leq 0)$ are computed using the empirical cdf with $N = 10^6$ scenarios.

B. Demonstration of joint chance-constraints

We first show how the algorithm performs with different sample sizes N. Table I presents the results of running 10 replications of the algorithm. The computation times are given in seconds; these times include the total time for the binary search algorithm to find the right-hand side t.

First, note that the variability in the objective value decreases with increasing sample size, as expected; however, the variability is small even for $N = 100$ scenarios. This indicates that when using a small number of scenarios, the NLP approach performs favourably. Second, the solutions returned for cases 14, 57, and 118 (with $\zeta = 0.01$) are always feasible for the true problem and, on average, are close to the prescribed risk level. For case 118 with $\zeta = 0.05$, all solutions are feasible for a sample size of at least 200. For $N = 100$, the solutions obtained by the NLP approach on three of these instances are not feasible for the out-of-sample approximation of (4d). We believe that this happens because the number of

Case 14	N = 100	N = 200	N = 500	N = 1000
Min. obj ($)	2.0868	2.0783	2.0824	2.0810
Avg. obj ($)	2.0958	2.0976	2.0979	2.0911
Max. obj ($)	2.1291	2.1549	2.1385	2.1189
Min. prob	0.950	0.950	0.950	0.950
Avg. prob	0.950	0.950	0.950	0.951
Max. prob	0.950	0.950	0.950	0.957
Min. time (s)	0.3361	0.6295	4.4579	8.3221
Avg. time (s)	1.7767	2.6536	6.397	11.178
Max. time (s)	4.0661	5.8929	11.199	17.912
Avg. t ($\times 10^{-3}$)	-5.2656	-5.6562	-3.6797	-5.5625

Case 57	N = 100	N = 200	N = 500	N = 1000
Min. obj ($)	35.336	35.235	35.328	35.307
Avg. obj ($)	35.503	35.408	35.374	35.344
Max. obj ($)	35.606	35.490	35.429	35.385
Min. prob	0.950	0.950	0.950	0.950
Avg. prob	0.958	0.954	0.950	0.950
Max. prob	0.983	0.969	0.950	0.950
Min. time (s)	2.3305	8.4275	13.770	29.068
Avg. time (s)	7.8157	14.559	39.345	157.87
Max. time (s)	13.457	30.283	60.585	319.16
Avg. t ($\times 10^{-2}$)	-1.8922	-2.2641	-1.2156	-1.0281

Case 118 ($\zeta = 0.01$)	N = 100	N = 200	N = 500	N = 1000
Min. obj ($)	112.43	112.13	112.03	111.86
Avg. obj ($)	112.61	112.30	112.11	111.98
Max. obj ($)	112.79	112.67	112.31	112.08
Min. prob	0.950	0.950	0.950	0.950
Avg. prob	0.950	0.950	0.950	0.950
Max. prob	0.950	0.950	0.950	0.950
Min. time (s)	11.461	71.176	266.90	1308.7
Avg. time (s)	58.891	110.95	866.48	2585.7
Max. time (s)	101.18	144.20	1439.2	4244.0
Avg. t ($\times 10^{-3}$)	-5.2969	-8.7227	-5.7422	-3.8906

Case 118 ($\zeta = 0.05$)	N = 100*	N = 200	N = 500	N = 1000
Min. obj	116.42	116.09	115.96	115.55
Avg. obj	116.67	116.31	116.11	115.87
Max. obj	117.06	116.76	116.22	116.09
Min. prob	0.950	0.950	0.950	0.950
Avg. prob	0.953	0.950	0.950	0.950
Max. prob	0.965	0.950	0.950	0.950
Min. time	70.303	162.56	1287.4	2913.7
Avg. time	102.94	277.82	1573.5	3939.9
Max. time	127.07	533.59	2694.8	5174.4
Avg. t ($\times 10^{-3}$)	-4.64	-0.33	5.047	3.609

TABLE I: Results from Algorithm 1 using $\hat{\epsilon}_{14} = 3.7(10)^{-2}$; $\hat{\epsilon}_{57} = 1.8(10)^{-1}$; $\hat{\epsilon}_{118} = 7.5(10)^{-2}$ ($\zeta = 0.01$); $\hat{\epsilon}_{118} = 8.5(10)^{-2}$ ($\zeta = 0.05$). ***:** Statistics of feasible instances.

scenarios is insufficient for the level of variability. Finally, although many of the solutions returned by the algorithm achieve the target risk-level, the ones that do not is because the algorithm gets stuck in a spurious local minimum.

C. Comparison of the NLP and scenario approaches

This section compares our solutions to those obtained from the scenario approach (SA) [19], which approximates (4) as

$$\min_{g,\alpha} \quad c(g) \tag{19a}$$

$$\text{s.t.} \quad f_{ij}^{\text{LB}} \leq \Phi\hat{p}(\omega_s) \leq f_{ij}^{\text{UB}}, \quad \forall\, ij \in \mathcal{L},\ \forall s \in [N^{\text{SA}}] \tag{19b}$$

$$g_i^{\text{LB}} \leq g_i - \beta_i \Omega_s \leq g_i^{\text{UB}}, \ \forall\, i \in \mathcal{G},\ \forall s \in [N^{\text{SA}}] \tag{19c}$$

$$\text{Eqns. (4b), (4c).} \tag{19d}$$

where N^{SA} is a pre-specified number of scenarios. The goal of the SA is to specify a minimum number of scenarios N^{SA} such that a solution to (19), which is feasible for all N^{SA} scenarios, is also feasible for the probabilistic constraint (4d) with a probability of at least $1 - \sigma$.

We use the sample size given in [35] to select N^{SA},

$$N^{\text{SA}} \geq \frac{2}{\alpha}\left(\ln\left(\frac{1}{\sigma}\right) + n\right),$$

for $n = 2|\,\mathcal{G}\,|$, in this case. We select $\sigma = 10^{-4}$.

Comparisons between the solutions obtained from the SA and NLP methods are presented in Table II. The SA problem is solved for 10 different samples. We report the minimum, average, and maximum values of the objective and the out-of-sample probability of the returned solutions. If at least one of the instances is infeasible, the minimum probability is considered to be zero and the maximum objective function is marked as Inf. The average reported in the table does not consider the instances where the problem is infeasible. The number of infeasible instances using the SA approach for the different cases are: (1) Case 14: 8, (2) Case 57: 1, (3) Case 118 ($\zeta = 0.01$): 0, and (4) Case 118 ($\zeta = 0.05$): 4.

First, it can be seen that despite the nominal solution being the least expensive, the solutions obtained when the uncertainty is ignored are far from being feasible for cases 57 and 118. Notice that there are many instances for which the SA algorithm cannot obtain a feasible solution to the approximated problem (19). On the other hand, the NLP approach can always find a solution to the approximated problem (11); yet, for three instances of case 118, with $\zeta = 0.05$ and $N = 100$, the solutions to the approximated problem are only feasible $93 - 94\%$ to the true problem. Second, it can be seen that the best solution obtained from the SA approach can be up to 16% more expensive than the worst NLP solution (see case 14). This demonstrates the NLP approach's ability to provide solutions that are feasible without being overly conservative. Finally, when comparing both 118 cases for low-variance regimes ($\zeta = 0.01$), the NLP approach works comparably to the SA. However, as the variance increases, Algorithm 1 returns solutions that are at least 4% cheaper than using SA. When allowing for more computation time in order to use $N = 200$ scenarios, the advantage of the NLP approach becomes apparent even for the low-variance regimes.

VII. Conclusions and Outlook

This paper has developed a sample-based NLP algorithm for solving DC-OPF problems with joint chance constraints. By tuning two parameters in this algorithm using a proposed heuristic approach, the solutions obtained via this algorithm balance feasibility of the chance constraints and operational costs. Empirical results on several IEEE test cases demonstrate the algorithm's ability to jointly enforce chance constraints while being significantly less conservative with respect to operational costs than the alternative "scenario approach" proposed in prior literature. Our ongoing work is extending this approach to AC-OPF problems with joint chance constraints.

References

[1] P. Panciatici, M. C. Campi, S. Garatti, S. H. Low, D. K. Molzahn, A. X. Sun, and L. Wehenkel, "Advanced optimization methods for power systems," in *Power Syst. Comput. Conf. (PSCC)*, 2014.

Case 14	Nominal	SA	NLP(100)	NLP(200)
Min. obj	-	2.4706	2.0868	2.0783
Avg. obj	2.0515	2.4767	2.0958	2.0976
Max. obj	-	Inf	2.1291	2.1549
Min. prob	-	0	0.950	0.950
Avg. prob	0.945	0.994	0.950	0.950
Max. prob	-	0.994	0.950	0.950
Min. time	-	0.0124	0.3361	0.6295
Avg. time	0.0036	0.0381	1.7767	2.6536
Max. time	-	0.0415	4.0661	5.8929
N^{SA}/Avg. t	1	516	-0.0053	-0.0057

Case 57	Nominal	SA	NLP(100)	NLP(200)
Min. obj	-	35.568	35.336	35.235
Avg. obj	34.773	35.649	35.503	35.408
Max. obj	-	Inf	35.606	35.490
Min. prob	-	0	0.950	0.950
Avg. prob	0.376	0.996	0.958	0.954
Max. prob	-	0.998	0.983	0.969
Min. time	-	0.2054	2.3305	8.4275
Avg. time	0.0041	0.2395	7.8157	14.559
Max. time	-	0.2639	13.457	30.283
N^{SA}/Avg. t	1	637	-0.0189	-0.0226

Case 118 ($\zeta = 0.01$)	Nominal	SA	NLP(100)	NLP(200)
Min. obj	-	112.39	112.43	112.13
Avg. obj	109.79	112.69	112.61	112.30
Max. obj	-	113.15	112.79	112.67
Min. prob	-	0.995	0.950	0.950
Avg. prob	0.112	0.997	0.950	0.950
Max. prob	-	0.999	0.950	0.950
Min. time	-	81.086	11.461	71.176
Avg. time	0.0071	87.485	58.891	110.95
Max. time	-	91.676	101.18	144.20
N^{SA}/Avg. t	1	2998	-0.0053	-0.0087

Case 118 ($\zeta = 0.05$)	Nominal	SA	NLP(100)*	NLP(200)
Min. obj	-	122.77	116.42	116.09
Avg. obj	109.79	127.32	116.67	116.31
Max. obj	-	Inf	117.06	116.76
Min. prob	-	0	0.950	0.950
Avg. prob	0.112	0.995	0.953	0.950
Max. prob	-	0.997	0.965	0.950
Min. time	-	19.779	70.303	162.56
Avg. time	0.0071	90.534	102.94	277.82
Max. time	-	94.012	127.07	533.59
N^{SA}/Avg. t	1	2998	-0.0046	-0.0003

TABLE II: **Nominal:** Solution of (4) for $\omega = 0$. **SA:** SA with $\sigma = 10^{-4}$. **NLP(100):** Algorithm 1 with $N = 100$. **NLP(200):** Algorithm 1 with $N = 200$. ***:** Statistics of feasible instances.

[2] F. Bouffard and F. D. Galiana, "Stochastic security for operations planning with significant wind power generation," *IEEE Trans. Power Syst.*, vol. 23, pp. 306–316, May 2008.

[3] J. Morales, A. Conejo, and J. Perez-Ruiz, "Economic valuation of reserves in power systems with high penetration of wind power," *IEEE Trans. Power Syst.*, vol. 24, no. 2, pp. 900–910, May 2009.

[4] A. Papavasiliou and S. S. Oren, "Multi-area stochastic unit commitment for high wind penetration in a transmission constrained network," *Operations Research*, vol. 61, no. 3, pp. 578–592, 2013.

[5] P. Panciatici, Y. Hassaine, S. Fliscounakis, L. Platbrood, M. Ortega-Vazquez, J. Martinez-Ramos, and L. Wehenkel, "Security management under uncertainty: From day-ahead planning to intraday operation," in *IREP Symposium*, Buzios, Brazil, Aug 2010, pp. 1–8.

[6] R. A. Jabr, "Adjustable robust OPF with renewable energy sources," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 4742–4751, Nov 2013.

[7] J. Warrington, P. J. Goulart, S. Mariethoz, and M. Morari, "Policy-based reserves for power systems," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 4427–4437, 2013.

[8] A. Lorca and X. A. Sun, "Adaptive robust optimization with dynamic uncertainty sets for multi-period economic dispatch under significant wind," *IEEE Trans. Power Syst.*, vol. 30, no. 4, pp. 1702–1713, July 2015.

[9] H. Zhang and P. Li, "Chance constrained programming for optimal power flow under uncertainty," *IEEE Trans. Power Syst.*, vol. 26, no. 4, pp. 2417–2424, Nov. 2011.

[10] M. Vrakopoulou, K. Margellos, J. Lygeros, and G. Andersson, "A probabilistic framework for reserve scheduling and N-1 security assessment of systems with high wind power penetration," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 3885–3896, 2013.

[11] L. Roald, F. Oldewurtel, T. Krause, and G. Andersson, "Analytical reformulation of security constrained optimal power flow with probabilistic constraints," in *IEEE Grenoble PowerTech*, 6 2013, pp. 1–6.

[12] D. Bienstock, M. Chertkov, and S. Harnett, "Chance-constrained optimal power flow: Risk-aware network control under uncertainty," *SIAM Review*, vol. 56, no. 3, pp. 461–495, 2014.

[13] T. Summers, J. Warrington, M. Morari, and J. Lygeros, "Stochastic optimal power flow based on convex approximations of chance constraints," in *Power Syst. Comput. Conf. (PSCC)*, Aug. 2014.

[14] K. Baker and A. Bernstein, "Joint chance constraints in AC optimal power flow: Improving bounds through learning," *IEEE Trans. Smart Grid*, 2019.

[15] X. Geng and L. Xie, "Data-driven decision making with probabilistic guarantees (part 1): A schematic overview of chance-constrained optimization," *arXiv preprint arXiv:1903.10621*, 2019.

[16] R. Henrion, "Structural properties of linear probabilistic constraints," *Optimization*, vol. 56, no. 4, pp. 425–440, 2007.

[17] W. Chen, M. Sim, J. Sun, and C.-P. Teo, "From CVaR to uncertainty set: Implications in joint chance-constrained optimization," *Operations research*, vol. 58, no. 2, pp. 470–485, 2010.

[18] L. Roald, S. Misra, T. Krause, and G. Andersson, "Corrective control to handle forecast uncertainty: A chance constrained optimal power flow," *IEEE Trans. Power Syst.*, vol. 32, no. 2, pp. 1626–1637, 2016.

[19] G. Calafiore and M. C. Campi, "Uncertain convex programs: Randomized solutions and confidence levels," *Mathematical Programming*, vol. 102, no. 1, pp. 25–46, 2005.

[20] G. C. Calafiore and M. C. Campi, "The scenario approach to robust control design," *IEEE Trans. Autom. Control*, vol. 51, no. 5, pp. 742–753, 2006.

[21] A. Nemirovski and A. Shapiro, "Scenario approximations of chance constraints," in *Probabilistic and Randomized Methods for Design Under Uncertainty*. Springer, 2006, pp. 3–47.

[22] J. Luedtke and S. Ahmed, "A sample approximation approach for optimization with probabilistic constraints," *SIAM Journal on Optimization*, vol. 19, no. 2, pp. 674–699, 2008.

[23] L. Roald and G. Andersson, "Chance-constrained AC optimal power flow: Reformulations and efficient algorithms," *IEEE Trans. Power Syst.*, vol. 33, no. 3, pp. 2906–2918, 2017.

[24] M. S. Modarresi, L. Xie, M. Campi, S. Garatti, A. Carè, A. Thatte, and P. Kumar, "Scenario-based economic dispatch with tunable risk levels in high-renewable power systems," *IEEE Trans. Power Syst.*, 2018.

[25] A. Peña-Ordieres, J. R. Luedtke, and A. Wächter, "Solving chance-constrained problems via a smooth sample-based nonlinear approximation," *arXiv:1905.07377*, May 2019.

[26] A. J. Wood, B. F. Wollenberg, and G. B. Sheble, *Power Generation, Operation and Control*, 3rd ed. John Wiley and Sons, Inc., 2013.

[27] A. Geletu, A. Hoffmann, M. Kloppel, and P. Li, "An inner-outer approximation approach to chance constrained optimization," *SIAM Journal on Optimization*, vol. 27, no. 3, pp. 1834–1857, 2017.

[28] A. Shapiro, D. Dentcheva, and A. Ruszczyński, *Lectures on stochastic programming: Modeling and theory*, 2nd ed. SIAM, 2014.

[29] D. W. Scott, R. A. Tapia, and J. R. Thompson, "Kernel density estimation revisited," *Nonlinear Analysis: Theory, Methods & Applications*, vol. 1, no. 4, pp. 339–372, 1977.

[30] J.-B. Hiriart-Urruty and C. Lemarechal, *Convex Analysis and Minimization Algorithms I: Fundamentals*. Springer Science & Business Media, 1996, vol. 305.

[31] F. E. Curtis and M. L. Overton, "A sequential quadratic programming algorithm for nonconvex, nonsmooth constrained optimization," *SIAM Journal on Optimization*, vol. 22, no. 2, pp. 474–500, 2012.

[32] J. Nocedal and S. Wright, *Numerical optimization*. Springer Science & Business Media, 2006.

[33] A. Azzalini, "A note on the estimation of a distribution function and quantiles by a kernel method," *Biometrika*, vol. 68, no. 1, pp. 326–328, 1981.

[34] IEEE PES Task Force on Benchmarks for Validation of Emerging Power System Algorithms, "The Power Grid Library for Benchmarking AC Optimal Power Flow Algorithms," *arXiv:1908.02788*, Aug. 2019.

[35] M. C. Campi, S. Garatti, and M. Prandini, "The scenario approach for systems and control design," *Annual Reviews in Control*, vol. 33, no. 2, pp. 149–157, 2009.