Scalable Solutions for Security-Constrained Optimal Power Flow with Multiple Time Steps

Hussein Sharadga, Javad Mohammadi, Constance Crozier, Kyri Baker

***Abstract*—This work introduces an innovative approach to scaling security-constrained optimal power flow problems to large power grids with multi-timestep, addressing the significant challenges associated with managing millions of continuous and integer optimization variables as well as nonlinear and nonconvex constraints. Through a strategic combination of problem reformulation, linearization methods, constraint-relaxation techniques, and sequential optimization, the complexities inherent to large power grid optimization are effectively navigated. The proposed methodology enables the resolution of complex power grid models with strict time constraints while attaining high-quality solutions. Demonstrating remarkable robustness, the novel approach consistently surpasses established benchmark methods.**

***Index Terms*—Power grid simulation, security-constrained optimal power flow, large-scale grid optimization.**

I. Introduction

A. Motivation

The electrical grid remains a remarkable engineering achievement of the 20th century [1], [2]. Yet, maintaining effective power distribution across the grid continues to be a significant challenge. With the increasing complexity of modern power systems, the pursuit of robust solutions becomes paramount. In efforts to improve grid security, the U.S. Department of Energy (DOE) is actively supporting initiatives to advance grid technologies [3]. These initiatives focus on modernizing grid operating models and tools, integrating innovative methodologies and technologies to address the grid's evolving complexities. This ensures a reliable, resilient, and secure power grid.

The optimal power flow (OPF) model serves as the foundation of electrical grid operations [4], [5], which optimizes the setting for generators, load consumption, energy storage, and parameters related to grid control. By integrating security considerations and incorporating contingencies into the operational framework, security-constrained optimal power flow (SCOPF) enhances the security of the power grid [6].

Manuscript received 11-Apr-2024.

Financial support for this research is provided by the US Advanced Research Projects Agency-Energy (ARPA-E), under grant number DE-AR0001646.

Hussein Sharadga is with the School of Engineering, Texas A&M International University, Laredo, TX, USA, and the Department of Civil, Architectural, and Environmental Engineering, The University of Texas at Austin, Austin, TX, USA (e-mail: hssharadga@tamu.edu). Javad Mohammadi is with the Civil, Architectural and Environmental Engineering, The University of Texas at Austin, Austin, Texas, US (e-mail: javadm@utexas.edu). Constance Crozier is with Industrial and Systems Engineering, Georgia Institute of Technology, Georgia, USA (e-mail: ccrozier8@gatech.edu). Kyri Baker is with Civil, Environmental, and Architectural Engineering, University of Colorado, Boulder, Colorado, USA (e-mail: kyri@colorado.edu).

Established methods for solving SCOPF were developed during a period when computing power was limited and expensive. These methods often rely on linear approximations, particularly in the form of the DC optimal power flow (DCOPF), ignoring voltage and reactive power altogether [7]. Despite advancements, widely used optimization software and tools are still based on the linear assumptions of the classical DCOPF problems. Notably, commercial tools come short of solving the comprehensive AC power flow problem [3]. To address this, the DOE launched the Grid Optimization (GO) challenges, aiming to solve the complete AC power flow problem with unit commitment and security constraints [1].

B. Literature Review

Recent advancements in both computation and methodology present an opportunity for significant improvements in SCOPF calculations. The substantial progress in computational capabilities and optimization solvers has spurred research into novel methodologies for grid operation and innovative strategies to tackle SCOPF and associated grid challenges [8]. Commercial optimization solvers like GUROBI and CPLEX have seen a significant speed increase, exceeding three orders of magnitude on the same hardware configurations [9]. Cloud computing, with its capacity to harness these gains, is garnering increasing attention within the power grid operation community [10].

Concurrently with this trend and departing from conventional linear assumptions, ongoing research is exploring new techniques such as second-order cone programming (SOCP), quadratic convex (QC), and semi-definite (SDP) methods [11]–[15]. However, SDP relaxation often fails to produce meaningful physical solutions to OPF problem for practical systems [16]. Current solvers continue to face efficiency challenges, particularly when dealing with larger systems and mixed integer problems, as opposed to those tailored for linear problems and second-order cone formulations. Conversely, QC relaxation, as demonstrated in [17] and [18], offers greater computational efficiency and ease of implementation. The proposed QC relaxation techniques in [17] and [18] can generate solutions that are nearly optimal when tested using 95% of the grid networks in the literature. In a recent study [19], a new semidefinite bound tightening method is proposed to address the remaining percentage. To further tighten state-of-the-art relaxations, a new tight quadratic relaxation was proposed in [20] to relax the trigonometric functions of the OPF problems. Authors in [21] propose another high-quality relaxation technique that is based on adaptive piecewise relax-

ation. Furthermore, Castillo et al. [22] utilized the outer approximation method to address the nonconvexity of AC power flow constraints. This method relaxes nonconvex problems into linear forms. It was applied to AC unit commitment problems and tested on the IEEE 118-bus system.

Many investigations indicate that addressing SCOPF problems encompassing both pre-contingency and post-contingency conditions can be enhanced by advancements in decomposition algorithms [23] and stochastic optimization methods [24]. The proposed methods utilize a similar dual-stage problem-solving process [25], creating opportunities to operate complex power grids more effectively. For instance, the authors in [26] addressed security-constrained unit commitment with AC power flow using Benders decomposition, which separates the unit commitment (UC) problem from the AC security constraints. The approach is demonstrated using a six-bus system and the IEEE 118-bus system. Nogales et al. proposes a decomposition methodology that breaks down the optimal power flow problem into multi-areas [27]. This methodology presents a simple procedure for computing a suboptimal coordinated but decentralized solution for electric energy systems with 708 buses, particularly suitable for independent system operators. A similar multi-area optimal power flow method is introduced in [28], leveraging a modified column-and-constraint generation algorithm to manage a 1008-bus power grid. However, further enhancements are necessary to effectively address the complex interconnected structures inherent in real power grid systems.

Recent research has explored an extended formulation of the AC optimal power flow problem with security constraints and multiple time steps, applied to 60- and 304-bus systems, as discussed in [29], [30]. Recent studies also focuses on tackling the GO challenges. The authors in [31] demonstrated that a straightforward algorithm designed solely to satisfy unit commitment, reserve, and AC power balance constraints can achieve high-quality solutions for the AC unit commitment problems within the GO challenges. Chevalier developed a parallelized Adam-based numerical solver to tackle these challenges in [32]. This solver parallelizes backpropagation and variable projection processes, leveraging parallel computing hardware to solve large-scale problems quickly and is designed to be hyper-scalable.

C. Contributions

This paper introduces our innovative decomposition strategy combined with a hybrid solver approach to address the challenges associated with multi-timestep security-constrained optimal power flow problems in large-scale power grids, which are part of the Grid Optimization (GO) challenges, a US Department of Energy (DOE) initiative [33]. Our contributions include the development of a unique two-stage strategy to deliver a rapid, and robust solution for the entire AC optimal power flow problems, while simultaneously solving for both real and reactive power, as well as device switching. This approach does not utilize any of the aforementioned ACOPF relaxations, as it solves the complete AC problem directly, adhering to the strict tolerance of 10^{-8} for constraint violations. The proposed approach incorporates an innovative decomposition technique, which partitions the problem into two modules (the DC module and the AC module), which are solved sequentially. Through the utilization of solvers tailored to each stage and the integration of reformulation techniques, significant improvements in efficiency are achieved in our solution process.

The DC module, which is linear and involves integer variables, is solved by Gurobi, while the AC module, which is nonconvex but involves only continuous variables, is solved by IPOPT. This hybrid method leverages the complementary strengths of fast DC linear approximation (solved by Gurobi) and a detailed, comprehensive AC formulation (solved by IPOPT), resulting in improved performance compared to existing methods.

In a nutshell, our novelty lies in strategic decomposition, the tailored application of solvers, and the innovative reformulation techniques that facilitate solver use, leading to significant advancements in handling complex power flow problems.

II. The SCOPF Problem

A. Problem Formulation

The problem formulation utilized in this article is sourced from the GO challenge, which is an initiative by the Advanced Research Projects Agency-Energy (ARPA-E). This formulation, which is notably comprehensive, consists of a primary formulation spanning 53 pages [33], supplemented by 22 pages of additional information. It encompasses a diverse array of optimization variables and constraints, such as voltage constraints at bus nodes, limits on real and reactive power, zone-specific reserves, and considerations related to device operation and reliability. This involves device statuses, intricate switching considerations, factors governing device uptime and downtime, counts of device starts over multiple time intervals, specifications on maximum and minimum energy generation/consumption over specific multiple time intervals, and constraints on ramping down and up, among others (as illustrated in Figure 1 in [1]). Importantly, it should be noted that these constraints apply to both base and contingency states. The incorporation of contingency constraints ensures the grid's security in mitigating outages and unforeseen events. The OPF problem is succinctly outlined as:

$$\begin{aligned} \max_{p_{jt},\theta_{it}} \quad & z\ (p_{jt},\theta_{it},u_{jt},\ldots) \\ \text{s.t.} \quad & G(p_{jt},\theta_{it},u_{jt},\ldots)=0, \quad H(p_{jt},\theta_{it},u_{jt},\ldots)\leq 0 \end{aligned} \tag{1}$$

The problem formulation aims to maximize the total market surplus, represented by z. Here, p_{jt} is the power of device with index j at time step t, θ_{it} represents the voltage angle of the bus with index i, and the binary variable u_{jt} represents the device switching status. The set of equality constraints is denoted by G, while the set of inequality constraints is denoted by H.

B. Problem Complexity, Data, and Testing Platform

The problem dataset consists of 9 network models, including configurations with 73, 617, 1576, 2000, 4224, 6049, 6708,

6717, and 8316 buses. The 6708-bus network is the only real-world network; the others are synthetic. The dataset is available at [34]. The 73-bus network is relatively simple, involving 72 thousand continuous variables and 11 thousand binary variables. In contrast, the 8316-bus network represents a highly complex test case, requiring the optimization of over 9 million continuous variables and more than 2 million binary variables.

To develop our solution strategy, we utilized the testing platform developed by the Pacific Northwest National Laboratory (PNNL), which takes in GitHub repositories and then executes the code repository on a computing node made available by PNNL. The computing node is equipped with dual 32-core CPUs and local memory. The testing platform calculates the cost function for the returned solution, verifies solution feasibility, and sets the time limit. It is worth noting that our code repository is written in Python.

The testing criteria for the power grid scheduling code are divided into three categories, each with its own set of requirements. Table I outlines the scheduling requirements for the power grid in each category.

TABLE I: Power Grid Scheduling Categories and Testing Criteria Developed by ARPA-E

Category	Time Horizon	Time Limit
C1	8 hours ahead	10 mins
C2	48 hours ahead	2 hours
C3	7-day (168 hours) ahead	3 hours

The problem datasets are divided into Scenario Sets to ensure the code handles diverse scenarios and prevents it from memorizing the testing instances [34]. Scenario Set 3, which corresponds to the E3 set, builds on Scenario Set 2 (E2 set) but includes more diversity in the testing cases.

For scenario indexing in Scenario Sets 3 and 2, the scenarios within each set are indexed based on network size and category. We start with the smallest network and then iterate through each category. Within each category, the scenario with the smallest number is indexed first.

III. The Proposed Solving Strategy

To manage the complexity of a large-scale SCOPF problem and to expedite the solution procedure, we decompose the SCOPF problem into two modules—the DC and AC components—which are solved sequentially. In the first modules, we apply the DC assumptions, where optimization variables consist of both binary and continuous types. Following the resolution of the first subproblem, the binary variables obtained are held constant, while disregarding the continuous variables.

In the AC module, all constraints are incorporated, accurately simulating a realistic grid. Conversely, we opt not to include all constraints in the DC module in order to reduce solving time for the first module. Nonetheless, essential constraints for the AC module feasibility, such as device ramping constraints, are included in the DC module. Omitting these constraints in the DC module could lead to sudden device shutdowns. As a result, the AC module, which considers ramping constraints, may deem the decision to shut down as violating the ramp down limit and therefore infeasible.

The DC module is transformed into a mixed-integer linear problem (MILP), which is then addressed by the Gurobi solver (version 10.0.2). The AC module, however, presents nonconvex and nonlinear (NLP) problems, which are effectively addressed by the IPOPT solver (version 3.12.13).

By decomposing the problem into two modules and then employing specialized solvers for each module, we improve computational efficiency. Figure 1 depicts the two-stage solution approach we propose. In the following sections, novelties introduced for solving each module are discussed.

A. Handling Computations Workload within the DC Module

To efficiently manage the computations workload within the DC module, we employ a variety of methods to enhance computation efficiency and reduce problem complexity.

a) Relaxation of Nonconvex Constraints: This entails utilizing the substitution method and relaxing the equality constraint. More precisely, shunt power (shunt is indicated by 'sh') is expressed as [33]:

$$p_{jt} = g^{sh}_{jt} v^2_{it} \quad \forall j \in j^{sh}_i \tag{2}$$

The energy-balance constraint is expressed as [33]:

$$\sum_{j \in j^{cs,pr}_i} p_{jt} + \sum_{j \in j^{sh}_i} p_{jt} + \sum_{j \in j^{fr}_i} p^{fr}_{jt} + \sum_{j \in j^{to}_i} p^{to}_{jt} = p_{it} \tag{3}$$

Equation (2) is non-linear and non-convex. To address this, we first substitute (2) into (3):

$$\sum_{j \in j^{cs,pr}_i} p_{jt} + \sum_{j \in j^{sh}_i} g^{sh}_{jt} v^2_{it} + \sum_{j \in j^{fr}_i} p^{fr}_{jt} + \sum_{j \in j^{to}_i} p^{to}_{jt} = p_{it} \tag{4}$$

This new constraint, however, is nonconvex. To transform it into a convex, we relax the equality constraint into an inequality constraint:

$$\sum_{j \in j^{cs,pr}_i} p_{jt} + \sum_{j \in j^{sh}_i} g^{sh}_{jt} v^2_{it} + \sum_{j \in j^{fr}_i} p^{fr}_{jt} + \sum_{j \in j^{to}_i} p^{to}_{jt} \leq p_{it} \tag{5}$$

In (5), the right side is the power mismatch, denoted as p_{it}, and given the substantial mismatch penalty, the solver works on minimizing it. When p_{it} matches the counterpart of (5), minimization occurs.

b) Linearizing Convex Constraints: First, we reformulate the convex constraints into quadratic forms. Second, we utilize McCormick envelope relaxation to transform these quadratic terms into linear. McCormick envelope relaxation is explained in [13]. These two steps together significantly reduce computation time, with potential savings of up to 75% for small power grids and yielding even higher efficiency gains for larger power grids. In particular, the branch flow-limit constraint takes the following second-order cone problem (SOCP) format:

$$||p^{fr}_{jt} + q^{fr}_{jt}||_2 \leq s^{max}_j + s^{+}_{jt} \quad \forall t \in T,\, j \in j^{ac} \tag{6}$$

In (6), the right side is linear, while the left side represents the norm function, known for its convexity. Therefore, (6) is

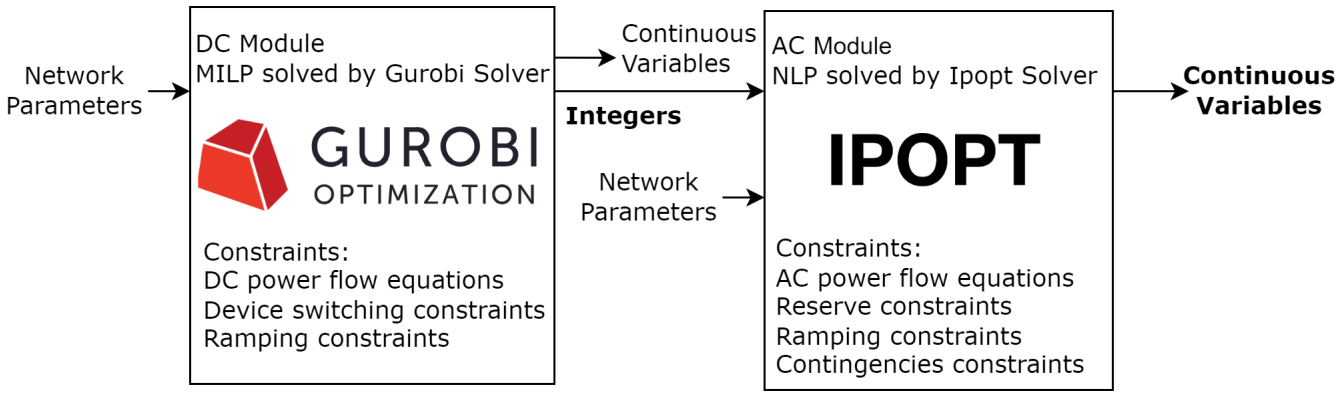


Fig. 1: Our two-stage solution strategy includes DC and AC modules.

considered convex. To transform (6) into a quadratic form, we square both sides as follows:

$$(p_{jt}^{fr})^2 + (q_{jt}^{fr})^2 \leq (s_j^{max})^2 + 2\, s_j^{max}\, s_{jt}^{+} + (s_{jt}^{+})^2 \quad \forall t \in T,\ j \in j^{ac} \tag{7}$$

The constraint in (7) is nonconvex. However, McCormick envelope relaxation linearizes the quadratic terms.

c) MILP Problem with Sparse Constraint Matrix and Slack Variables: Inherent feasibility characterizes the MILP as formulated; however, the incorporation of slack variables enhances the solver's convergence speed. These are present in a limited number constraints, leading to a more sparsely populated constraint matrix with increased zero entries. This sparsity promotes faster convergence for the solver. For instance, the problem can be solved by setting the power mismatch and branch flow limits violation to zero and removing them from the problem formulation. However, it has been observed that leaving them as variables helps the solver converge faster.

d) Reduction of Lengthy Linear Expressions: Numerous devices are connected to the same bus, leading to a long power-balance constraint expressed as a lengthy linear equation. To address this issue, additional variables are introduced. Although this expands the pool of optimization variables, it enhances the convergence speed. For instance, (3) is updated as follows:

$$P_{jt}^{cs,pr} + P_{jt}^{sh} + P_{jt}^{fr} + P_{jt}^{to} = p_{it} \tag{8}$$

where:

$$\begin{aligned} P_{jt}^{cs,pr} &= \sum_{j \in j_i^{cs,pr}} p_{jt} \\ P_{jt}^{sh} &= \sum_{j \in j_i^{sh}} p_{jt} \\ P_{jt}^{fr} &= \sum_{j \in j_i^{fr}} p_{jt}^{fr} \\ P_{jt}^{to} &= \sum_{j \in j_i^{to}} p_{jt}^{to} \end{aligned} \tag{9}$$

The introduction of these new variables results in a more concise representation of the power balance constraint (3), thereby improving computational efficiency and speeding up convergence.

e) Reformulation of Downtime-Dependent Device Startup Costs: We structured these costs using clique constraints. Our selection criteria were not based solely on minimizing the number of variables and constraints, but rather on the alignment with Gurobi's workflow. In particular, we chose the clique constraints because of its compatibility with Gurobi's pre-solver, which leverages the structure of the optimization problem to eliminate variables and constraints.

The process of determining downtime-dependent startup costs is analogous to finding the first occurrence of an item in a vector [33]. Assuming the vector x represents binary variables indicating shutdown status, we introduce another binary variable B. B will be 1 the first time the value 1 appears in x and 0 otherwise. B can be formulated using clique constraints:

$$\begin{aligned} B_t &\leq x_t \quad \forall t \\ B_t &\geq x_t - \sum_{\tau<t} x_\tau \quad \forall t \\ B_t &\leq 1 - \sum_{\tau<t} B_\tau \quad \forall t \end{aligned} \tag{10}$$

f) Max/Min Constraint Functions Reformulation: There are two approaches to formulating maximum/minimum equality constraints for compatibility with the Gurobi solver. The first method involves linearizing these constraints, as we will demonstrate. Alternatively, the constraints can be formulated into MIP problem. In this investigation, we found that the MIP formulation facilitates faster convergence for the Gurobi solver, as it aligns more effectively with Gurobi's solution workflow.

For instance, consider the synchronized reserve requirement for reserve zone n as given below:

$$P_{nt}^{scr,req} = \sigma_n^{scr} \max_{j \in j_n^{pr}} p_{jt} \tag{11}$$

This constraint is nonconvex; the nonlinear equality constraint is nonconvex. Gurobi's solver replaces the max equality constraint with an equivalent MIP formulation. Another proposed method is a linear version of the model as shown below. The reserve balance constraint is stated as follows:

$$\sum_{j \in j_n^{pr,cs}} (p_{jt}^{rgu} + p_{jt}^{scr}) + p_{nt}^{\text{scr, }+} \geq p_{nt}^{\text{rgu, req}} + p_{nt}^{\text{scr, req}} \tag{12}$$

By substituting (11) into (12), (11) is eliminated from the problem formulation:

$$\sum_{j\in j_n^{pr,cs}} (p_{jt}^{rgu} + p_{jt}^{scr}) + p_{nt}^{\text{scr, }+} \geq p_{nt}^{\text{rgu, req}} + \sigma_n^{scr} \max_{j\in j_n^{pr}} p_{jt} \quad (13)$$

A linearized version is shown below:

$$\sum_{j\in j_n^{pr,cs}} (p_{jt}^{rgu} + p_{jt}^{scr}) + p_{nt}^{\text{scr, }+} \geq p_{nt}^{\text{rgu, req}} + \sigma_n^{scr} p_{jt} \quad \forall j \in j_n^{pr} \quad (14)$$

However, the solver converges faster if the max equality constraint is formulated as an MIP problem using Gurobi. There are two ways to reformulate the max function into a MIP problem: one method is based on using the big M-value; however, Gurobi utilizes the SOS condition to eliminate the need for introducing the big M-value [35].

B. Handling Computations Workload within the AC Module

In this section, we discuss how we manage the computational workload of the AC module by implementing various techniques. Importantly, the complete AC model is addressed without resorting to relaxation or linearization, ensuring a thorough and precise portrayal of the power grid. Below, we provide detailed insights into these techniques:

a) Managing Uncontrollable Loads: We thoroughly examined the devices to identify the uncontrollable variables. Then, these variables are eliminated to reduce the problem size. Specifically, the device power is partitioned into bid blocks $(m \in M)$, with each block assigned a corresponding cost. As a result, The device total power is determined by adding up the power values for every block:

$$p_{jt} = \sum_{m\in M_{jt}} p_{jtm} \quad \forall j \in j_i^{cs,pr} \quad (15)$$

Consider an example of a consumption device with the following bid blocks:

$$\text{Bid Blocks} = \begin{bmatrix} 10,000 & 0.1 \\ 5,000 & 0.2 \\ 1,000 & 0.3 \\ 500 & 0.4 \end{bmatrix} \quad (16)$$

The left column represents the cost of purchase, and the right column represents the maximum power of each bid block. Our goal is to maximize the objective function, so we begin by processing the bid blocks, restoring the block with the highest purchase value on top.

In this context, if the device's lower power bound exceeds the maximum power that can be allocated to the first block $(m = 1)$ in the bid blocks, the power allocation within this block becomes uncontrollable. Consequently, the power for the first block is set to its upper limit. For this device, the lower bound is given to be 0.15, and the upper bound is 0.5. Since the lower bound is 0.15 and the maximum power that can be allocated to the first block is 0.1, p_{jtm} for the first block $(m = 1)$ is 0.1. Consequently, no optimization variable is introduced for the first bid block. This indicates that the power in the first bid block is uncontrollable. To respect the device power bounds, we set the minimum power in the second bid to 0.05, the maximum power in the third bid to 0.2, and the power in the last bid to zero. Therefore, no optimization variable is introduced for the last bid block. This approach allows us to reduce the number of optimization variables needed specifically to represent this device's bid blocks by 50%.

b) Optimization of Power Reserves- Post-Simulation: Initially, reserves are excluded from the primary optimization process. Instead, following IPOPT's simulation completion, power values are adjusted to meet reserve requirements. This approach significantly reduces the complexity of the problem. To determine the reserve requirements for each zone, an offline calculation was performed. Then the available reserves were deployed from least to highest cost until either the requirements were met or the reserve product run out. In the GO challenges there were several swaps that could be made, for example replacing spinning reserves for non-spinning reserves. Therefore, the reserve products were allocated from lowest to highest quality (as the lowest quality products could not be used for another service). In the vast majority of cases, this was sufficient to meet all reserve requirements. However, it should be noted that the success of this strategy relies on an abundance of reserve products being available.

c) Analytical Derivatives Computation: The derivatives of the Hessian and Jacobian matrices are calculated analytically. Utilizing these analytical derivatives accelerates IPOPT's convergence compared to relying on automatic differentiation. However, generating these analytical-based matrices requires more manual code development.

d) Structural Representation of Jacobian & Hessian Matrices: When providing the analytical Hessian and Jacobian matrices, we only include data about non-zero elements and its corresponding coordinates, outlined by the matrix structure. This method avoids the necessity of specifying details for zero elements, thereby minimizing both the computations workload and data storage requirements. Consequently, it expedites the convergence.

e) Utilizing Vectorized Form: This approach reduces the amount of information that needs to be stored, speeds up data access, and eliminates the need for if statements and loops, thereby improving computation speed. This optimization becomes especially crucial when dealing with complex and large equations, where efficient data handling is paramount for performance. As an illustrative example, consider the derivative of power in the AC branch originating from bus i with respect to voltage [33]:

$$\frac{dp_{jt}^{fr}}{dv} = -2u_{jt}^{on}(g_j^{sr} + g_j^{fr})\frac{v_{it}}{\tau_{jt}^2} \quad \forall j \quad (17)$$

The above equation is part of the following equation:

$$\sum_{j\in j_i^{cs,pr}} \frac{dp_{jt}}{dv} + \sum_{j\in j_i^{sh}} \frac{dp_{jt}}{dv} + \sum_{j\in j_i^{fr}} \frac{dp_{jt}^{fr}}{dv} + \sum_{j\in j_i^{to}} \frac{dp_{jt}^{to}}{dv} = \frac{dp_{it}}{dv} \quad (18)$$

Equation (17) applies to every branch j. The subscript i refers to the bus to which the branch is connected. To iterate over branches and reduce the number of information calls, we

introduce a matrix-vector storing the branches originating from the buses, which we refer to as the "bus-branch" matrix. Rows represent different buses, and each column corresponds to a branch index. The element in row r and column c is set to one if the bus with index r is connected to the branch with index c and power is originating from bus r, otherwise, it is set to zero. A similar matrix is concatenated for each time step t in the control horizon. (17) is updated as follows:

$$\frac{dp_{jt}^{fr}}{dv} = -2 \times \textit{bus-branch} \cdot (u_{jt}^{on}(g_j^{sr} + g_j^{fr}) \times \frac{1}{\tau_{jt}^2}) \times v_{it} \quad (19)$$

This streamlined approach not only enhances computational efficiency but also simplifies code readability, facilitating maintenance and future development. By leveraging vectorized form, we optimize performance while maintaining code clarity and conciseness.

f) Managing Ramping Bounds in Sequential Optimization: This technique allows problem-solving through sequential optimization, particularly for networks with a significant scale, typically exceeding 2000 buses. Recognizing the impracticality of tackling such large networks within the allocated time windows as a whole, we adopt a sequential approach. By breaking down the problem into manageable steps, we can effectively address it. To maintain adherence to ramping limits, we recursively update the bounds for each step. Figs. 2 and 3 illustrate this updating process, which involves dual steps.

The bounds are adjusted in stage 1 as we progress forward, employing the following equations:

$$\overline{P}_{t,new} = \min[\overline{P}_t, \overline{P}_{t+1} + P_{rd}] \quad (20)$$

$$\underline{P}_{t,new} = \max[\underline{P}_t, \underline{P}_{t+1} - P_{ru}] \quad (21)$$

The bounds undergo a second stage of adjustment, moving backward:

$$\overline{P}_{t,final} = \min[\overline{P}_{t,new}, \overline{P}_{t-1,new} + P_{ru}] \quad (22)$$

$$\underline{P}_{t,final} = \max[\underline{P}_{t,new}, \underline{P}_{t-1,new} - P_{rd}] \quad (23)$$

g) Selective Inclusion of Contingency Constraints: Due to the problem's size and constraints imposed by computational limits, we prioritized the inclusion of only the contingency constraints related to the energy-balance equation. This decision was made because there is a high penalty on power mismatch, denoted as p_{it} in (3). These contingency constraints are formulated explicitly within the AC module.

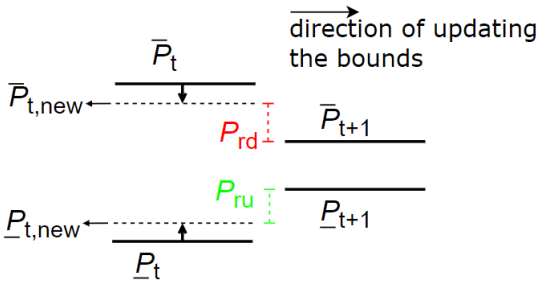


Fig. 2: Adjusting the bounds in step 1 to maintain compliance with the ramping limit while solving the problem sequentially.

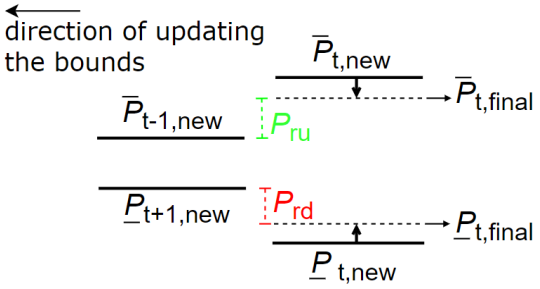


Fig. 3: Adjusting the bounds in step 2 to maintain compliance with the ramping limit while solving the problem sequentially.

C. Optimal Allocation of Computing Resources

Compared to the process of solving the DC module, solving the nonconvex AC module requires substantial computational time. To refine the proposed two-stage strategy, we allocate third of the computation time to solve the DC module, followed sequentially by two-thirds dedicated to addressing the complexities of the AC module. This allocation maintains effective coordination between the DC and AC modules, enabling timely problem resolution.

D. Sequential Optimization

For systems with over 2000 buses, the scale of the problem prohibits its resolution as a whole. Similar to smaller networks, the problem is first decomposed into DC/AC subproblems. The DC subproblem is solved first, followed by the AC subproblem. However, in these large networks, the AC subproblem is divided into stages; the control horizon of the AC module is divided into stages, with each stage's output serving as the starting state for the next stage.

In the DC module for these large networks, the devices are allowed two switching events: the initial one at the start of the control horizon, and the second after the minimum uptime constraint has been met. The minimum uptime constraint ensures that that devices remain operational for a set duration before they are switched off. By limiting the number of switching events to two, we reduce the number of binary variables needed to represent the device status.

To prevent ramping limits violations, the bounds are recursively updated before initiating the sequential optimization process.

IV. Performance Evaluation

The solving strategy we proposed was tested with over 300 scenarios, which were meticulously chosen by domain experts and research teams from multiple organizations [3]. The score assessment aims to create a model that can generate high-quality solutions within specified time constraints across various scenarios. We aimed to find a balance between achieving high scores and meeting stringent time limitations.

To achieve this, we introduce the scaled score metric, representing the ratio between the objective function generated by our code and the best objective function achieved by other available codes. Scaled scores are calculated for Scenario Set 3 and Scenario Set 2, as shown in Figures 4 and 5. A scaled

score of 1 represents the best-known solution, whereas a score of 0.99 indicates a deviation of 1% from the highest achieved score.

A. Score Comparison

Upon examination of Figs. 4 and 5, it becomes apparent that the scaled score for most scenarios exceeds 0.95. Furthermore, the average scaled score exceeds 0.98, which highlights the resilience of our strategy in adeptly managing a wide range of scenarios.

More precisely, our code was evaluated using Scenario Set 3 and Scenario Set 2, which includes 137 scenarios and 197 scenarios, respectively. In Scenario Set 3, the scaled score was 0.99 or higher in 42% of scenarios, compared to 53% in Scenario Set 2. Additionally, the scaled score was 0.95 or higher in 70% of scenarios for both Scenario Set 3 and Scenario Set 2. The average scaled score across all scenarios in Scenario Set 3 is 0.983, and for Scenario Set 2, it is 0.9992. Table II provides the distribution of these scores.

The dataset encompassed a variety of network models, spanning different sizes and levels of sophistication. It comprised configurations with 73, 617, 1576, 2000, 4224, 6049, 6708, 6717, and 8316 buses, representing a broad spectrum of network scales. Notably, the network with 6708 buses stands as the sole real-world network, while the remainder are simulated. Across 45 scenarios within the industrial network (as depicted in Fig. 6), the scaled score approached near optimality (close to 1) in approximately 39 scenarios. In this context, a scaled score close to 1 indicates the proposed model effectively converges to the available best solution. Out of the the six remaining scenarios, four achieved scaled scores ranging from 0.90 to 0.95, while the remaining two scenarios had scaled scores slightly below 0.9. With an average scaled score of 0.9719, our solution further demonstrates its robust performance in addressing large-scale industrial networks.

B. Performance Across Network Sizes

The distribution of scaled scores across different network sizes in Scenario Set 3 is depicted in Fig. 7. Networks with 73 and 1576 buses exhibit wider distributions, indicating greater variability in the scores. Conversely, the distributions for the

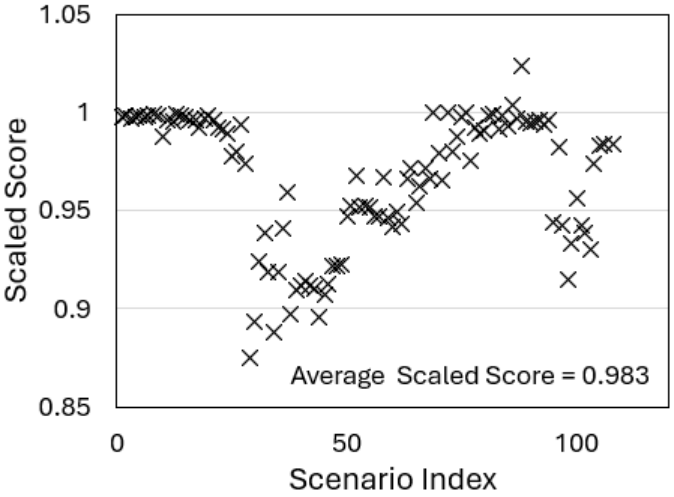
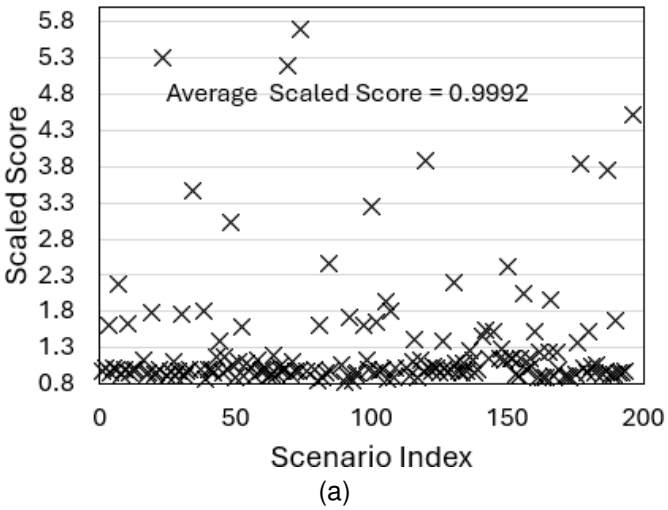


Fig. 4: Proposed approach performance in Scenario Set 3 across various network sizes and scenarios. Scaled score: Our code's score / top score achieved.



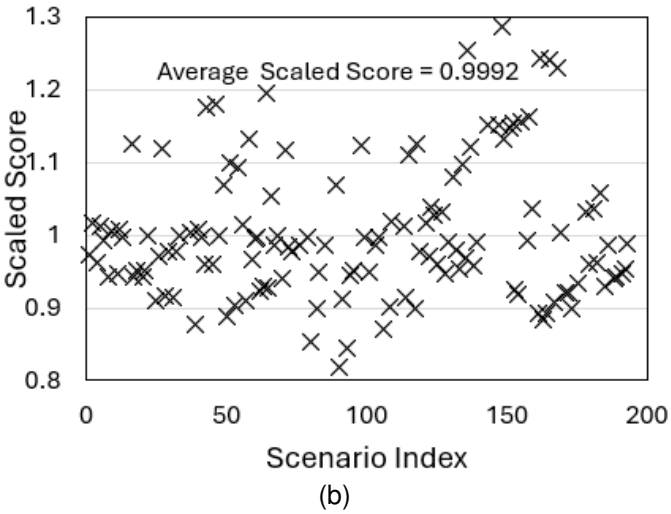


Fig. 5: Proposed approach performance (a) scenarios set 2, (b) a magnified view of scenarios set 2, compared with top score.

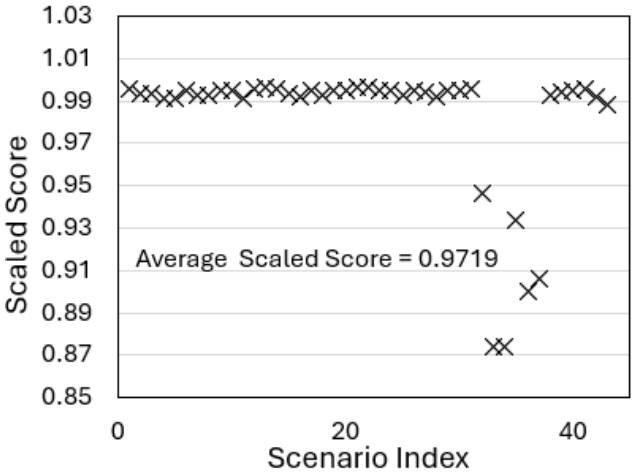


Fig. 6: Proposed approach performance in 6708-bus industrial grid, compared to the top score.

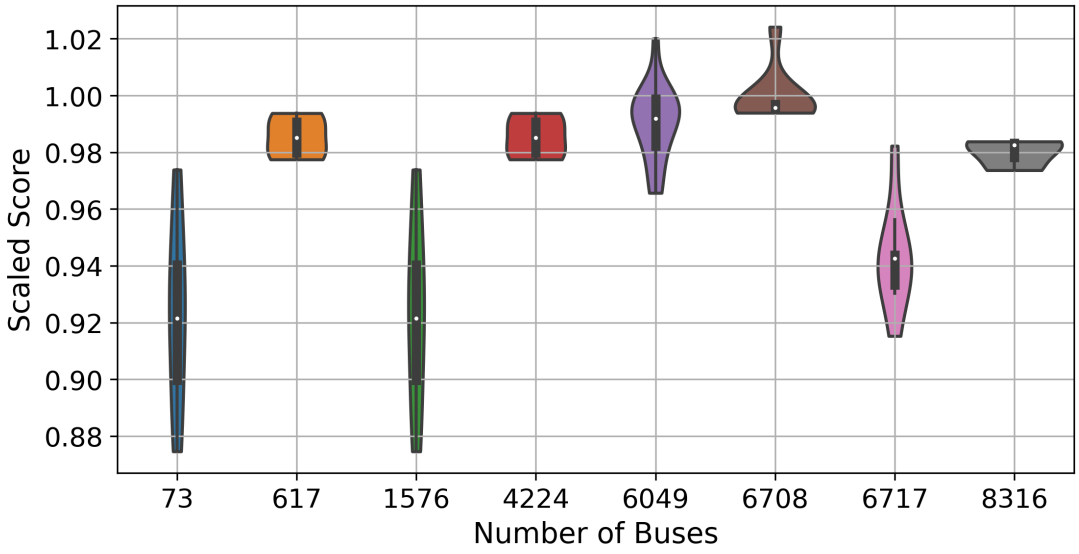


Fig. 7: Exploring scaled score distributions across different network sizes using violin plots for Scenario Set 3: insights into the proposed tool performance and scalability.

remaining networks tend to be narrower and more concentrated around the median, suggesting a more consistent performance.

Fig. 8 illustrates the distribution of scaled scores across different network sizes in Scenario Set 2. Notably, the tool consistently produces scaled scores with an average value of 1, indicating its robustness across various scenarios and network sizes. Additionally, for certain scenarios, the tool yields relatively high scores that significantly exceed the best available solution, as evidenced by the elongation of the violin plot. This suggests the tool's capability to achieve exceptional performance under specific conditions, offering valuable insights into its effectiveness and potential for optimizing the power grid.

C. Time Analysis Across Network Sizes

The time for the solver completing the task within three categories (C1, C2, and C3) for different network sizes is visually depicted in Fig. 9. Notably, the network with 1576 buses requires the highest time across all categories. It is worth mentioning that networks with more than 2000 buses consume relatively less time compared to smaller networks. This can be attributed to breaking the control horizon for these networks into steps and solving them using sequential optimization, reducing the complexity of the problem and thus computation time. Although we still need to solve for every stage, it is faster than solving the problem as a whole.

These insights provide valuable guidance into the relationship between the network size and performance metrics, aiding in the optimization of resource allocation and operational efficiency within the transmission system.

TABLE II: Scaled Score Distribution for Scenario Sets 3 & 2 [1]

Range	% **Scenarios**	
	Set 3	**Set 2**
Scaled Score $\geq$ 0.99	42	53
0.99 $>$ Scaled Score $\geq$ 0.97	16	10
0.97 $>$ Scaled Score $\geq$ 0.95	12	9
0.95 $>$ Scaled Score $\geq$ 0.93	12	9
0.93 $>$ Scaled Score $\geq$ 0.90	13	11
Scaled Score $<$ 0.9	5	8

D. Algorithm Design Insights

The computation speed-up resulting from deploying different techniques to solve our optimization problem for category C1 on the 73-bus system is presented in Table III. For this analysis, we used a personal machine (a Lenovo X1 Carbon laptop with an Intel(R) Core(TM) i7-8665U CPU @ 1.90 GHz, 2.11 GHz and 16.00 GB RAM). In this context, the speed-up is defined as the ratio of the computation time of the original formulation to the computation time of the proposed formulation. The problem formulation for the 73-bus network involves 72 thousand continuous variables, 11 thousand binary variables, and 94 thousand constraints. The proposed techniques significantly reduce the required computation time in the C1 category. Moreover, the speed-up is even more significant in categories C2 and C3, given the problem size. The speed-up is also more substantial for larger networks.

While the proposed model proves to be fast and robust across a wide range of scenarios and network sizes, the quality of the solutions varies across different scenarios. This inconsistency may stem from differences in non-linearity levels and the length of the control horizon. In our future work, we will explore mitigation strategies to enhance the reliability of our proposed solution.

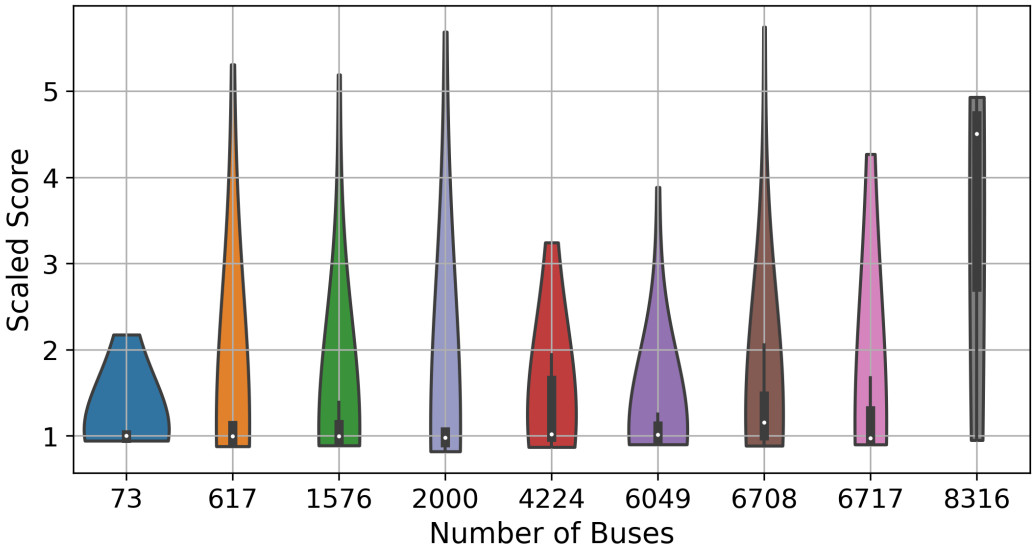


Fig. 8: Exploring scaled score distributions across different network sizes using violin plots for Scenario Set 2: insights into the proposed tool performance and scalability.

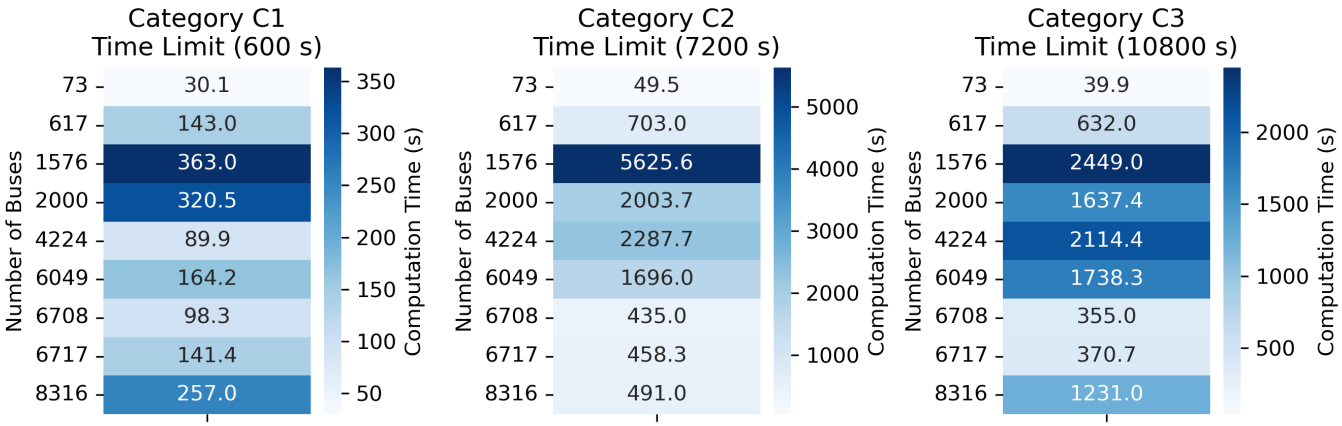


Fig. 9: Comparison of computation time across varying network sizes and testing categories: the tool demonstrates robustness in adhering to time constraints. The computation time displayed in the heatmap represents the maximum computation time recorded across different scenarios for each network and within a given category.

TABLE III: Speed-Up Achieved by Employing Different Techniques. We considered the C1 Category on the 73-bus system. The speed-up is defined as the solution time of the original formulation divided by the run time of the revised problem.

Employed Techniques	**Speed-Up**
Nonconvex Constraints Relaxation	12.25x
Convex Constraints Linearization	4.12x
Slack Variables	1.75x
Reduction of Lengthy Linear Expressions	3.75x
Downtime-Dependent Startup Costs Reformulation	1.78x
Max/Min Constraint Reformulation	1.95x
Managing Uncontrollable Loads	4.32x
Analytical Derivatives	9.89x

V. Conclusion

In conclusion, our study focuses on the development of a highly effective approach for tackling the complexities of power grid optimization. Scaling security-constrained optimal power flow in large power grids with multi-timestep presents significant challenges, primarily attributable to the inclusion of millions of variables and constraints, involving integer variables and intricate, nonconvex characteristics. Our proposed strategy, which integrates innovative methodologies such as problem reformulation, linearization methods, constraint-relaxation techniques, and sequential optimization, has emerged as a robust solution for managing these complexities within strict time limitations, and providing high-quality solutions.

By strategically dividing the problem into two modules, namely the DC module and the AC module, and carefully managing computational workload, we have significantly enhanced the proposed strategy's efficiency. The thorough perfor-

mance evaluation conducted validates the proposed strategy's resilience in managing the challenges intrinsic in optimizing large-scale power grids. Importantly, our approach has demonstrated its capability to handle diverse power grid sizes, affirming its versatility and suitability for real-world applications.

ACKNOWLEDGMENTS

During the preparation of this work, the authors used ChatGPT to enhance readability. Subsequently, they thoroughly reviewed and edited the content as necessary, taking full responsibility for the publication's content.

REFERENCES

[1] H. Sharadga, J. Mohammadi, C. Crozier, and K. Baker, "Optimizing Multi-Timestep Security-Constrained Optimal Power Flow for Large Power Grids," *2024 IEEE Texas Power Energy Conf.*, 2024, doi: 10.1109/TPEC60005.2024.10472229.

[2] G. B. Giannakis, V. Kekatos, N. Gatsis, S. J. Kim, H. Zhu, and B. F. Wollenberg, "Monitoring and optimization for power grids: a signal processing perspective," *IEEE Signal Process. Mag.*, vol. 30, no. 5, pp. 107–128, 2013, doi: 10.1109/MSP.2013.2245726.

[3] ARPA-E, "Grid optimization challenge 3," https://gocompetition.energy.gov/; last accessed March 12, 2024.

[4] M. Gao, J. Yu, Z. Yang, and J. Zhao, "A Physics-Guided Graph Convolution Neural Network for Optimal Power Flow," *IEEE Trans. Power Syst.*, vol. 39, no. 1, pp. 380–390, 2024, doi: 10.1109/TPWRS.2023.3238377.

[5] J. Mohammadi, G. Hug, and S. Kar, "Agent-based distributed security constrained optimal power flow," *IEEE Trans. Smart Grid*, vol. 9, no. 2, pp. 1118–1130, 2018, doi: 10.1109/TSG.2016.2577684.

[6] F. Capitanescu *et al.*, "State-of-the-art, challenges, and future trends in security constrained optimal power flow," *Electr. Power Syst. Res.*, vol. 81, no. 8, pp. 1731–1741, 2011, doi: 10.1016/j.epsr.2011.04.003.

[7] J. K. Skolfield and A. R. Escobedo, "Operations research in optimal power flow: A guide to recent and emerging methodologies and applications," *Eur. J. Oper. Res.*, vol. 300, no. 2, pp. 387–404, 2022, doi: 10.1016/j.ejor.2021.10.003.

[8] P. Panciatici *et al.*, "Advanced optimization methods for power systems," *Proc. - 2014 Power Syst. Comput. Conf. PSCC 2014*, pp. 1–18, 2014, doi: 10.1109/PSCC.2014.7038504.

[9] T. Koch *et al.*, "MIPLIB 2010: Mixed integer programming library version 5," *Math. Program. Comput.*, vol. 3, no. 2, pp. 103–163, 2011, doi: 10.1007/s12532-011-0025-9.

[10] M. Yigit, V. C. Gungor, and S. Baktir, "Cloud Computing for Smart Grid applications," *Comput. Networks*, vol. 70, pp. 312–329, 2014, doi: 10.1016/j.comnet.2014.06.007.

[11] J. Lavaei and S. H. Low, "Zero duality gap in optimal power flow problem," *IEEE Trans. Power Syst.*, vol. 27, no. 1, pp. 92–107, 2012, doi: 10.1109/TPWRS.2011.2160974.

[12] S. H. Low, "Convex relaxation of optimal power flow - Part i: Formulations and equivalence," *IEEE Trans. Control Netw. Syst.*, vol. 1, no. 1, pp. 15–27, 2014, doi: 10.1109/TCNS.2014.2309732.

[13] A. Miro´ *et al.*, "Deterministic global optimization algorithm based on outer approximation for the parameter estimation of nonlinear dynamic biological systems," *BMC Bioinformatics*, vol. 13, no. 90, 2012. doi: 10.1186/1471-2105-13-90.

[14] R. Madani, S. Sojoudi, and J. Lavaei, "Convex relaxation for optimal power flow problem: Mesh networks," *IEEE Trans. Power Syst.*, vol. 30, no. 1, pp. 199–211, 2015, doi: 10.1109/TPWRS.2014.2322051.

[15] D. K. Molzahn, J. T. Holzer, B. C. Lesieutre, and C. L. DeMarco, "Implementation of a large-scale optimal power flow solver based on semidefinite programming," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 3987–3998, 2013, doi: 10.1109/TPWRS.2013.2258044.

[16] B. C. Lesieutre, D. K. Molzahn, A. R. Borden, and C. L. DeMarco, "Examining the limits of the application of semidefinite programming to power flow problems," *2011 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 1492–1499, 2011, doi: 10.1109/Allerton.2011.6120344.

[17] C. Coffrin, H. L. Hijazi, and P. Van Hentenryck, "The QC Relaxation: A Theoretical and Computational Study on Optimal Power Flow," *IEEE Trans. Power Syst.*, vol. 31, no. 4, pp. 3008–3018, 2016, doi: 10.1109/TPWRS.2015.2463111.

[18] H. Hijazi, C. Coffrin, and P. Van Hentenryck, "Convex quadratic relaxations for mixed-integer nonlinear programs in power systems," *Math. Program. Comput.*, vol. 9, no. 3, pp. 321–367, 2017, doi: 10.1007/s12532-016-0112-z.

[19] S. Gopinath *et al.*, "Proving global optimality of ACOPF solutions," *Electr. Power Syst. Res.*, vol. 189, p. 10668, 2020.

[20] K. Bestuzheva, H. Hijazi, & C. Coffrin, "Convex relaxations for quadratic on/off constraints and applications to optimal transmission switching," *INFORMS J. Comput.*, vol. 32, no. 3, pp. 682–696, 2020, doi: 10.1287/ijoc.2019.0900.

[21] M. Lu, H. Nagarajan, R. Bent, S. D. Eksioglu, and S. J. Mason, "Tight piecewise convex relaxations for global optimization of optimal power flow," *IEEE Power Syst. Comput. Conf. PSCC 2018*, pp. 1–7, 2018, doi: 10.23919/PSCC.2018.8442456.

[22] A. Castillo, C. Laird, C. A. Silva-Monroy, J.-P. Watson, and R. P. O'Neill, "The Unit Commitment Problem With AC Optimal Power Flow Constraints," *IEEE Trans. Power Syst.*, vol. 31, no. 6, pp. 4853–4866, Nov. 2016, doi: 10.1109/TPWRS.2015.2511010.

[23] H. T. Kahraman, M. Akbel, & S. Duman, "Optimization of Optimal Power Flow Problem Using Multi-Objective Manta Ray Foraging Optimizer," *Appl. Soft Comput.*, vol. 116, p. 108334, 2022, doi: 10.1016/j.asoc.2021.108334.

[24] D. Liu, C. Zhang, G. Chen, Y. Xu, & Z. Y. Dong, "Stochastic security-constrained optimal power flow for a microgrid considering tie-line switching," *Int. J. Electr. Power Energy Syst.*, vol. 134, p. 107357, 2021, doi: 10.1016/j.ijepes.2021.107357.

[25] D. E. Olivares, C. A. Canizares, & M. Kazerani, "A centralized energy management system for isolated microgrids," *IEEE Trans. Smart Grid*, vol. 5, no. 4, pp. 1864–1875, 2014, doi: 10.1109/TSG.2013.2294187.

[26] Y. Fu, M. Shahidehpour, and Z. Li, "Security-Constrained Unit Commitment With AC Constraints," *IEEE Trans. Power Syst.*, vol. 20, no. 2, pp. 1001–1013, 2005, doi: 10.1109/TPWRS.2005.846076.

[27] F. J. Nogales, F. J. Prieto, & A. J. Conejo, "A Decomposition Methodology Applied to the Multi-Area Optimal Power Flow Problem," *Ann. Oper. Res.*, vol. 120, pp. 99–116, 2003, doi: 10.1023/A:1023374312364.

[28] W. Huang, W. Zheng, & D. J. Hill, "Distributionally Robust Optimal Power Flow in Multi-Microgrids with Decomposition and Guaranteed Convergence," *IEEE Trans. Smart Grid*, vol. 12, no. 1, pp. 43–55, 2021, doi: 10.1109/TSG.2020.3012025.

[29] M. I. Alizadeh and F. Capitanescu, "A tractable linearization-based approximated solution methodology to stochastic multi-period AC security-constrained optimal power flow," *IEEE Trans. Power Syst.*, vol. 38, no. 6, pp. 5896-5908, doi: 10.1109/TPWRS.2022.3220283.

[30] M. I. Alizadeh, M. Usman, and F. Capitanescu, "Envisioning security control in renewable dominated power systems through stochastic multi-period AC security constrained optimal power flow," *Int. J. Electr. Power Energy Syst.*, vol. 139, no. 107992, doi: 10.1016/j.ijepes.2022.107992.

[31] R. Parker and C. Coffrin, "Managing Power Balance and Reserve Feasibility in the AC Unit Commitment Problem," *Electric Power Syst. Res.*, vol. 234, no. 110670, 2024, doi:10.1016/j.epsr.2024.110670.

[32] S. Chevalier, "A Parallelized, Adam-Based Solver for Reserve and Security Constrained AC Unit Commitment," *Electric Power Syst. Res.*, vol. 235, no. 110685, 2024, doi: 10.1016/j.epsr.2024.110685.

[33] J. Holzer *et al.*, "Grid optimization challenge 3 problem formulation," https://gocompetition.energy.gov/challenges/challenge-3/formulation; last accessed March 12, 2024.

[34] ARPA-E, "Grid optimization challenge 3 datasets," https://gocompetition.energy.gov/challenges/600650/datasets; last accessed July 28, 2024.

[35] Gurobi Optimization. Documentation. *General Constraint Functions*, 2024. Available online: https://www.gurobi.com/documentation/9.0/refman/constraints.html#subsubsection:GenConstrFunction; last accessed March 14, 2024.