Restoring AC Power Flow Feasibility for Solutions to Relaxed and Approximated Optimal Power Flow Problems

Babak Taheri and Daniel K. Molzahn
School of Electrical and Computer Engineering
Georgia Institute of Technology
{taheri, molzahn}@gatech.edu

Abstract

To address computational challenges associated with power flow nonconvexities, significant research efforts over the last decade have developed convex relaxations and approximations of optimal power flow (OPF) problems. However, benefits associated with the convexity of these relaxations and approximations can have tradeoffs in terms of solution accuracy since they may yield voltage phasors that are inconsistent with the power injections and line flows, limiting their usefulness for some applications. Inspired by state estimation techniques, this paper proposes a new method for obtaining an AC power flow feasible point from the solution to a relaxed or approximated OPF problem. By treating the inconsistent voltage phasors, power injections, and line flows analogously to noisy measurements in a state estimation algorithm, the proposed method yields power injections and voltage phasors that are feasible with respect to the AC power flow equations while incorporating information from many quantities in the solution to a relaxed or approximated OPF problem. We improve this method by adjusting weighting terms with a machine learning based approach. We demonstrate the proposed method using several relaxations and approximations. The result show up to several orders of magnitude improvement in accuracy over traditional methods.

Keywords: Optimal power flow, state estimation, operating point restoration, machine learning.

1. Introduction

Optimal power flow (OPF) problems are central to many tasks in power systems. OPF problems optimize a specified objective function, such as generation cost, while satisfying inequality constraints on quantities including the generators' power outputs, the voltage magnitudes, and the line flows as well as equality constraints from a power flow model. The AC power flow equations accurately model the steady-state behavior of a power system by relating the complex power injections and line flows to the voltage phasors.

OPF problems that use an AC power flow model are nonconvex and NP-Hard [1, 2, 3]. The computational challenges from power flow nonconvexities are further compounded when solving OPF problems that consider, e.g., discreteness and uncertainty [4, 5]. To address these challenges, OPF problems are often simplified using convex relaxations and approximations of the AC power flow relaxations. Common relaxations and approximations yield semidefinite programming [6], second-order cone programming [7, 8], and linear programming [9] formulations; see [10] for a survey.

Convex relaxations can certify problem infeasibility, obtain bounds on the optimal objective values, and yield globally optimal decision variables when the relaxation is tight [10]. When applied appropriately, power flow approximations can also provide useful results. Accordingly, power flow relaxations and approximations are frequently used to convexify power systems optimization problems; see, e.g., [5, 10, 11, 12].

The benefits of power flow relaxations and approximations can come at the expense of accuracy since the results may be infeasible or suboptimal with respect to the nonconvex OPF problem. This occurs due to inaccuracies in the power flow model, i.e., the voltage phasors from the solution to the relaxed or approximated OPF problem may be inconsistent with the power injections and line flows. This is problematic for many practical applications that require a solution which satisfies the power flow equations. Prior work includes

conditions which guarantee that some relaxations will always provide globally optimal decision variables [10, 13], but these conditions are limited to special classes of problems that are not relevant to many practical settings.

These inaccuracies motivate the development of methods to restore voltage phasors and power injections that satisfy the AC power flow equations from the solutions to OPF problems that use relaxed or approximated power flow models. Previous papers on convex relaxations and approximations use straightforward methods for restoring AC power flow feasibility. For instance, one method simply fixes the active power injections and voltage magnitudes at (non-slack) generator buses to formulate power flow problems that are solved with traditional Newton-based methods [14]. Alternatively, one could solve the power flow problem resulting from fixing the active and reactive power injections at (non-slack) generator buses, or one could substitute the voltage magnitudes and angles from the relaxed or approximated solutions into the power flow equations to get consistent values for active and reactive power generation.[1]

Crucially, any such method may lead to values for some variables that do not match the solution to the relaxed or approximated problem. Moreover, the resulting point may violate the OPF problem's inequality constraints. Nevertheless, it is often important to get an AC power flow feasible point, ideally one that is as close as possible to the true OPF solution.

This paper focuses on improving upon the existing methods for restoring an AC power flow feasible point by leveraging more of the available information in the relaxed or approximated OPF solution. To accomplish this, we propose a restoration method inspired by state estimation techniques. Rather than selecting a subset of variables to fix, we formulate an unconstrained optimization problem that considers information regarding voltage phasors, power injections, and line flows from the relaxed or approximated solution. The proposed method treats inconsistencies in these quantities due to power flow model inaccuracies analogously to measurement errors in state estimation. The use of more information from the relaxed or approximated solution enables restoration of a point that is closer to the true OPF solution while also enabling us to borrow mathematical machinery from existing state estimation algorithms.

---

[1]We also note various penalization approaches which seek to modify the objective functions of relaxed OPF problems with the goal of finding a feasible and nearly optimal OPF solution [15]. We consider a different setting which seeks to quickly restore a point which satisfies the AC power flow equations (but may violate inequality constraints) directly from a relaxed or approximated OPF solution, that is, without recomputing a penalized relaxation of the OPF problem.

Moreover, the proposed method provides the freedom to choose weights that are analogous to the variances of the sensors' noise levels in state estimation algorithms. Rather than physically informed quantities obtained from sensor accuracies, the weights in our method are parameters that we can select based on the inconsistencies (with respect to the AC power flow equations) among various quantities in the solution to the relaxed or approximated OPF problem. This choice has a significant impact on the results. Since the inconsistencies in the solutions to the relaxation or approximation are not known a priori, choosing good values for these weights is challenging. We therefore take a machine learning approach for determining these weights. We solve a large number of OPF problems and their relaxations or approximations to get a training dataset. We then use a gradient descent algorithm to iteratively update the weights based on sensitivity information from the proposed restoration method. These updates drive the weights to values that minimize a loss function formulated as the squared difference between the true OPF solutions and restored points across the full set of training data.

We demonstrate the proposed restoration method using various convex relaxations and approximations of OPF problems, with the results showing several orders of magnitude improvements in some instances.

To summarize, this paper's main contributions are:

- Developing a method inspired by state estimation techniques in order to restore AC power flow feasibility from the solutions to relaxed and approximated OPF problems.

- Improving this method by tuning weighting parameters via a machine learning approach.

- Demonstrating the performance of the proposed method using several power flow relaxations and approximations.

The remainder of this paper is organized as follows. Section 2 describes the OPF problem and several convex relaxations and approximations. Section 3 presents the proposed operating point restoration method. Section 4 demonstrates the proposed method with numerical experiments. Finally, Section 5 concludes the paper and discusses future work.

2. Preliminaries

This section reviews the OPF problem along with the SDP, SOCP, and QC relaxations and the LPAC approximation. A survey of these and other relaxations and approximations is provided in [10].

2.1. AC Optimal Power Flow

We first introduce notation. Let $\mathcal{N}$ and $\mathcal{E}$ denote the sets of buses and lines, respectively. Each bus $i \in \mathcal{N}$ has a voltage phasor V_i, complex power demand S_i^d, shunt admittance Y_i^S, and complex power generation S_i^g. (Buses without generators are modeled with zero generation limits.) Complex power flows into each terminal for each line $(j,k) \in \mathcal{E}$ are denoted as S_{jk} and S_{kj}. Each line $(j,k) \in \mathcal{E}$ has admittance parameters Y_{jk} and Y_{kj}. Real and imaginary parts of a complex number are denoted as $\Re(\cdot)$ and $\Im(\cdot)$, $(\cdot)^*$ indicates the complex conjugate, and $(\cdot)^T$ indicates the transpose. Upper and lower bounds are denoted as $(\overline{\cdot})$ and $(\underline{\cdot})$, respectively. The OPF problem is:

$$\min \sum_{i \in N} c_{2i}\left(\Re\left(S_i^g\right)\right)^2 + c_{1i}\Re\left(S_i^g\right) + c_{0i} \tag{1a}$$

s.t. $(\forall i \in \mathcal{N}, \ \forall (j,k) \in \mathcal{E})$

$$\mathbf{W}_{jk} = V_j V_k^*, \ \mathbf{W}_{kj} = V_k V_j^*, \ \mathbf{W}_{ii} = V_i V_i^* \tag{1b}$$

$$\left(\underline{V}_i\right)^2 \leq \mathbf{W}_{ii} \leq \left(\overline{V}_i\right)^2 \tag{1c}$$

$$\underline{S}_i^g \leq S_i^g \leq \overline{S}_i^g \tag{1d}$$

$$|S_{jk}| \leq \overline{S}_{jk}, \ |S_{kj}| \leq \overline{S}_{jk} \tag{1e}$$

$$S_i^g - S_i^d - \left(Y_i^S\right)^* \mathbf{W}_{ii} = \sum_{(i,j) \in \mathcal{E}} S_{ij} + \sum_{(k,i) \in \mathcal{E}} S_{ki} \tag{1f}$$

$$S_{jk} = Y_{jk}^* \mathbf{W}_{jj} - Y_{jk}^* \mathbf{W}_{jk} \tag{1g}$$

$$S_{kj} = Y_{kj}^* \mathbf{W}_{kk} - Y_{kj}^* \mathbf{W}_{kj}^* \tag{1h}$$

$$\tan\left(-\overline{\theta}_{jk}\right)\Im\left(\mathbf{W}_{jk}\right) \leq \Re\left(\mathbf{W}_{jk}\right) \leq \tan\left(\overline{\theta}_{jk}\right)\Im\left(\mathbf{W}_{jk}\right). \tag{1i}$$

The OPF problem minimizes an objective function, here chosen to be generation cost in (1a) with quadratic coefficients c_{2i}, c_{1i}, and c_{0i}. The products of voltage phasors are collected in a Hermitian matrix $\mathbf{W}$ as described in (1b). Voltage magnitude limits are imposed in (1c). The generators' output limits are enforced in (1d). Apparent power flow limits are specified in (1e). Constraint (1f)) enforces complex power balance at each bus and complex power flows are defined in (1g) and (1h) for each line. Limits on phase angle differences across lines are imposed in (1i). All nonconvexities in (1) are associated with the products in (1b).

2.2. Semidefinite Programming (SDP) Relaxation

The nonconvex constraint (1b) can be equivalently represented by the pair of constraints $\mathbf{W} \succeq 0$ (i.e., $\mathbf{W}$ is positive semidefinite) and $\text{rank}(\mathbf{W}) = 1$. Neglecting the rank constraint yields a semidefinite programming relaxation of the OPF problem [6]:

$$\min (1a) \ \text{s.t.} \ (1c)\text{–}(1i), \mathbf{W} \succeq 0. \tag{2}$$

2.3. Second-Order Cone Programming (SOCP) Relaxation

Rather than jointly considering all elements in the matrix $\mathbf{W}$, the SOCP relaxation separately convexifies individual elements from (1b) using the fact that all positive semidefinite matrices have nonnegative principal minors [7]. Considering the principal minors associated with both the diagonal entries and the 2×2 submatrices yields:

$$\min (1a) \ \text{s.t.} \ (1c)\text{–}(1i), |\mathbf{W}_{jk}|^2 \leq W_{jj} W_{kk}, \ \mathbf{W}_{ii} \geq 0. \tag{3}$$

2.4. Quadratic Convex (QC) Relaxation

The QC relaxation described in [8] introduces additional variables and constraints to better model phase angle relationships and current flows. Define new variables v_i, θ_i, and l_{jk} to represent the voltage magnitude and phase angle at each bus $i \in \mathcal{N}$ and the squared magnitude of the current flow on line $(j,k) \in \mathcal{E}$. Let $\langle \cdot \rangle^T$, $\langle \cdot \rangle^M$, $\langle \cdot \rangle^C$, and $\langle \cdot \rangle^S$ denote convex envelopes associated with the square, bilinear product, cosine, and sine functions, respectively, as defined in [8]. Define an impedance parameter Z_{jk} associated with each line $(j,k) \in \mathcal{E}$. The QC relaxation is:

$$\min (1a) \ \text{s.t.} \ (1c)\text{–}(1i) \tag{4a}$$

$$\mathbf{W}_{ii} \in \left\langle v_i^2 \right\rangle^T \tag{4b}$$

$$\Re\left(\mathbf{W}_{jk}\right) \in \left\langle \langle v_j v_k \rangle^M \langle \cos\left(\theta_j - \theta_k\right)\rangle^C \right\rangle^M \tag{4c}$$

$$\Im\left(\mathbf{W}_{jk}\right) \in \left\langle \langle v_j v_k \rangle^M \langle \sin\left(\theta_j - \theta_k\right)\rangle^S \right\rangle^M \tag{4d}$$

$$S_{jk} + S_{kj} = Z_{jk} l_{jk} \tag{4e}$$

$$|S_{jk}|^2 \leq \mathbf{W}_{jj} l_{jk}. \tag{4f}$$

2.5. Linear Programming AC (LPAC) Approximation

The LPAC approximation linearizes the sine function and makes other approximations regarding near-nominal voltage magnitudes to model active and reactive power injections. Similar to the QC relaxation, the LPAC approximation also introduces voltage

magnitude and phase angle variables v_i and θ_i for each bus $i \in \mathcal{N}$ along with lifted variables ϕ_{jk} representing $\cos(\theta_j - \theta_k)$ using convex envelopes $\langle \cdot \rangle^{LPAC}$ that are related to the QC relaxation's cosine envelopes $\langle \cdot \rangle^C$. Define the conductance and susceptance of each line $(j, k) \in \mathcal{E}$ as g_{jk} and b_{jk}, respectively. The cold-start version of the LPAC approximation is:

$$\min (1a) \text{ s.t. } (1d)\text{–}(1f) \tag{5a}$$

$$\Re(S_{jk}) = g_{jk} - (g_{jk}\phi_{jk} + b_{jk}(\theta_j - \theta_k)) \tag{5b}$$

$$\Im(S_{jk}) = -b_{jk}(1 + v_j - v_k) - (g_{jk}(\theta_j - \theta_k) - b_{jk}\phi_{jk}) \tag{5c}$$

$$\phi_{jk} \in \langle \cos(\theta_j - \theta_k) \rangle^{LPAC} \tag{5d}$$

$$-\overline{\theta}_{jk} \le \theta_j - \theta_k \le \overline{\theta}_{jk}. \tag{5e}$$

3. Restoring AC Power Flow Feasibility

Each of the formulations in Section 2 have their own advantages and disadvantages. The OPF problem (1) gives a feasible solution if the solver converges, but the problem is nonconvex and NP-Hard, meaning that solvers may fail to converge or converge to a local solution. Moreover, the nonconvexity of this problem imposes significant challenges for solving extensions of (1) that involve discrete and stochastic variables. Conversely, the relaxations and approximations are convex but are not guaranteed to give a solution that satisfies the power flow equations. To address this issue, this section develops a method for restoring voltage phasors and power injections that satisfy the AC power flow equations using information from relaxed and approximated solutions.

This section presents a method that, compared to previous techniques, uses more information available in the solution by not fixing any variables to specific values in order to achieve better accuracy. This method draws on ideas from state estimation with the voltage phasors, power injections, and line flows from a relaxation or approximation playing a role analogous to noisy measurements. We emphasize that we are *not* suggesting to use measured quantities from the actual system. Rather, similar to the manner by which state estimation algorithms resolve inconsistencies between noisy measurements, our method seeks the set of voltage phasors that most closely matches the voltage phasors, power injections, and line flows that result from a power flow relaxation or approximation.

Analogous to how state estimation algorithms use the variation associated with sensor noise to weight measured quantities, our method includes weighting parameters associated with the outputs of each quantity
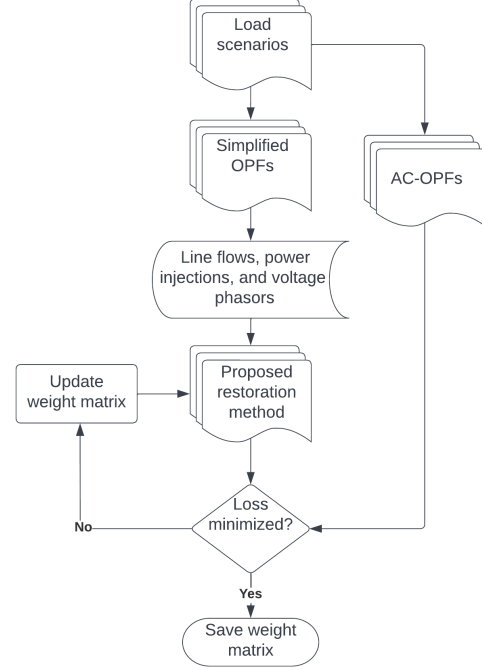

Figure 1: Flowchart of the proposed method for determining weighting parameters.

from the relaxed or approximated OPF solution. However, rather than being determined by the physical characteristics of a sensor, we can customize our weighting parameters based on the inconsistencies (with regard to the AC power flow equations) between various quantities in the solution to the relaxed or approximated problem. To compute good values for these weighting parameters, we propose a gradient descent based technique. A flowchart of the proposed method for determining the weights is shown in Fig. 1.

3.1. Restoration Method

We next propose our method for restoring AC power flow feasible points from the solutions to convex relaxations and approximations. This method seeks to find voltage phasors that are as close as possible to the OPF voltage phasors based on the voltage phasors, power injections, and line flows from the relaxed or approximated OPF solution. As summarized in Table 1, the proposed method is inspired by state estimation.

We first introduce notation based on typical presentations of state estimation algorithms to illustrate how we leverage the associated mathematical machinery, while again emphasizing that we do not use measured quantities from physical sensors. We seek the

Table 1: Analogy between the proposed method and state estimation

Proposed restoration method	State estimation
Solutions from relaxed or approximated models	Measurements from physical sensors
Inconsistencies in relaxed or approximated solutions	Noise from physical sensors
Weighting parameters	Variance of the measurement noise

voltage phasors, denoted as x, that are most consistent with the voltage magnitudes, phase angles, power flows, and power injections from the solution to a relaxed or approximated OPF solution, which we gather into a vector z. Let m denote the number of these quantities, i.e., the length of z, and n denote the number of voltage magnitudes plus the number of (non-slack) voltage angles, i.e., the length of x. We relate x and z through the network model, denoted as $h(x)$:

$$z_i = h_i(x) + e_i, \quad i = 1, \ldots, m, \qquad (6)$$

where e_i indicates the inconsistency between x and z relative to the power flow model $h(x)$.

Similar to typical state estimation algorithms, our method uses a weighted least squares formulation:

$$\min J(x) = \sum_{i=1}^{n} \left(\frac{z_i - h_i(x)}{\sigma_i} \right)^2 = e^T \Sigma e, \quad (7)$$

where σ_i is a weighting parameter associated with z_i and $\Sigma = \text{diag}\left(\cdots, \sigma_i, \cdots\right)$. (In state estimation, σ_i would be the sensor noise variance for measurement i and Σ would be the covariance matrix for the sensor noise.) We will discuss a method for choosing σ in the following subsection.

Computing (7) is accomplished by solving:

$$g(x) = \frac{\partial J(x)}{\partial x} = -(H(x))^T \Sigma (z - h(x)) = 0, \quad (8)$$

where $H(x)$ is the Jacobian matrix associated with the network model $h(x)$, i.e., $H(x) = \frac{\partial h(x)}{\partial x}$. Applying a Newton-Raphson method to (8) yields the iterates:

$$x^{k+1} = x^k - (G(x))^{-1} g(x), \qquad (9)$$

where $G(x) = \frac{\partial g(x)}{\partial x}$ and k denotes the iteration number. Thus, the solution to (7) is obtained with the following

iterative algorithm:

$$x^{k+1} = x^k +$$
$$\left((H(x^k))^T \Sigma H(x^k)\right)^{-1} \left((H(x^k))^T \Sigma \left(z - h\left(x^k\right)\right)\right). \tag{10}$$

In contrast to existing restoration methods, observe that the value of x depends on all quantities from the relaxed or approximated OPF solution in z. Thus, we leverage more information available in the solution. As we will show numerically in Section 4, this facilitates more accurate restorations of AC power flow feasibility.

3.2. Determining the Weighting Parameters

The weighting parameters σ can have a significant impact on the accuracy of the point obtained from the algorithm (10). Conceptually, we would like to choose larger values of σ_i for quantities z_i from the relaxed or approximated solution which more accurately represent the true solution to the OPF problem. However, the accuracy of a particular z_i with respect to the true OPF solution is unknown a priori and difficult to estimate. Accordingly, we employ a machine learning method to determine the weighting parameters as shown in Fig. 1. This method first solves a randomly generated representative set of OPF problems along with the relaxation or approximation of interest to create a training dataset. The method then uses a gradient descent method that iterates between solving the proposed restoration method and computing steps which minimize a loss function based on the difference between the restored point and the true OPF solution across the training dataset.

The sensitivities of the restored point x with respect to the weighting parameters σ are key to this gradient descent method. These sensitivities $\frac{\partial x_R}{\partial \Sigma}$ are:

$$\frac{vec(\partial x_R)}{vec(\partial \Sigma)} = \left((z - h) - (H(H^T \Sigma H)^{-1} \right.$$
$$\left. H^T \Sigma (z - h) \right) \otimes \left((H^T \Sigma H)^{-1} H^T \right)^T, \tag{11}$$

where $\otimes$ is the Kronecker product and vec($\cdot$) denotes the vectorization of a matrix. (Note that the dependencies on x are suppressed for notational brevity in (11).) The vectors z and $h(x)$ have length m and the matrix H has size $m \times n$. Therefore, the resulting sensitivities are described by a $n \times m^2$ matrix. The appendix provides the mathematical derivation for (11).

Using these sensitivities, we seek weighting parameters that minimize a loss function formulated

as the squared difference between the OPF solutions, whose voltage magnitudes and angles are denoted as x_{AC}, and points from our restoration method, whose voltage magnitudes and angles are denoted as x_R, across the set of training data. Formally, our loss function is:

$$F(\Sigma) = \frac{1}{2}\Psi^T(\Sigma)\Psi(\Sigma), \qquad (12)$$

where $\Psi(\Sigma) = x_R(\Sigma) - x_{AC}$.

To minimize this loss function, we use a gradient descent algorithm. The gradient of the objective with respect to the weighting parameters is denoted as q_t:

$$q_t = \nabla_\Sigma F(\Sigma) = J_\Psi^T \Psi = \frac{\partial x_R}{\partial \Sigma}(x_R - x_{AC}) \quad (13)$$

Using this gradient, one can find the optimal weighting parameters Σ using an iterative method. There are many variants of gradient descent algorithms, such as batch gradient, momentum, AdaGrad, Adam, improved Adam, etc., each of which has their own advantages and disadvantages. We use the Adaptive Gradient (AdaGrad) as shown in (14) since we empirically found it to perform best for this application:

$$Q_t = Q_{t-1} + q_t^2 \qquad (14a)$$

$$\Delta\Sigma = -\eta \frac{q_t}{\sqrt{Q_t + \epsilon}} \qquad (14b)$$

$$\Sigma_{t+1} = \Sigma_t + \Delta\Sigma \qquad (14c)$$

where η is a learning rate (step size), t is the iteration counter, ϵ is a parameter, and Q_t is the sum of the squares of the gradients up to iteration t. Using the gradient descent method shown in Algorithm 1, the optimal weighting parameters are obtained iteratively over the training dataset. The training dataset is created by randomly sampling different loading conditions within the forecast range of demands expected over the next time period (e.g., the variation predicted within the next day's load forecast). After the offline execution of Algorithm 1, the resulting weighting parameters are applied to restore AC power flow feasibility for solutions to OPF relaxations and approximations computed online.

4. Experimental Results and Discussion

This section presents numerical results for evaluating the performance of the proposed restoration algorithm for several convex relaxations and approximations.

Algorithm 1: Computing Weight Parameters

Input: η, ϵ: Gradient descent parameters
$\quad\quad\quad \Sigma^{init}$: Initial weighting parameters
Output: Σ^{opt}: Optimal weighting parameters
Data: OPF problem data

1 Generate many loading scenarios s and store in the set $\mathcal{S}$
2 Solve OPF problems (1) for each scenario and store the results in x_{AC}
3 Solve relaxed or approximated OPF problems for each scenario and store the results in x_R
4 $\Sigma^1 \leftarrow \Sigma^{init}$
5 **while** $iter \leq max\_iter$ **do**
6 $\quad$ $\nabla F(\Sigma) = 0$
7 $\quad$ **for** $s \in \mathcal{S}$ **do**
8 $\quad\quad$ Run restoration method (10) to an accuracy of ε and store solution in x_R
9 $\quad\quad$ $\nabla F(\Sigma) \leftarrow \nabla F(\Sigma) + \frac{\partial x_R}{\partial \Sigma}(x_R - x_{AC})$
10 $\quad\quad$ Store solutions: $X_R \leftarrow x_R$
11 $\quad$ $\Delta\Sigma = -\eta \frac{q_t}{\sqrt{Q_t + \epsilon}}$
12 $\quad$ $\Sigma_{t+1} = \Sigma_t + \Delta\Sigma$
13 $\quad$ $\mathcal{L} = \sum_{s \in \mathcal{S}} \sum_{i \in \mathcal{N}} \left(\|X_R^{i,s} - X_{AC}^{i,s}\|_2^2 \right)$
$\quad$ $iter \leftarrow iter + 1$
14 $\Sigma^{opt} \leftarrow \Sigma$

4.1. Experiment Setup

We evaluate the performance of proposed restoration method by applying the SOCP, SDP, and QC relaxations and LPAC approximation to the PJM five-bus system and the IEEE 14-bus system [16]. Datasets of 10,000 scenarios for training and testing purposes were generated by multiplicatively perturbing the nominal loads in these test cases by a normal random variable with zero mean and standard deviation of 0.1. The OPF problems as well as the relaxations and approximations were computed using `PowerModels.jl` [17] with the solvers Ipopt [18] and SCS [19] on a Apple laptop with a 10-core M1 Pro CPU and 32 GB of RAM. The restoration method was implemented in Python 3.0 using a Jupyter notebook [20].

4.2. Benchmarking Approach

To assess the performance of various restoration methods, we consider the two-norm of the difference between the OPF solution's voltage phasors and the restored solutions voltage phasors across all samples in

the dataset, denoted as $\mathcal{L}$:

$$\mathcal{L} = \sum_{s \in \mathcal{S}} \sum_{i \in \mathcal{N}} \left(||v_R^{i,s} - v_{AC}^{i,s}||_2^2 + ||\theta_R^{i,s} - \theta_{AC}^{i,s}||_2^2 \right),$$

(15)

where v_R and θ_R denote the restored voltage magnitudes and angles, respectively. We consider four restoration methods. The first simply compares the voltage magnitudes and angles from the relaxed or approximated solution directly (without any processing) to the OPF solution. Note that this method typically does not yield an AC power flow feasible point and is thus not suitable for many practical applications. Also note that this method is inapplicable to the SDP and SOCP relaxations which lack variables corresponding to the voltage phase angles. Note that the voltage angles are in radians. The second method, denoted as "benchmark" in the discussion below, solves the power flow problem that results from fixing the voltage magnitudes at all generator buses and the active power injections at non-slack generator buses to the output of the relaxation of approximation, as discussed in [14]. The third is our proposed restoration method with the weighting parameters set to (heuristically determined) initial values of Σ that weight the voltage magnitudes and phase angles with values of $\frac{1}{0.001^2}$ and the power injections and line flows with values of $\frac{1}{0.01^2}$. The fourth is our restoration method with the weighting parameters computed using Algorithm 1, trained using 8,000 test scenarios. We used 2,000 test scenarios to evaluate each method.

4.2.1. Training the Weighting Parameters

To compare the performance of different gradient descent algorithms for computing the weighting parameters, including batch gradient, momentum base gradient, Adagrad, Adam, and improved Adam methods, we ran a sample loss function using each of these methods for 20 training scenarios for $1,000$ iterations with the heuristically determined initialization described above. The results are shown in Fig. 2. While each of these methods reduces the loss function, the Adagrad method outperforms the other methods.

The weighting parameters resulting from Algorithm 1 for the five-bus system with the QC, SOCP, and SDP relaxations and the LPAC approximation are visualized in Fig. 3. We observe that some quantities are assigned much larger weights than others, motivating further investigation into how we might exploit knowledge of these weights to improve the relaxations and approximations.
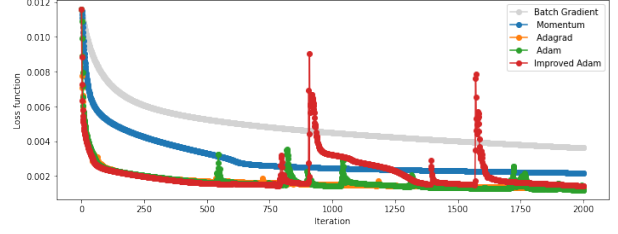

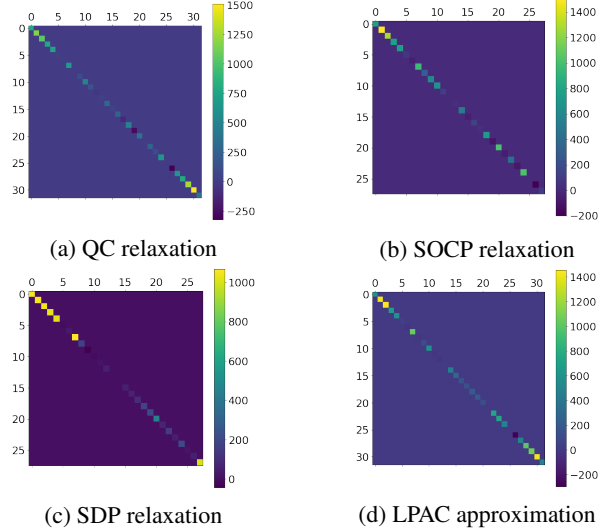
Figure 2: The loss function for training scenarios.

(a) QC relaxation

(b) SOCP relaxation

(c) SDP relaxation

(d) LPAC approximation

Figure 3: Trained weight matrices for QC, SOCP, and SDP relaxations and LPAC approximation.

4.2.2. Performance Evaluation

We next show the effectiveness of the proposed restoration method using the test datasets that were unseen during the weighting parameter calculation in Algorithm 1 (i.e., the 2,000 test scenarios). The values of the loss function (15) for each solution recovery method are given in Table 2. To further illustrate the performance of the proposed method, the voltage phasors from OPF solutions and the restored point the proposed method (with optimized weighting parameters) are compared in Fig. 4 for the SDP relaxation.

As can be seen from both Table 2 and Fig. 4, the proposed restoration method obtains high-quality AC power flow feasible points from the solutions to relaxed and approximated OPF problems. The loss functions from the proposed method are several orders of magnitude smaller than the other approaches, including the benchmark approach. Moreover, using the optimized weighting parameters significantly improves the loss function relative to the heuristically determined weighting parameters. The loss functions when computed with the initial weighting parameters for the

Table 2: Loss Function Evaluated for Different Test Cases Using Various Relaxations and Approximations

	Modified PJM 5-Bus				IEEE 14-Bus			
	QC	SOCP	SDP	LPAC	QC	SOCP	SDP	LPAC
Relaxed or approximated solution	6.0378	—	—	4.4964	102.4015	—	—	15.2709
Benchmark	5.4617	153.4819	101.6931	4.2735	0.0240	107.0302	105.4196	5.2313
SE with Σ^{init}	2.5974	2.1195	9.7564	2.4278	6.1669	6.6353	6.8579	16.4865
SE with Σ^{opt}	0.3305	0.3658	0.0493	0.2896	0.0494	0.0416	0.0361	2.1471

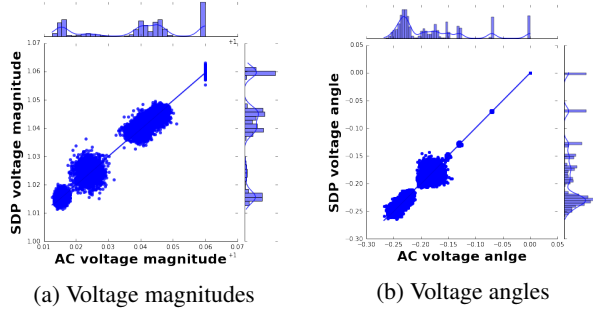(a) Voltage magnitudes   (b) Voltage angles

Figure 4: Comparison of the voltage magnitudes and angles for the OPF solution vs. the proposed method with the optimized weighting parameters for the SDP relaxation.

Figure 5: The loss function with different training scenarios.

IEEE 14-bus system in comparison with the optimized weighting parameters are 8, 80, 159, and 190 times larger for LPAC, QC, SOCP, and SDP formulations, respectively.

Furthermore, to show how the proposed method's performance scales with different numbers of training scenarios, we optimized the weighting parameters using varying number of scenarios for the five-bus test system with the QC relaxation. Fig. 5 shows the obtained loss function on the test dataset for varying numbers of training scenarios. With 200 training scenarios for this test case, the restoration method can efficiently learn the weighting parameters. The results show little improvement (and even some minor degradation) when using more scenarios. Our ongoing work aims to assess the performance and sample complexity tradeoffs as we scale to larger systems.

Also, while this exploratory work focused primarily on the gains in accuracy from the proposed method, we also show the execution time per scenario with optimized weighting parameters in Table 3. We note that the main computational burden in this problem is the training of the weighting parameters, which is done offline. In this offline training, solving the relaxed and approximated OPFs and running
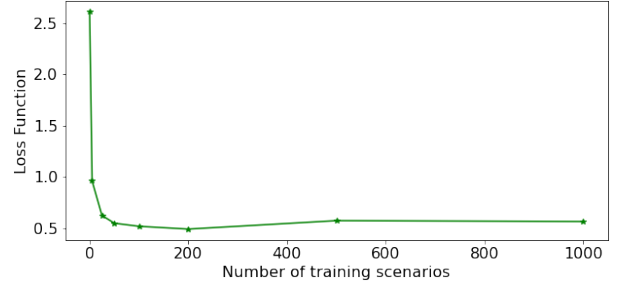
the restoration model to learn the weighting terms are the most time-consuming steps in the algorithm. Our ongoing work aims to create a more efficient implementation of our method and to assess variants such as stochastic gradient descent that may lead to substantial computational improvements.

5.   Conclusion and Future Work

To recover AC power flow feasible points from the solutions to relaxed and approximated OPF problems, this paper proposes a solution restoration method inspired by state estimation techniques. Treating quantities from the solutions of relaxed or approximated OPF problems (i.e., voltage phasors, power injections, and line flows) analogous to measurements from physical sensors in a state estimation algorithm, the proposed method finds an operating point that is feasible with respect to the AC power flow equations. Moreover, we improve this method by adjusting weighting parameters using a gradient descent algorithm. After computing the weighting parameters based on training datasets, application of the method to the unseen test datasets demonstrate that the solution restoration method can achieve several orders of magnitude improvement in accuracy relative to existing methods.

Our ongoing work includes 1) improving

Table 3: Average Execution Time per Scenario (seconds)

	Modified PJM 5-Bus				IEEE 14-Bus			
	QC	SOCP	SDP	LPAC	QC	SOCP	SDP	LPAC
Average solution time per scenario	0.0867	0.0757	0.0642	0.0644	0.4265	0.3090	0.3120	0.3823

computational tractability using, for instance, stochastic gradient descent techniques, 2) extending to consider restoration of AC power flow feasibility for the outputs of machine learning algorithms such as [21], 3) incorporating more information from the relaxed or approximated solution from, e.g., dual variables, 4) considering more general representations of the weighting parameters by extending the diagonal matrices considered here to full matrices, and 5) studying whether the optimal weighting parameter values could be useful for characterizing the behavior of a power flow relaxation or approximation.

6. Appendix

The sensitivity of the voltage phasors obtained from the state estimation inspired algorithm (i.e., x_R) with respect to the weight matrix (i.e., Σ) is computed by (11), which is derived as follows:

$$Y = \left(H^T \Sigma H\right)^{-1} H^T \Sigma \left(z - h\left(x^k\right)\right) \tag{16}$$

$$dY = \overbrace{d\left(\left(H^T \Sigma H\right)^{-1} H^T\right) \Sigma \left(z - h\left(x^k\right)\right)}^{I} \\ + \underbrace{\left(H^T \Sigma H\right)^{-1} H^T d\left(\Sigma \left(z - h\left(x^k\right)\right)\right)}_{U} \tag{17}$$

$$U = \left(H^T \Sigma H\right)^{-1} H^T d\Sigma \left(z - h\left(x^k\right)\right) \tag{18}$$

$$I = d\left(\left(H^T \Sigma H\right)^{-1}\right) H^T \Sigma \left(z - h\left(x^k\right)\right) \tag{19}$$

$$I = \left(-\left(H^T \Sigma H\right)^{-1} d\left(H^T \Sigma H\right)\left(H^T \Sigma H\right)^{-1}\right) \\ H^T \Sigma \left(z - h\left(x^k\right)\right) \tag{20}$$

$$I = \left(-\left(H^T \Sigma H\right)^{-1} H^T d\Sigma H\left(H^T \Sigma H\right)^{-1}\right) \\ H^T \Sigma \left(z - h\left(x^k\right)\right) \tag{21}$$

$$dY = -\left(H^T \Sigma H\right)^{-1} H^T d\Sigma H\left(H^T \Sigma H\right)^{-1} H^T \Sigma \\ \left(z - h\left(x^k\right)\right) + \left(H^T \Sigma H\right)^{-1} H^T d\Sigma \left(z - h\left(x^k\right)\right) \tag{22}$$

$$vec(dY) = vec\left[-\left(H^T \Sigma H\right)^{-1} H^T d\Sigma H \\ \left(H^T \Sigma H\right)^{-1} H^T \Sigma \left(z - h\left(x^k\right)\right)\right] \\ + vec\left[\left(H^T \Sigma H\right)^{-1} H^T d\Sigma \left(z - h\left(x^k\right)\right)\right] \tag{23}$$

$$vec(dY) = \left[-\left(H\left(H^T \Sigma H\right)^{-1} H^T \Sigma \left(z - h\left(x^k\right)\right)\right)^T \\ \otimes \left(\left(H^T \Sigma H\right)^{-1} H^T\right)\right] vec(d\Sigma) \\ + \left[(z - h)^T \otimes \left(\left(H^T \Sigma H\right)^{-1} H^T\right)\right] vec(d\Sigma) \tag{24}$$

$$vec(dY) = \left[(z - h)^T \otimes \left(\left(H^T \Sigma H\right)^{-1} H^T\right) \\ -\left(H\left(H^T \Sigma H\right)^{-1} H^T \Sigma \left(z - h\left(x^k\right)\right)\right)^T \\ \otimes \left(\left(H^T \Sigma H\right)^{-1} H^T\right)\right] vec(d\Sigma) \tag{25}$$

$$\frac{vec(\partial Y)}{vec(\partial \Sigma)} = \left((z - h) - (H(H^T \Sigma H)^{-1} H^T \\ \Sigma(z - h)\right) \otimes \left((H^T \Sigma H)^{-1} H^T\right)^T \tag{26}$$

References

[1] W. A Bukhsh et al. "Local solutions of the optimal power flow problem". In: *IEEE Transactions on Power Systems* 28.4 (2013), pp. 4780–4788.

[2] D. K. Molzahn. "Computing the feasible spaces of optimal power flow problems". In: *IEEE Transactions on Power Systems* 32.6 (2017), pp. 4752–4763.

[3] D. Bienstock and A. Verma. "Strong NP-hardness of AC power flows feasibility". In: *Operations Research Letters* 47.6 (2019), pp. 494–501.

[4] C. Barrows, S. Blumsack, and P. Hines. "Correcting optimal transmission switching for AC power flows". In: *47th Hawaii International Conference on System Sciences*. Jan. 2014, pp. 2374–2379.

[5] L. A. Roald et al. "Power systems optimization under uncertainty: A review of methods and applications". In: to appear in *Electric Power Systems Research,* to be presented at the *22nd Power Systems Computation Conference (PSCC)* (2022).

[6] J. Lavaei and S. H. Low. "Zero duality gap in optimal power flow problem". In: *IEEE Transactions on Power Systems* 27.1 (2011), pp. 92–107.

[7] R. A Jabr. "Radial distribution load flow using conic programming". In: *IEEE transactions on power systems* 21.3 (2006), pp. 1458–1459.

[8] C. Coffrin, H. L. Hijazi, and P. Van Hentenryck. "The QC relaxation: A theoretical and computational study on optimal power flow". In: *IEEE Transactions on Power Systems* 31.4 (2015), pp. 3008–3018.

[9] C. Coffrin and P. Van Hentenryck. "A linear-programming approximation of AC power flows". In: *INFORMS Journal on Computing* 26.4 (2014), pp. 718–734.

[10] D. K. Molzahn, I. A. Hiskens, et al. "A survey of relaxations and approximations of the power flow equations". In: *Foundations and Trends in Electric Energy Systems* 4.1-2 (2019), pp. 1–221.

[11] A. Venzke et al. "Convex relaxations of chance constrained AC optimal power flow". In: *IEEE Transactions on Power Systems* 33.3 (May 2018), pp. 2829–2841.

[12] K. Bestuzheva, H. Hijazi, and C. Coffrin. "Convex relaxations for quadratic on/off constraints and applications to optimal transmission switching". In: *INFORMS Journal on Computing* 32.3 (2020), pp. 682–696.

[13] S. H. Low. "Convex relaxation of optimal power flow-Part II: Exactness". In: *IEEE Transactions on Control of Network Systems* 1.2 (2014), pp. 177–189.

[14] A. Venzke, S. Chatzivasileiadis, and D. K. Molzahn. "Inexact convex relaxations for AC optimal power flow: Towards AC feasibility".

In: *Electric Power Systems Research* 187 (2020), p. 106480.

[15] R. Madani, S. Sojoudi, and J. Lavaei. "Convex relaxation for optimal power flow problem: Mesh networks". In: *IEEE Transactions on Power Systems* 30.1 (2014), pp. 199–211.

[16] S. Babaeinejadsarookolaee et al. *The Power Grid Library for benchmarking AC optimal power flow algorithms. arXiv:1908.02788*. Aug. 2019.

[17] C. Coffrin et al. "PowerModels.jl: An open-source framework for exploring power flow formulations". In: *20th Power Systems Computation Conference (PSCC)*. 2018, pp. 1–8.

[18] A. Wächter and L. T. Biegler. "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming". In: *Mathematical programming* 106.1 (2006), pp. 25–57.

[19] B. O'Donoghue et al. "Conic optimization via operator splitting and homogeneous self-dual embedding". In: *Journal of Optimization Theory and Applications* 169.3 (June 2016), pp. 1042–1068.

[20] T. Kluyver et al. "Jupyter Notebooks – A publishing format for reproducible computational workflows". In: *Positioning and Power in Academic Publishing: Players, Agents and Agendas*. Ed. by F. Loizides and B. Schmidt. IOS Press. 2016, pp. 87–90.

[21] T. W. K. Mak et al. "Learning regionally decentralized AC optimal power flows with ADMM". In: *arXiv:2205.03787* (May 2022).