

# 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 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  $S\ell_1$ QP-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  $\alpha$  is a specified acceptable violation probability.

A. Peña-Ordieres and A. Wächter are with the Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL, 60208. alejandra.pena@u.northwestern.edu, andreas.waechter@northwestern.edu

D. Molzahn is with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332. molzahn@gatech.edu

L. Roald is with the Department of Electrical and Computer Engineering, University of Wisconsin–Madison, Madison, WI, 53706. roald@wisc.edu

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 second-order cone programs 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 SCCs, 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 SCC 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, these solutions are 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 DC OPF problem.

The formulation is based on a sample average approximation (SAA) which gives rise to a continuous non-linear programming (NLP) problem. The algorithm is an adaptation of the JCC algorithm presented in [25]. Making this algorithm applicable to electric power systems requires careful consideration of the formulation and several non-trivial modifications to the algorithm. To improve computational performance, we (1) select a DC OPF formulation that avoids the need to replicate certain variables for each scenario, (2) approximate the Hessian of the quantile function defined in [25] in order to solve a more tractable convex problem at each iteration of the algorithm, and (3) develop a lazy constraint generation algorithm to exploit the observation that a small number of line flow limits are binding at the solutions to typical DC OPF problems [26]. To improve the solution quality relative to a naïve application of [25], we propose methods for adaptively tuning the key parameters introduced in the quantile approximation. This tuning improves the out-of-sample feasibility of the resulting solutions.

The NLP approach has several advantages: (1) *Sample-based*: Similar to the SA and MIP methods, 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] depends on certain parameters that impact the conservativeness of the solution. We propose two methods that adaptively and automatically tune these parameters such that the resulting solution accurately satisfies 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 in the JCC-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 an 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  $\omega$  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 makes the following assumptions that are standard to all DC power flow formulations [27]: (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  $\omega$  is a random variable with zero mean; this can be interpreted as the sum of the forecasted value  $d$  and its fluctuation  $\omega$ . Due to the nature of the renewable energy uncertainty, we model  $\omega$  as a continuous random variable. We note that the random variable  $\omega$  models the uncertainty in the net load, i.e., the load demands minus the outputs of stochastic renewable generators.

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,

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

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

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

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

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  $\omega \neq 0$ .

If  $g(\omega)$  did not follow the affine functional form in (1), a copy of the variable  $g(\omega)$  would need to be introduced for each realization of  $\omega$  to satisfy (2). For a sample-based approach, this implies that the number of variables in the problem would depend on the number of scenarios considered in the sample. Thus, the choice of the functional form of  $g(\omega)$  is relevant to decreasing the complexity of the JCC-OPF problem.

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  $p(\omega) = g(\omega) - d(\omega)$ :

$$f_{ij}(\omega) = \Phi_{(.,ij)} p(\omega). \quad (3)$$

The matrix  $\Phi$  denotes the DC power transfer distribution factors (DC-PTDFs) [28], with  $\Phi_{(.,ij)}$  referring to the row of  $\Phi$  corresponding to the line  $ij$ . We note that the DC-PTDF formulation of the DC power flow equations, i.e., the combination of (2) and (3), implicitly ensures power balance at every bus in the system [12]. The DC-PTDF formulation of the DC OPF problem used in this paper is equivalent to alternative formulations that explicitly include variables for the voltage angles at every bus.

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

$$\text{Cost}(x) = \frac{1}{2}x^T Mx + v^T x + k_0, \quad (4)$$

where  $M$  is a diagonal matrix with non-negative entries. We minimize the expected generation cost  $\mathbb{E}[\text{Cost}(g(\omega))]$ . Substituting (1) and taking the expectation in (4), we obtain

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

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

$$\min_{g, \beta} \text{Cost}(g) + \frac{1}{2}\mathbb{V}(\Omega)\beta^T M\beta \quad (5a)$$

$$\text{s.t. } \sum_{i \in \mathcal{G}} g_i - \sum_{i \in \mathcal{B}} d_i = 0, \quad (5b)$$

$$\sum_{i \in \mathcal{G}} \beta_i = 1, \quad (5c)$$

$$\mathbb{P}\left(\begin{array}{l} f_{ij}^{\text{LB}} \leq \Phi p(\omega) \leq f_{ij}^{\text{UB}}, \quad \forall i, j \in \mathcal{L} \\ g_i^{\text{LB}} \leq g_i - \beta_i \Omega \leq g_i^{\text{UB}}, \quad \forall i \in \mathcal{G} \end{array}\right) \geq 1 - \alpha. \quad (5d)$$

The objective (5a) minimizes the expected cost. The deterministic constraints (5b), (5c) ensure power balance. The JCC (5d) enforces bounds on the line flows and generator outputs,  $f_{ij}^{\text{LB}}, f_{ij}^{\text{UB}}$  and  $g_i^{\text{LB}}, g_i^{\text{UB}}$ , with probability  $1 - \alpha$ . Here,  $\alpha$  represents the acceptable violation probability.

### III. REPRESENTATION OF CHANCE CONSTRAINTS

Constraint (5d) results in 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 smooth sample-based approximation of chance constraints to efficiently solve chance-constrained problems. This section summarizes the method from [25] and discusses modifications needed to address (5). We refer to one realization of  $\omega$  as a “scenario” and a set of scenarios as a “sample”.

We begin by defining the  $(1 - \alpha)$ -quantile of a generic 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  $\widehat{Y} = \max_{j=1, \dots, m}\{Y_j\}$  and consider  $Q^{1-\alpha}(\widehat{Y})$  instead.

We denote the probabilistic constraint (5d) as  $\mathbb{P}(c(g, \beta; \omega) \leq 0) \geq 1 - \alpha$ , where the random constraint vector  $c(g, \beta; \omega)$  is defined by

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

Then,  $c(g, \beta; \omega) \in \mathbb{R}^m$ , where  $m = 2|\mathcal{L}| + 2|\mathcal{G}|$ . Since constraints  $c_j(g, \beta; \omega) \leq 0$  for  $j = 1, \dots, m$  are equivalent

to  $C(g, \beta; \omega) = \max_{j=1, \dots, m}\{c_j(g, \beta; \omega)\} \leq 0$ , the single chance constraint  $\mathbb{P}(C(g, \beta; \omega) \leq 0) \geq 1 - \alpha$  is equivalent to the joint chance constraint  $\mathbb{P}(c_j(g, \beta; \omega) \leq 0, j = 1, \dots, m) \geq 1 - \alpha$ . Reformulating (5) yields

$$\begin{aligned} \min_{g, \beta} \quad & \text{Cost}(g) + \frac{1}{2}\mathbb{V}(\Omega)\beta^T M\beta \\ \text{s.t.} \quad & Q^{1-\alpha}(C(g, \beta; \omega)) \leq 0, \\ & \text{Eqns. (5b), (5c).} \end{aligned} \quad (6)$$

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  $(g, \beta)$ , an approximation of the quantile can be obtained from an approximation of the cdf.

For the rest of Section III, we assume that  $C(g, \beta; \omega)$  defines a smooth function. This is clearly not the case because  $C(g, \beta; \omega)$  is the maximum of the linear constraints given by the vector  $c(g, \beta; \omega)$ . However, we postpone the discussion of the non-smoothness of  $C(g, \beta; \omega)$  to Section IV in order to introduce the approximation of the quantile via the cdf in a simplified manner.

One way to approximate the cdf is to consider a sample  $\{\omega_1, \dots, \omega_N\}$  of the random variable  $\omega$ . 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

$$F^N(t; g, \beta) = \frac{1}{N} \sum_{i=1}^N \mathbb{1}(C(g, \beta; \omega_i) \leq t), \quad (7)$$

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  $(g, \alpha)$ , we follow an approach similar to [29], [30] by defining

$$F_\epsilon^N(t; g, \beta) = \frac{1}{N} \sum_{i=1}^N \Gamma_\epsilon(C(g, \beta; \omega_i) - t), \quad (8)$$

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

$$\Gamma_\epsilon(y) = \begin{cases} 1, & y \leq -\epsilon \\ \gamma_\epsilon(y), & -\epsilon < y < \epsilon \\ 0, & y \geq \epsilon \end{cases} \quad (9)$$

and  $\gamma_\epsilon : [-\epsilon, \epsilon] \rightarrow [0, 1]$  is a symmetric and strictly decreasing function such that  $\Gamma_\epsilon$  is continuously differentiable. With this choice of  $\gamma_\epsilon$ ,  $F_\epsilon^N(t; g, \beta)$  is a differentiable approximation of the empirical cdf,  $F^N(t; g, \beta)$  (see Fig. 1). We use the following  $\gamma_\epsilon$  function based on the quartic kernel [31, p. 353], which makes (9) twice continuously differentiable:

$$\gamma_\epsilon(y) = \frac{15}{16} \left( -\frac{1}{5} \left( \frac{y}{\epsilon} \right)^5 + \frac{2}{3} \left( \frac{y}{\epsilon} \right)^3 - \left( \frac{y}{\epsilon} \right) + \frac{8}{15} \right). \quad (10)$$

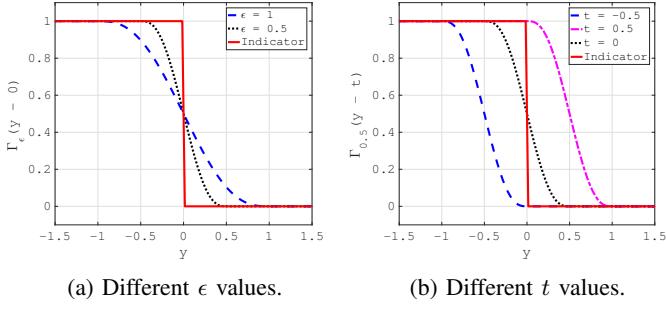


Fig. 1: Function  $\Gamma_\epsilon(y - t)$ .

For a fixed  $g$  and  $\beta$ , the approximation of  $Q^{1-\alpha}$  can be computed as the inverse of  $F_\epsilon^N$  at  $1 - \alpha$ . The inverse can be obtained from the value  $Q_\epsilon$  such that

$$\sum_{i=1}^N \Gamma_\epsilon(C(g, \alpha; \omega_i) - Q_\epsilon) = N(1 - \alpha). \quad (11)$$

Reference [25] shows that  $Q_\epsilon$  results in an approximation of  $Q^{1-\alpha}$  at  $(g, \alpha)$ . It also shows that the value  $Q_\epsilon$  is unique, under mild conditions, and that it defines a function that maps the vector  $C^N(g, \beta) = [C(g, \beta; \omega_1), \dots, C(g, \beta; \omega_N)] \in \mathbb{R}^N$  to the root of (11). We denote this function as  $Q_\epsilon(C^N(g, \beta))$ .

Hence, we propose the following approximation to (5):

$$\min_{g, \beta} \text{Cost}(g) + \frac{1}{2} \mathbb{V}(\Omega) \beta^T M \beta \quad (12a)$$

$$\text{s. t. } \sum_{i \in \mathcal{G}} g_i - \sum_{i \in \mathcal{B}} d_i = 0, \quad (12b)$$

$$\sum_{i \in \mathcal{G}} \beta_i = 1, \quad (12c)$$

$$Q_\epsilon(C^N(g, \beta)) \leq 0. \quad (12d)$$

Notice that  $Q_\epsilon$  has taken the place of the chance constraint (5d) following the formulation in (6). Reference [25] discusses the convergence and feasibility of this approximation with respect to the solutions of the original problem (5) with increasing sample size. Section V details the choice of  $\epsilon$  and  $t$ .

Since  $C(g, \beta; \omega)$  is the maximum of the entries of the vector  $c(g, \beta; \omega)$ , (12d) results in a non-smooth function; this raises algorithmic challenges. Section IV discusses how to solve (12).

#### IV. SOLUTION ALGORITHM

Adopting from the approach in [25], this section proposes an algorithm for solving (12). To avoid having a non-smooth constraint, namely (12d), we first reformulate (12) as an equivalent unconstrained optimization problem in which the constraints are added to the objective function via terms that penalize infeasible solutions. To address the challenges arising from 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 that make standard solvers compute the updates faster.

#### A. $\ell_1$ -penalty function

Let  $\pi > 0$  be a penalty parameter and  $[x]^+ = \max\{0, x\}$ . We propose an  $\ell_1$ -penalty function in order to solve (12):

$$\phi_\pi(g, \beta) = \text{Cost}(g) + \frac{1}{2} \mathbb{V}(\Omega) \beta^T M \beta + \pi \|V(g, \beta)\|_1 \quad (13)$$

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  $(g^*, \beta^*)$  is a local minimizer of  $\phi_\pi$  for  $\pi > 0$  and is feasible for problem (12), then  $(g^*, \beta^*)$  solves (12) [32, p. 299]. Moreover, Theorem 2.1 in [33] shows that, under standard assumptions, there exists  $\pi^* > 0$  such that the minimization of (13) yields a solution for (12) for all  $\pi \geq \pi^*$ . This property makes the performance of exact penalty methods less dependent on the strategy for updating the penalty parameter than other penalty methods [34, p. 507].

#### B. Minimizing the $\ell_1$ -penalty function

To minimize (13), we propose an  $S\ell_1$ QP-type trust-region algorithm that solves a sequence of quadratic programs (QP). At each iteration  $k$ ,  $\phi_\pi$  is approximated with a piecewise quadratic function that depends on  $(g, \beta)$  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 penalty function  $\phi_\pi$ . In each iteration of the algorithm, a trial step is computed as the minimizer of the model within the trust region. If sufficient progress is made, the trial step is accepted. Otherwise, the trust region radius is reduced and a new trial step is computed.

To be able to prove global convergence of the algorithm, the model must approximate the penalty function to first order. The following non-standard piecewise quadratic model was developed in [25] specifically for problems such as (12) using a chain-rule-type approach,

$$\begin{aligned} m(g, \beta, H; \delta) = & \text{Cost}(g) + \frac{1}{2} \mathbb{V}(\Omega) \beta^T M \beta + \nabla \text{Cost}(g)^T \delta_g \\ & + \mathbb{V}(\Omega) \beta^T M \delta_\beta + \frac{1}{2} \delta^T H \delta + \pi \left( \left| \sum_{i \in \mathcal{G}} (g_i + \delta_{g_i}) - \sum_{i \in \mathcal{B}} d_i \right| \right. \\ & \left. + \left| \sum_{i \in \mathcal{G}} (\beta_i + \delta_{\beta_i}) - 1 \right| + [\tilde{Q}_{\epsilon, C}(g, \beta; \delta)]^+ \right), \end{aligned} \quad (14)$$

where  $\delta = [\delta_g; \delta_\beta] \in \mathbb{R}^{2|\mathcal{G}|}$  represents the trial step taken from the current point  $(g, \beta)$ ,  $H \in \mathbb{R}^{2|\mathcal{G}| \times 2|\mathcal{G}|}$  is a symmetric matrix, and

$$\tilde{Q}_{\epsilon, C}(g, \beta; \delta) = \tilde{Q}_\epsilon(C^N(g, \beta); \tilde{C}^N(g, \beta; \delta) - C^N(g, \beta)),$$

$$\tilde{Q}_\epsilon(z; p) = Q_\epsilon(z) + \nabla Q_\epsilon(z)^T p,$$

$$[\tilde{C}^N]_j(g, \beta; \delta) = \max_j \{c_j(g, \beta; \omega_i) + \nabla c_j(g, \beta; \omega_i)^T \delta\}.$$

Each iteration  $k$  finds a descent step,  $\delta^k$ , for  $\phi_\pi$  by minimizing the model  $m(g^k, \beta^k, H^k; \delta)$  within a radius  $\Delta^k$  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 that measure the infeasibility of the

constraints. Hence, (14) is rewritten as a smooth constrained QP by introducing slack variables  $u$ ,  $v$ , and  $w$ :

$$\begin{aligned} \min_{\delta, z, u, v, w} & \text{Cost}(g^k) + \frac{1}{2} \mathbb{V}(\Omega)(\beta^k)^T M \beta^k + \nabla \text{Cost}(g^k)^T \delta_g \\ & + \mathbb{V}(\Omega)(\beta^k)^T M \delta_\beta + \frac{1}{2} \delta^T H^k \delta + \pi [(u + v)^T \mathbf{1}_2 + w] \end{aligned} \quad (15a)$$

$$\text{s.t. } \sum_{i \in \mathcal{G}} (g_i^k + \delta_{g_i}) - \sum_{i \in \mathcal{B}} d_i = u_1 - v_1 \quad (15b)$$

$$\sum_{i \in \mathcal{G}} (\beta_i^k + \delta_{\beta_i}) = 1 + u_2 - v_2 \quad (15c)$$

$$c(g^k, \beta^k; \omega_i) + \nabla c(g^k, \beta^k; \omega_i)^T \delta \leq z_i \mathbf{1}_m, \forall i \in [N] \quad (15d)$$

$$\begin{aligned} \nabla Q_\epsilon(C^N(g^k, \beta^k))^T (z - C^N(g^k, \beta^k)) \\ + Q_\epsilon(C^N(g^k, \beta^k)) \leq w, \end{aligned} \quad (15e)$$

$$t, u, w \geq 0, \quad \|\delta\|_\infty \leq \Delta^k, \quad (15f)$$

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 the linearization of (12b), (12c) and (12d), given by (15b), (15c) and (15e), respectively. The  $z$  variable in (15d) represents the maximum of the linearization of  $c$ , i.e.,  $z_i = [\bar{C}^N]_i(g, \beta; \delta)$ . Thus, (15) is indeed equivalent to minimizing  $m(g^k, \beta^k, H^k; \delta)$  with the addition of the trust-region constraint (15f).

A step  $\delta^k$  obtained from solving (15) is accepted if it results in sufficient decrease of  $\phi_\pi$ , i.e., we move in the  $\delta^k$  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  $\Delta^{k+1} \geq \Delta^k$ ; otherwise, the iterates are not accepted and we choose  $\Delta^{k+1} < \Delta^k$ .

For fast local convergence,  $H^k$  is chosen as

$$\begin{aligned} H^k &= H_{\mathbb{E}} \\ &+ \lambda^k \nabla \bar{C}^N(g^k, \beta^k) [\nabla^2 Q_\epsilon(C^N(g^k, \beta^k))] [\nabla \bar{C}^N(g^k, \beta^k)]^T, \end{aligned} \quad (16)$$

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

$$H_{\mathbb{E}} = \begin{pmatrix} M & 0 \\ 0 & \mathbb{V}(\Omega)M \end{pmatrix},$$

$\nabla \bar{C}^N(g^k, \beta^k)$  represents the transpose of the Jacobian of a smooth approximation of  $C^N(g^k, \beta^k)$  (see [25]) obtained from

$$\begin{aligned} [\nabla \bar{C}^N(g^k, \beta^k)]_{..i} &= \nabla c(g^k, \beta^k; \omega_i) \bar{\mu}_i^k, \forall i \in [N], \\ [\bar{\mu}_i^k]_j &= \frac{[\mu_i^k]_j}{\lambda^k [\nabla Q_\epsilon(C^N(g^k, \beta^k))]_i}, \forall j \in [m], \forall i \in [N], \end{aligned}$$

and  $\lambda^k$  and  $\mu_i^k$  are the multipliers corresponding to (15e) and (15d), respectively, from the previous iteration. If  $\lambda^k = 0$  or  $[\nabla Q_\epsilon(C^N(g^k, \beta^k))]_i = 0$ , 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  $\ell \neq j$ .

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

$$\begin{aligned} \nabla \bar{\mathcal{L}}(g^k, \beta^k, \nu^k, \lambda^k) &= \nabla \text{Cost}_{\mathbb{E}}(g^k, \beta^k) + \nu_1^k e_{\mathcal{G}} + \nu_2^k e_{\beta} \\ &+ \lambda^k \sum_{i=1}^N [\nabla Q_\epsilon(C^N(g^k, \beta^k))]_i \nabla c(g^k, \beta^k; \omega_i) \bar{\mu}_i^k, \end{aligned} \quad (17)$$

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

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

$\nu_1$  and  $\nu_2$  are the multipliers associated with (15b) and (15c), respectively;  $\lambda$  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 (12) (see (5.10) in [25]). Thus, if  $\|\nabla \bar{\mathcal{L}}(g^*, \beta^*, \nu^*, \lambda^*)\|$  is less than a small convergence tolerance, then the point  $(g^*, \beta^*, \nu^*, \lambda^*)$  is returned as a stationary point of (13). If constraints (12b)–(12d) are satisfied by  $(g^*, \beta^*)$ , we conclude that  $(g^*, \beta^*)$  is a stationary point of (12).

### C. Improving the computation time

Directly applying the algorithm described thus far has limited tractability since the number of constraints imposed in large DC OPF problems with many scenarios leads to computationally challenging instances. We next propose two extensions that improve the algorithm's computational scalability. The first extension is a Hessian approximation that enables application of faster convex QP solvers. The second extension is a lazy constraint generation technique.

1) *Convex Hessian approximation:* Since  $Q_\epsilon$  is a non-convex function, the associated Hessian matrix  $H^k$  in (16) is not necessarily positive semi-definite. Thus, (15) might not be convex. In general, non-convex QPs are more challenging to solve than convex QPs. Our experiments show that the times for finding a global minimizer with 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 next.

First, notice that if  $\nabla^2 Q_\epsilon(C^N(g^k, \beta^k))$  is positive semi-definite, then (16) is positive semi-definite. Thus, we replace  $\nabla^2 Q_\epsilon(C^N(g^k, \beta^k))$  by a positive semi-definite approximation  $\hat{Q}^k$  in which all negative eigenvalues are replaced by zero [34, 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  $\hat{Q}^k$  as

$$\hat{Q}^k = A (\Lambda + \text{diag}(\tau_i)) A^T,$$

where

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

and  $\lambda_i$  represents the  $i$ th eigenvalue of  $\nabla^2 Q_\epsilon(C^N(g^k, \beta^k))$ . When we replace  $\nabla^2 Q_\epsilon$  by  $\hat{Q}^k$ , the semi-definite approximation of (16) is

$$\hat{H}^k = H_{\mathbb{E}} + \lambda^k \nabla \bar{C}^N(g^k, \beta^k) \hat{Q}^k [\nabla \bar{C}^N(g^k, \beta^k)]^T. \quad (18)$$

This modification ensures that  $\hat{H}^k$  is positive semi-definite at every iteration. Substituting  $\hat{H}^k$  in (15) 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  $\phi_\pi$ . Hence, while the number of iterates that the algorithm performs might increase, the approximation

$H^k$  proposed in (18) 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 of a nonconvex model.

2) *Lazy constraint generation:* To further improve tractability, we utilize a lazy constraint generation technique. Motivated by the observation that only a small fraction of the inequality constraints in the QP (15) are active at the optimal solution [26], we first solve a version of (15) that only includes those inequalities that are either infeasible or within a certain threshold of becoming infeasible for the solution of the deterministic problem with  $\omega = 0$ . In an iterative fashion, we check which of the original inequalities are violated by the optimal solution of the reduced QP, add those to the QP, and solve the augmented problem until all of the original constraints are satisfied. For the next instance of (15), we start with the most recent set of inequalities. This procedure results in a reduction of up to 57% of the total computation time for the test cases we considered.

To summarize, Algorithm 1 describes our proposed approach for solving (12) for a given smoothing parameter  $\epsilon > 0$ . In Algorithm 1 we represent the standard trust-region parameters by  $\hat{\Delta} > 0$ , the maximum trust-region radius;  $\Delta_0 \in (0, \hat{\Delta})$ , the initial trust-region radius;  $\eta \in (0, 1)$ , the actual reduction ratio; and  $\tau_1 \in (0, 1)$  and  $\tau_2 > 1$ , the contraction and expansion coefficients of the trust-region radius. For the experiments in Section VI, we set the values of these parameters to standard values of trust-region algorithms (see, e.g., [34, 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.

## V. SELECTING THE SMOOTH-QUANTILE PARAMETERS

The smooth approximation of the quantile  $Q_\epsilon$  is motivated by a kernel estimation of the cdf resulting in (10) [35, p. 256]. The properties of the kernel approximation of the cdf can be extended to those of the quantile [36]. These properties imply that large values of the smoothing parameter,  $\epsilon$ , reduce the variance among the estimators obtained from different samples, but can lead to biased estimators that are either consistently infeasible or consistently conservative.

To illustrate this, consider the following example in  $\mathbb{R}^2$ ,

$$c(x_1, x_2; \omega_1, \omega_2) = x_1\omega_1 + x_2\omega_2 - 1, \quad (19)$$

where  $\omega_i \sim N(0, 1)$  are independent random variables. The true feasible region of constraint  $\mathbb{P}(c(x_1, x_2; \omega_1, \omega_2) \leq 0) \geq 0.95$  as well as its empirical and smooth approximations for a sample of size  $N = 100$  are shown in Fig. 2.

Fig. 2 illustrates how the feasible region changes for different  $\epsilon$ . We observe that large values of  $\epsilon$  introduce a bias in the feasible region of the smooth approximation by making it more conservative than the true feasible region. However, increasing the value of  $\epsilon$  also decreases the existence of spurious “non-convexities” in the feasible region. These “non-convexities” are not inherent to the problem but introduced by the discrete nature of the sample approximation, and they may cause local optimization algorithms (such as ours) to converge to local optima with worse objective values. To avoid local optima and to increase the consistency of the solutions over different

---

**Algorithm 1**  $S\ell_1$ QP 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  $k \leftarrow 0$

- 1: Let  $\mathcal{J} = \{(j, i) \mid c_j(g^0, \beta^0; \omega_i) > -\kappa_2\}$ .
- 2: **while**  $\|\nabla \tilde{\mathcal{L}}(g^k, \beta^k, \nu^k, \lambda^k)\|_\infty > \kappa_1$  **or**  $\|V(g^k, \beta^k)\|_\infty > \kappa_1$  **do**
- 3:   Set  $C^N(g^k, \beta^k; \omega_i) = \max_{j=1, \dots, m} \{c_j(g^k, \beta^k; \omega_i)\}$  for all scenarios, compute  $Q_\epsilon(C^N(g^k, \beta^k))$ ,  $\nabla Q_\epsilon(C^N(g^k, \beta^k))$ , and  $\hat{H}^k$ .
- 4:   Obtain  $(\delta, \nu, \mu, \lambda)$  by solving (15) with the constraints of the type (15d) given in  $\mathcal{J}$  (**if**  $\delta^k = 0$  **stop**, stationary point of  $\phi_\pi$  reached).
- 5:   **while** There exists  $(j, i)$  such that  $c_j(g^k + \delta_g, \beta^k + \delta_\beta; \omega_i) > -\kappa_2$  **do**
- 6:      $\mathcal{J} = \mathcal{J} \cup \{(j, i) \mid c_j(g^k + \delta_g, \beta^k + \delta_\beta; \omega_i) > -\kappa_2\}$
- 7:     Resolve (15) with the constraints of the type (15d) given in  $\mathcal{J}$ .
- 8:   **end while**
- 9:   Compute the ratio  $\rho^k = \frac{\phi_\pi(g^k, \beta^k) - \phi_\pi(g^k + \delta_g, \beta^k + \delta_\beta)}{m(g^k, \beta^k, \hat{H}^k; 0) - m(g^k, \beta^k, \hat{H}^k; \delta^k)}$
- 10:   **if**  $\rho^k < \eta$  **then**
- 11:      $\Delta^{k+1} = \tau_1 \min\{\Delta^k, \|d^k\|_\infty\}$
- 12:      $g^{k+1} = g^k; \beta^{k+1} = \beta^k$
- 13:      $\nu^{k+1} = \nu^k; \lambda^{k+1} = \lambda^k; \bar{\mu}^{k+1} = \bar{\mu}^k$
- 14:   **else**
- 15:      $g^{k+1} = g^k + \delta_g; \beta^{k+1} = \beta^k + \delta_\beta$
- 16:     Set  $\bar{\mu}_i = \frac{\mu_i}{\lambda[\nabla Q_\epsilon(C^N(g^k, \beta^k))]_i}$ , for all  $i = [N]$ .
- 17:      $\nu^{k+1} = \nu^k; \lambda^{k+1} = \lambda^k; \bar{\mu}^{k+1} = \bar{\mu}^k$
- 18:     **if**  $\rho^k \geq \eta$  **and**  $\|d^k\|_\infty = \Delta^k$  **then**
- 19:        $\Delta^{k+1} = \min\{\tau_2 \Delta^k, \hat{\Delta}\}$
- 20:     **else**
- 21:        $\Delta^{k+1} = \Delta^k$
- 22:     **end if**
- 23:   **end if**
- 24:    $k = k + 1$
- 25: **end while**

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

---

samples, using larger values of  $\epsilon$  is advantageous. However, it is important to consider how the bias introduced by a large choice of  $\epsilon$  can be counteracted.

To counteract the bias introduced by our choice of  $\epsilon$ , we propose to relax or strengthen the quantile constraint (12d) by adjusting the right-hand side by  $t \in \mathbb{R}$  as follows:

$$Q_\epsilon(C^N(g, \beta)) \leq t. \quad (20)$$

If  $t < 0$ , (20) is more restrictive than (12d); if  $t > 0$ , (12d) is relaxed. Hence, the feasible region of the approximated problem (12) gets smaller as the right-hand side  $t$  decreases (see Fig. 3), which implies that the optimal objective value will increase as  $t$  decreases. Given this monotone behavior of the approximated problem with respect to the right-hand side of (20), we propose a binary search method in order to find a value of  $t$  such that the solution of (12) attains an out-of-sample probability of  $1 - \alpha$  for a given  $\epsilon$  and  $N$ .

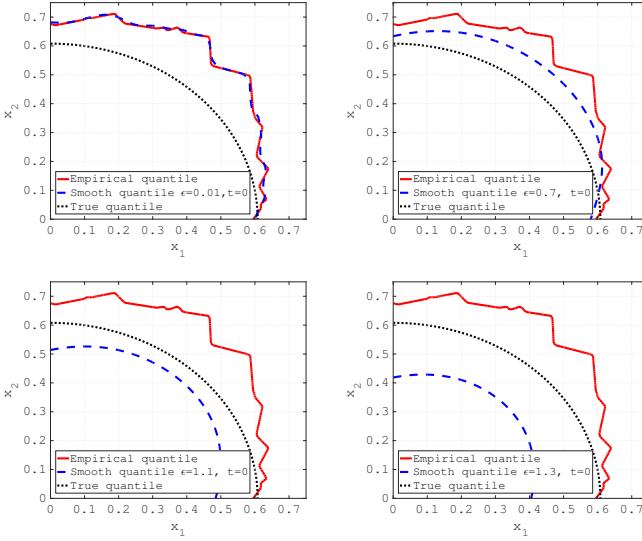


Fig. 2: Comparison of different values of the smoothing parameter  $\epsilon$ .

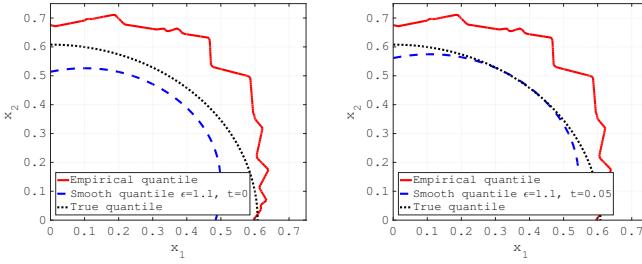


Fig. 3: Comparison of different values of  $t$ .

In Fig. 3, it can be seen that for  $N = 100$  and  $\epsilon = 1.1$ , the smooth quantile (blue dashed line) is slightly more conservative than the true quantile (black pointed line). However, if we keep  $\epsilon = 1.1$  and adjust the right-hand side to  $t = 0.05$ , the smooth approximation is still conservative, in the sense that all feasible solutions for the approximated problem are feasible for the true problem, but now the bias is reduced.

Our previous work in [25] did not consider adjustments to  $t$ . Here, we propose a new strategy that exploits the flexibility provided by the parameter  $t$  in combination with the parameter  $\epsilon$  in order to better tune the performance of our algorithm. We determine appropriate values for  $\epsilon$  and  $t$  in two stages. First, for a given problem class (i.e., a certain network), we empirically determine a suitable value of  $\epsilon$  for a particular sample size. This value is then adjusted to different sample sizes using a statistical result obtained from the theory of kernel estimators [36] (see Section V-A). After this, for each particular instance, we tune the value of  $t$  to obtain the best feasible point (see Section V-B).

#### A. Procedure for choosing the value of $\epsilon$

For a given sample size  $\hat{N}$ , we want to select a value  $\hat{\epsilon}$  that eliminates spurious “non-convexities” in the feasible region, while making sure that the bias introduced by this choice of

the smoothing parameter is not too large (see Fig. 2). We propose Algorithm 2 in order to select an appropriate  $\hat{\epsilon}$  value.

---

#### Algorithm 2 Binary search for the smoothing parameter $\hat{\epsilon}$

---

**Inputs:**  $\hat{N} \in \mathbb{N}; \hat{\epsilon}_0 > 0; \tau_\epsilon > 0; \hat{M} \in \mathbb{N}$   
**for**  $m = 1, \dots, \hat{M}$  **do**  
    Set  $\epsilon_0 = \hat{\epsilon}_0$ ,  $\epsilon_{LB} = 0$ ,  $\epsilon_{UB} = \infty$ , and  $\ell = 0$ .  
    **repeat**  
        Obtain  $(g^*, \beta^*)$  by solving (12) with  $\hat{N}$  samples,  $t = 0$ , and  $\epsilon_\ell$ .  
        Approximate  $p_\ell \approx \mathbb{P}(C(g^*, \beta^*; \omega) \leq 0)$ .  
        **if**  $p_\ell > 1 - \alpha$  **then**  
             $\epsilon_{UB} = \epsilon_\ell$   
             $\epsilon_{\ell+1} = (\epsilon_{LB} + \epsilon_\ell)/2$   
        **else**  
             $\epsilon_{LB} = \epsilon_\ell$   
            **if**  $\epsilon_{UB} = \infty$  **then**  $\epsilon_{\ell+1} = 2\epsilon_\ell$   
            **else**  $\epsilon_{\ell+1} = (\epsilon_{UB} + \epsilon_\ell)/2$  **end if**  
        **end if**  
        **until**  $|p_\ell - (1 - \alpha)| \leq \tau_\epsilon$  **or**  $\epsilon_{UB} - \epsilon_{LB} \leq \tau_\epsilon$   
         $\bar{\epsilon}_m = \epsilon_\ell$   
    **end for**  
     $\hat{\epsilon} = \max_{m=1, \dots, \hat{M}} \{\bar{\epsilon}_m\}$   
**Return:**  $\hat{\epsilon}$ .

---

The goal of Algorithm 2 is to choose the most conservative approximation of the smoothing parameter  $\epsilon$  from a set of  $\hat{M}$  samples. This value is obtained by running  $\hat{M}$  replications of the binary search algorithm described in [25] for a sample of size  $\hat{N}$  each. Then, since we have observed empirically that larger values of  $\epsilon$  result in a more conservative approximation, we choose the maximum value of  $\epsilon$  observed from the replications above in order to obtain the most conservative approximation of this parameter. In this paper, we choose  $\hat{N} = 100$  scenarios per sample and  $\hat{M} = 10$  samples.

To avoid repeating Algorithm 2 for different sample sizes  $N$ , we use a result from [36] that approximates  $\epsilon$  for different sample sizes. Reference [36] proves that, asymptotically, the optimal choice of  $\epsilon$  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  $\hat{N}$ . Thus, for a given sample size  $N$ , we consider  $\epsilon = \frac{(\hat{N}^{1/3})\hat{\epsilon}}{N^{1/3}}$ .

#### B. Binary search to determine $t$

After choosing the smoothing parameter  $\epsilon$  based on  $N$ , as described in Section V-A, we tune the parameter  $t$  using the binary search described in Algorithm 3 for each separate sample. The parameter  $\tau_t > 0$  in this algorithm determines the maximum difference allowed between the probability attained by the solution  $(g^*, \beta^*)$  and the target probability  $1 - \alpha$ .

When implementing Algorithm 3, we use the optimal solution and multipliers obtained when solving for the right-hand side  $t^\ell$  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  $\ell$ th 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 of  $t$ .

**Algorithm 3** 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  $(g^*, \beta^*)$  by solving (12) with  $t_\ell$ .
- Approximate  $p_\ell \approx \mathbb{P}(C(g^*, \beta^*; \omega) \leq 0)$ .
- if**  $p_\ell > 1 - \alpha$  **then**

  - $t_{UB} = t_\ell$
  - if**  $t_{LB} = -\infty$  **then**  $t_{\ell+1} = t_\ell - \kappa_t$  **end if**
  - if**  $t_{LB} > -\infty$  **then**  $t_{\ell+1} = (t_{LB} + t_\ell)/2$  **end if**

- else**

  - $t_{LB} = t_\ell$
  - if**  $t_{UB} = \infty$  **then**  $t_{\ell+1} = t_\ell + \kappa_t$  **end if**
  - if**  $t_{UB} < \infty$  **then**  $t_{\ell+1} = (t_{UB} + t_\ell)/2$  **end if**

- end if**
- until**  $|p_\ell - (1 - \alpha)| \leq \tau_t$  **or**  $t_{UB} - t_{LB} \leq \tau_t$

**Return:**  $t_\ell, (g^*, \beta^*)$ , and  $p_\ell$ .

## 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` [37]. We compare our method against the scenario-approach [19], [21], [38] 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 (15). We set the CPLEX parameter `barrier.colnonzeros` to 1. We use the parameters  $\pi = 10$ ,  $\hat{\Delta} = 10^6$ ,  $\Delta_0 = 1$ ,  $\eta = 10^{-8}$ ,  $\tau_1 = 1/2$ ,  $\tau_2 = 2$ ,  $\kappa_1 = 10^{-6}$ ,  $\kappa_2 = 0.1$ ,  $\tau_t = 10^{-4}$  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 (5) 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  $t^\ell$ , we initiate  $g^0$ ,  $\beta^0$ ,  $\lambda^0$ , 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 loads, i.e.,  $\omega \sim N(\vec{0}, \Sigma)$ , where  $\Sigma$  represents the covariance matrix. To create the covariance matrix, we generated a  $|\mathcal{B}| \times |\mathcal{B}|$  matrix,  $A$ , with entries taken from a uniform random variable with support in  $[-1, 1]$ . Then, we obtained a positive definite matrix via  $\hat{A} = AA^T$ . Finally, we scaled each entry of  $\hat{A}$  to ensure that  $\Sigma_{ii} = \zeta d_i$  by defining

$$\Sigma_{ij} = \zeta \frac{\hat{A}_{ij}}{\sqrt{\hat{A}_{ii}\hat{A}_{jj}}} \left( \sqrt{d_i d_j} \right).$$

Here,  $\zeta$  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  $\zeta$ . 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 the algorithm’s performance 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  $t$ .

	N = 100	N = 200	N = 500	N = 1000
Case 14				
Min. obj (\$)	2,106.2	2,120.1	2,104.6	2,105.4
Avg. obj (\$)	2,127.8	2,138.4	2,117.0	2,115.5
Max. obj (\$)	2,191.5	2,238.2	2,128.9	2,128.5
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)	1.6812	4.6733	6.6583	28.4767
Avg. time (s)	3.2486	5.4729	15.2696	42.0556
Max. time (s)	6.5607	6.0644	35.0050	73.0396
Avg. $t (\times 10^{-3})$	17.219	8.3281	3.1875	-1.7344
Case 57	N = 100	N = 200	N = 500	N = 1000
Min. obj (\$)	35,342	35,318	35,327	35,307
Avg. obj (\$)	35,413	35,394	35,358	35,337
Max. obj (\$)	35,508	35,471	35,381	35,357
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)	2.5178	8.4458	11.586	64.078
Avg. time (s)	16.657	28.405	44.837	121.47
Max. time (s)	23.800	52.813	100.045	199.53
Avg. $t (\times 10^{-2})$	4.0375	2.7172	1.5625	0.9938
Case 118 ( $\zeta = 0.01$ )	N = 100	N = 200	N = 500	N = 1000
Min. obj (\$)	112,346	112,231	112,131	111,979
Avg. obj (\$)	112,594	112,469	112,231	112,038
Max. obj (\$)	112,891	112,626	112,453	112,160
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)	13.143	45.256	268.08	658.34
Avg. time (s)	46.460	84.236	470.45	1127.1
Max. time (s)	86.017	181.70	879.82	1801.1
Avg. $t (\times 10^{-3})$	7.5234	3.9297	2.5156	4.7109
Case 118 ( $\zeta = 0.05$ )	N = 100*	N = 200	N = 500	N = 1000
Min. obj (\$)	116,257	116,178	116,092	116,074
Avg. obj (\$)	116,615	116,315	116,138	116,107
Max. obj (\$)	117,183	116,670	116,168	116,165
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)	26.842	104.34	581.31	1618.3
Avg. time (s)	50.336	142.75	867.83	2172.8
Max. time (s)	60.821	179.73	1925.9	3585.8
Avg. $t (\times 10^{-2})$	6.4642	5.1920	3.2102	2.0457

TABLE I: Results from Algorithm 1 using  $\hat{\epsilon}_{14} = 6.7(10)^{-2}$ ;  $\hat{\epsilon}_{57} = 1.9(10)^{-1}$ ;  $\hat{\epsilon}_{118} = 7.(10)^{-2}$  ( $\zeta = 0.01$ );  $\hat{\epsilon}_{118} = 1.9(10)^{-1}$  ( $\zeta = 0.05$ ). \*: Statistics of feasible instances.

Notice that the variability in the objective value decreases with increasing sample size. For example, in case 118 with  $\zeta = 0.05$ , the difference between the maximum and minimum costs decreases from 926 for sample size  $N = 100$  to 76 for  $N = 1,000$  scenarios, a 90% decrease on the variability with

respect to the sample size. However, even for  $N = 100$ , the variability between samples is no greater than 5.6% for case 14 and 0.8% for the rest of the cases. This indicates that the NLP approach performs favourably even when using a small number of scenarios. Furthermore, the solutions returned for cases 14, 57 and 118 (with  $\zeta = 0.01$ ) are always feasible for the true problem and accurately achieve the prescribed risk level of 95%. For case 118 with  $\zeta = 0.05$ , all solutions are feasible for a sample size of at least 200. For  $N = 100$ , the solution obtained by the NLP approach on one instance is not feasible for the out-of-sample approximation of (5d). The achieved risk level is 92.1% for this instance. We believe that this happens because the number of scenarios is insufficient for the level of variability.

For larger sample sizes,  $N = 500$  and  $N = 1,000$ , the computation times can be improved by setting the CPLEX parameter `barrier.colnonzeros` = 0. This has shown a decrease of up to 55% in the computation times. However, Tables I and II only show the times obtained using `barrier.colnonzeros` = 1 for consistency in the paper.

### C. Comparison of the NLP and scenario approaches

This section compares our solutions to those obtained from the scenario approach (SA) [19], which approximates (5) as

$$\min_{g, \alpha} c(g) \quad (21a)$$

$$\text{s. t. } f_{ij}^{\text{LB}} \leq \Phi \hat{p}(\omega_s) \leq f_{ij}^{\text{UB}}, \quad \forall ij \in \mathcal{L}, \forall s \in [N^{\text{SA}}] \quad (21b)$$

$$g_i^{\text{LB}} \leq g_i - \beta_i \Omega_s \leq g_i^{\text{UB}}, \quad \forall i \in \mathcal{G}, \forall s \in [N^{\text{SA}}] \quad (21c)$$

$$\text{Eqns. (5b), (5c).} \quad (21d)$$

where  $N^{\text{SA}}$  is a pre-specified number of scenarios. SA specifies a minimum number of scenarios  $N^{\text{SA}}$  such that a solution to (21), which is feasible for all  $N^{\text{SA}}$  scenarios, is also feasible for the probabilistic constraint (5d) with a probability of at least  $1 - \sigma$ . An attractive feature of SA is that it results in a large-scale convex optimization problem, yet it tends to produce conservative results [39].

We use the sample size given in [38] to select  $N^{\text{SA}}$ ,

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

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: 7, (2) Case 57: 1, (3) Case 118 ( $\zeta = 0.01$ ): 0, and (4) Case 118 ( $\zeta = 0.05$ ): 3. We note that, since SA is a conservative approach, an infeasible SA instance does not indicate that the JCC problem is infeasible.

First, observe that while the nominal solution is the least expensive, the solutions obtained when ignoring uncertainty are far from being feasible for cases 57 and 118; for these two cases, the nominal solution is feasible at most 38% of the time. In addition, there are many instances for which the SA

algorithm cannot obtain a feasible solution to the approximated problem (21), even though the solutions obtained for the other samples clearly indicate that the problem is feasible. Of the 40 instances shown in the table, the SA approach was only able to return a solution for 29 of them. On the other hand, the NLP approach can always find a solution to the approximated problem (12), and out-of-sample testing verifies feasibility of the solutions to almost all of the problems we considered. The sole exception is one instance of case 118, with  $\zeta = 0.05$  and  $N = 100$ , that is infeasible with respect to the true problem; the out-of-sample risk level attained for this sample is 92.1%.

Table II shows that the best solution from the SA can be up to 10% more expensive than the worst NLP solution. For example, in case 14 the worst solution obtained by the NLP is 2,238.2, while the best solution for the SA is 2,461.5. This demonstrates that the NLP approach provides solutions that are feasible without being overly conservative.

As several specific comparisons, the results for case 118 with  $\zeta = 0.01$  and  $N = 100$  show that the slowest computation time of the NLP method is faster than the fastest time of the SA, and that the average objective value from the NLP method is better. The results for case 118 with  $\zeta = 0.05$  and  $N = 100$  show that the best objective value from the SA is more expensive than the worst objective value from the NLP, and that the average computation time of the NLP is faster than that of the SA. Hence, there are instances for which the NLP method returns better solutions in less time than the SA.

Finally, as the number of generators increases, the SA method prescribes a larger number of scenarios,  $N^{\text{SA}}$ , to be satisfied in order to guarantee feasibility. This significantly impacts the size of the SA problem and, as a consequence, the solution time of the SA method may become worse than the NLP (see case 118,  $\zeta = 0.01$ ). For that reason, the advantages of the NLP method relative to the SA method are expected to be particularly pronounced for systems with many generators.

## VII. CONCLUSIONS AND OUTLOOK

This paper has developed a sample-based NLP algorithm for solving DC-OPF problems with JCC. 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 JCC as well as contingency constraints to model the possibility of component failures.

## ACKNOWLEDGMENTS

The work by Alejandra Peña-Ordieres was supported by the U.S. Department of Energy, Office of Electricity Delivery and Energy Reliability under contract DE-AC-02-06CH11357 and the National Science Foundation grant DMS-1522747.

The work by Daniel Molzahn was supported by the U.S. Department of Energy, Office of Electricity Delivery and Energy Reliability under contract DE-AC-02-06CH11357.

The work by Line Roald was supported by the Department of Energy, Office of Science, Office of Advanced Scientific

Case 14	Nominal	SA	NLP(100)	NLP(200)
Min. obj (\$)	-	2,461.5	2,106.2	2,120.1
Avg. obj (\$)	2,051.5	2,489.0	2,127.8	2,138.4
Max. obj (\$)	-	Inf	2,191.5	2,238.2
Min. prob	-	0	0.950	0.950
Avg. prob	0.939	0.993	0.950	0.950
Max. prob	-	0.994	0.950	0.950
Min. time (s)	-	0.0325	1.6812	4.6733
Avg. time (s)	0.0036	0.0425	3.2486	5.4729
Max. time (s)	-	0.0492	6.5607	6.0644
$N^{\text{SA}}/\text{Avg. } t$	1	516	0.0172	0.0083
Case 57	Nominal	SA	NLP(100)	NLP(200)
Min. obj (\$)	-	35,493	35,342	35,318
Avg. obj (\$)	34,773	35,625	35,413	35,394
Max. obj (\$)	-	Inf	35,508	35,471
Min. prob	-	0	0.950	0.950
Avg. prob	0.382	0.995	0.950	0.950
Max. prob	-	0.999	0.950	0.950
Min. time (s)	-	0.2228	2.5178	8.4458
Avg. time (s)	0.0041	0.2776	16.657	28.405
Max. time (s)	-	0.3761	23.800	52.813
$N^{\text{SA}}/\text{Avg. } t$	1	637	0.0404	0.0272
Case 118 ( $\zeta = 0.01$ )	Nominal	SA	NLP(100)	NLP(200)
Min. obj (\$)	-	112,496	112,346	112,231
Avg. obj (\$)	109,791	112,824	112,594	112,469
Max. obj (\$)	-	113,118	112,891	112,626
Min. prob	-	0.996	0.950	0.950
Avg. prob	0.114	0.997	0.950	0.950
Max. prob	-	0.999	0.950	0.950
Min. time (s)	-	86.022	13.143	45.256
Avg. time (s)	0.0071	88.546	46.460	84.236
Max. time (s)	-	94.845	86.017	181.70
$N^{\text{SA}}/\text{Avg. } t$	1	2998	0.0075	0.0039
Case 118 ( $\zeta = 0.05$ )	Nominal	SA	NLP(100)*	NLP(200)
Min. obj (\$)	-	118,304	116,257	116,178
Avg. obj (\$)	109,791	122,553	116,615	116,315
Max. obj (\$)	-	Inf	117,183	116,670
Min. prob	-	0	0.950	0.950
Avg. prob	0.057	0.996	0.950	0.950
Max. prob	-	0.997	0.950	0.950
Min. time (s)	-	21.222	26.842	104.34
Avg. time (s)	0.0071	77.609	50.336	142.75
Max. time (s)	-	94.166	60.821	179.73
$N^{\text{SA}}/\text{Avg. } t$	1	2998	0.0646	0.0519

TABLE II: **Nominal:** Solution of (5) 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.

Computing Research, Applied Mathematics program under contract DE-AC-02-06CH11347.

The work by Andreas Wächter was supported by National Science Foundation grant DMS-1522747 and Los Alamos National Laboratory, its Center for Nonlinear Studies as the Ulam Scholarship program. Los Alamos National Laboratory is operated by Triad National Security, LLC, for the National Nuclear Security Administration of U.S. Department of Energy (Contract No. 89233218CNA000001) under the auspices of the NNSA of the U.S. DOE at LANL under contract DE-AC-52-06NA25396.

## 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 *18th Power Syst. Comput. Conf. (PSCC)*, Aug. 2014.
- [2] F. Bouffard and F. D. Galiana, “Stochastic security for operations planning with significant wind power generation,” in *IEEE Power and Energy Society General Meeting (PESGM)*, 2008.
- [3] J. M. Morales, A. J. Conejo, and J. Pérez-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, “Multiarea stochastic unit commitment for high wind penetration in a transmission constrained network,” *Oper. Res.*, 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 Bulk Power System Dynamics and Control-VIII (IREP)*, Aug. 2010.
- [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. Goulart, S. Mariéthoz, and M. Morari, “Policy-based reserves for power systems,” *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 4427–4437, Nov. 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, Nov. 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*, June 2013.
- [12] D. Bienstock, M. Chertkov, and S. Harnett, “Chance-constrained optimal power flow: Risk-aware network control under uncertainty,” *SIAM Rev.*, vol. 56, no. 3, p. 461495, 2014.
- [13] T. Summers, J. Warrington, M. Morari, and J. Lygeros, “Stochastic optimal power flow based on convex approximations of chance constraints,” in *18th 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*, vol. 10, no. 6, pp. 6376–6385, Nov. 2019.
- [15] X. Geng and L. Xie, “Data-driven decision making with probabilistic guarantees (part 1): A schematic overview of chance-constrained optimization,” *arXiv:1903.10621*, Mar. 2019.
- [16] R. Henrion, “Structural properties of linear probabilistic constraints,” *Optimiz.*, 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,” *Oper. Res.*, 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, Mar. 2017.
- [19] G. Calafiore and M. C. Campi, “Uncertain convex programs: Randomized solutions and confidence levels,” *Math. Prog.*, 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. Automat. 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 J. Optimiz.*, 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. C. Campi, S. Garatti, A. Carè, A. A. Thatte, and P. Kumar, “Scenario-based economic dispatch with tunable risk levels in high-renewable power systems,” *IEEE Trans. Power Syst.*, vol. 34, no. 6, pp. 5103–5114, Nov. 2019.
- [25] A. Peña-Ordieres, J. R. Luedtke, and A. Wächter, “Solving chance-constrained problems via a smooth sample-based nonlinear approximation,” *arXiv:1905.07377*, 2019.
- [26] L. A. Roald and D. K. Molzahn, “Implied Constraint Satisfaction in Power System Optimization: The Impacts of Load Variations,” in *57th Allerton Conf. Commun., Control, Comput.*, Sept. 2019.
- [27] B. Stott, J. Jardim, and O. Alsaq, “DC Power Flow Revisited,” *IEEE Trans. Power Syst.*, vol. 24, no. 3, pp. 1290–1300, Aug. 2009.
- [28] A. J. Wood, B. F. Wollenberg, and G. B. Sheblé, *Power Generation, Operation, and Control*. John Wiley & Sons, 2013.

- [29] A. Geletu, A. Hoffmann, M. Kloppel, and P. Li, "An inner-outer approximation approach to chance constrained optimization," *SIAM J. Optimiz.*, vol. 27, no. 3, pp. 1834–1857, 2017.
- [30] A. Shapiro, D. Dentcheva, and A. Ruszczyński, *Lectures on Stochastic Programming: Modeling and Theory*. SIAM, 2014.
- [31] 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.
- [32] J.-B. Hiriart-Urruty and C. Lemarechal, *Convex Analysis and Minimization Algorithms I: Fundamentals*. Springer, 1996.
- [33] F. E. Curtis and M. L. Overton, "A sequential quadratic programming algorithm for nonconvex, nonsmooth constrained optimization," *SIAM J. Optimiz.*, vol. 22, no. 2, p. 474500, 2012.
- [34] J. Nocedal and S. Wright, *Numerical Optimization*. Springer, 2006.
- [35] R. Alemany, C. Bolancé, and M. Guillén, "A nonparametric approach to calculating value-at-risk," *Insurance: Mathematics and Economics*, vol. 52, no. 2, pp. 255–262, 2013.
- [36] 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.
- [37] 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.
- [38] M. C. Campi, S. Garatti, and M. Prandini, "The scenario approach for systems and control design," *Ann. Rev. Control*, vol. 33, no. 2, pp. 149–157, 2009.
- [39] X. Zhang, S. Grammatico, G. Schildbach, P. Goulart, and J. Lygeros, "On the sample size of random convex programs with structured dependence on the uncertainty," *Automatica*, vol. 60, pp. 182–188, 2015.



**Alejandra Peña-Ordieres** is a senior software engineer at The MathWorks. She received the M.S. and Ph.D. degrees in Industrial Engineering and Management Sciences from Northwestern University in 2016 and 2020, and the B.S. degree in Applied Mathematics from the Instituto Tecnológico Autónomo de México (ITAM) in 2013. Her research interests include stochastic programming and large-scale nonlinear optimization algorithms with a focus on optimal power flow problems.



**Daniel K. Molzahn** (S'09-M'13-SM'19) is an Assistant Professor in the School of Electrical and Computer Engineering at the Georgia Institute of Technology. He also holds an appointment as a computational engineer in the Energy Systems Division at Argonne National Laboratory and is a fellow of the Strategic Energy Institute at Georgia Tech. He was a Dow Postdoctoral Fellow in Sustainability at the University of Michigan, Ann Arbor. He received the B.S., M.S., and Ph.D. degrees in electrical engineering and the Masters of Public Affairs degree from the University of Wisconsin-Madison, where he was a National Science Foundation Graduate Research Fellow. His research focuses on optimization and control of electric power systems.



from the University of Wisconsin-Madison, where he was a National Science Foundation Graduate Research Fellow. His research focuses on optimization and control of electric power systems.

**Line Roald** is an Assistant Professor and Grainger Institute Fellow in the Department of Electrical and Computer Engineering in University of Wisconsin-Madison. She received her Ph.D. degree in Electrical Engineering (2016) and M.Sc. and B.Sc. degrees in Mechanical Engineering from ETH Zurich, Zurich, Switzerland. She was a postdoctoral research fellow with the Center of Non-Linear Studies at Los Alamos National Laboratory. Her research interests focus on modeling and optimization of energy systems, with a particular focus on managing uncertainty and risk from renewable energy variability and component failures.



**Andreas Wächter** is a Professor at the Industrial Engineering and Management Sciences Department at Northwestern University. He obtained his master's degree in Mathematics at the University of Cologne, Germany, in 1997, and this Ph.D. in Chemical Engineering at Carnegie Mellon University in 2002. Before joining Northwestern University in 2011, he was a Research Staff Member in the Department of Mathematical Sciences at IBM Research in Yorktown Heights, NY. His research interests include the design, analysis, implementation and application of numerical algorithms for nonlinear continuous and mixed-integer optimization. He is a recipient of the 2011 Wilkinson Prize for Numerical Software and the 2009 Informs Computing Society Prize for his work on the open-source optimization package Ipopt. He is currently spending a year at Los Alamos National Laboratory as the Ulam Fellow.