Optimizing Multi-Timestep Security-Constrained Optimal Power Flow for Large Power Grids

Hussein Sharadga
Civil, Architectural and Environmental Engineering
The University of Texas at Austin
Austin, Texas, USA
hssharadga@utexas.edu

Javad Mohammadi
Civil, Architectural and Environmental Engineering
The University of Texas at Austin
Austin, Texas, USA
javadm@utexas.edu

Constance Crozier
Industrial and Systems Engineering
Georgia Institute of Technology
Georgia, USA
ccrozier8@gatech.edu

Kyri Baker
Civil, Environmental, and Architectural Engineering
University of Colorado
Boulder, Colorado, USA
Kyri.Baker@colorado.edu

Abstract—This work proposes a novel method for scaling multi-timestep security-constrained optimal power flow in large power grids. The challenge arises from dealing with millions of variables and constraints, including binary variables and non-convex, nonlinear characteristics. To navigate these complexities, techniques such as constraint relaxation, linearization, sequential optimization, and problem reformulation are employed. By leveraging these methods, complex power grid problems are solved while achieving high-quality solutions and meeting time constraints. The innovative solution approach showcases great robustness and consistently outperforms benchmark standards.

Index Terms—power grid optimization, power grid software, security-constrained optimal power flow, large-scale optimization

I. Introduction

The power grid stands as one of the most significant engineering achievements of the 20th century [1]. However, optimizing power grid dispatch presents a substantial challenge in ensuring efficient and reliable energy distribution. As modern power systems grow increasingly complex, finding robust solutions becomes imperative. To address these challenges and enhance grid resilience, the US Department of Energy (DOE) has been actively involved in initiatives focused on advancing grid technologies. These efforts aim to modernize grid operating models and software, integrating new technologies and methodologies to adapt to the complexities of the evolving grid, ensuring its reliability, resilience, and security.

The optimal power flow (OPF) problem is the core optimization challenge in grid planning and operations [2]. It involves finding the best settings for power generation, demand flexibility, energy storage, and grid control to maximize grid objectives. Security-constrained optimal power flow (SCOPF) adds grid resiliency constraints to this optimization.

Today's industry-standard SCOPF methods were developed in an era of less capable and costlier computers, when general-purpose optimization solvers were in their infancy. To simplify calculations, most often, linearizing assumptions were made, ignoring voltage and reactive power—referred to as the DC optimal power flow (DCOPF) [3]. Despite various improvements, widely-used optimization software, including production cost models, security-constrained unit commitment (SCUC), and security-constrained economic dispatch (SCED) tools, continue to rely on linear OPF assumptions similar to those in a classical DCOPF problem. There are currently no widely-adopted industry tools that use the full AC power flow equations, without linearization, while simultaneously optimizing both real and reactive power generation [4].

Recent computational and methodological advancements suggest the potential for substantial enhancements in SCOPF methodologies. The significant leaps in computational capabilities and the evolution of optimization solvers in recent times have fuelled investigations into innovative strategies for grid operations and novel approaches to address SCOPF and related grid challenges [5]. Moving away from traditional linear approximations, current research focuses on innovative methods such as quadratic convex (QC), semi-definite (SDP), and second-order cone programming techniques [6]–[9]. Nevertheless, it has been observed that SDP relaxation fails to yield a solution with meaningful physical implications when applied to solve OPF for numerous practical systems [10]. Additionally, the existing solvers are not as efficient as those employed for solving linear programming (LP) and second-order cone programming (SOCP) formulations, especially for larger systems and problems involving mixed integer aspects. On the other hand, QC relaxation, as in [11], is more straightforward to implement and computationally efficient.

Also, recent studies suggest that SCOPF problems addressing both pre-contingency and post-contingency states could benefit from emerging research in decomposition methods [12] or stochastic optimization [13] algorithms. These approaches aim to exploit a similar two-stage problem structure, present-

ing new avenues for improving the efficiency and effectiveness of solving complex power system optimization challenges.

In this paper, we present our innovative approach to address the challenges of security-constrained optimal power flow (SCOPF) within the framework of the US DOE's Grid Optimization (GO) initiatives. Our contributions include leveraging a two-pronged approach to provide a fast high-quality solution for the complete AC power flow equations and optimizing both real and reactive power generation simultaneously. Our novel decomposition strategy breaks down the SCOPF problem into sequential sub-problems: the DC and AC models. By employing specialized solvers for each stage and incorporating advanced optimization techniques such as quadratic convex relaxation and decomposition methods, we significantly improve the efficiency and accuracy of the solution process.

II. PROBLEM FORMULATION

In this paper, we use the problem statement developed for the DOE Advanced Research Projects Agency - Energy (ARPA-E) GO challenge. This problem is quite extensive, with a main formulation that spans 53 pages [14], plus an additional 22 pages of supplementary information. There are a wide range of optimization constraints and variables, including limits on real and reactive power at various bus nodes, voltage constraints, zone-specific reserves, device statuses, switching intricacies, considerations related to device downtime and uptime, the number of device starts over multiple intervals, ramping up and down constraints, and stipulations on minimum and maximum energy levels across multiple time intervals, to name a few (See Fig. 1). It's important to note that all these constraints appear in both the base and contingency conditions. The contingency constraints serve to guarantee system reliability when addressing unforeseen events and outages. The compact form of the OPF problem is:

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

Where z is the total market surplus, p_{jt} represents the power value for device j at time t, θ_{jt} is the voltage angle, and u_{jt} is a binary variable representing device switching decisions. In these equations, G denotes the set of equality constraints, while H denotes the set of inequality constraints.

III. PROBLEM COMPLEXITY AND TESTING PLATFORM

The network sizes span from a relatively small-scale 73 buses to a highly complex 8316 buses test system. Our dataset is comprised of 9 network models; 73-, 617-, 1576-, 2000-, 4224-, 6049-, 6708-, 6717-, and 8316-bus configurations. For larger-scale test cases, we need to deal with over 2 million binary variables and more than 9 million continuous variables. We will use the testing platform that is made available by the Pacific Northwest National Laboratory (PNNL). This testing platform consists of a single computing node with local memory storage, equipped with dual 32-core CPUs (64 threads).

The testing criteria consists of three categories, each with unique power grid scheduling requirements:

- Category 1: Optimal scheduling for an 8-hour period, with a maximum decision time of 10 minutes.
- Category 2: Optimal scheduling for a 48-hour window, with a maximum decision time of 2 hours.
- Category 3: Optimal scheduling for 7 days (168 hours), with a maximum decision time of 3 hours.

A summary of the power grid scheduling requirements for each category is given in Table I.

IV. PROPOSED STRATEGY

To address the complexities of solving large-scale SCOPF problem and reducing the solution time, we propose breaking the original problem into two sequential sub-problems: the DC and AC modules. The initial subproblem involves the DC assumption, wherein binary and continuous are optimization variables. The binary variables returned by the first subproblem are fixed, while the continuous variables are discarded.

In the AC module, our aim is to include all relevant constraints to accurately represent a realistic grid. However, in the DC module, we choose not to include all constraints to reduce the computation time of the first subproblem (module). Nevertheless, we do incorporate constraints essential for ensuring the feasibility of the AC subproblem (module), e.g., we include ramping limits. Without these limits in the DC module, it might prompt abrupt device shutdowns. Consequently, the AC module, which accounts for ramping limits, may identify these shutdowns as infeasible to ramp down.

The DC model is formulated as a mixed-integer linear problem (MILP) and is solved using the Gurobi solver. On the other hand, the AC model is a nonlinear (NLP) and nonconvex problem, which is efficiently solved by the IPOPT solver. By decomposing the problem and leveraging specialized solvers for the sub-problem of each module, we increase computational effectiveness and overall efficiency. Fig. 2 illustrates our proposed two-stage solution approach and it what follows we will discuss the novelties of each module.

A. Managing Computational Workload in the DC Model

In managing the computational workload within the DC model, we implement a series of techniques to reduce complexity and improve efficiency.

a) Eliminating Quadratic Terms by Reformulation: While quadratic terms can typically be linearized or relaxed by adding more variables and constraints, we achieved the transformation of these terms into linear expressions without

TABLE I: Power Grid Scheduling Requirements by Category (i.e., Testing Criteria)

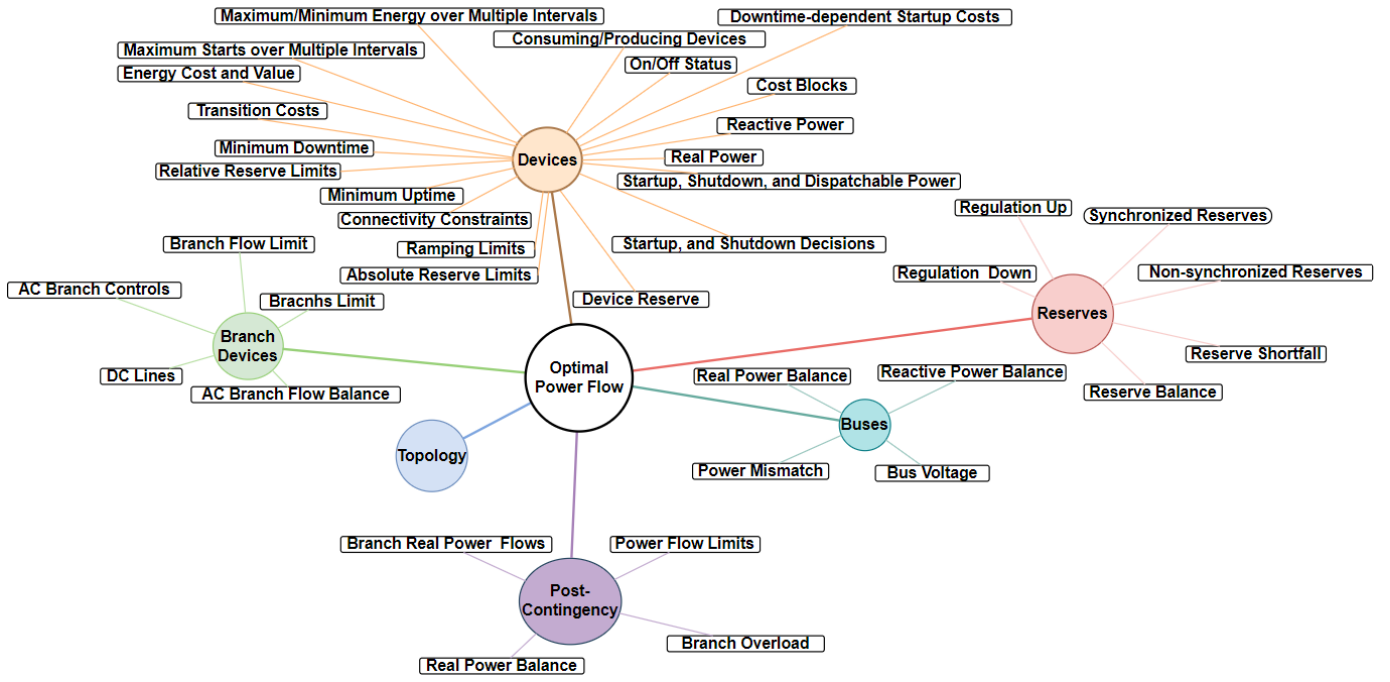| Category	Horizon	Time Constraints
1	8 hours look ahead	10 mins
2	48-hour look ahead	2 hours
3	7-day (168-hour) look ahead	3 hours

Fig. 1: The visual representation of Optimal Power Flow Problem formulation developed by the ARPA-E for the Grid Optimization challenge. Circles represent major components of this formulation while boxes illustrate the associated constraints.
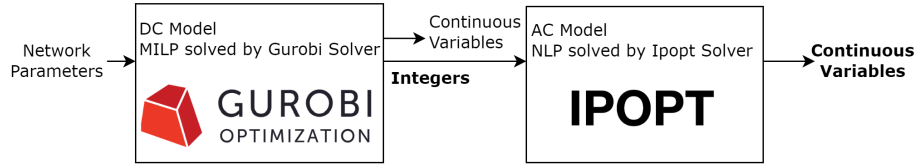

Fig. 2: Our two-pronged solution approach which includes a DC and an AC module.

introducing new variables or constraints. This artful reformulation significantly reduces computation burden. For example, the energy balance constraint is represented as [4]:

$$\sum_{j\in j_i^{cs,pr}} p_{jt} u_{jt}^{on} + \sum_{j\in j_i^{sh}} p_{jt} + \sum_{j\in j_i^{fr}} p_{jt}^{fr} + \sum_{j\in j_i^{to}} p_{jt}^{to} = p_{it} \quad (2)$$

Here, p_{jt} in the first term of (2) denotes the device power, as a generator (denoted by the superscript 'pr') or a consumer (denoted by the superscript 'cs'). The power limit is given as:

$$p_{jt,min} \leq p_{jt} \leq p_{jt,max} \quad \forall j \in j_i^{cs,pr} \quad (3)$$

The u_{jt}^{on} is the device on/off status. The quadratic term $p_{jt} u_{jt}^{on}$ can be linearized by introducing a new variable and four constraints. However, our approach reformulates the problem into a linear format without the need for additional variables or constraints. Equations (2) and (3) are updated as:

$$\sum_{j\in j_i^{cs,pr}} p_{jt} + \sum_{j\in j_i^{sh}} p_{jt} + \sum_{j\in j_i^{fr}} p_{jt}^{fr} + \sum_{j\in j_i^{to}} p_{jt}^{to} = p_{it} \quad (4)$$

$$p_{jt,min} u_{jt}^{on} \leq p_{jt} \leq p_{jt,max} u_{jt}^{on} \quad \forall j \in j_i^{cs,pr} \quad (5)$$

b) Nonconvex Constraints Relaxation: This involves the utilization of the substitution method and the relaxation of the equality constraint within the nonlinear constraints into an inequality constraint. For instance, shunt power (denoted by the superscript 'sh') is represented as [4]:

$$p_{jt} = g_{jt}^{sh} v_{it}^2 \quad \forall j \in j_i^{sh} \quad (6)$$

Equation (6) is non-linear. In an effort to eliminate the non-linearity, we will first substitute (6) into (4):

$$\sum_{j\in j_i^{cs,pr}} p_{jt} + \sum_{j\in j_i^{sh}} g_{jt}^{sh} v_{it}^2 + \sum_{j\in j_i^{fr}} p_{jt}^{fr} + \sum_{j\in j_i^{to}} p_{jt}^{to} = p_{it} \quad (7)$$

However, the resulting constraint is nonconvex. The new constraint can be transformed into a convex constraint by relaxing the equality constraint into an inequality constraint:

$$\sum_{j\in j_i^{cs,pr}} p_{jt} + \sum_{j\in j_i^{sh}} g_{jt}^{sh} v_{it}^2 + \sum_{j\in j_i^{fr}} p_{jt}^{fr} + \sum_{j\in j_i^{to}} p_{jt}^{to} \leq p_{it} \quad (8)$$

Because the right side of (8) represents the power mismatch, p_{it}, and due to a significant penalty imposed on this mismatch, the solver strives to minimize it. The mismatch will be minimized when it equals the value on the left side of (8).

c) Convex Constraints Linearization: First, we reformulate the convex constraints into a quadratic problem. Next, we relax the quadratic constraints into linear problems using McCormick envelope relaxation [7]. This reduction in complexity leads to a significant reduction in computation time for small networks and even greater time savings for larger networks. Specifically, the branch flow limit and penalty are represented as a second-order cone problem taking the following format:

$$||Ax_i + By_i||_2 \leq Cz_i + D \tag{9}$$

Here, the left side is the norm function which is a convex function. Since the right side is linear, (9) is convex. We reformulated (9) into quadratic by squaring both parts as,

$$A^2x_i^2 + B^2y_i^2 + 2ABx_iy_i \leq C^2z_i^2 + D^2 + 2CDz_i \tag{10}$$

While (10) is nonconvex, the bilinear and quadratic terms can be relaxed into linear using a McCormick envelope.

d) Sparse Constraint-matrix: The MILP problem, given this formulation, is always feasible. Nevertheless, introducing slack variables aids in faster convergence for the solver. These slack variables appear in only a few constraints, resulting in a sparser constraint matrix with more zeros, which facilitates quicker convergence for the solver.

e) Reduction of Lengthy Linear Expressions: A significant number of devices are connected to the same bus, resulting in an extensive power balance constraint in the form of a lengthy linear expression. This complexity can lead to numerical issues for the solver. To mitigate this, we introduce new variables. Even though this increases optimization variables, our adjustment enhances convergence speed.

f) Reformulating Downtime-Dependent Startup Costs: We formulated downtime-dependent startup costs using seven different models, which include three conventional models, the polyhedron representation transformation algorithm, and three versions of the clique constraint [15]. Rather than selecting the model with the fewest variables and constraints, our choice is based on compatibility with the Gurobi workflow. Specifically, we opted for the clique constraint model due to its alignment with Gurobi's pre-solver, which uses problem structure for eliminating variables and constraints.

g) Max/Min Constraint Functions Reformulation: The maximum and minimum equality constraints can be reformulated in two ways: (1) as a linear model or (2) as an MIP problem. Our analysis suggests that the MIP formulation aligns more effectively with Gurobi's structure and workflow.

B. Managing Computational Workload in the AC Model

This subsection's efforts are focused on managing the computational workload in the AC module using various techniques. Notably, we handle the entire AC model without any linearization or relaxation. This ensures a comprehensive and accurate representation of the power grid system. The details of these techniques are presented below:

a) Handling Controllable and Uncontrollable Load: Every device has a set of cost blocks $(m \in M)$ [4]. We investigated every device to identify pseudo-variables that cannot be controlled. This approach reduces the number of variables associated with the cost blocks and eliminates the need for introducing binary variables. More precisely, the power is split into bid blocks $(m \in M)$, and a cost is applied to each block. Therefore, the total power of the device is the sum of the power values in each block:

$$p_{jt} = \sum_{m \in M_{jt}} p_{jtm} \quad \forall j \in j_i^{cs,pr} \tag{11}$$

For instance, if the minimum power value of the device is higher than the power in the first block $(m = 1)$, then the power value in the first block is uncontrollable. Hence, the power value in the first block is equal to its maximum power.

b) Post-processing Reserves Calculations: Reserves are not incorporated into the main optimization framework. Instead, after IPOPT completes the simulation, power values are processed to meet the reserve requirements. This modification significantly reduces the problem's complexity. Another approach we considered was incorporating the reserve constraints in the DC section and then using the obtained power values as the maximum power for the AC section.

c) Exact Hessian and Jacobian Matrix: We compute the first and second derivatives of the Hessian and Jacobian analytically. Without these matrices, the solver resorts to less efficient numerical methods for estimation. Supplying the matrices analytically expedites IPOPT's convergence but necessitates a more hands-on code development, rather than relying on IPOPT's numerical techniques.

d) Providing Hessian and Jacobian Structure: When supplying the exact Hessian and Jacobian matrices, we provide information exclusively for non-zero elements along with their respective locations, as indicated by the structure. This approach eliminates the need to provide information for zero-valued elements, effectively reducing the computational workload and the volume of stored data. As a result, it contributes to faster convergence.

e) Utilizing Vectorized Form and Sparse Matrix in Coordinate format: This minimizes the required storage capacity. However, when not using the sparse matrix in coordinate format, we ran out of memory.

f) Receding Ramping Bounds: This approach facilitates solving the problem through sequential optimization. The problem is addressed sequentially for large networks because it is not feasible to solve these extensive networks within the given time constraints if we attempt to tackle the entire problem without breaking it down into steps. Therefore, we recursively update the bounds to ensure that we stay within the ramping limits when solving each step individually. The process of updating the bounds is undertaken in two stages, as illustrated in Figs. 3 and 4. The bounds are updated in stage 1 using the following equations as we move forward:

$$\overline{P}_{t,new} = \min[\overline{P}_t, \overline{P}_{t+1} + P_{rd}] \tag{12}$$
$$\underline{P}_{t,new} = \max[\underline{P}_t, \underline{P}_{t+1} - P_{ru}] \tag{13}$$

The second stage bounds are updated as we move backward:

$$\overline{P}_{t,final} = \min[\overline{P}_{t,new}, \overline{P}_{t-1,new} + P_{ru}] \quad (14)$$
$$\underline{P}_{t,final} = \max[\underline{P}_{t,new}, \underline{P}_{t-1,new} - P_{rd}] \quad (15)$$

C. Efficient Resource Sharing between DC and AC Modules

Addressing the complex nonconvex nature of the AC model demands a considerable amount of computational resources compared to the MILP problem. To optimize our two-pronged approach, we divide our time allocation as follows: One-third of the computational resources are dedicated to solving the DC model, while the remaining two-thirds are devoted to tackling the challenges posed by the AC model. This balanced allocation ensures efficient coordination between the DC and AC modules, allowing us to solve these problems in time.

D. Sequential Optimization

Solving the problem in a single shot is impractical given the problem scale, typically those with more than 2000 buses. In such cases, the control horizon is subdivided into stages, where the outcomes of one stage serve as the initial conditions for the subsequent one. In the context of the mixed-integer programming (MIP) problem, devices are permitted to undergo two switching events: the first occurs at the initiation of the control horizon. However, a crucial constraint to consider is the minimum uptime requirement, which necessitates that the device remains operational for a certain duration before it can be turned off again. As a result, the second switching event can only occur once this minimum uptime has been satisfied.

To ensure compliance with ramping limits and prevent violations, the bounds are adjusted in a recursive manner before executing the sequential optimization process.

V. Performance Evaluation

Our method was tested using over 300 scenarios meticulously chosen by domain experts and research teams from multiple organizations [4]. The code was evaluated based on two criteria: the total score and the number of best scores achieved. The score assessment aims to create a model that can generate optimal solutions within specified time constraints across various scenarios. We aimed to find a balance between achieving high scores and meeting stringent time limitations. The scaled scores for scenarios set 3 and scenarios set 2 are displayed in Figs. 5 and 6. The scaled score is calculated as the achieved score divided by the highest score recorded among available codes. A scaled score of 1.0 indicates the highest
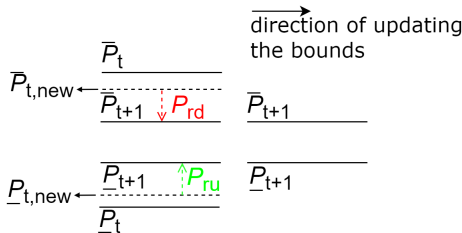
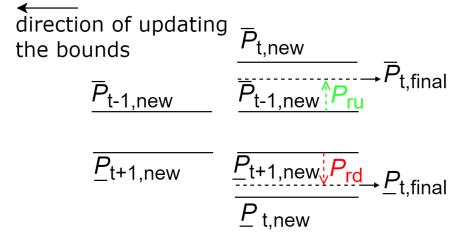Fig. 3: Bound update in stage 1 to prevent ramping violations in sequential solving.

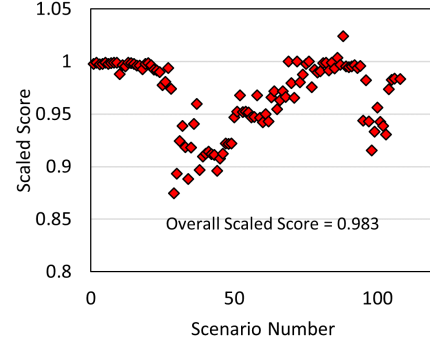Fig. 4: Update bounds in stage 2 to avoid ramping violations during sequential solving.

Fig. 5: Performance of the proposed solution strategy in scenarios set 3, evaluated against the best available solution. The scaled score represents the ratio to the best score.

possible achievement, while a score of 0.99 signifies a 1% deviation from the top performance.

After reviewing Fig. 5 and Fig. 6, it's evident that the scaled score for the majority of scenarios is higher than 0.95, with an overall scaled score exceeding 0.98. This demonstrates the robustness of the proposed strategy in effectively handling various scenarios. More precisely, in scenarios set 3, the code was tested in 137 scenarios, while in scenarios set 2, it was tested in 196 scenarios. In scenarios set 3, the scaled score was at least 0.99 in 42% of the scenarios, whereas in scenarios set 2, this was the case in 53% of the scenarios. Importantly, the scaled score exceeded 0.95 in 70% of the scenarios for both scenarios set 3 and scenarios set 2. The overall scaled scores for both scenarios set 3 and scenarios set 2 are 0.983 and 0.9992, respectively. A detailed distribution of these score ranges can be found in Table II.

The dataset included diverse network models, varying in size and complexity. These models included configurations with 73, 617, 1576, 2000, 4224, 6049, 6708, 6717, and 8316 buses, showcasing a wide range of network scales. The 6708-bus network is the only industrial network; the rest are synthesized. In the industrial network scenarios, totaling

TABLE II: Distribution of Scaled Scores for Set 3 and 2

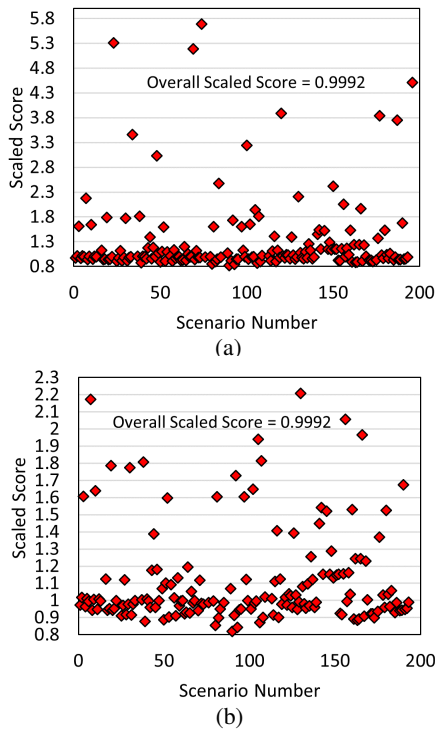| Range	% Scenarios	
	Set 3	Set 2
Scaled Score ≥ 0.99	42	53
$0.99 >$ Scaled Score ≥ 0.97	16	10
$0.97 >$ Scaled Score ≥ 0.95	12	9
$0.95 >$ Scaled Score ≥ 0.93	12	9
$0.93 >$ Scaled Score ≥ 0.90	13	11
Scaled Score < 0.9	5	8

Fig. 6: Performance of the proposed solution strategy in (a) scenarios set 2 and (b) zoomed-in view of scenarios set 2, evaluated against the best available solution.
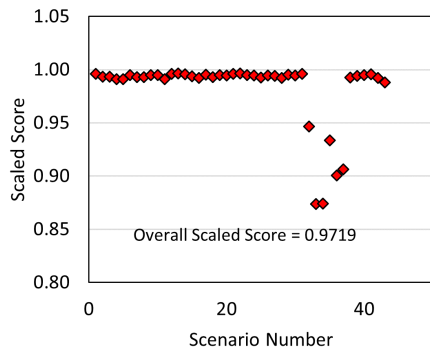

Fig. 7: Performance of the proposed strategy on 6708-bus industrial network, evaluated against the best available solution.

approximately 45 (as shown in Fig. 7), the scaled score was nearly optimal (close to 1) in about 39 scenarios. In this context, a scaled score close to 1 indicates the proposed model effectively converges to the available best solution. Among the remaining 6 scenarios, 4 exhibited scaled scores between 0.95 and 0.9, while the other 2 scenarios fell just below the 0.9 threshold. With an overall scaled score of 0.9719, our solution further demonstrates its robust performance in addressing this large-scale industrial real network.

VI. CONCLUSION

In conclusion, addressing the intricate challenge of optimizing the power grid requires a highly effective strategy. The endeavor to scale multi-timestep security-constrained optimal power flow in large power grids demands innovative solutions due to the involvement of millions of variables and constraints, including binary variables and nonconvex, nonlinear characteristics. Our solution approach, grounded in techniques such as constraint relaxation, linearization, sequential optimization, and problem reformulation, has proven to be efficient in navigating the complexities of the problem while adhering to time constraints and maintaining optimality. The strategic division of the problem into sequential sub-problems, namely the DC model and the AC model, along with the careful management of computational workload, proved essential in enhancing the efficiency and effectiveness of our approach. The performance evaluation highlights the robustness of our strategy in addressing the intricate challenges inherent in power grid optimization on a large scale.

REFERENCES

[1] 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.

[2] 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. PP, pp. 1–11, 2023, doi: 10.1109/tpwrs.2023.3238377.

[3] 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.

[4] ARPA-E, "Grid optimization challenge 3." https://gocompetition.energy. gov/(accessed Oct. 28, 2023)

[5] 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.

[6] 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.

[7] A. R. Aldik and B. Venkatesh, "Fast QC relaxation of the optimal power flow using the line-wise model for representing meshed transmission networks," IEEE Access, vol. 11, no. October 2022, pp. 2775–2786, 2023, doi: 10.1109/ACCESS.2022.3233633.

[8] 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.

[9] 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.

[10] 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," in Proc. 49th Annu. Allerton Conf. Commun., Control, Comput. (Allerton), Sep. 2011, pp. 1492–1499

[11] 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, Jul. 2016

[12] H. T. Kahraman, M. Akbel, and 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.

[13] D. Liu, C. Zhang, G. Chen, Y. Xu, and Z. Y. Dong, "Stochastic security-constrained optimal power flow for a microgrid considering tie-line switching," Int. J. Electr. Power Energy Syst., vol. 134, no. May 2021, p. 107357, 2021, doi: 10.1016/j.ijepes.2021.107357.

[14] J. Holzer et al., "Grid optimization challenge 3 problem formulation." https://gocompetition.energy.gov/challenges/challenge-3/formulation (accessed Oct. 28, 2023).

[15] H. Sharadga , How to find the index of the item, the first time appears?, https://or.stackexchange.com/q/9440 (accessed Nov. 14, 2023).