Improving the Accuracy of DC Optimal Power Flow Formulations via Parameter Optimization

Babak Taheri and Daniel K. Molzahn

***Abstract*—DC Optimal Power Flow (DC-OPF) problems optimize the generators' active power setpoints while satisfying constraints based on the DC power flow linearization. The computational tractability advantages of DC-OPF problems come at the expense of inaccuracies relative to AC Optimal Power Flow (AC-OPF) problems which accurately model the nonlinear steady-state behavior of power grids. This paper proposes an algorithm that significantly improves the accuracy of the generators' active power setpoints from DC-OPF problems with respect to the corresponding AC-OPF problems over a specified range of operating conditions. Using sensitivity information in a machine learning-inspired methodology, this algorithm tunes coefficient and bias parameters in the DC power flow approximation to improve the accuracy of the resulting DC-OPF solutions. Employing the Truncated Newton Conjugate-Gradient (TNC) method—a Quasi-Newton optimization technique—this parameter tuning occurs during an offline training phase, with the resulting parameters then used in online computations. Numerical results underscore the algorithm's efficacy with accuracy improvements in squared two-norm and ∞-norm losses of up to 90% and 79%, respectively, relative to traditional DC-OPF formulations.**

***Index Terms*—DC optimal power flow (DC-OPF), AC-OPF, machine learning, parameter optimization.**

NOMENCLATURE

$\mathcal{N}$	Set of buses
$\mathcal{E}$	Set of branches
$\mathbf{j}$	Imaginary unit $\mathbf{j} = \sqrt{-1}$
$\mathbf{s}^{\mathrm{d}}$	Complex power demand; $\mathbf{s}^{\mathrm{d}} = \mathbf{p}^{\mathrm{d}} + \mathbf{j}\mathbf{q}^{\mathrm{d}}$
Y^s	Bus shunt admittance
Y	Complex branch line admittance
Y^c	Complex branch shunt admittance
$\bar{s}$	Thermal branch limit
$\underline{\mathrm{s}}^{\mathrm{g}}, \overline{\mathrm{s}}^{\mathrm{g}}$	Complex power generation bounds
$\underline{\mathrm{V}}, \overline{\mathrm{V}}$	Voltage magnitude bounds
$\mathbf{z}$	Branch impedance; $\mathbf{z}{=}\mathbf{r} + \mathbf{j}\mathbf{x}$
$\mathbf{A}$	Branch-bus incidence matrix
$\mathbf{s}^{\mathrm{g}}$	Complex power generation; $\mathbf{s}^{\mathrm{g}}{=}\mathbf{p}^{\mathrm{g}} + \mathbf{j}\mathbf{q}^{\mathrm{g}}$
$\mathbf{S}$	Complex power flow; $\mathbf{S} = \mathbf{P} + \mathbf{j}\mathbf{Q}$
$\mathbf{V}$	Complex voltage; $\mathbf{V} = \lvert\mathbf{V}\rvert\angle\boldsymbol{\theta}$
c	Active power generation cost vector
$\mathbf{b}$	DC power flow coefficient parameter vector
$\boldsymbol{\gamma}, \boldsymbol{\rho}$	DC power flow bias parameter vectors

I. INTRODUCTION

Optimal Power Flow (OPF) is a fundamental tool for power system design and operation. OPF problems optimize system performance (e.g., minimizing generation costs or reducing voltage deviations) while satisfying both equality constraints from a power flow model and inequality constraints from operational limits. In addition to significant economic implications [1], OPF problems are at the heart of many algorithms for addressing uncertainty [2], planning infrastructure investments [3], improving stability margins [4], etc.

The AC power flow equations accurately model the steady-state behavior of power systems by relating the complex power injections and line flows with the voltage phasors. This motivates solution of AC Optimal Power Flow (AC-OPF) problems that incorporate the AC power flow equations. However, AC-OPF problems are computationally challenging, with these problems being non-convex and NP-hard in the worst-case [5]–[7]. These challenges are exacerbated for extensions of OPF problems that consider discrete characteristics, such as those related to transmission switching [8], [9] and unit commitment [10], as well as uncertainties due to variable renewable energy generation and flexible loads [2].

Since first being formulated by Carpentier in 1962 [11], OPF algorithms have undergone continual development [12]–[15]. Most existing methods seek a local optimum due to the nonconvex nature of OPF problems that results from the AC power flow equations. Machine learning (ML) techniques are emerging as tools to address OPF challenges, offering computational speedups but often lacking interpretability and consistency with physical intuition [16].

Due to these challenges, engineers often employ power flow approximations [17] to obtain more tractable convex OPF formulations, sometimes by optimizing the linearization point to account for supply and demand uncertainty [18]. Using a common linear approximation known as the DC power flow [19] yields the DC Optimal Power Flow (DC-OPF) problem, which is widely used for both short- and long-term planning purposes. However, the DC-OPF problem's tractability advantages come with the drawback of inaccuracies relative to the AC-OPF problem. DC power flow approximations linearize the AC power flow equations by assuming a flat voltage profile, applying a small-angle approximation of the sine function, and neglecting reactive power flows and network losses, thus introducing approximation errors when these assumptions do not hold. Inaccuracies from these assumptions are particularly pronounced when reactive power flows and voltage variations play a significant role, such as in heavily loaded systems or systems with high penetrations of renewable generators. In such cases, the DC-OPF problem may not adequately capture the complex behavior of the power system, leading to suboptimal or infeasible decisions in operational and planning contexts [8], [20]–[22].

These limitations motivate enhancements to the DC-OPF

B. Taheri and D.K. Molzahn are with the School of Electrical and Computer Engineering, Georgia Institute of Technology. {btaheri6, molzahn}@gatech.edu. Support from NSF award #2145564.

problem that aim to improve accuracy while maintaining tractability. The DC power flow approximation is foundational to the DC-OPF problem. This approximation relates active power flow between buses i and j, denoted p_{ij}, to the phase angle difference $\theta_i - \theta_j$ through a proportionality coefficient b_{ij}: $p_{ij} = b_{ij}(\theta_i - \theta_j)$. While typically derived from line parameters such as resistance r_{ij} and reactance x_{ij} via various heuristics, tailoring these parameter values to specific system and operating range characteristics can yield significant accuracy improvements. Common heuristics choose $b_{ij} = 1/x_{ij}$ or $b_{ij} = -\Im(1/(r_{ij} + jx_{ij}))$, where $\Im(\cdot)$ extracts the imaginary part of a complex argument. When resistance is nonzero ($r_{ij} \neq 0$), these choices for b_{ij} produce slightly different results, impacting the accuracy of the DC power flow approximation. Selecting the optimal b_{ij} for a given application is not straightforward and depends on system characteristics and objectives [19].

Additionally, the introduction of bias parameters allows further flexibility by adjusting power injections and flows to account for shunts, HVDC infeeds, phase shifts, and line losses [19]. Two variants of the DC power flow—cold-start and hot-start—differ in their reliance on prior information when determining the b_{ij} coefficients and bias parameters.

Our previous work in [23] introduced an algorithm that optimizes both the coefficient and bias parameters to improve DC power flow accuracy across a range of operating conditions. This optimization selects the DC power flow parameter values which minimize the mismatch between the line flows from the DC and AC power flow models over a training set of scenarios.

While this previous paper focused on optimizing the coefficients and biases within the DC power flow approximation itself [23], specialization of the parameters to specific applications can yield even larger accuracy advantages. Accordingly, this paper extends the concepts from [23] to optimize parameters for the DC power flow approximation in order to improve accuracy of DC-OPF solutions relative to AC-OPF solutions (as opposed to the focus on DC power flow versus AC power flow in our prior work). Specifically, we seek the DC power flow parameters which best reduce the discrepancies of the generator active power setpoints associated with the DC-OPF solutions relative to those from the AC-OPF solutions across a range of load demands. This introduces new challenges since the loss function considers the optimal operational decisions from the DC-OPF problem, resulting in a bilevel structure.

To address these challenges, this paper introduces a new algorithm for the adaptive selection of coefficients b_{ij} and bias parameters within the context of the DC-OPF problem. Building on the approach from our previous work [23], we focus on enhancing the DC-OPF model by leveraging machine learning-inspired techniques for parameter optimization. The algorithm performs offline fine-tuning of b_{ij} and bias parameters using a large training dataset of AC-OPF solutions under various operating conditions. Leveraging differentiable optimization techniques originally developed for deep learning architectures [24], [25], we compute the sensitivities of the DC-OPF solution with respect to these parameters. This enables us to iteratively adjust the parameter values during an offline computation phase by using a gradient-based optimization method, specifically the Truncated Newton Conjugate-Gradient (TNC) algorithm, to minimize the difference between DC-OPF's generator active power setpoints and their AC-OPF counterparts across the training dataset. The resulting optimized parameter values are then applied to improve the accuracy of the DC-OPF problems during online computations.

To summarize, the main contributions of this paper are:

1) Introducing an optimization algorithm designed to adaptively choose the coefficients b_{ij} and bias parameters of the DC power flow inside the DC-OPF problem. These parameters are instrumental in modeling line losses and incorporating factors such as shunts, HVDC infeeds, phase shifts, and line losses.
2) Computing gradients for the DC-OPF solution with respect to both coefficient and bias parameters to obtain sensitivities of the generator set points as well as our defined loss function.
3) Utilizing a quasi-Newton method (TNC) to scale our proposed algorithm to large power systems.
4) Providing numerical results that demonstrate the superior accuracy of our proposed algorithm over a range of operating conditions.

The rest of the paper is structured as follows. Section II reviews the AC-OPF and DC-OPF problems. Section III presents our algorithm for optimizing parameters of DC-OPF problems. Section IV provides numerical results to demonstrate the algorithm's performance. Section V concludes the paper.

II. Optimal Power Flow Problem

To introduce notation and necessary background material, this section describes the AC-OPF formulation and the DC-OPF linear approximation that simplifies the AC-OPF problem to improve tractability at the cost of accuracy. Buses, lines and generators in the network are denoted by the sets $\mathcal{N}$, $\mathcal{E}$ and $\mathcal{G}$, respectively. Each bus $i \in \mathcal{N}$ has a voltage phasor V_i with phase angle θ_i, along with a shunt admittance Y_i^S, complex power demand $s_i^{\mathrm{d}} = \mathrm{p}_i^{\mathrm{d}} + \mathbf{j}\mathrm{q}_i^{d}$, and generated complex power $s_i^{\mathrm{g}} = p_i^{\mathrm{g}} + \mathbf{j}q_i^{\mathrm{g}}$, where $\mathbf{j}$ is the imaginary unit $\mathbf{j} = \sqrt{-1}$. Buses without generators are modeled as having upper and lower generation limits of zero. The complex power flows entering each end of line $(j,k) \in \mathcal{E}$ are represented by S_{jk} and S_{kj}. Series admittance parameters for each line (j,k) are Y_{jk} and Y_{kj} and the line's shunt admittance is Y_{jk}^c. The real and imaginary components of a complex number are indicated by $\Re(\,\cdot\,)$ and $\Im(\,\cdot\,)$, respectively, $\|\cdot\|_\infty$ denotes the L_∞-norm, and $\|\cdot\|_2$ denotes the L_2-norm. The complex conjugate and the transpose of a matrix are denoted by $(\,\cdot\,)^\star$ and $(\,\cdot\,)^\top$, respectively, and $\angle(\,\cdot\,)$ is the angle of a complex argument. Upper and lower limits are represented by $(\overline{\cdot})$ and $(\underline{\cdot})$.

A. The AC-OPF Problem

Model 1 presents the AC-OPF formulation. The objective (1a) minimizes the total generation cost by summing the costs for each generator $i \in \mathcal{N}$, modeled as quadratic functions (with coefficients c_{2i}, c_{1i}, and c_{0i}) of the real part of the complex power generation s_i^{g}. Constraint (1b) ensures that all voltage magnitudes are within specified limits. Constraint (1c) bounds the complex power outputs of each generator within its feasible operating range, where inequalities on complex

Model 1 The AC-OPF Problem

$$
\begin{aligned}
\min_{s_i^{\mathrm{g}}, V_i} \quad & \sum_{i \in \mathcal{N}} c_{2i}(\Re(s_i^{\mathrm{g}}))^2 + c_{1i}\Re(s_i^{\mathrm{g}}) + c_{0i} && \text{(1a)} \\
\text{s.t.} \quad & (\forall i \in \mathcal{N},\ \forall (j,k) \in \mathcal{E}) && \\
& \underline{\mathrm{V}}_i \leq |V_i| \leq \overline{\mathrm{V}}_i && \text{(1b)} \\
& \underline{s}_i^{\mathrm{g}} \leq s_i^{\mathrm{g}} \leq \overline{s}_i^{\mathrm{g}} && \text{(1c)} \\
& |S_{jk}| \leq \overline{S}_{jk},\ \ |S_{kj}| \leq \overline{S}_{jk} && \text{(1d)} \\
& s_i^{\mathrm{g}} - s_i^{\mathrm{d}} - (Y_i^s)^\star |V_i|^2 = \sum_{(i,j)\in\mathcal{E}} S_{ij} + \sum_{(k,i)\in\mathcal{E}} S_{ki} && \\
& && \text{(1e)} \\
& S_{jk} = (Y_{jk} + Y_{jk}^c)^\star V_j V_j^\star - Y_{jk}^\star V_j V_k^\star && \text{(1f)} \\
& S_{kj} = (Y_{kj} + Y_{kj}^c)^\star V_k V_k^\star - Y_{kj}^\star V_k V_j^\star && \text{(1g)} \\
& \theta_{ref} = 0 && \text{(1h)} \\
& -\overline{\theta}_{jk} \leq \angle(V_j V_k^\star) \leq \overline{\theta}_{jk} && \text{(1i)} \\
\textbf{variables}: \ & s_i^{\mathrm{g}}\ (\forall i \in \mathcal{N}), V_i\ (\forall i \in \mathcal{N}) &&
\end{aligned}
$$

quantities are interpreted as bounds on the real and imaginary parts. Constraint (1d) ensures that the power flows into each line terminal do not exceed the line's thermal limits.

The constraint (1e) ensures power balance for the power generated, consumed, and flowing through the network at each bus. Constraints (1f) and (1g) represent the power flow in each line of the network. The voltage angle at the reference bus is set to zero in (1h). Finally, constraint (1i) limits the phase angle difference across each line (often used as a proxy for transient stability). The AC-OPF problem is inherently nonlinear and non-convex due to the quadratic nature of the power flow equations. Using nonlinear optimization techniques such as interior-point methods, AC-OPF algorithms frequently find high-quality solutions [12]–[15].

B. The DC-OPF Problem

The DC-OPF problem is a linear approximation of the AC-OPF problem that assumes uniform voltage magnitudes (typically normalized to one per-unit), small voltage angle differences, negligible losses, and ignores reactive power [19]. These assumptions are most relevant to transmission systems. The DC-OPF model underlies various applications within electricity markets and is widely used in unit commitment, transmission network expansion planning, and optimal transmission switching problems [9], [22], [26]–[29].

Model 2 presents the DC-OPF problem. Analogous to (1a), the objective (2a) minimizes the total generation cost by summing quadratic functions of the generators' active power outputs. Similarly, (2b) bounds the generators' active power outputs, reminiscent of the dispatch bounds (1c). Analogous to (1d), constraint (2c) enforces thermal limits on transmission lines. Analogous to (1e), constraint (2d) balances active power at each bus by equating the net power injection with the net branch power flow. Constraint (2e) linearizes the line flow expressions by relating branch power flows to voltage angle differences according to the DC power flow approximation, thus offering a simplified counterpart to (1f) and (1g). Finally, (2f) sets the reference bus angle to zero. Note that

Model 2 The DC-OPF Problem

$$
\begin{aligned}
\min_{p_i^{\mathrm{g}}, \theta_i} \quad & \sum_{i \in \mathcal{N}} c_{2i}(p_i^{\mathrm{g}})^2 + c_{1i} p_i^{\mathrm{g}} + c_{0i} && \text{(2a)} \\
\text{s.t.} \quad & (\forall i \in \mathcal{N},\ \forall (j,k) \in \mathcal{E}) && \\
& \underline{\mathrm{p}}_i^{\mathrm{g}} \leq p_i^{\mathrm{g}} \leq \overline{\mathrm{p}}_i^{\mathrm{g}} && \text{(2b)} \\
& |P_{jk}| \leq \bar{S}_{jk} && \text{(2c)} \\
& p_i^{\mathrm{g}} - \mathrm{p}_i^{\mathrm{d}} - \gamma_i = \sum_{(i,j)\in\mathcal{E}} P_{ij} + \sum_{(k,i)\in\mathcal{E}} P_{ki} && \text{(2d)} \\
& P_{jk} = b_{jk}(\theta_k - \theta_j) + \rho_{jk} && \text{(2e)} \\
& \theta_{ref} = 0 && \text{(2f)} \\
\textbf{variables}: \ & p_i^{\mathrm{g}}\ (\forall i \in \mathcal{N}), \theta_i\ (\forall i \in \mathcal{N}) &&
\end{aligned}
$$

Model 3 The DC-OPF-M Problem

$$
\begin{aligned}
\min_{\mathbf{p}^{\mathrm{g}}, \boldsymbol{\theta}} \quad & {\mathbf{p}^{\mathrm{g}}}^\top \mathrm{diag}(\mathbf{c}_2)\mathbf{p}^{\mathrm{g}} + \mathbf{c}_1^\top \mathbf{p}^{\mathrm{g}} + \sum_{i \in \mathcal{N}} c_{0i} && \text{(3a)} \\
\text{s.t.} \quad & && \\
& \underline{\mathbf{p}}^{\mathrm{g}} \leq \mathbf{p}^{\mathrm{g}} \leq \overline{\mathbf{p}}^{\mathrm{g}} && \text{(3b)} \\
& \mathbf{p}^{\mathrm{g}} - \mathbf{p}^{\mathrm{d}} - \boldsymbol{\gamma} = \mathbf{A}^\top \Big(\mathrm{diag}(\mathbf{b})\mathbf{A}\boldsymbol{\theta} + \boldsymbol{\rho} \Big) && \text{(3c)} \\
& |\mathrm{diag}(\mathbf{b})\mathbf{A}\boldsymbol{\theta} + \boldsymbol{\rho}| \leq \bar{\mathbf{S}} && \text{(3d)} \\
& \theta_{ref} = 0 && \text{(3e)} \\
\textbf{variables}: \ & \mathbf{p}^{\mathrm{g}}, \boldsymbol{\theta} &&
\end{aligned}
$$

(2d) and (2e) include parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\rho}$. The parameter γ_i accounts for losses from shunts, HVDC infeeds, and injections modeling phase shifts and branch losses for lines connected to bus i. The parameter ρ_{ij} account for losses in line $(i,j) \in \mathcal{E}$.

Model 3 equivalently rewrites Model 2 in a more notationally friendly matrix form. In Model 3, we define $\mathbf{A}$ as the $|\mathcal{E}| \times |\mathcal{N}|$ incidence matrix describing the connections between the system's buses and branches and diag($\mathbf{b}$) as a diagonal matrix having the entries of $\mathbf{b}$ on its diagonal.

The DC-OPF problem's accuracy relative to the AC-OPF problem is dictated by the parameters $\mathbf{b}$, $\boldsymbol{\gamma}$, and $\boldsymbol{\rho}$. Traditionally, there are two DC-OPF versions, cold-start and hot-start, that rely on different amounts of prior information to select these parameters.

1) Cold-start DC-OPF: In this version, the coefficient and bias parameters are selected without relying on a nominal operating point. For instance, the coefficient values b_{ij} can be selected as the imaginary part of the line susceptance:

$$
b_{ij}^{cold} = \Im\left(\frac{-1}{r_{ij} + \mathbf{j}x_{ij}}\right). \tag{4}
$$

We note that the alternative selection of $b_{ij}^{cold} = 1/x_{ij}$ yields very similar numerical results to the choice of b_{ij}^{cold} in (4), so we only present results for the latter for the sake of brevity.

The cold-start version assigns values of zero to the bias parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\rho}$. The simplicity provided by the cold-start DC-OPF problem comes at the cost of accuracy in settings where information on a nominal operating point is available.

2) Hot-start DC-OPF: Hot-start DC-OPF problems leverage information from a nominal AC power flow solution to improve accuracy. There are many hot-start DC-OPF variants [19]; for example, the "localized loss modeling" variant of the hot-start DC power flow model in [19] selects the coefficient and bias parameters as:

$$b_{ij}^{hot} = b_{ij} v_i^\bullet v_j^\bullet \sin(\theta_i^\bullet - \theta_j^\bullet)/(\theta_i^\bullet - \theta_j^\bullet), \tag{5a}$$

$$\gamma_i^{hot} = \sum_{(i,j)\in\mathcal{E}} \Re(Y_{ij}) v_i^\bullet (v_i^\bullet - v_j^\bullet \cos(\theta_i^\bullet - \theta_j^\bullet)), \tag{5b}$$

$$\rho_{ij}^{hot} = \Re(Y_{ij}) v_i^\bullet (v_i^\bullet - v_j^\bullet \cos(\theta_i^\bullet - \theta_j^\bullet)), \tag{5c}$$

where the notation $(\,\cdot\,)^\bullet$ signifies quantities associated with a nominal AC power flow solution. The bias $\boldsymbol{\gamma}^{hot}$ aims to capture the effects of branch losses on phase angles and $\boldsymbol{\rho}^{hot}$ incorporates the branch losses in the line flow expressions.

3) Optimized Parameters for DC Power Flow (DCPF) [23]: Our recent work in [23] proposes an alternative for selecting the $\mathbf{b}$, $\boldsymbol{\gamma}$, and $\boldsymbol{\rho}$ parameters by solving an optimization problem. Providing a conceptual foundation for the approach we will propose in this paper, the algorithm in [23] also optimizes these parameters over a range of operation. However, there is a key distinction between the algorithm in [23] and the parameter optimization algorithm in this paper. Specifically, the approach in [23] does not explicitly consider the impacts of the optimized parameters on the solution to a DC-OPF problem, but rather seeks to balance the DC power flow's accuracy with respect to all operating points within a specified range. Conversely, our proposed algorithm in this paper leverages machine learning techniques for training models with differentiable optimization layers to tailor the DC power flow parameter values so that the DC *optimal* power flow solutions align with the AC-OPF solutions. As shown numerically in Section IV, this enables further accuracy improvements relative to [23] in settings where the optimized DC power flow parameters are used in optimization problems like DC-OPF.

We next introduce our machine learning-inspired algorithm for optimizing the coefficient ($\mathbf{b}$) and bias ($\boldsymbol{\gamma}$ and $\boldsymbol{\rho}$) parameters. Our algorithm seeks to match the generators' active power outputs from the DC-OPF solution to corresponding values from the AC-OPF problem.

III. Parameter Optimization Algorithm

Our parameter optimization algorithm, depicted in Fig. 1, has offline and online phases. The offline phase computes DC-OPF parameters ($\mathbf{b}$, $\boldsymbol{\gamma}$, and $\boldsymbol{\rho}$) to achieve a high-fidelity approximation of the AC-OPF solution over a range of operational conditions. The online phase employs these parameters to efficiently solve DC-OPF problems in various applications. While our numerical results focus on DC-OPF problems, the optimized DC-OPF parameters are applicable to many computationally intensive optimization problems that incorporate DC power flow models, such as unit commitment [9], [26], expansion planning [3], and grid reliability enhancement [30].

Our algorithm minimizes a loss function that quantifies the discrepancy between DC-OPF and AC-OPF generator active power setpoints over various load scenarios. To perform this minimization, we compute the loss function's gradients with respect to the parameters $\mathbf{b}$, $\boldsymbol{\gamma}$, and $\boldsymbol{\rho}$ for use in a nonlinear optimization method. Based on our numerical tests comparing various nonlinear optimization methods, the TNC optimization method best used these gradients to achieve scalable and effective parameter optimization. TNC's capability to approximate the Hessian matrix from gradient information ensures efficient handling of large-scale power systems.

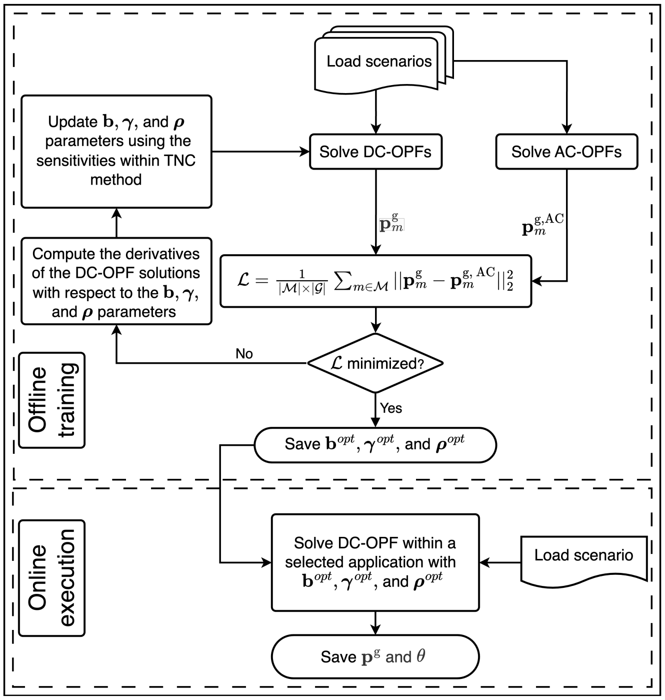

Load scenarios
Update $\mathbf{b}, \boldsymbol{\gamma}$, and $\boldsymbol{\rho}$ parameters using the sensitivities within TNC method
Solve DC-OPFs
Solve AC-OPFs
$\mathbf{p}_m^{\mathrm{g}}$
$\mathbf{p}_m^{\mathrm{g,AC}}$
Compute the derivatives of the DC-OPF solutions with respect to the $\mathbf{b}, \boldsymbol{\gamma}$, and $\boldsymbol{\rho}$ parameters
$\mathcal{L} = \frac{1}{|\mathcal{M}|\times|\mathcal{G}|}\sum_{m\in\mathcal{M}}||\mathbf{p}_m^{\mathrm{g}} - \mathbf{p}_m^{\mathrm{g,\,AC}}||_2^2$
No
$\mathcal{L}$ minimized?
Offline training
Yes
Save $\mathbf{b}^{opt}, \boldsymbol{\gamma}^{opt}$, and $\boldsymbol{\rho}^{opt}$
Online execution
Solve DC-OPF within a selected application with $\mathbf{b}^{opt}, \boldsymbol{\gamma}^{opt}$, and $\boldsymbol{\rho}^{opt}$
Load scenario
Save $\mathbf{p}^{\mathrm{g}}$ and θ

Fig. 1. Flowchart of the proposed algorithm.

A. Loss Function

We first introduce a loss function $\mathcal{L}$ that is the sum of squared two-norm discrepancies between the generators' active power setpoints obtained from the AC-OPF problems ($\mathbf{p}_m^{\mathrm{g,AC}}$) and the DC-OPF problems ($\mathbf{p}_m^{\mathrm{g}}$) across a specified set of load scenarios $\mathcal{M}$:

$$\mathcal{L}(\mathbf{b}, \boldsymbol{\gamma}, \boldsymbol{\rho}) = \frac{1}{|\mathcal{G}| \times |\mathcal{M}|} \sum_{m\in\mathcal{M}} ||\mathbf{p}_m^{\mathrm{g}}(\mathbf{b}, \boldsymbol{\gamma}, \boldsymbol{\rho}) - \mathbf{p}_m^{\mathrm{g,AC}}||_2^2, \tag{6}$$

where the constant $\frac{1}{|\mathcal{G}|\times|\mathcal{M}|}$ normalizes this function based on the number of generators and load scenarios. As shown in (6), $\mathbf{p}_m^{\mathrm{g}}$ (and thus $\mathcal{L}(\mathbf{b}, \boldsymbol{\gamma}, \boldsymbol{\rho})$) is a function of the coefficient parameters $\mathbf{b}$ and the bias parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\rho}$.

We note that this two-norm loss formulation is typical in machine learning for its robustness and differentiability. Moreover, this loss function penalizes larger deviations more heavily, which is well aligned with typical applications where a small number of severe approximation errors would be more problematic than a large number of minor errors. One could instead use other norms without major conceptual changes.

With this loss function, the optimization problem to find the best coefficient and bias parameters is formulated as:

$$\min_{\mathbf{b}, \boldsymbol{\gamma}, \boldsymbol{\rho}} \quad \mathcal{L}(\mathbf{b}, \boldsymbol{\gamma}, \boldsymbol{\rho}). \tag{7}$$

Using the sensitivity analysis which we describe in the next subsection, we adjust the coefficient and bias parameters ($\mathbf{b}$,

Model 4 Optimizing DC-OPF Problem

$$\min_{\mathbf{b},\boldsymbol{\gamma},\boldsymbol{\rho}} \quad \mathcal{L} = \frac{1}{|\mathcal{G}| \times |\mathcal{M}|} \sum_{m\in\mathcal{M}} ||\mathbf{p}^{\mathrm{g}}_m(\mathbf{b},\boldsymbol{\gamma},\boldsymbol{\rho}) - \mathbf{p}^{\mathrm{g,AC}}_m||_2^2 \tag{8a}$$

s.t.

$$\min_{\mathbf{p}^{\mathrm{g}}_m,\boldsymbol{\theta}_m} \quad {\mathbf{p}^{\mathrm{g}}_m}^\top \mathrm{diag}(\mathbf{c}_2)\mathbf{p}^{\mathrm{g}}_m + \mathbf{c}_1^\top \mathbf{p}^{\mathrm{g}}_m + \sum_{i\in\mathcal{G}} c_{0i} \tag{8b}$$

s.t.

$$\mathbf{p}^{\mathrm{g}}_m - \mathbf{p}^{\mathrm{d}}_m - \boldsymbol{\gamma} = \mathbf{A}^\top \Big(\mathrm{diag}(\mathbf{b})\mathbf{A}\boldsymbol{\theta}_m + \boldsymbol{\rho}\Big) \tag{8c}$$

$$|\mathrm{diag}(\mathbf{b})\mathbf{A}\boldsymbol{\theta}_m + \boldsymbol{\rho}| \leq \bar{\mathbf{S}} \tag{8d}$$

$$\underline{\mathbf{p}}^{\mathrm{g}} \leq \mathbf{p}^{\mathrm{g}}_m \leq \bar{\mathbf{p}}^{\mathrm{g}} \tag{8e}$$

$$\theta_{ref} = 0 \tag{8f}$$

$$\forall m \in \mathcal{M}$$

$\boldsymbol{\gamma}$, and $\boldsymbol{\rho}$) to optimize the DC-OPF problem's parameters. As detailed in Model 4, this task takes the form of a bilevel optimization problem, with the upper-level fine-tuning parameters based on outcomes from lower-level DC-OPF problems under a variety of operational scenarios.

B. Sensitivity Analysis of Coefficient and Bias Parameters

This subsection describes the sensitivity analyses crucial for our parameter optimization algorithm. We utilize a gradient-based method, TNC, which requires the gradient of the loss function with respect to $\mathbf{b}$, $\boldsymbol{\gamma}$, and $\boldsymbol{\rho}$. These gradients, denoted as $\mathbf{g}^b$, $\mathbf{g}^\gamma$, and $\mathbf{g}^\rho$, are computed as follows:

$$\mathbf{g}^b = \frac{\partial \mathcal{L}}{\partial \mathbf{b}} = \frac{2}{|\mathcal{G}| \times |\mathcal{M}|} \sum_{m\in\mathcal{M}} \left.\frac{\partial \mathbf{p}^{\mathrm{g}}}{\partial \mathbf{b}}\right|_{\mathbf{p}^{\mathrm{g}}_m} \left(\mathbf{p}^{\mathrm{g}}_m - \mathbf{p}^{\mathrm{g},AC}_m\right), \tag{9a}$$

$$\mathbf{g}^\gamma = \frac{\partial \mathcal{L}}{\partial \boldsymbol{\gamma}} = \frac{2}{|\mathcal{G}| \times |\mathcal{M}|} \sum_{m\in\mathcal{M}} \left.\frac{\partial \mathbf{p}^{\mathrm{g}}}{\partial \boldsymbol{\gamma}}\right|_{\mathbf{p}^{\mathrm{g}}_m} \left(\mathbf{p}^{\mathrm{g}}_m - \mathbf{p}^{\mathrm{g},AC}_m\right), \tag{9b}$$

$$\mathbf{g}^\rho = \frac{\partial \mathcal{L}}{\partial \boldsymbol{\rho}} = \frac{2}{|\mathcal{G}| \times |\mathcal{M}|} \sum_{m\in\mathcal{M}} \left.\frac{\partial \mathbf{p}^{\mathrm{g}}}{\partial \boldsymbol{\rho}}\right|_{\mathbf{p}^{\mathrm{g}}_m} \left(\mathbf{p}^{\mathrm{g}}_m - \mathbf{p}^{\mathrm{g},AC}_m\right). \tag{9c}$$

The derivatives $\frac{\partial \mathbf{p}^{\mathrm{g}}}{\partial \mathbf{b}}$, $\frac{\partial \mathbf{p}^{\mathrm{g}}}{\partial \boldsymbol{\gamma}}$, and $\frac{\partial \mathbf{p}^{\mathrm{g}}}{\partial \boldsymbol{\rho}}$ are computed from the KKT conditions of the lower-level DC-OPF problems (8b)–(8f) in Model 4. Differentiating through convex optimization problems like DC-OPF involves implicit differentiation of their optimality conditions [24], [31], [32]. To perform these implicit derivative computations, we leverage techniques developed for embedding convex optimization layers within deep learning architectures; specifically, we utilize `cvxpylayers` [25], which allows for automatic differentiation through convex optimization problems. This integration facilitates efficient and scalable gradient computations necessary for our optimization framework. In detail, we define the DC-OPF problem using CVXPY's modeling syntax, specifying the objective function and constraints as per Model 3. This CVXPY model is then converted into a differentiable layer using `cvxpylayers`, enabling it to be embedded within our optimization pipeline. During the forward pass, cvxpylayer solves the DC-OPF problem to obtain $\mathbf{p}^{\mathrm{g}}$. In the backward pass, cvxpylayer automatically computes the derivatives $\frac{\partial \mathbf{p}^{\mathrm{g}}}{\partial \mathbf{b}}$, $\frac{\partial \mathbf{p}^{\mathrm{g}}}{\partial \boldsymbol{\gamma}}$, and $\frac{\partial \mathbf{p}^{\mathrm{g}}}{\partial \boldsymbol{\rho}}$, leveraging automatic differentiation provided by `cvxpylayers`; we refer the reader to [24], [25] for further details and efficient computational implementations.

Algorithm 1: Truncated Newton (TNC) Method

```
Input: $\mathbf{x}_0 = [\mathbf{b}_0^\top, \boldsymbol{\gamma}_0^\top, \boldsymbol{\rho}_0^\top]^\top$: Initial guess
       $\epsilon$: Tolerance for convergence
       max_iter: Maximum iterations
       $\mathcal{L}(\mathbf{x}_k)$: Loss function
       $\nabla\mathcal{L}(\mathbf{x}_k) = \mathbf{g}$
       $\mathbf{M}$: Preconditioning matrix (often diagonal)
       $\mathbf{H}$: Hessian or its approximation function
       $\alpha_1$: Armijo condition constant (e.g., $10^{-4}$)
       $\alpha_2$: Curvature condition constant, between $\alpha_1$ and 1
Output: Optimized parameters $\mathbf{x}^*$
Initialize $\mathbf{x}_k \leftarrow \mathbf{x}_0$, $k \leftarrow 0$
while $k \leq$ max_iter and $\|\nabla\mathcal{L}(\mathbf{x}_k)\| > \epsilon$ do
    $\mathbf{g} \leftarrow \nabla\mathcal{L}(\mathbf{x}_k)$
    $\mathbf{r} \leftarrow -\mathbf{g}$
    $\mathbf{z} \leftarrow \mathbf{M}^{-1}\mathbf{r}$
    $\mathbf{p} \leftarrow \mathbf{z}$
    $\mathbf{s} \leftarrow \mathbf{0}$
    $\rho_{old} \leftarrow \mathbf{r}^\top\mathbf{z}$
    while $\|\mathbf{r}\| > \epsilon$ do
        $\mathbf{q} \leftarrow \mathbf{H}\mathbf{p}$
        $\alpha \leftarrow \frac{\rho_{old}}{\mathbf{p}^\top\mathbf{q}}$
        $\mathbf{s} \leftarrow \mathbf{s} + \alpha\mathbf{p}$
        $\mathbf{r} \leftarrow \mathbf{r} - \alpha\mathbf{q}$
        $\mathbf{z} \leftarrow \mathbf{M}^{-1}\mathbf{r}$
        $\rho_{new} \leftarrow \mathbf{r}^\top\mathbf{z}$
        $\eta \leftarrow \frac{\rho_{new}}{\rho_{old}}$
        $\mathbf{p} \leftarrow \mathbf{z} + \eta\mathbf{p}$
        $\rho_{old} \leftarrow \rho_{new}$
        // Wolfe Line Search to determine $\beta$
    $\beta \leftarrow 1$
    while True do
        if $\mathcal{L}(\mathbf{x}_k + \beta\mathbf{s}) \leq \mathcal{L}(\mathbf{x}_k) + \alpha_1\beta\mathbf{g}^\top\mathbf{s}$ and
          $\|\nabla\mathcal{L}(\mathbf{x}_k + \beta\mathbf{s})^\top\mathbf{s}\| \leq \alpha_2\|\mathbf{g}^\top\mathbf{s}\|$ then
            break
        $\beta \leftarrow \beta/2$
    $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \beta\mathbf{s}$
    $k \leftarrow k + 1$
$\mathbf{x}^* \leftarrow \mathbf{x}_k$
```

Next, we will use the collected gradient information defined as $\nabla\mathcal{L}(\mathbf{b},\boldsymbol{\gamma},\boldsymbol{\rho}) = \mathbf{g} = [{\mathbf{g}^b}^\top, {\mathbf{g}^\gamma}^\top, {\mathbf{g}^\rho}^\top]^\top$ in the TNC optimization method to update the coefficient and bias parameters.

C. Optimization Formulation and Solution Method

To solve Model 4, we employ the gradient-based TNC optimization method that uses the calculated parameter sensitivities [33], [34]. TNC combines the advantages of Newton's method for fast convergence with the conjugate gradient approach to efficiently handle large-scale problems without the need for explicit Hessian computations. Compared to alternatives such as Broyden-Fletcher-Goldfarb-Shanno (BFGS) and limited-memory BFGS [34], the TNC method demonstrates superior performance in speed and reliability for our purposes.

As shown in Algorithm 1, the TNC method iteratively updates the solution vector by approximating the Newton direction using truncated conjugate gradients with preconditioning. The process begins with an initial guess for the solution vector, followed by the preparation of an initial diagonal approximation to the Hessian matrix. At each iteration, the gradient of the objective function is computed, and the conjugate gradient method is applied to approximate the Newton direction while ensuring that the preconditioner improves numerical stability and accelerates convergence. The stopping criterion for the inner conjugate gradient loop is based on the reduction of the gradient's magnitude. The optimal step length along the search direction is then determined using a line search that satisfies the Wolfe conditions, ensuring sufficient decrease in the objective function. The solution vector is updated accordingly, and the process repeats until the outer convergence criteria are satisfied.

Note that we use a standard implementation of the TNC algorithm to solve the bilevel optimization problem in Model 4. Other optimization methods could be used without conceptual changes. The novelty of this work lies in the formulation and overall solution approach for the bilevel problem, rather than the application of TNC itself. For a more detailed discussion on the standard TNC method, refer to [33].

IV. Numerical Experiments

Using systems from the PGLib-OPF test case archive [35], including `IEEE 14-bus`, `EPRI 39-bus`, `IEEE 57-bus`, `IEEE 118-bus`, `Illinois 200-bus`, and `Texas 2000-bus` [35], this section numerically evaluates our parameter optimization algorithm's performance via comparison against traditional DC-OPF and AC-OPF models. For each test case, we generated 2,020 load scenarios, allocating 20 for offline training and $2,000$ for testing, by multiplicatively scaling the nominal loads with normally distributed random variables, each load independently by its own variable, with mean $\mu = 1$ and standard deviation $\sigma = 15\%$. The active and reactive power demands at each bus were scaled by the same multiplicative factor to maintain the loads' nominal power factors.

All computations were conducted on a 24-core, 32 GB RAM computing node at Georgia Tech's Partnership for an Advanced Computing Environment (PACE). The AC-OPF problems were solved using `PowerModels.jl` [36], and we implemented our algorithm in PyTorch with derivatives $\frac{\partial \mathbf{p}^{\text{g}}}{\partial \mathbf{b}}$, $\frac{\partial \mathbf{p}^{\text{g}}}{\partial \boldsymbol{\gamma}}$, and $\frac{\partial \mathbf{p}^{\text{g}}}{\partial \boldsymbol{\rho}}$ calculated via cvxpylayers [25]. The TNC optimization method from the `scipy.optimize.minimize` library was used to minimize the loss function (6) based on the sensitivities $\mathbf{g}^{b}$, $\mathbf{g}^{\gamma}$, and $\mathbf{g}^{\rho}$.

We benchmark the accuracy achieved by our optimized parameters by comparing the generators' active power setpoints across different DC-OPF models (i.e., $\mathbf{p}^{[\text{model}]}$ where [model] stands for either the optimized parameters from our proposed algorithm, the cold-start DC-OPF (4), the hot-start DC-OPF (5), or the optimized DC power flow (DCPF) parameters from [23]) against the true values from the AC-OPF solutions (i.e., $\mathbf{p}^{\text{g, AC}}$). The discrepancies are measured using both the mean square error (MSE) as defined in the loss function (6) and the maximum error metric, with all values in the per unit (p.u.) system with a 100 MVA base power:

$$\varepsilon^{[\text{model}]}_{\text{MSE}} = \frac{1}{|\mathcal{G}| \times |\mathcal{M}|} \sum_{m \in \mathcal{M}} \|\mathbf{p}^{[\text{model}]}_m - \mathbf{p}^{g,AC}_m\|_2^2, \qquad (10)$$

$$\varepsilon^{[\text{model}]}_{\text{max}} = \|\mathbf{p}^{[\text{model}]}_m - \mathbf{p}^{\text{g, AC}}_m\|_\infty, \qquad (11)$$

where $\|\cdot\|_\infty$ is the L_∞-norm, and $\|\cdot\|_2$ is the L_2-norm.

A. Comparison of Parameter Values Across Selection Methods

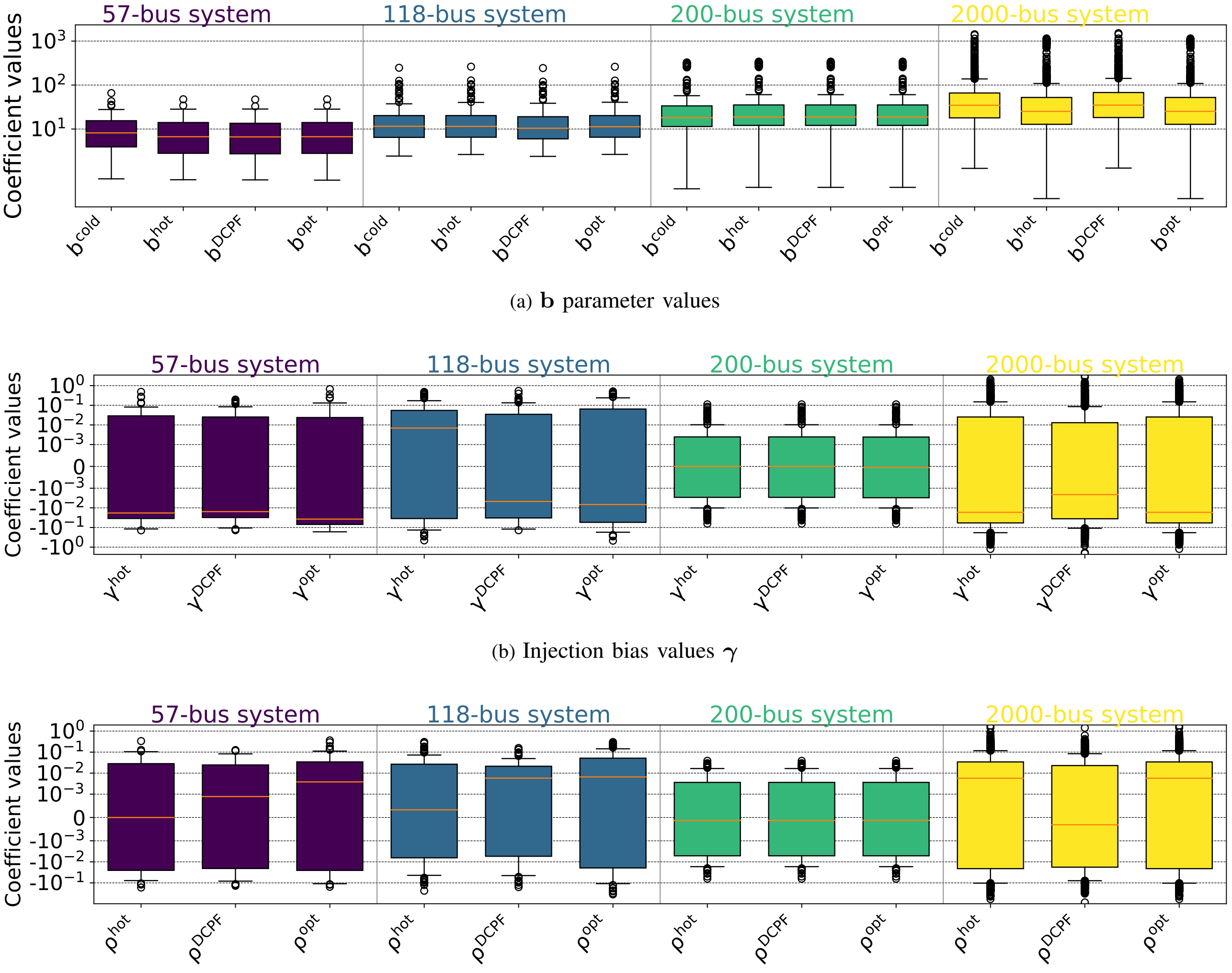
This section presents the distributions of the cold-start, hot-start, and optimized parameters $\mathbf{b}$, $\boldsymbol{\gamma}$, and $\boldsymbol{\rho}$ for different test systems using box plots and scatter plots. The box plots show the spread of parameter values across the buses and lines in the system, providing insight into how the parameter values vary within the system. The scatter plots compare the hot-start and optimized parameters for each bus or line, demonstrating the consistency of the optimized parameters with those used in a common DC power flow approximation.

1) Box Plot Analysis: Fig. 2 illustrates the distribution of the parameter values using box plots. The interquartile range (IQR), which represents the middle 50% of the data, is shown in each box, with the central line marking the median value. Whiskers extend to 1.5 times the IQR, and data points outside this range are considered outliers, marked individually. The box plots for each test system capture the cold-start values ($\mathbf{b}^{cold}$), hot-start values ($\mathbf{b}^{hot}$, $\boldsymbol{\gamma}^{hot}$, and $\boldsymbol{\rho}^{hot}$), and the optimized values ($\mathbf{b}^{opt}$, $\boldsymbol{\gamma}^{opt}$, and $\boldsymbol{\rho}^{opt}$). All values are presented on a logarithmic scale to accommodate variations across multiple orders of magnitude. The box plots in Fig. 2 show that the optimized parameter values closely align with those derived from traditional heuristics, particularly for the coefficient parameters. This suggests consistency with long-standing power engineering intuition, where line susceptances are known to be a key factor in determining power flows.

2) Scatter Plot Analysis: Complementing the box plots, the scatter plots (Fig. 3) directly compare the hot-start and optimized parameter values. Each plot includes a 45° red dashed line to indicate one-to-one correspondence. Data points falling along this line suggest a close match between the hot-start and optimized values, while deviations highlight discrepancies. Fig. 3a compares the coefficient values $\mathbf{b}^{hot}$ and $\mathbf{b}^{opt}$, while Fig. 3b compares the injection bias values $\boldsymbol{\gamma}^{hot}$ and $\boldsymbol{\gamma}^{opt}$, and Fig. 3c compares the flow bias values $\boldsymbol{\rho}^{hot}$ and $\boldsymbol{\rho}^{opt}$. These scatter plots reveal that the optimized parameters align well with hot-start values, except for a few outliers where targeted adjustments to $\mathbf{b}$, $\boldsymbol{\gamma}$, and $\boldsymbol{\rho}$ improve the DC-OPF approximation's accuracy.

Table I gives the squared two-norm and ∞-norm errors for several test cases when using the optimized and conventional DC-OPF parameters over $2,000$ testing scenarios. Notably, in the `118-bus` test case, our algorithm achieved a substantial reduction in the squared two-norm loss to 0.0123, compared to the cold-start, hot-start, and optimized DCPF parameters with 0.1295, 0.1221, and 0.1251, respectively. This represents an order of magnitude improvement over all prior methods, highlighting the effectiveness of the proposed algorithm.

The improvement column in Table I shows the percentage improvement from using the optimized parameters in the DC-OPF models compared to using the cold-start and hot-start pa-



(a) $\mathbf{b}$ parameter values

(b) Injection bias values $\boldsymbol{\gamma}$

(c) Flow bias values $\boldsymbol{\rho}$

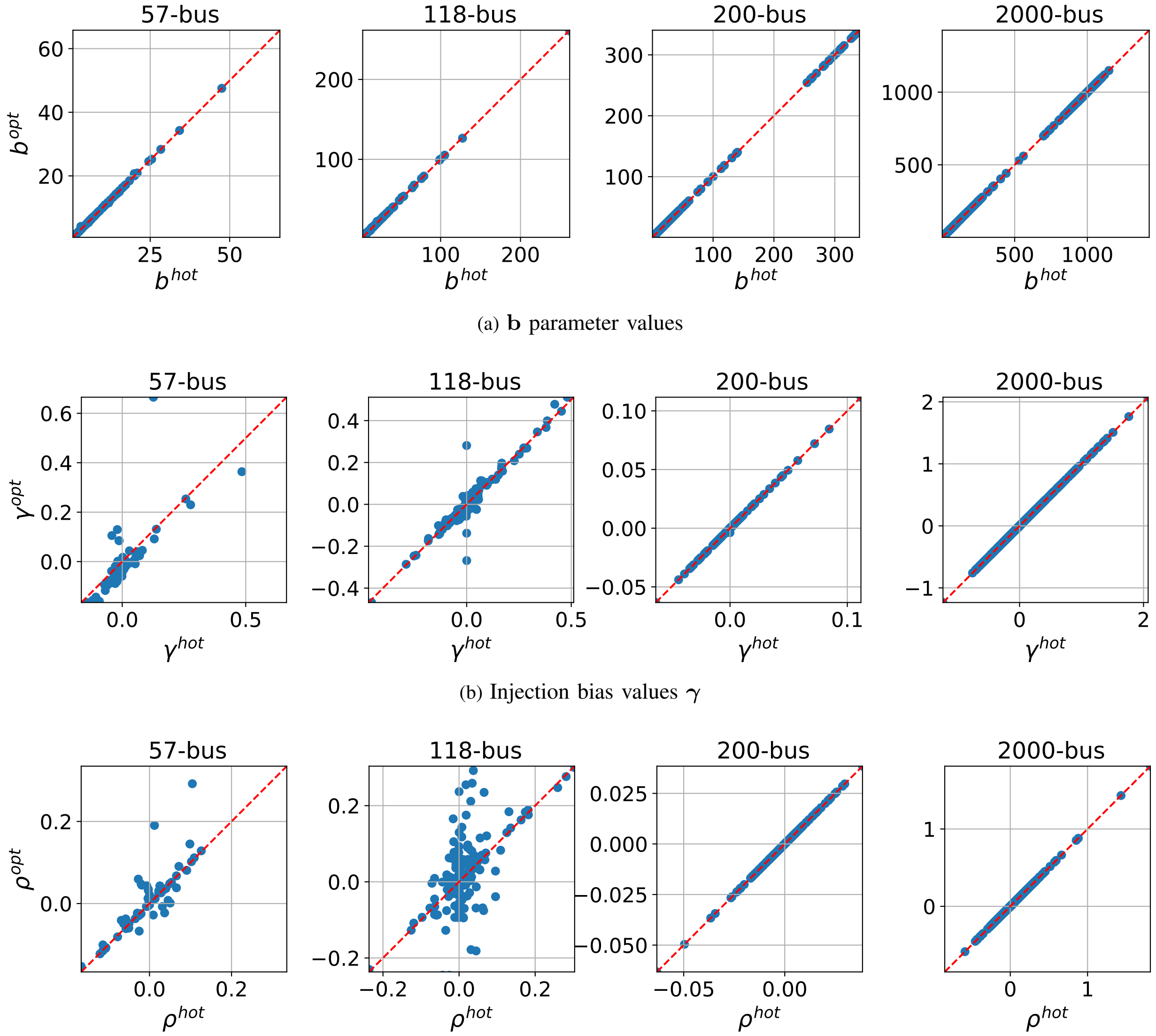
Fig. 2. Boxplots showing the distributions of the parameter values ($\mathbf{b}$, $\boldsymbol{\gamma}$, and $\boldsymbol{\rho}$) across various test cases for cold-start, hot-start, DCPF, and optimal methods.

rameters. The results show up to 90% and 79% improvements in the squared two-norm and ∞-norm errors, respectively.

The box plot in Fig. 4 compares the mean generator outputs for the IEEE 118-bus system across five formulations: cold-start, hot-start, DCPF, DC-OPF with optimal parameters, and AC-OPF. The vertical axis represents the generator output in per unit, and the horizontal axis shows the different parameter choices. Each box illustrates the distribution of mean generator outputs for non-zero generators across 2,000 test scenarios, highlighting the variability and median for each method. The figure shows that the DC-OPF with optimized parameters ($\mathbf{b}^{opt}$, $\boldsymbol{\gamma}^{opt}$, and $\boldsymbol{\rho}^{opt}$) aligns closely with the AC-OPF, as indicated by similar medians and ranges.

The graphical analyses provided in Figs. 5 and 6 offer further insights into the performance of the proposed algorithm. In Fig. 5, the generator setpoints for the IEEE 118-bus test system are presented, where we compare the AC-OPF solutions (i.e., $\mathbf{p}^{\rm g,AC}$) with the DC-OPF solutions derived from models with cold-start, hot-start, DCPF, and optimized DC-OPF parameters. Fig. 5a displays the setpoints of individual generators, excluding those that consistently reached their maximum output limits across all parameter choices. The alignment of setpoints with the 45-degree equality line demonstrates how closely the DC-OPF models, particularly the one with optimized parameters, mimic the AC-OPF results. Fig. 5b shows the average generator setpoints (p.u.) with standard deviation ellipses across $2,000$ scenarios. These ellipses illustrate the variability and performance of each DC-OPF method, highlighting the consistency of the optimized DC-OPF in closely matching the AC-OPF setpoints. The inset focuses on generators with lower setpoints, providing a detailed view of their alignment and variability. These results demonstrate that our optimized DC-OPF parameters achieve setpoints comparable to the more complex AC-OPF while offering enhanced computational efficiency.

Fig. 6 shows the cumulative proportion of the absolute errors between AC-OPF and DC-OPF solutions for the IEEE 118-bus system. The graph compares four scenarios: usage of cold-start $\mathbf{b}$, hot-start $\mathbf{b}^{hot}$, $\boldsymbol{\gamma}^{hot}$, $\boldsymbol{\rho}^{hot}$, DCPF $\mathbf{b}^{DCPF}$, $\boldsymbol{\gamma}^{DCPF}$, $\boldsymbol{\rho}^{DCPF}$, and optimized $\mathbf{b}^{opt}$, $\boldsymbol{\gamma}^{opt}$, $\boldsymbol{\rho}^{opt}$ parameters, shown on a logarithmic scale. Points located towards the upper left represent better performance, indicating a larger proportion of smaller errors, which demonstrates the improved accuracy



(a) $\mathbf{b}$ parameter values

(b) Injection bias values $\boldsymbol{\gamma}$

(c) Flow bias values $\boldsymbol{\rho}$

Fig. 3. Scatter plots comparing hot-start ($\mathbf{b}^{hot}$, $\boldsymbol{\gamma}^{hot}$, $\boldsymbol{\rho}^{hot}$) and optimized ($\mathbf{b}^{opt}$, $\boldsymbol{\gamma}^{opt}$, $\boldsymbol{\rho}^{opt}$) parameter values across various test cases. The red dashed line indicates a one-to-one correspondence.

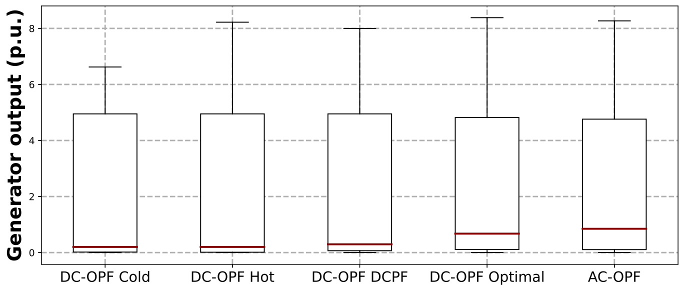


Fig. 4. Comparison of the generator outputs for the IEEE 118-bus system over 2,000 test scenarios. The box plot compares the performance of the AC-OPF vs DC-OPF with four parameter sets: cold-start ($\mathbf{b}^{cold}$, $\boldsymbol{\gamma}^{cold}$, and $\boldsymbol{\rho}^{cold}$), hot-start ($\mathbf{b}^{hot}$, $\boldsymbol{\gamma}^{hot}$, and $\boldsymbol{\rho}^{hot}$), DCPF-based parameters ($\mathbf{b}^{DCPF}$, $\boldsymbol{\gamma}^{DCPF}$, and $\boldsymbol{\rho}^{DCPF}$), and optimized parameters ($\mathbf{b}^{opt}$, $\boldsymbol{\gamma}^{opt}$, and $\boldsymbol{\rho}^{opt}$).

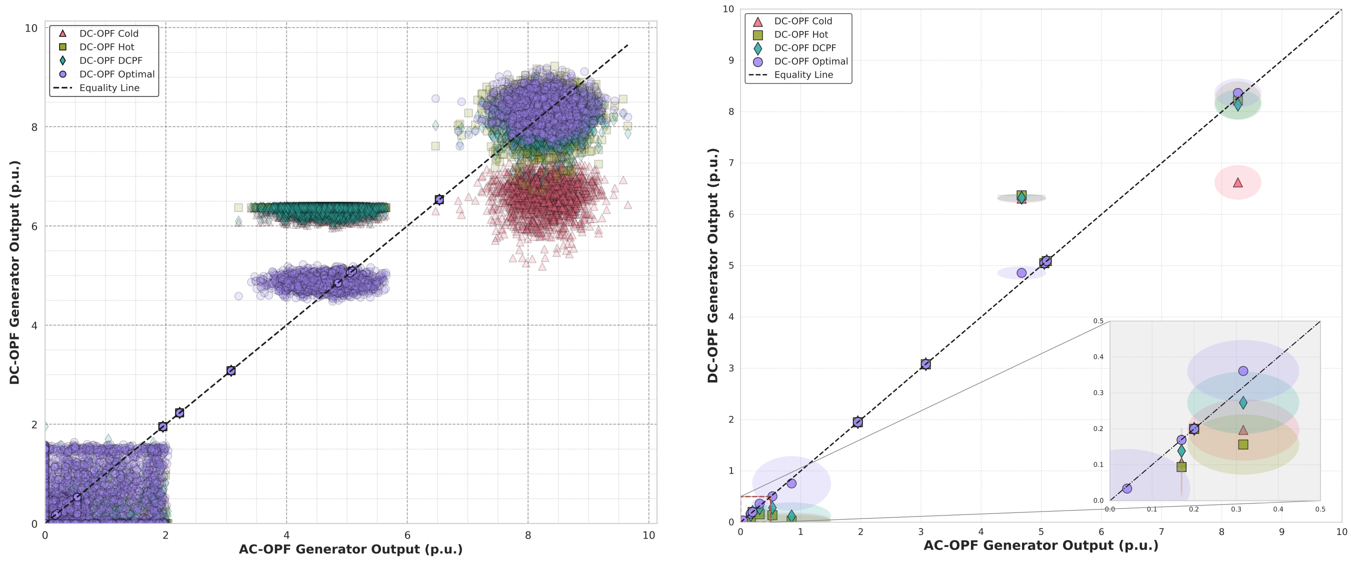
of the DC-OPF solutions using the optimized parameters.

Observe that the maximum errors are less than 1.918 per unit for the solution obtained from the optimized DC-OPF model versus errors up to 3.133, 3.170, and 3.146 per unit resulting from the cold-start DC-OPF with $\mathbf{b}^{cold}$, the hot-start DC-OPF with $\mathbf{b}^{hot}$, $\boldsymbol{\gamma}^{hot}$, and $\boldsymbol{\rho}^{hot}$, and the DCPF with $\mathbf{b}^{DCPF}$, $\boldsymbol{\gamma}^{DCPF}$, and $\boldsymbol{\rho}^{DCPF}$, respectively.

Finally, Table II summarizes the training, testing, and AC-OPF model computation times. Training times, denoted as t_{train}, range from 13 to $9,357$ seconds. This represents a one-time, upfront effort that is within reasonable ranges for offline computations (minutes to hours for larger systems). We note that the primary bottlenecks in training computations are solving the DC-OPF problems and computing the gradients of the loss function with respect to the parameters. On average, solving the DC-OPF problems accounts for approximately 45% of the total training time, while computing the gradients contributes to 55% of the training time. We note that an implementation that utilizes a stochastic gradient descent method, as is often employed when training machine learning

TABLE I
SQUARED TWO-NORM AND ∞-NORM LOSS FUNCTIONS USING $2,000$ TEST SCENARIO DATASET

Test case	**Squared Two-Norm Loss**					**∞-Norm Loss**				
	$\varepsilon^{cold}_{\text{MSE}}$	$\varepsilon^{hot}_{\text{MSE}}$	$\varepsilon^{DCPF}_{\text{MSE}}$ [23]	$\varepsilon^{opt}_{\text{MSE}}$	Improv. (%)	$\varepsilon^{cold}_{\max}$	$\varepsilon^{hot}_{\max}$	$\varepsilon^{DCPF}_{\max}$ [23]	$\varepsilon^{opt}_{\max}$	Improv. (%)
`14-bus`	0.0070	0.0069	0.0069	**0.0030**	(57, 57, 57)	0.590	0.590	0.590	0.590	(0, 0, 0)
`39-bus`	0.3223	0.4248	0.3046	**0.3029**	(6, 28, 1)	5.877	6.215	5.784	**5.585**	(5, 10, 3)
`57-bus`	0.7445	0.6311	0.2054	**0.1765**	(76, 72, 14)	4.671	4.590	3.658	**3.120**	(33, 32, 15)
`118-bus`	0.1295	0.1221	0.1251	**0.0123**	(90, 90, 90)	3.133	3.170	3.146	**1.918**	(39, 39, 39)
`200-bus`	0.0031	0.0003	0.0003	**0.0002**	(93, 33, 33)	0.145	0.033	0.035	**0.031**	(79, 6, 11)
`500-bus`	0.0073	0.0069	0.0054	**0.0030**	(59, 56, 44)	1.356	1.000	0.950	**0.890**	(34, 11, 6)
`2000-bus`	0.0149	0.0161	0.0156	**0.0069**	(54, 57, 56)	0.829	1.024	0.967	**0.623**	(25, 39, 36)



(a) Generator setpoints

(b) Average generator setpoints

Fig. 5. (a) Generator setpoints (p.u.) obtained by solving the AC-OPF problem and DC-OPF problems with cold-start, hot-start, DCPF, and optimized parameters for the `IEEE 118-bus` test system. Generators that consistently reached their maximum limits across all models are excluded. The 45-degree equality line indicates perfect alignment between the AC-OPF and DC-OPF setpoints, serving as a benchmark for comparison. (b) Average generator setpoints (p.u.) with standard deviation ellipses across 2,000 scenarios.

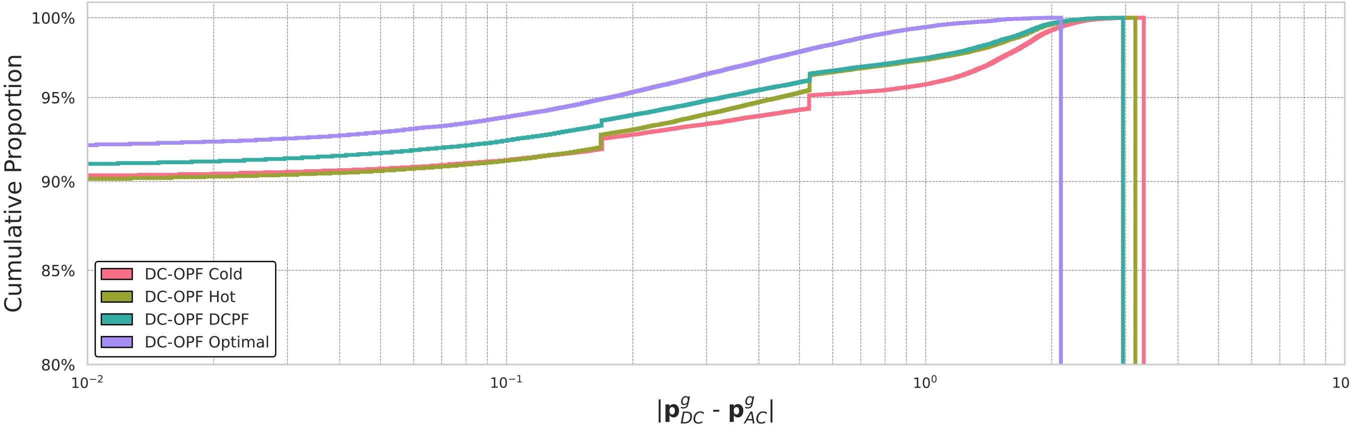


Fig. 6. Cumulative proportion of the absolute error between AC-OPF and DC-OPF (in per unit) for the `IEEE 118-bus` system over $2,000$ test scenarios. The graph compares four scenarios: usage of cold-start ($\mathbf{b}^{cold}$, $\boldsymbol{\gamma}^{cold}$, and $\boldsymbol{\rho}^{cold}$), hot-start ($\mathbf{b}^{hot}$, $\boldsymbol{\gamma}^{hot}$, and $\boldsymbol{\rho}^{hot}$), DCPF ($\mathbf{b}^{DCPF}$, $\boldsymbol{\gamma}^{DCPF}$, and $\boldsymbol{\rho}^{DCPF}$), and optimized ($\mathbf{b}^{opt}$, $\boldsymbol{\gamma}^{opt}$, and $\boldsymbol{\rho}^{opt}$) parameters, shown on a logarithmic scale.

models, could reduce both components of the training time.

Post-training, the solution times for the DC-OPF model (t_{DCOPF}) range from 0.009 to 1.304 seconds per scenario, significantly faster than the AC-OPF model due to the sim-

TABLE II
CPU Time in Seconds

Test case	14-bus	39-bus	57-bus	118-bus	200-bus	500-bus	2000-bus
t_{train}	13	17	41	52	58	903	9357
t_{DCOPF}	0.009	0.021	0.024	0.064	0.055	0.354	1.304
t_{ACOPF}	0.030	0.071	0.091	0.295	0.289	1.650	8.708

t_{train}: Offline computation time to train the parameters.
t_{DCOPF}: Online per-scenario computation time averaged over 100 scenarios.
t_{ACOPF}: Online per-scenario computation time averaged over 100 scenarios.

plicity of solving linear programs versus nonlinear programs. Moreover, the variation in DC power flow parameters does not impact DC-OPF solution times, which remain consistent to within 2% across different parameter choices.

V. Conclusion

This paper presented an optimization algorithm for significantly enhancing the accuracy of DC-OPF problems by selecting coefficient and bias parameters in the DC power flow approximation based on a sensitivity analysis. Employing the TNC method for optimization alongside efficient gradient computation techniques, an offline training phase refines DC power flow parameters across a range of loading scenarios. With squared two-norm and ∞-norm losses of up to 90% and 79%, respectively, lower than traditional parameter selection methods for DC-OPF problems, our numerical tests demonstrate a substantial improvement in DC-OPF solution accuracy relative to the AC-OPF problem.

Our future work is evaluating the performance of the optimized parameters in other applications, such as optimal transmission switching, unit commitment, and infrastructure hardening. Additionally, we plan to explore security-constrained variants of these problems, with a particular focus on how well the trained parameters generalize across related topologies, such as those associated with the $N-1$ line contingencies.

References

[1] M. B. Cain, R. P. O'Neill, and A. Castillo, "History of optimal power flow and formulations (OPF Paper 1)," *Federal Energy Regulatory Commission*, December 2012.

[2] L. A. Roald, D. Pozo, A. Papavasiliou, D. K. Molzahn, J. Kazempour, and A. Conejo, "Power systems optimization under uncertainty: A review of methods and applications," *Elect. Power Syst. Res.*, vol. 214, p. 108725, 2023, presented at the *22nd Power Syst. Comput. Conf. (PSCC)*.

[3] M. Majidi-Qadikolai and R. Baldick, "Stochastic transmission capacity expansion planning with special scenario selection for integrating $N-1$ contingency analysis," *IEEE Trans. Power Syst.*, vol. 31, no. 6, pp. 4901–4912, 2016.

[4] S. Abhyankar, G. Geng, M. Anitescu, X. Wang, and V. Dinavahi, "Solution Techniques for Transient Stability-Constrained Optimal Power Flow – Parts I and II," *IET Generation, Transmission & Distribution*, vol. 11, pp. 3177–3193, August 2017.

[5] D. Bienstock and A. Verma, "Strong NP-hardness of AC power flows feasibility," *Oper. Res. Lett.*, vol. 47, no. 6, pp. 494–501, 2019.

[6] D. K. Molzahn, "Computing the feasible spaces of optimal power flow problems," *IEEE Trans. Power Syst.*, vol. 32, no. 6, pp. 4752–4763, 2017.

[7] W. A. Bukhsh, A. Grothey, K. I. McKinnon, and P. A. Trodden, "Local solutions of the optimal power flow problem," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 4780–4788, 2013.

[8] C. Barrows, S. Blumsack, and P. Hines, "Correcting optimal transmission switching for AC power flows," in *47th Hawaii Int. Conf. Syst. Sci.*, January 2014, pp. 2374–2379.

[9] W. van Ackooij, I. Danti Lopez, A. Frangioni, F. Lacalandra, and M. Tahanan, "Large-scale unit commitment under uncertainty: An updated literature survey," *Annals Oper. Res.*, vol. 271, no. 1, pp. 11–85, December 2018.

[10] N. P. Padhy, "Unit commitment–a bibliographical survey," *IEEE Trans. Power Syst.*, vol. 19, no. 2, pp. 1196–1205, May 2004.

[11] J. Carpentier, "Contribution to the economic dispatch problem," *Bull. Soc. Franc. Elect.*, vol. 3, no. 8, pp. 431–447, 1962.

[12] J. Momoh, R. Adapa, and M. El-Hawary, "A review of selected optimal power flow literature to 1993. I. Nonlinear and quadratic programming approaches," *IEEE Trans. Power Syst.*, vol. 14, no. 1, pp. 96–104, 1999.

[13] J. Momoh, M. El-Hawary, and R. Adapa, "A review of selected optimal power flow literature to 1993. II. Newton, linear programming and interior point methods," *IEEE Trans. Power Syst.*, vol. 14, no. 1, pp. 105–111, 1999.

[14] I. Aravena *et al.*, "Recent developments in security-constrained AC optimal power flow: Overview of Challenge 1 in the ARPA-E Grid Optimization Competition," *Oper. Res.*, vol. 71, no. 6, pp. 1977–2014, November-December 2023.

[15] S. Gopinath and H. L. Hijazi, "Benchmarking large-scale ACOPF solutions and optimality bounds," in *IEEE Power & Energy Society General Meeting (PESGM)*, 2022.

[16] L. Duchesne, E. Karangelos, and L. Wehenkel, "Recent developments in machine learning for energy systems reliability management," *Proc. IEEE*, vol. 108, no. 9, pp. 1656–1676, 2020.

[17] D. K. Molzahn and I. A. Hiskens, "A survey of relaxations and approximations of the power flow equations," *Found. Trends Electr. Energy Syst.*, vol. 4, no. 1-2, pp. 1–221, 2019.

[18] M. Hohmann, J. Warrington, and J. Lygeros, "Optimal linearizations of power systems with uncertain supply and demand," *IEEE Trans. Power Syst.*, vol. 34, no. 2, pp. 1504–1512, March 2019.

[19] B. Stott, J. Jardim, and O. Alsaç, "DC power flow revisited," *IEEE Trans. Power Syst.*, vol. 24, no. 3, pp. 1290–1300, 2009.

[20] K. Dvijotham and D. K. Molzahn, "Error bounds on the DC power flow approximations: A convex relaxation approach," *55th IEEE Conf. Decis. Control (CDC)*, December 2016.

[21] K. Baker, "Solutions of DC OPF are never AC feasible," in *12th ACM Int. Conf. Future Energy Syst.*, 2021, pp. 264–268.

[22] A. Khodaei, M. Shahidehpour, and S. Kamalinia, "Transmission switching in expansion planning," *IEEE Trans. Power Syst.*, vol. 25, no. 3, pp. 1722–1733, 2010.

[23] B. Taheri and D. K. Molzahn, "Optimizing parameters of the DC power flow," *Elect. Power Syst. Res.*, vol. 235, p. 110719, 2024, presented at the *23rd Power Syst. Comput. Conf. (PSCC)*.

[24] B. Amos and J. Z. Kolter, "Optnet: Differentiable optimization as a layer in neural networks," in *Int. Conf. Machine Learning (ICML)*, 2017, pp. 136–145.

[25] A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and Z. Kolter, "Differentiable convex optimization layers," in *Adv. Neural Information Processing Syst. 32 (NeurIPS)*, 2019.

[26] B. Knueven, J. Ostrowski, and J.-P. Watson, "On mixed-integer programming formulations for the unit commitment problem," *INFORMS J. Comput.*, vol. 32, no. 4, pp. 857–876, 2020.

[27] E. B. Fisher, R. P. O'Neill, and M. C. Ferris, "Optimal transmission switching," *IEEE Trans. Power Syst.*, vol. 23, no. 3, pp. 1346–1355, 2008.

[28] K. W. Hedman, R. P. O'Neill, E. B. Fisher, and S. S. Oren, "Optimal transmission switching–Sensitivity analysis and extensions," *IEEE Trans. Power Syst.*, vol. 23, no. 3, pp. 1469–1479, 2008.

[29] C. Guo, H. Nagarajan, and M. Bodur, "Tightening quadratic convex relaxations for the AC optimal transmission switching problem," *arXiv:2212.12097*, 2022.

[30] Y. Liu and C. Singh, "Reliability evaluation of composite power systems using Markov cut-set method," *IEEE Trans. Power Syst.*, vol. 25, no. 2, pp. 777–785, 2010.

[31] A. V. Fiacco and G. P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. SIAM, 1990.

[32] S. M. Robinson, "Strongly regular generalized equations," *Math. Oper. Res.*, vol. 5, no. 1, pp. 43–62, 1980.

[33] S. G. Nash, "A survey of truncated-Newton methods," *J. Computat. Appl. Math.*, vol. 124, no. 1-2, pp. 45–59, 2000.

[34] J. Nocedal and S. Wright, *Numerical Optimization*. New York, NY: Springer Science & Business Media, 2006.

[35] IEEE PES Task Force on Benchmarks for Validation of Emerging Power System Algorithms, "The Power Grid Library for benchmarking AC optimal power flow algorithms," August 2019, *arXiv:1908.02788*.

[36] C. Coffrin, R. Bent, K. Sundar, Y. Ng, and M. Lubin, "PowerModels.jl: An open-source framework for exploring power flow formulations," in *20th Power Syst. Comput. Conf. (PSCC)*, 2018.