Prescribed Robustness in Optimal Power Flow

Robert Mieth, H. Vincent Poor

Department of Electrical and Computer Engineering, Princeton University, Princeton, USA.

Abstract—**For a timely decarbonization of our economy, power systems need to accommodate increasing numbers of clean but stochastic resources. This requires new operational methods that internalize this stochasticity to ensure safety and efficiency. This paper proposes a novel approach to compute adaptive safety intervals for each stochastic resource that internalize power flow physics and optimize the expected cost of system operations, making them "prescriptive". The resulting intervals are interpretable and can be used in a tractable robust optimal power flow problem as uncertainty sets. We use stochastic gradient descent with differentiable optimization layers to compute a mapping that obtains these intervals from a given vector of context parameters that captures the expected system state. We demonstrate and discuss the proposed approach on two case studies.**

I. INTRODUCTION

The ongoing deployment of clean but stochastic energy resources challenges established power system operations by threatening their ability to ensure system security and economic efficiency. To tackle this, numerous effective stochastic optimizations methods have been proposed, covering, for example, system scheduling and control [1]–[3] and electricity markets and the unit commitment process [4]–[6]. However, the adoption of [1]–[6] and similar proposals is obstructed by the necessity to significantly alter established operations, an expensive and risky process for system operators. Moreover, stochastic approaches to power system operations typically optimize reserve allocations and control policies implicitly based on probabilistic models and risk parameters defined by the system operator. This complicates a transparent communication towards power system stakeholders (e.g., owners of generation assets, electricity market participants who are interested in anticipating price formation and scheduling procedures). To overcome these barriers and further facilitate the adoption of clean energy technology, actionable methods for power system operation and planning that smartly internalize resource stochasticity while remaining interpretable and largely compatible with established processes are required.

Motivated by this requirement, this paper proposes a data-driven approach inspired by [7] to compute uncertainty sets for stochastic resources that (i) robustify generation and reserve allocation, (ii) internalize system physics, and (iii) minimize risk-adjusted cost while avoiding the need for a stochastic decision-making process. Extending [7], we introduce an additional prescription step to dynamically adjust the uncertainty

sets for a given problem context. We show that this leads to significant performance increases.

A. General problem formulation

Consider power system operations as a two-stage optimization problem with uncertain parameters $\boldsymbol{\xi}$, e.g., real-time nodal demand or renewable energy injection. Given a current parametrization of the system $\boldsymbol{\zeta}$ (e.g., net-demand forecast, generator availability, grid topology information), the first stage computes a resource allocation $\boldsymbol{x}$ (e.g., generation schedules, reserve allocations). The second stage then decides on recourse actions $\boldsymbol{y}$ depending on the first-stage allocation $\boldsymbol{x}^*$ and the realization of $\boldsymbol{\xi}$. In current industry practice, the two stages are solved deterministically and independently using fixed security parameters $\boldsymbol{\theta}$ (e.g., reserve requirements) in the first stage:

$$\text{1st stage:} \quad \mathcal{P}(\boldsymbol{\zeta}, \boldsymbol{\theta}) = \min_{\boldsymbol{x} \in \mathcal{X}(\boldsymbol{\zeta}, \boldsymbol{\theta})} \mathcal{C}^{\text{F}}(\boldsymbol{x}, \boldsymbol{\zeta}, \boldsymbol{\theta}) \quad (1)$$

$$\text{2nd stage:} \quad \mathcal{Q}(\boldsymbol{x}^*, \boldsymbol{\zeta}, \boldsymbol{\xi}) = \min_{\boldsymbol{y} \in \mathcal{Y}(\boldsymbol{x}^*, \boldsymbol{\zeta}, \boldsymbol{\xi})} \mathcal{C}^{\text{S}}(\boldsymbol{x}^*, \boldsymbol{\zeta}, \boldsymbol{\xi}) \quad (2)$$

While this process is computationally efficient and transparent, it ignores any probabilistic information that my be available from models or historic observations of $\boldsymbol{\xi}$.

As an alternative to replacing this deterministic two-step procedure with probabilistic optimization (see discussion above), this paper proposes an adaptive computation of parameters $\boldsymbol{\theta}$ that internalizes information on the stochasticity of $\boldsymbol{\xi}$ and is aware of $\boldsymbol{\zeta}$. We highlight that vector $\boldsymbol{\zeta}$ can be considered richer than just containing problem parameters, but may also contain additional covariates of $\boldsymbol{\xi}$ (e.g., weather information, forecasts). We therefore call $\boldsymbol{\zeta}$ *context* parameter. Given $\boldsymbol{\zeta}$, the optimal choice of $\boldsymbol{\theta}$ in the first stage is

$$\boldsymbol{\theta}^*(\boldsymbol{\zeta}) \in \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{\xi}|\boldsymbol{\zeta})}\big[\mathcal{P}(\boldsymbol{\zeta}, \boldsymbol{\theta}) + \mathcal{Q}(\boldsymbol{x}^*, \boldsymbol{\zeta}, \boldsymbol{\xi})\big], \quad (3)$$

Following [8], we call $\boldsymbol{\theta}^*$ *prescriptive* in the context of $\boldsymbol{\zeta}$, as it provides a parametrization that minimizes the conditional expectation over $\boldsymbol{\xi}$. To avoid re-solving (3) every time (which might be hard or impossible), the objective of this paper is to obtain a mapping $\mathcal{M}$ that computes $\boldsymbol{\theta}^*$ from $\boldsymbol{\zeta}$, i.e.:

$$\min_{\mathcal{M}} \mathbb{E}_{(\boldsymbol{\zeta}, \boldsymbol{\xi})}\big[\mathcal{P}(\boldsymbol{\zeta}, \mathcal{M}(\boldsymbol{\zeta})) + \mathcal{Q}(\boldsymbol{x}^*, \boldsymbol{\zeta}, \boldsymbol{\xi})\big]. \quad (4)$$

Solving problem (4) is hard in general and its tractability depends on the definition of the operational problem $\mathcal{P}$, $\mathcal{Q}$ and the mapping $\mathcal{M}$. This paper studies the following approach:

- For $\mathcal{P}$, $\mathcal{Q}$ we study a DC optimal power flow (OPF) problem where the first stage computes a reserve allocation alongside a second-stage control policy that ensures system safety for

predefined uncertainty regions given by $\boldsymbol{\theta}$. The resulting problem is a robust optimization problem for which we compute uncertainty sets that avoid being overly conservative through a decision-aware tuning of $\boldsymbol{\theta}$. We deliberately chose to study a DC OPF with affine recourse in this paper due to its popularity and wide range of applications [1], [3], [9], [10], allowing us to focus on the derivation and analysis of the prescriptive and decision-dependent uncertainty set tuning. Section II explains the model in detail.

- For $\mathcal{M}$ we focus on an interpretable linear model $\mathcal{M}_{\boldsymbol{w}}(\boldsymbol{\zeta}) = \boldsymbol{M}\boldsymbol{\zeta} + \boldsymbol{m}$ parametrized by $\boldsymbol{w} = (\boldsymbol{M}, \boldsymbol{m})$ similar to the approach in [11]. We will refer to $\boldsymbol{w}$ as *weights* throughout this paper to avoid confusion with *parameters* $\boldsymbol{\zeta}$ and $\boldsymbol{\theta}$.
- The resulting problem is a bi-level program that we solve using stochastic gradient descent with differentiable optimization layers. We discuss this in Section III.

B. Related literature

Robust optimization for OPF has been studied alongside numerous proposals for general stochastic OPF. Mainly introduced by the seminal work in [5], robust OPF can provide security with manageable computational complexity and transparent communication of the considered uncertain region for each resource. More recent variants include data-driven approaches [12] and tractable extensions to AC-OPF [2], [13], [14]. However, defining uncertainty sets that ensure high-quality decisions remains tricky [15]. Adaptive robust programming approaches, e.g., [5], [16], [17], allow constraint violation estimates, but cannot internalize the impact of the set definition on the decision outcome and cost. Popular chance-constraint approaches, e.g., [3], that rely on pre-defined ellipsoidal sets have the same weakness.

Adoption barriers for stochastic optimization in power systems have been previously highlighted in [18], [19] and options for approximating their performance with no or little alteration of current industry practice have been explored in [11], [18]–[21]. While [19], [20] study more flexible reserve products and reserve requirements informed by an auxiliary stochastic program, respectively, [11], [18], [21] propose methods that prescribe intentionally biased input parameters (forecasts) to the first stage problem. This approach fits our formulations in (3) and (4). In [18] the authors solve a bi-level program for each instance of the first stage to obtain a (prescriptive) alternative wind power forecast. Building on this idea, [11] computes a mapping from the original to a prescribed net-demand forecast. To this end, the authors include the first stage optimality conditions in the second stage to compute the map using in a single stochastic program. Pursuing the similar objective of obtaining optimally biased forecasts that reflect the asymmetric power system cost structure (generation excess can typically be handled more cheaply than shortage), [21] propose a bi-level program with a scaleable solution heuristic.

Models (e.g., for forecasting) that minimize the loss of a downstream optimization task have gained general popularity as *end-to-end learning* [22] or *smart predict-and-optimize*

TABLE I
NOMENCLATURE

Operation model:	
D	Number of uncertain parameters
K	Number of constraints
G	Number of generators
V	Number of nodes (vertices) in the network
$\boldsymbol{A}$	Matrix of balancing control variables
$\boldsymbol{B}$	Linear maps from net-injections to line flows
$\boldsymbol{c}$	Vector of cost parameters
$\boldsymbol{d}$	Vector of active power demand
$\boldsymbol{f}$	Vector of power flows
$\boldsymbol{p}$	Vector of active power generation
$\boldsymbol{r}^+/\boldsymbol{r}^-$	Vector of upward/downward generation reserves
$\boldsymbol{u}$	Vector of forecasts of uncertain injections
$\boldsymbol{x}$	Vector collecting model decision variables
$\boldsymbol{\xi}$	Vector of forecast errors
Other variables and parameters:	
G	Subgradient of loss function
H	Empirical CVaR
L	Loss function
$\boldsymbol{M}$	Affine part of linear prescription map
$\mathcal{M}_{\boldsymbol{w}}$	Prescription map parameterized by $\boldsymbol{w}$
N	Number of samples
$\boldsymbol{m}$	Constant part of linear prescription map
$\boldsymbol{w}$	Parameters (weights) of prescription map
$\boldsymbol{\Sigma}_{\boldsymbol{u}}$	Covariance matrix of $\boldsymbol{\xi}$ conditional to $\boldsymbol{u}$
γ	Target probability of exceedance
$\boldsymbol{\zeta}$	Vector of context parameters
κ	Step size for Lagrangian in CVaR loss
λ	Lagrangian multiplier for CVaR loss
$\boldsymbol{\mu}$	Center of box uncertainty set
ρ	Learning rate
$\boldsymbol{\sigma}$	Vector collecting width for each dimension of the box uncertainty set
τ	Auxiliary variable for CVaR
$\boldsymbol{\theta}$	Uncertainty set parameters
Other notation:	
$\cdot^v/\cdot^z$	Superscript indicating iteration epoch/batch
$[\cdot]^+$	$\max\{0, \cdot\}$
$\boldsymbol{1}_X$	Vector of ones of dimension X

[23]. Many exciting results in this direction have been unlocked by differentiable optimization frameworks, e.g., [24], that enable efficient iterative model training procedures that internalize optimization layers. For example, [25] train a demand forecast model that minimizes the expected cost of generation excess and shortage, [26] train a generative network to obtain adversarial forecast scenarios to improve reserve allocation, [27] create a wind power forecast model that minimizes wind spillage, and [28] tune their wind power prediction model to minimize forecast errors in resulting electricity prices.

II. OPERATION MODEL

We consider a short-term generator dispatch problem with balancing control (e.g., automatic generator control) with uncertain injections from stochastic wind generators. We model these uncertain injections as a D-dimensional vector $\boldsymbol{u}(\boldsymbol{\xi}) = \boldsymbol{u} + \boldsymbol{\xi}$, where $\boldsymbol{u}$ is a (deterministic) forecast and $\boldsymbol{\xi}$ is a vector of random forecast errors. Power imbalance caused by forecast errors $\boldsymbol{\xi}$ is corrected by controllable generators. The generator output is a G-dimensional vector $\boldsymbol{p}(\boldsymbol{\xi}) = \boldsymbol{p} - \boldsymbol{A}\boldsymbol{\xi}$, where $\boldsymbol{A}$ is a matrix of decision variables that defines an affine balancing control, i.e., it defines how each generator adjusts its output as

a reaction to imbalance $\boldsymbol{\xi}$. Ensuring $\boldsymbol{A}^{\mathsf{T}}\mathbf{1}_G = \mathbf{1}_D$, where $\mathbf{1}_D$ is a D-dimensional vector of ones, ensures system balance.

A. Robust optimal power flow with affine recourse

To immunize the system against uncertain injections $\boldsymbol{u}(\boldsymbol{\xi})$, $\boldsymbol{p}(\boldsymbol{\xi})$ and the resulting uncertain power flows, the system operator defines an uncertainty set $\mathcal{U}$ that captures all outcomes of $\boldsymbol{\xi}$ for which all system constrains should hold. Given a parametrization $\boldsymbol{\zeta}$ (i.e., forecast $\boldsymbol{u}$ and demand vector $\boldsymbol{d}$—we assume in this paper that cost and system topology remain constant) the system operator solves the following robust OPF problem to decide on the generator dispatch $\boldsymbol{p}$, generator reserves $\boldsymbol{r}^+$, $\boldsymbol{r}^-$, and transmission line security margins $\boldsymbol{f}^{\mathrm{RAM}+}, \boldsymbol{f}^{\mathrm{RAM}-}$:

$$\min \quad (\boldsymbol{c}^{\mathrm{E}})^{\mathsf{T}}\boldsymbol{p} + (\boldsymbol{c}^{\mathrm{R}})^{\mathsf{T}}(\boldsymbol{r}^+ + \boldsymbol{r}^-) \tag{5a}$$

$$\text{s.t.} \quad \mathbf{1}_G^{\mathsf{T}}\boldsymbol{p} = \mathbf{1}_V^{\mathsf{T}}\boldsymbol{d} - \mathbf{1}_D^{\mathsf{T}}\boldsymbol{u} \tag{5b}$$

$$\boldsymbol{A}^{\mathsf{T}}\mathbf{1}_G = \mathbf{1}_D \tag{5c}$$

$$\boldsymbol{p} + \boldsymbol{r}^+ \le \boldsymbol{p}^{\max} \tag{5d}$$

$$\boldsymbol{p} - \boldsymbol{r}^- \ge \boldsymbol{p}^{\min} \tag{5e}$$

$$\boldsymbol{B}^{\mathrm{G}}\boldsymbol{p} + \boldsymbol{B}^{\mathrm{W}}\boldsymbol{u} - \boldsymbol{B}^{\mathrm{B}}\boldsymbol{d} \le \boldsymbol{f}^{\max} - \boldsymbol{f}^{\mathrm{RAM}+} \tag{5f}$$

$$-(\boldsymbol{B}^{\mathrm{G}}\boldsymbol{p} + \boldsymbol{B}^{\mathrm{W}}\boldsymbol{u} - \boldsymbol{B}^{\mathrm{B}}\boldsymbol{d}) \le \boldsymbol{f}^{\max} - \boldsymbol{f}^{\mathrm{RAM}-} \tag{5g}$$

$$-\boldsymbol{A}\boldsymbol{\xi} \le \boldsymbol{r}^+ \quad\quad \forall\boldsymbol{\xi} \in \mathcal{U} \tag{5h}$$

$$\boldsymbol{A}\boldsymbol{\xi} \le \boldsymbol{r}^- \quad\quad \forall\boldsymbol{\xi} \in \mathcal{U} \tag{5i}$$

$$(\boldsymbol{B}^{\mathrm{W}} - \boldsymbol{B}^{\mathrm{G}}\boldsymbol{A})\boldsymbol{\xi} \le \boldsymbol{f}^{\mathrm{RAM}+} \quad\quad \forall\boldsymbol{\xi} \in \mathcal{U} \tag{5j}$$

$$-(\boldsymbol{B}^{\mathrm{W}} - \boldsymbol{B}^{\mathrm{G}}\boldsymbol{A})\boldsymbol{\xi} \le \boldsymbol{f}^{\mathrm{RAM}-} \quad\quad \forall\boldsymbol{\xi} \in \mathcal{U} \tag{5k}$$

The objective (5a) minimizes system cost given energy cost and reserve provision cost vectors $\boldsymbol{c}^{\mathrm{E}}$ and $\boldsymbol{c}^{\mathrm{R}}$. Energy balance (5b) ensures that the total generator injections equals the total system demand $\boldsymbol{d}$ and $\boldsymbol{u}$. Similarly, (5c) ensures that all forecast errors are balanced. Constraints (5d) and (5e) enforce the technical production limits of each controllable generator. Constraints (5f) and (5g) map the power injections and withdrawals of each resource and load to a resulting power flow via linear maps $\boldsymbol{B}^{\mathrm{G}}$, $\boldsymbol{B}^{\mathrm{W}}$, $\boldsymbol{B}^{\mathrm{B}}$, e.g., obtained from the DC power flow approximation [29]. Vectors $\boldsymbol{f}^{\mathrm{RAM}+}$ and $\boldsymbol{f}^{\mathrm{RAM}-}$ are the remaining available margins for each power transmission line, i.e., the difference between the power flow caused by the forecast injections and the upper and lower line limits. Constraints (5h)–(5k) enforce robust constraints on the system response to uncertain forecast errors $\boldsymbol{\xi}$. Constraints (5h) and (5i) ensure that generator balancing responses do not exceed the available reserves and (5j) and (5k) ensure that the resulting power flow changes do not exceed the remaining available margins on each power line for any $\boldsymbol{\xi} \in \mathcal{U}$.

Problem (5) can be re-written in a more concise form. We collect all decision variables in a vector $\boldsymbol{x}$, cost vectors $\boldsymbol{c}^{\mathrm{E}}$, $\boldsymbol{c}^{\mathrm{R}}$ in a vector $\boldsymbol{c}$, denote the feasible space defined by constraints (5b)–(5g) as $\mathcal{F}(\boldsymbol{\zeta})$ and write:

$$\min_{\boldsymbol{x} \in \mathcal{F}(\boldsymbol{\zeta})} \boldsymbol{c}^{\mathsf{T}}\boldsymbol{x} \tag{6a}$$

$$\text{s.t.} \quad \max_{k=1,...,K}[\boldsymbol{a}_k^{\mathsf{T}}\boldsymbol{\xi} + b_k] \le 0, \quad \forall\boldsymbol{\xi} \in \mathcal{U} \tag{6b}$$

where $\boldsymbol{a}_k$ and b_k are, respectively, the k-th row and k-th entry of the $K \times D$ matrix and $K \times 1$ vector

$$\begin{bmatrix} -\boldsymbol{A} \\ \boldsymbol{A} \\ (\boldsymbol{B}^{\mathrm{W}} - \boldsymbol{B}^{\mathrm{G}}\boldsymbol{A}) \\ -(\boldsymbol{B}^{\mathrm{W}} - \boldsymbol{B}^{\mathrm{G}}\boldsymbol{A}) \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} -\boldsymbol{r}^+ \\ -\boldsymbol{r}^- \\ -\boldsymbol{f}^{\mathrm{RAM}+} \\ -\boldsymbol{f}^{\mathrm{RAM}-} \end{bmatrix}.$$

Note that (6b) is an exact reformulation of (5h)–(5k) as the maximum of K affine functions.

B. Uncertainty set formulation

Model (5) cannot be solved directly but requires a definition of $\mathcal{U}$ alongside a tractable reformulation of constraints (5h)–(5k). This paper focuses on box uncertainty sets, as they provide a clear safety region for each uncertain resource. Other formulations are possible [30], [31]. Box uncertainty sets ensure that constraints are feasible for a security interval along each dimension of the uncertain vector $\boldsymbol{\xi}$. We can define such a set as $\mathcal{U}^{\mathrm{box}}(\boldsymbol{\theta}) = \{\xi_j | \xi_j \in [\mu_j - \sigma_j, \mu_j + \sigma_j], \; j = 1,...,D\}$ parametrized by $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\sigma})$ with $\boldsymbol{\mu} = [\mu_1,...,\mu_D]$ and $\boldsymbol{\sigma} = [\sigma_1,...,\sigma_D]$. Parameter $\boldsymbol{\mu}$ is the center of the security interval and can be interpreted as a forecast error bias. Parameter $\boldsymbol{\sigma}$ defines the width of the interval. Using $\mathcal{U} = \mathcal{U}^{\mathrm{box}}$ and introducing auxiliary variable $\boldsymbol{t}_k$, (6b) becomes

$$\boldsymbol{a}_k^{\mathsf{T}}\boldsymbol{\mu} + \boldsymbol{t}_k^{\mathsf{T}}\boldsymbol{\sigma} \le b_k, \; \boldsymbol{t}_k \ge |\boldsymbol{a}_k|, \quad \forall k = 1,..,K. \tag{7}$$

C. Real-time cost and security

The choice of uncertainty set parameters $\boldsymbol{\theta}$ implies a trade-off between security in real-time and cost in the first-stage decision. For example, choosing $\boldsymbol{\theta}$ such that $\mathcal{U}$ is large and covers all potential outcomes of $\boldsymbol{\xi}$ will lead to high security in real-time but also to high first-stage cost. A good choice of $\boldsymbol{\theta}$ will balance this trade-off by minimizing the combined first- and second-stage cost as defined in (3). We consider two relevant approaches to quantify security of the robust problem from Section II-A in real-time.

1) Cost of exceedance

We can define the real-time cost as cost of exceedance by imposing a penalty for insufficient reserves $\boldsymbol{r}^+$, $\boldsymbol{r}^-$, $\boldsymbol{f}^{\mathrm{RAM}+}$, $\boldsymbol{f}^{\mathrm{RAM}-}$. Using the notation from (6), we compute this cost as

$$\mathcal{C}^{\mathrm{S}}(\boldsymbol{x}, \boldsymbol{\xi}) = \sum_{k=1}^{K} c_k^{\mathrm{viol}}\left[\boldsymbol{a}_k^{\mathsf{T}}\boldsymbol{\xi} + b_k\right]^+, \tag{8}$$

where $[\cdot]^+ = \max\{\cdot, 0\}$ and c_k^{viol} is the cost for exceeding the reserve given by b_k. Cost c_k^{viol} could, for example, reflect the cost of procuring emergency resources or load shedding.

2) Probability of exceedance

Instead of minimizing cost of exceedance, the system operator may be interested in a probabilistic guarantee that real-time operations do not exceed reserves, i.e.:

$$\mathbb{P}\left[\max_{k=1,...,K}[\boldsymbol{a}_k^{\mathsf{T}}\boldsymbol{\xi} + b_k] < 0\right] \ge 1 - \gamma. \tag{9}$$

Here, $(1 - \gamma)$ defines the target probability of no constraint exceeding its limits and γ is a small risk factor.

III. SOLUTION APPROACH

We now show an iterative method inspired by the results in [7] to obtain the desired mapping $\mathcal{M}_{\boldsymbol{w}}(\boldsymbol{\zeta}) = \boldsymbol{M}\boldsymbol{\zeta} + \boldsymbol{m}$ that returns uncertainty set parameters $\boldsymbol{\theta}$, which, in turn, optimally parametrize the first-stage uncertainty set with respect to the cost and probability of exceedance discussed in Section II-C.

A. Cost of exceedance

The problem to compute the optimal choice of weights $\boldsymbol{w} = (\boldsymbol{M}, \boldsymbol{m})$ that minimize the combined expected first- and second-stage cost is the bi-level problem:

$$\min_{\boldsymbol{w}} \quad \mathbb{E}_{(\boldsymbol{\zeta}, \boldsymbol{\xi})}\big[\boldsymbol{c}^\mathsf{T}\boldsymbol{x}^* + \mathcal{C}^{\mathrm{S}}(\boldsymbol{x}^*, \boldsymbol{\xi})\big] \tag{10a}$$

$$\text{s.t.} \quad \boldsymbol{\theta} = \mathcal{M}_{\boldsymbol{w}}(\boldsymbol{\zeta}) \tag{10b}$$

$$\boldsymbol{x}^*(\boldsymbol{\zeta}, \boldsymbol{\theta}) \in \begin{cases} \arg\min_{\boldsymbol{x} \in \mathcal{F}(\boldsymbol{\zeta})} \boldsymbol{c}^\mathsf{T}\boldsymbol{x} \\ \text{s.t.} \ \max_k[\boldsymbol{a}_k^\mathsf{T}\boldsymbol{\xi} + b_k] \leq 0, \ \forall \boldsymbol{\xi} \in \mathcal{U}(\boldsymbol{\theta}). \end{cases} \tag{10c}$$

We solve (10) using a stochastic gradient descent approach (see, e.g., [32]) with $v^{\max}$ outer steps (epochs) indexed by v and $z^{\max}$ inner steps ("mini-batches" [33]) indexed by z. For each step z we assume having access to an *individual* context parameter sample $\boldsymbol{\zeta}^z$ (e.g., obtained from historic observations or a sample generation mechanism) and a *set of N_z samples* $\mathcal{X}^z = \{\boldsymbol{\xi}_i^z\}_{i=1}^{N_z}$ of $\boldsymbol{\xi}$ (again, either obtained from historical observations or a sample generation mechanism). We note that $\mathcal{X}^z$ may be conditional to $\boldsymbol{\zeta}^z$. See, for example, [34] who also provide an approach to sample $\mathcal{X}^z$ for a given wind power forecast $\boldsymbol{u}^z$ (which is part of the parametrization $\boldsymbol{\zeta}^z$). We provide additional discussion on the relationship between $\boldsymbol{\zeta}^z$ and $\mathcal{X}^z$ in the case study below. We define $\boldsymbol{x}^z = \boldsymbol{x}^*(\boldsymbol{\zeta}^z, \boldsymbol{\theta}^z)$ as the solution to the inner problem (10c). The loss function $L(\boldsymbol{w}^z; \boldsymbol{\zeta}^z, \boldsymbol{\xi}^z)$ corresponds to the problem objective in (10c) and we compute it as the empirical mean given $\mathcal{X}^z$:

$$L^{\mathrm{C}}(\boldsymbol{w}^z; \boldsymbol{\zeta}^z, \mathcal{X}^z) = \boldsymbol{c}^\mathsf{T}\boldsymbol{x}^z + \frac{1}{N_z}\sum_{i=1}^{N_z}\sum_{k=1}^{K} c_k\big[(\boldsymbol{a}_k^z)^\mathsf{T}\boldsymbol{\xi}_i^z + b_k^z\big]^+. \tag{11}$$

Recall from (6b) that $\boldsymbol{a}_k^z$ and b_k^z are part of $\boldsymbol{x}^z$. Following [7] we estimate the derivative of $L^{\mathrm{C}}(\boldsymbol{w}^z; \boldsymbol{\zeta}^z, \boldsymbol{\xi}^z)$ using a subgradient $G^{\mathrm{C}}(\boldsymbol{w}^z; \boldsymbol{\zeta}^z, \mathcal{X}^z)$ computed over samples of $\boldsymbol{\zeta}$ and $\boldsymbol{\xi}$. Notably, the computation of $G^{\mathrm{C}}(\boldsymbol{w}^z; \boldsymbol{\zeta}^z, \mathcal{X}^z)$ requires computing gradient $\nabla_{\boldsymbol{\theta}}\boldsymbol{x}^*(\boldsymbol{\zeta}, \boldsymbol{\theta})$, i.e., the derivative of the decision variables of the inner problem over the uncertainty set parameters. We discuss an efficient approach to obtain this gradient in Section III-C below. Choosing initial weights $(\boldsymbol{M}^{\mathrm{init}}, \boldsymbol{m}^{\mathrm{init}})$ and a learning rate ρ the resulting solution steps are itemized in Algorithm 1.

B. Probability of exceedance

The problem to compute the optimal choice of weights $\boldsymbol{w}$ that ensure real-time operations do not exceed their limits with a probability of at least $(1 - \gamma)$ is the bi-level problem:

$$\min_{\boldsymbol{w}} \quad \mathbb{E}_{(\boldsymbol{\zeta}, \boldsymbol{\xi})}\big[\boldsymbol{c}^\mathsf{T}\boldsymbol{x}^*\big] \tag{12a}$$

$$\text{s.t.} \quad \mathbb{P}\Big[\max_{k=1,\dots,K}[(\boldsymbol{a}_k^*)^\mathsf{T}\boldsymbol{\xi} + b_k^*] < 0\Big] \geq 1 - \gamma \tag{12b}$$

Algorithm 1 Stochastic gradient descent to solve (10)

1: **given** $\boldsymbol{w}^0 = (\boldsymbol{M}^{\mathrm{init}}, \boldsymbol{m}^{\mathrm{init}})$, learning rate ρ
2: **for** $v = 1, \dots, v^{\max}$ **do** ▷ *outer epoch*
3:    **for** $z = 1, \dots, z^{\max}$ **do** ▷ *inner mini-batch*
4:       $\boldsymbol{\zeta}^z \leftarrow$ sample context parameter
5:       $\mathcal{X}^z \leftarrow$ sample forecast errors conditional to $\boldsymbol{\zeta}^z$
6:       $\boldsymbol{\theta}^z \leftarrow \mathcal{M}_{\boldsymbol{w}^{v-1}}(\boldsymbol{\zeta}^z)$ ▷ *prescribe parameters*
7:       $\boldsymbol{x}^z \leftarrow$ solve inner problem (10c)
8:       $\boldsymbol{g}^z \leftarrow G^{\mathrm{C}}(\boldsymbol{w}^{v-1}; \boldsymbol{\zeta}^z, \mathcal{X}^z)$
9:    $\boldsymbol{g}^v \leftarrow \frac{1}{z^{\max}}\sum_{z=1}^{z^{\max}} \boldsymbol{g}^z$ ▷ *update gradient*
10:   $\boldsymbol{w}^v \leftarrow \boldsymbol{w}^{v-1} - \rho\boldsymbol{g}^v$ ▷ *update variables*
11: **return** $\boldsymbol{w}^{v^{\max}}$

$$\big[\boldsymbol{\theta} \text{ and } \boldsymbol{x}^*(\boldsymbol{\zeta}, \boldsymbol{\theta}) \text{ as in (10b) and (10c)}\big]. \tag{12c}$$

Constraint (12b) is generally non-convex and complicates the solution of (12). To create a tractable problem, we reformulate (12b) using conditional value-at-risk [7], [35]:

$$\mathbb{P}\big[\max_{k=1,\dots,K}[\boldsymbol{a}_k^\mathsf{T}\boldsymbol{\xi} + b_k \leq 0]\big] \geq 1 - \gamma$$
$$\Leftrightarrow \quad \mathrm{VaR}_\gamma\big(\max_{k=1,\dots,K}[\boldsymbol{a}_k^\mathsf{T}\boldsymbol{\xi} + b_k]\big) \leq 0 \tag{13}$$
$$\Leftarrow \quad \mathrm{CVaR}_\gamma\big(\max_{k=1,\dots,K}[\boldsymbol{a}_k^\mathsf{T}\boldsymbol{\xi} + b_k]\big) \leq 0$$

where VaR_γ and CVaR_γ are the value-at-risk and conditional value-at-risk at risk level γ. Reformulation (13) utilizes the fact that limiting CVaR implies a limit on VaR [36]. CVaR further allows the convex reformulation [36]

$$\mathrm{CVaR}_\gamma = \inf_\tau \Big\{\mathbb{E}\big[\frac{1}{\gamma}\big[\max_{k=1,\dots,K}[\boldsymbol{a}_k^\mathsf{T}\boldsymbol{\xi} + b_k] - \tau\big]^+ + \tau\big]\Big\}, \tag{14}$$

which we use to define

$$h(\boldsymbol{x}^*, \tau, \boldsymbol{\xi}) = \frac{1}{\gamma}\big[\max_{k=1,\dots,K}[(\boldsymbol{a}_k^*)^\mathsf{T}\boldsymbol{\xi} + b_k^*] - \tau\big]^+ + \tau, \tag{15}$$

and to reformulate (12) as [7]:

$$\min_{\boldsymbol{w}, \tau} \quad \mathbb{E}_{(\boldsymbol{\zeta}, \boldsymbol{\xi})}\big[\boldsymbol{c}^\mathsf{T}\boldsymbol{x}^*\big] \tag{16a}$$

$$\text{s.t.} \quad \mathbb{E}_{(\boldsymbol{\zeta}, \boldsymbol{\xi})}\big[h(\boldsymbol{x}^*, \tau, \boldsymbol{\xi})\big] = 0 \tag{16b}$$

$$\big[\boldsymbol{\theta} \text{ and } \boldsymbol{x}^*(\boldsymbol{\zeta}, \boldsymbol{\theta}) \text{ as in (10b) and (10c)}\big]. \tag{16c}$$

We note that (16) has an additional auxiliary variable τ related to the CVaR reformulation (14). Also, following the logic in [7], we note that (16b) is an equality constraint, because zero is the optimal CVaR target given (13) and the convexity of (14). Finally, we note that for fixed $\boldsymbol{x}^*$ the term $\max_k[(\boldsymbol{a}_k^*)^\mathsf{T}\boldsymbol{\xi} + b_k^*]$ in (15) computes the constraint violation resulting from a given $\boldsymbol{\xi}$ and may be larger than zero.

Similar to he procedure outlined in [7], [37] we can now solve the equality-constrained problem (16) by introducing Lagrangian multiplier λ and defining the loss function

$$L^{\mathrm{P}}((\boldsymbol{w}, \tau); \lambda, \boldsymbol{\zeta}, \boldsymbol{\xi}) = \mathbb{E}_{(\boldsymbol{\zeta}, \boldsymbol{\xi})}\big[\boldsymbol{c}^\mathsf{T}\boldsymbol{x}^*\big] + \lambda\mathbb{E}_{(\boldsymbol{\zeta}, \boldsymbol{\xi})}\big[h(\boldsymbol{x}^*, \tau, \boldsymbol{\xi})\big]. \tag{17}$$

Using the notation introduced in Section III-A above, we compute (17) in each step z as

$$L^{\mathrm{P}}((\boldsymbol{w}^z, \tau^z); \lambda^z, \boldsymbol{\zeta}^z, \mathcal{X}^z) = \tag{18}$$
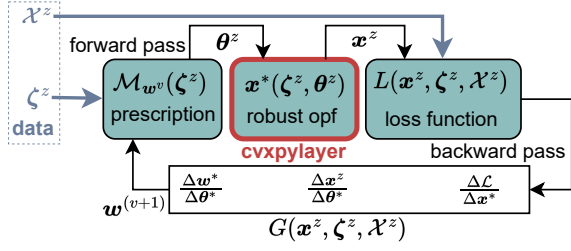
PSCC 2024

Fig. 1. Computation pipeline for the steps itemized in Algorithms 1 and 2. The differentiable optimization layer using `cvxpylayers` [24], allows the computation of subgradient $G = \nabla_{\boldsymbol{w}} L$ with a single backwards pass.

$$\boldsymbol{c}^{\mathsf{T}}\boldsymbol{x}^z + \lambda^z \underbrace{\frac{1}{N_z}\sum_{i=1}^{N_z}\Big(\frac{1}{\gamma}\big[\max_{k=1,...,K}[(\boldsymbol{a}_k^z)^{\mathsf{T}}\boldsymbol{\xi}_i^z + b_k^z] - \tau^z\big]^+ + \tau^z\Big)}_{H((\boldsymbol{w}^z,\tau^z);\boldsymbol{\zeta}^z,\mathcal{X}^z)}.$$

Denoting $G^{\mathrm{P}}((\boldsymbol{w}^z,\tau^z);\lambda^z,\boldsymbol{\zeta}^z,\mathcal{X}^z)$ as the gradient of (18), τ^{init}, λ^{init} as initial values for τ, λ, and κ as the step size for λ the resulting solution steps are itemized in Algorithm 2.

Algorithm 2 Stochastic gradient descent to solve (12)

1: **given** $\boldsymbol{w}^0 = (\boldsymbol{M}^{\mathrm{init}}, \boldsymbol{m}^{\mathrm{init}})$, $\tau^0 = \tau^{\mathrm{init}}$, $\lambda^0 = \lambda^{\mathrm{init}}$, ρ, κ
2:   **for** $v = 1, ..., v^{\max}$ **do**   ▷ *outer epoch*
3:     **for** $z = 1, ..., z^{\max}$ **do**   ▷ *inner mini-batch*
4:       $\boldsymbol{\zeta}^z \leftarrow$ sample context parameter
5:       $\mathcal{X}^z \leftarrow$ sample forecast errors conditional to $\boldsymbol{\zeta}^z$
6:       $\boldsymbol{\theta}^z \leftarrow \mathcal{M}_{\boldsymbol{w}^{v-1}}(\boldsymbol{\zeta}^z)$ ▷ *prescribe parameters*
7:       $\boldsymbol{x}^z \leftarrow$ solve inner problem (10c)
8:       $\boldsymbol{g}^z \leftarrow G^{\mathrm{C}}((\boldsymbol{w}^{v-1},\tau^{v-1});\lambda^{v-1},\boldsymbol{\zeta}^z,\mathcal{X}^z)$
9:       $H^v \leftarrow H((\boldsymbol{w}^z,\tau^z),\boldsymbol{\zeta}^z,\mathcal{X}^z)$ ▷ *See* (18)
10:     $\boldsymbol{g}^v \leftarrow \frac{1}{z^{\max}}\sum_{z=1}^{z^{\max}}\boldsymbol{g}^z$ ▷ *update gradient*
11:     $(\boldsymbol{w}^v,\tau^v) \leftarrow (\boldsymbol{w}^{v-1},\tau^{v-1}) - \rho\boldsymbol{g}^v$ ▷ *update variables*
12:     $H^v = \frac{1}{z^{\max}}\sum_{z=1}^{z^{\max}}H^z$ ▷ *compute current CVaR*
13:     $\lambda^v \leftarrow \lambda^{v-1} + \kappa H^v$ ▷ *update Lagrangian multiplier*
14: **return** $(\boldsymbol{w}^{v^{\max}}, \tau^{v^{\max}})$

Remark 1. We highlight that the mapping $\mathcal{M}_{\boldsymbol{w}}$ is only used to define the uncertainty set $\mathcal{U}$. Once this set is obtained, problem (5) can be solved by off-the-shelf solvers using reformulation (7). This guarantees optimality and feasibility for a given $\mathcal{U}$.

C. Implementation

We implement Algorithms 1 and 2 in Python using state-of-the-art machine learning packages (PyTorch) and results from [24] (package Cvxpylayers) that allow numerical differentiation through the inner optimization (10c). The computation steps are illustrated in Fig. 1. The differentiable optimization layer obtains the required gradient $\nabla_{\boldsymbol{\theta}}\boldsymbol{x}^*(\boldsymbol{\zeta},\boldsymbol{\theta})$ by differentiating through the Karush-Kuhn-Tucker conditions of the inner problem at optimality. This is performed efficiently by utilizing the implicit function theorem and by solving an inner quadratic program to obtain the required matrix inversion [38].

We add the following implementation remarks:

IR1: Formulation (5) limits the decision variables $\boldsymbol{A}$ to the interval $[0, 1]$. This leads to $\boldsymbol{A}$ taking small values compared to the other decision variables causing conditioning problems in the Cvxpylayers solver ECOS [39]. Introducing a scaling factor for $\boldsymbol{A}$ solves this problem consistently.

IR2: The inner problem has to be always feasible. To ensure this, we add a slack variable to (5b) to allow the curtailment of wind power $\boldsymbol{u}$ if needed, and a slack variable to (6b) to account for the case that the required uncertainty set can not be met with the available reserves. Both sets of slack variables are penalized in the problem objective and conditioned using a scaling factor as in IR1.

IR3: Formulation (5) is a linear problem and as such may not generally be differentiable with respect to its parameters [40]. This can be avoided by introducing a regularization $\pi \|\boldsymbol{x}\|_2^2$ with regularization factor π to the objective of the problem. We note that in contrast to [40], our problem is linear with *continuous* variables and adding a regularization term with a small π did not significantly impact our results.

Lastly, we note that we implement $\mathcal{M}_{\boldsymbol{w}}$ as two linear models $\mathcal{M}^{\boldsymbol{\mu}} = \boldsymbol{M}^{\mu}\boldsymbol{\zeta} + \boldsymbol{m}^{\mu}$ and $\mathcal{M}^{\boldsymbol{\sigma}} = \boldsymbol{M}^{\sigma}\boldsymbol{\zeta} + \boldsymbol{m}^{\sigma}$.

IV. NUMERICAL EXPERIMENTS

A. Illustrative 5-bus case

We first illustrate the suggested approach using synthetic data on the small-scale "case5" data set from MATPOWER [41]. The system topology with two configurations of wind generators is shown in Fig. 2. Its parameters follow [35] with $\boldsymbol{f}^{\max} = (3.2, 1.52, 1.76, 0.8, 0.8, 1.92)$ p.u., $\boldsymbol{c}^{\mathrm{E}} = (14, 15, 30, 40, 10)$\$/MW, and $\boldsymbol{c}^{\mathrm{R}} = (80, 80, 15, 30, 80)$\$/MW.

For training and testing the proposed approach we create a collection of $N = 2000$ samples of $\boldsymbol{\zeta} = (\boldsymbol{d}, \boldsymbol{u})$ with corresponding samples of $\boldsymbol{\xi}$ as follows. First, we define the nominal demand as $\boldsymbol{d}_0 = (0.0, 3.0, 3.0, 4.0, 0.0)$ p.u. and the nominal wind forecast as $\boldsymbol{u}_0 = (1.0, 1.5)$ p.u.. We then create N samples of $\boldsymbol{d}$ by uniformly drawing from the set $\{\boldsymbol{d} \in \mathbb{R} \mid 0.5\boldsymbol{d}_0 \leq \boldsymbol{d} \leq 1.1\boldsymbol{d}_0\}$ and N samples of $\boldsymbol{u}$ by uniformly drawing from the set $\{\boldsymbol{u} \in \mathbb{R} \mid 0.5\boldsymbol{u}_0 \leq \boldsymbol{u} \leq 1.1\boldsymbol{u}_0\}$. Next, for each available sample of $\boldsymbol{\zeta} = (\boldsymbol{d}, \boldsymbol{u})$ we draw a sample of forecast errors $\boldsymbol{\xi}$ from a multivariate normal distribution with a uniform correlation coefficient of $\phi = 0.5$ between all wind farms. Following the observations in [34], we model the forecast error standard deviation to be conditional to the forecast and the forecast error mean to be zero. The resulting normal distribution is given as:

$$\boldsymbol{\xi} \mid \boldsymbol{u} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma_u}), \quad \boldsymbol{\Sigma_u} = \begin{bmatrix} (0.15u_1)^2 & 0.15^2\phi u_1 u_2 \\ 0.15^2\phi u_1 u_2 & (0.15u_2)^2 \end{bmatrix}. \quad (19)$$

Finally, we set the maximum wind farm capacity to $\boldsymbol{u}^{\max} = (2.0, 3.0)$ p.u. and truncate all generated samples of $\boldsymbol{\xi} \mid \boldsymbol{u}$ such that $\boldsymbol{\xi} \geq -\boldsymbol{u}$ and $\boldsymbol{\xi} \leq \boldsymbol{u}^{\max} - \boldsymbol{u}$. The resulting collection of samples of $\boldsymbol{\zeta}$ with a single sample of $\boldsymbol{\xi}$ corresponds to what would be available in practice. We use 1500 of these generated samples for training and 500 for testing.

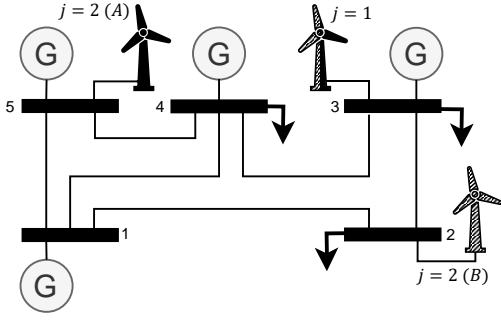We define the following 5 cases that we use to demonstrate the proposed prescribed robust sets:

Fig. 2. Schematic of the 5 bus test system in configuration A (wind farms at buses 3 and 5) and B (wind farms at buses 2 and 3).
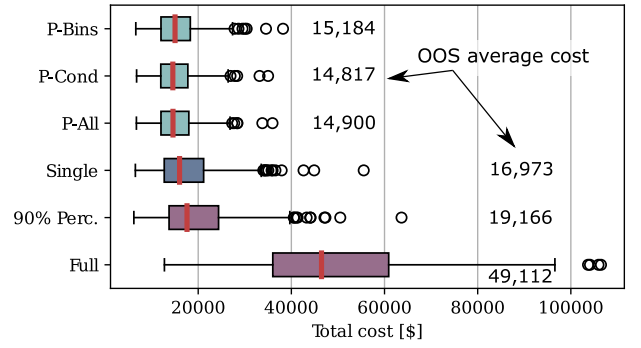

Fig. 3. Out-of-sample (OOS) results for 5-bus system in configuration A. Optimized for expected cost of exceedance. The boxes extend from the first quartile to the third quartile. Red line shows the median. The whiskers extend from the box to the farthest data point within 1.5 times the inter-quartile range from the box.
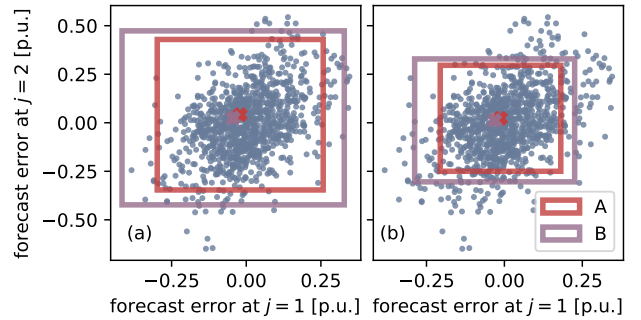

Fig. 4. Prescribed sets from case *P-All* (optimized for cost of exceedance) for (a) $\boldsymbol{\zeta} = (\boldsymbol{d}_0, \boldsymbol{u}_0)$ and (b) $\boldsymbol{\zeta} = 0.7(\boldsymbol{d}_0, \boldsymbol{u}_0)$, each for the two network configurations shown in Fig. 2.

- *Full*: Reference case for which the security interval of each wind farm covers the entire empirical forecast error support.
- *90 Perc*: Reference case for which the security interval of each wind farm is fixed between the 10 % and 90 % percentile of the forecast error training data.
- *Single*: Learning a single fixed uncertainty set without prescription, i.e., $\mathcal{M}^\mu = \boldsymbol{m}^\mu$ and $\mathcal{M}^\sigma = \boldsymbol{m}^\sigma$.
- *P-All*: Learning $\boldsymbol{w}$ using all available forecast errors. As a result, the training is ignorant to the error distribution being conditional to $\boldsymbol{u}$.
- *P-Cond*: Learning $\boldsymbol{w}$ assuming access to samples from the true conditional distribution. In each step z, given $\boldsymbol{u}^z$, we generated 200 new samples of $\boldsymbol{\zeta}$ using (19).
- *P-Bins*: Learning $\boldsymbol{w}$ with each set of forecast error samples $\mathcal{X}^z$ obtained by separating the available training samples of $\boldsymbol{u}$ into 10 bins of equal width, as in [34], collecting the forecast error of each bin, and assigning each sample $\boldsymbol{u}^z$ the set of forecast error samples corresponding to its bin.

Cases *Full*, *90 Perc.*, and *Single* offer a comparison to methods with fixed robust sets. We set $v^{\max} = 100$, although we typically observed satisfying convergence within 40 iterations. We used a mini-batch size of $z^{\max} = 20$ and a learning rate of $\tau = 10^{-6}$. Finally, we set $\boldsymbol{w}^{\text{init}}$ such that all entries of $\boldsymbol{M}^\mu$, $\boldsymbol{M}^\sigma$, and $\boldsymbol{m}^\mu$ are zero and the entries $\boldsymbol{m}^\sigma$ correspond to two times the empirical standard deviation of the training data. All experiments are implemented in Python (see also Section III-C) and available online [42]. We used a standard PC workstation with 16 GB memory and an Intel i5 processor.

1) Cost-based box uncertainty set

We first train $\mathcal{M}_{\boldsymbol{w}}$ to optimize the expected cost of constraint exceedance (see Section III-A). We set $c_k^{\text{viol}} = 20k\,^\$/\text{MW}$, $\forall k$, which corresponds to the value of lost load estimated by New York ISO [43]. The average computation time for each epoch across all cases was around $0.25\,\text{s}$, including sampling, solving the inner optimization problems, and computing the loss function and its gradients.

Fig. 3 shows the out-of-sample (OOS) average cost for the studied cases. The fully robust approach (*Full*) is overly conservative, which we can infer from the fact that the lowest cost on this case are around the 25-percentile of all other cases. Also, it is often infeasible, leading to higher cost from infeasibility penalties (see Section III-C IR2). The fixed *90*

Perc. case is less conservative and mostly avoids infeasibility, but is outperformed by the other cases. The *Single* uncertainty set improves average cost relative to *90 Perc*, but the set remains too small, as it tries to avoid infeasibility penalties. Introducing the prescription step overcomes this problem. The prescriptive cases *P-All*, *P-Cond*, and *P-Bins* further improve upon *Single* by 12.2 %, 12.7 %, and 10.5 %, respectively. Case *P-Cond* with access to the true conditional distribution slightly outperforms *Single*. Case *P-Bins* performs worse than *P-Single*, which we attribute to the loss of correlation information in the binning of the forecast errors. Re-running the experiment without correlation ($\phi = 0$) confirms this. Now, *P-All*, *P-Cond*, and *P-Bins* result in average OOS cost of 14,899 \$, 14,851 \$, and 14,981 \$, with *P-Bins* now outperforming *P-All* and only being slightly worse than the ideal *P-Cond*.

Fig. 4 shows how a change of the system topology impacts the prescribed uncertainty set, highlighting the relevance of making the training problem-aware. By moving wind farm $j = 2$ from bus 5 (configuration A) to bus 2 (configuration B), it can no longer rely on the direct balancing from the generator at bus 5. As a result, transmission lines are now more likely to exceed their remaining available margins in real time which leads to larger required safety intervals for both wind farms.
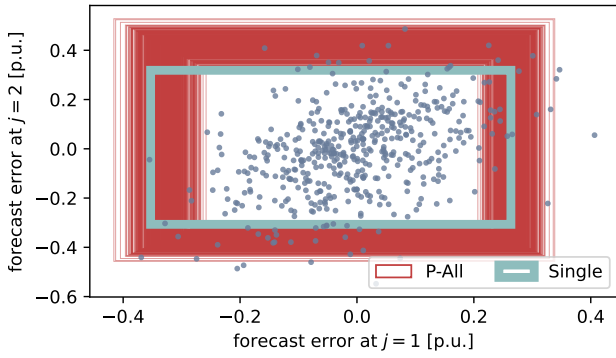
Fig. 5. Uncertainty sets for cases *P-All* and *P-Single* optimized for probability of exceedance. For *P-All* all prescriptions resulting from the test samples of ζ are shown. (System configuration A.)
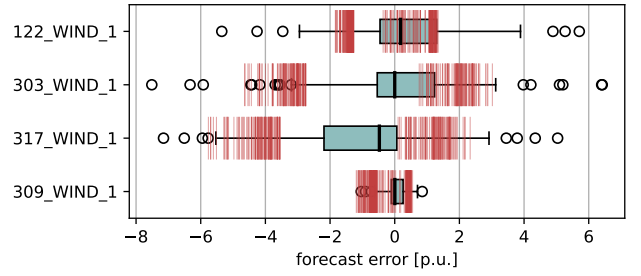

Fig. 6. OOS results for the RTS 96 bus case using the *P-All* training. Box plots show the distribution of the OOS forecast errors and red lines show the various security intervals obtained from the trained mapping. See Fig. 3 for box plot explanation.

2) Constraint-based box uncertainty set

We now train $\mathcal{M}_w$ such that the probability of constraint exceedance remains below 99 %, i.e., $\gamma = 0.01$, as described in Section III-B. For this experiment we increase the learning rate to $\phi = 10^{-5}$ and set $\lambda^{\text{init}} = 100$ and $\kappa = 0.1$. The time per iteration increased slightly to around 0.67 s per epoch, which we can mainly attribute to the more complex loss function. For *Single* the CVaR at convergence is 0.96 with an in-training probability of exceedance of 3.1% and a testing probability of exceedance of 13%. As discussed in Section IV-A1 above, this unsatisfying result can be explained with the algorithm avoiding infeasible uncertainty sets resulting in a set that is too small for many parametrizations (see Fig. 5). For *P-All*, on the other hand, the CVaR at convergence is 0.06 and we achieve an in-training probability of exceedance of 1% and a testing probability of exceedance of 1.8%. We note that the exact match of the in-training probability of exceedance with the target is a coincidence. The actual target of a CVaR equal to zero would lead to a smaller in-training probability of exceedance. However, the algorithm converges above this target as it again tries to avoid to create infeasible sets. Fig. 5 shows the resulting uncertainty sets for *Single* and *P-All* and highlights the improved ability of the set in the latter case to adapt to the current system state. Comparing Figs. 4 and 5 we see that the uncertainty sets resulting from the two different performance targets are qualitatively similar.

B. RTS 96-bus case

We now test the suggested approach on more realistic data using the system provided by the Reliability Test System Grid Modernization Lab Consortium (RTS-GLMC [44]). This update of the RTS-96 test system has 73 buses, 120 transmission lines, 73 conventional generators, 4 wind farms, and 76 other resources (hydro, PV). In our experiment we focus on the 4 wind farms as uncertain resources and treat hydro and PV injections as fixed negative demand, i.e. as part of d. The RTS-GLMC data set includes data for one year. To have a richer data set for the wind farms, we use the coordinates provided in the RTS-GLMC data set to map the 4 wind farms to the closest data points available in the extensive NREL WIND Toolkit [45]. We scale this data to fit the wind farms in the RTS-GLMC data set and obtain 7 years of wind power injections and realistic forecast errors. We select the data from 2012 to replace the wind data from the RTS-GLMC data set, as the yearly wind structure matches original data most closely (measured in terms of average deviation of hourly total wind injections). From the resulting 8760 available samples net-demand and wind-injection samples of the respective day-ahead data sets, we select 1500 for training and 500 for testing. We use all forecast errors for training as in *P-All* above and focus on the analysis of the cost of exceedance-based training.

We select the same meta-parameters as for the 5-bus case, but reduce the mini-batch size to $z^{\max} = 10$. Training requires an average of 28 s per epoch and we observe convergence after around 30 iterations. We note that around half of the time per iteration is spend on computing the gradient. This is expected because both larger parameter matrices M^σ, M^μ and inner optimization overproportionally increase the size of the computational graph from which the gradient is computed. However, because $\mathcal{M}_w$ has to be trained only once offline, training time and resources are not a critical limiting factor. In addition, any relevant parameters that change more frequently than the time needed for re-training $\mathcal{M}_w$, e.g., potential grid topology changes, can be included into context vector ζ.

In this case, the reference *90 Perc.* leads to a expected out of sample cost of 3,671,312 \$ while the *P-All* trained prescribed sets achieve a significant improvement of 1,249,725 \$. (We performed an additional grid search to find a better percentile-based uncertainty set. The best result was attained for the 78-percentile with 3,088,187 \$). Fig. 6 shows the OOS forecast errors alongside the collection of prescribed security intervals for the 4 wind farms in the system. We observe that the uncertainty sets are biased towards negative forecast errors. This can largely explained by the fact that the model chooses to curtail wind if the forecast is very high, which (i) leads to an insensitivity to upwards forecast errors and (ii) amplifies the fact that larger negative forecast errors are more likely at high wind power forecasts. This result highlights the advantage of internalizing the problem cost structure into the training.

V. CONCLUSION

This paper demonstrated an approach to compute uncertainty sets for robust optimal power flow that (i) are prescriptive, i.e., minimize the expected cost system cost, and (ii) are adaptive, i.e., are prescribed individually for each expected system state given by a vector of context parameters. Our approach to obtain these sets was inspired by [7] but additionally achieves property (ii).

The problem in (4) opens a wide range of future research [46]. For the approach studied in this paper we are pursuing the following avenues for further research. (i) Including richer context vectors. For example, in our data for the RTS96 case study, we observed a clear dependency of the forecast error distribution on the wind direction. Making the problem more context aware and better estimate conditional error distributions from real data should reveal impactful relations between system security and observable parameters, but is a non-trivial task [46]. (ii) AC power flow. Higher fidelity operational models, e.g., along the lines of [2], [14], promise improved applicability in practice and improved system security. (iii) Non-fixed recourse. Relaxing the second stage with a decision-making problem (or a tractable proxy, e.g., a trained neural network), should allow for a broader set of applications of the method. (iv) Investigating scalability, e.g., through parallelization or differentiable optimization advances, e.g., [47].

ACKNOWLEDGMENT

REFERENCES

[1] L. Roald and G. Andersson, "Chance-constrained ac optimal power flow," *IEEE Trans. Power Sys.*, vol. 33, no. 3, pp. 2906–2918, 2017.

[2] D. Lee *et al.*, "Robust ac optimal power flow with robust convex restriction," *IEEE Trans. Power Sys.*, vol. 36, no. 6, 2021.

[3] D. Bienstock *et al.*, "Chance-constrained optimal power flow," *SIAM Review*, vol. 56, no. 3, 2014.

[4] J. Kazempour *et al.*, "A stochastic market design with revenue adequacy and cost recovery by scenario: Benefits and costs," *IEEE Trans. Power Sys.*, vol. 33, no. 4, pp. 3531–3545, 2018.

[5] D. Bertsimas *et al.*, "Adaptive robust optimization for the security constrained unit commitment problem," *IEEE Trans. Power Sys.*, vol. 28, no. 1, pp. 52–63, 2012.

[6] Z. Liang *et al.*, "Inertia pricing in stochastic electricity markets," *IEEE Trans. Power Sys.*, vol. 38, no. 3, 2022.

[7] I. Wang *et al.*, "Learning for robust optimization," *arXiv:2305.19225*, 2023.

[8] D. Bertsimas and N. Kallus, "From predictive to prescriptive analytics," *Manag. Sci.*, vol. 66, no. 3, pp. 1025–1044, 2020.

[9] M. Lubin *et al.*, "A robust approach to chance constrained optimal power flow with renewable generation," *IEEE Transactions on Power Systems*, vol. 31, no. 5, pp. 3840–3849, 2015.

[10] R. Weinhold and R. Mieth, "Uncertainty-aware capacity allocation in flow-based market coupling," *IEEE Trans. Power Syst.*, 2023.

[11] J. M. Morales *et al.*, "Prescribing net demand for two-stage electricity generation scheduling," *Oper. Res. Perspect.*, vol. 10, p. 100268, 2023.

[12] D. Bertsimas *et al.*, "Data-driven robust optimization," *Math. Program.*, vol. 167, pp. 235–292, 2018.

[13] R. Louca and E. Bitar, "Robust ac optimal power flow," *IEEE Trans. Power Sys.*, vol. 34, no. 3, pp. 1669–1681, 2018.

[14] ——, "Stochastic ac optimal power flow with affine recourse," in *IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 2431–2436.

[15] F. Golestaneh *et al.*, "Polyhedral predictive regions for power system applications," *IEEE Trans. Power Sys.*, vol. 34, no. 1, 2018.

[16] A. Lorca *et al.*, "Multistage adaptive robust optimization for the unit commitment problem," *Oper. Res.*, vol. 64, no. 1, pp. 32–51, 2016.

[17] M. Isuru *et al.*, "A piecewise-affine decision rule based stochastic ac optimal power flow approach," in *Proc. of the IEEE PES Gen. Meeting*. IEEE, 2020.

[18] J. M. Morales *et al.*, "Electricity market clearing with improved scheduling of stochastic production," *Eur. J. Oper. Res.*, vol. 235, no. 3, 2014.

[19] V. Dvorkin *et al.*, "Setting reserve requirements to approximate the efficiency of the stochastic dispatch," *IEEE Trans. Power Sys.*, vol. 34, no. 2, pp. 1524–1536, 2018.

[20] B. Wang and B. F. Hobbs, "Fleiramp market design for real-time operations," in *Proc. of the IEEE PES Gen. Meeting*. IEEE, 2013.

[21] J. D. Garcia *et al.*, "Application-driven learning: A closed-loop prediction and optimization approach applied to dynamic reserves and demand forecasting," *arXiv:2102.13273*, 2021.

[22] J. Kotary *et al.*, "End-to-end constrained optimization learning: A survey," *arXiv:2103.16378*, 2021.

[23] A. N. Elmachtoub and P. Grigas, "Smart "predict, then optimize"," *Manag. Sci.*, vol. 68, no. 1, pp. 9–26, 2022.

[24] A. Agrawal *et al.*, "Differentiable convex optimization layers," *Advances in neural information processing systems*, vol. 32, 2019.

[25] P. Donti *et al.*, "Task-based end-to-end model learning in stochastic optimization," *Adv. Neural Inf. Process.*, vol. 30, 2017.

[26] Z. Liang *et al.*, "Operation-adversarial scenario generation," *Electric Power Systems Research*, vol. 212, p. 108451, 2022.

[27] D. Wahdany *et al.*, "More than accuracy: end-to-end wind power forecasting that optimises the energy system," *Electr. Power Syst. Res.*, vol. 221, p. 109384, 2023.

[28] V. Dvorkin and F. Fioretto, "Price-aware deep learning for electricity markets," *arXiv:2308.01436*, 2023.

[29] S. Bolognani and F. Dörfler, "Fast power system analysis via implicit linearization of the power flow manifold," in *Proc. of the 53rd Annual Allerton Conference*. IEEE, 2015, pp. 402–409.

[30] D. Bertsimas *et al.*, "Theory and applications of robust optimization," *SIAM review*, vol. 53, no. 3, pp. 464–501, 2011.

[31] B. L. Gorissen and D. Den Hertog, "Robust counterparts of inequalities containing sums of maxima of linear functions," *Eur. J. Oper. Res.*, vol. 227, no. 1, pp. 30–43, 2013.

[32] S.-i. Amari, "Backpropagation and stochastic gradient descent method," *Neurocomputing*, vol. 5, no. 4-5, pp. 185–196, 1993.

[33] R. M. Gower *et al.*, "Sgd: General analysis and improved rates," in *Int. Conf. on Machine Learning*. PMLR, 2019, pp. 5200–5209.

[34] Y. Dvorkin *et al.*, "Uncertainty sets for wind power generation," *IEEE Trans. Power Sys.*, vol. 31, no. 4, pp. 3326–3327, 2015.

[35] R. Mieth *et al.*, "Data valuation from data-driven optimization," *arXiv:2305.01775*, 2023.

[36] R. T. Rockafellar *et al.*, "Optimization of conditional value-at-risk," *Journal of Risk*, vol. 2, pp. 21–42, 2000.

[37] L. Zhang *et al.*, "Solving stochastic optimization with expectation constraints efficiently by a stochastic augmented lagrangian-type algorithm," *INFORMS Journal on Computing*, vol. 34, no. 6, pp. 2989–3006, 2022.

[38] A. Agrawal *et al.*, "Differentiating through a cone program," *arXiv:1904.09043*, 2019.

[39] A. Domahidi *et al.*, "ECOS: An SOCP solver for embedded systems," in *Proc. of the European Control Conference*, 2013, pp. 3071–3076.

[40] B. Wilder *et al.*, "Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization," in *Proc. of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019.

[41] MATPOWER. (2014) CASE5 Power flow data. [Online]. Available: https://matpower.org/docs/ref/matpower5.0/case5.html

[42] R. Mieth. (2023) Prescribed Robust DC-OPF Code Supplement. [Online]. Available: https://github.com/mieth-robert/p-robust_dcopf

[43] NYISO, "Ancillary services shortage pricing," NYISO, Tech. Rep., 2019.

[44] Reliability Test System - Grid Modernization Lab Consortium. [Online]. Available: github.com/GridMod/RTS-GMLC

[45] C. Draxl *et al.*, "Overview and meteorological validation of the wind integration national dataset toolkit," National Renewable Energy Lab. (NREL), Golden, CO (United States), Tech. Rep., 2015.

[46] U. Sadana *et al.*, "A survey of contextual optimization methods for decision making under uncertainty," *arXiv:2306.10374*, 2023.

[47] J. Kotary *et al.*, "Folded optimization for end-to-end model-based learning," *arXiv:2301.12047*, 2023.