Refining Unit Commitment in Large-Scale Power Grid Optimization

Hussein Sharadga
School of Engineering
Texas A&M International University
Laredo, Texas, USA
hssharadga@tamu.edu

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

***Abstract*—This paper introduces a set of relaxation and tightening techniques to enhance the efficiency of unit commitment optimization in large-scale power grids. The computational complexity of this problem stems from binary decision variables, millions of optimization variables, and the need for multi-period scheduling. To address these challenges, we propose methods such as constraint relaxation, tightened power bounds, tighter cost-block formulations, and lazy constraint techniques. Results demonstrate significant computational speedups while preserving solution quality, highlighting the practical applicability of these methods for real-world power grid operations.**

***Index Terms*—Power Grid Optimization, Power Grid Software, Unit Commitment, Optimal Power Flow, Large-scale Optimization**

I. Introduction

II. Problem Formulation

III. Data and Computing Machine

The dataset used in this study is available in [1]. The experiments were conducted on a computing machine with 256 GB of RAM and an Intel® Xeon® w9-3495X processor, which has 56 cores and 112 threads. The processor has a base clock speed of 1.90 GHz and a maximum turbo boost speed of 4.80 GHz. Gurobi version 12.0.1 [5] was used on Windows 11.

The 1576-bus and 4224-bus networks were adapted for this study, with a one-day control horizon (18 steps). The 4224-bus network optimization model has about 1 million continuous decision variables, approximately 116 thousand binary variables, and about 1 million constraints. The number of variables and constraints for the 1576-bus and 4224-bus networks are summarized in Table I.

TABLE I: Grid Network Parameters: All are scheduled with a control horizon of 18 steps.

Parameter	1576-Bus Network	4224-Bus Network
# of Buses	1,576	4,224
# of Binary Variables	111,564	116,154
# of Continuous Variables	546,811	966,220
# of Consraints	595,916	1,092,829

IV. Baseline Model

Our model from [2], [3] has been proven to be robust, achieving an average scaled score exceeding 0.98 when tested on different network sizes and for different control horizons. In this study, we enhance the unit commitment component, which is solved using the Gurobi solver, to improve computational efficiency.

V. Relaxation Methods

A. Eliminating Ramping Constraints with Tighter Bounds

This approach reduces the number of constraints by approximately 7% and provides tighter power bounds (i.e., a smaller search space), leading to a twofold speedup in the optimization model (i.e., it runs twice as fast). In the first stage, the power bounds of devices are updated as we move backward using Eqs. (1) and (2). In the second stage, the bounds are updated as we move forward using Eqs. (3) and (4):

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

$$\underline{P}_{t,new} = \max[\underline{P}_t, \underline{P}_{t-1} - P_{rd}] \tag{2}$$

$$\overline{P}_{t,final} = \min[\overline{P}_{t,new}, \overline{P}_{t+1,new} + P_{rd}] \tag{3}$$

$$\underline{P}_{t,final} = \max[\underline{P}_{t,new}, \underline{P}_{t+1,new} - P_{ru}] \tag{4}$$

B. Tighter Cost Blocks

This modification reduces the number of constraints by 9% and improves computational speed by 60%. The power produced or consumed by a device (p_j) is the sum of the power from different power blocks, where each block has a different cost:

$$p_{jt} = \sum_{m \in M_{jt}} p_{jtm} \quad \forall j \in j_i^{cs,pr} \tag{5}$$

The device power is a semi-continuous variable, which can be formulated as follows [2]:

$$p_{jt}^{min} u_{jt}^{on} \leq p_{jt} \leq p_{jt}^{max} u_{jt}^{on} \quad \forall j \in j_i^{cs,pr} \tag{6}$$

In this study, a tighter cost block formulation is proposed to replace (15), as follows:

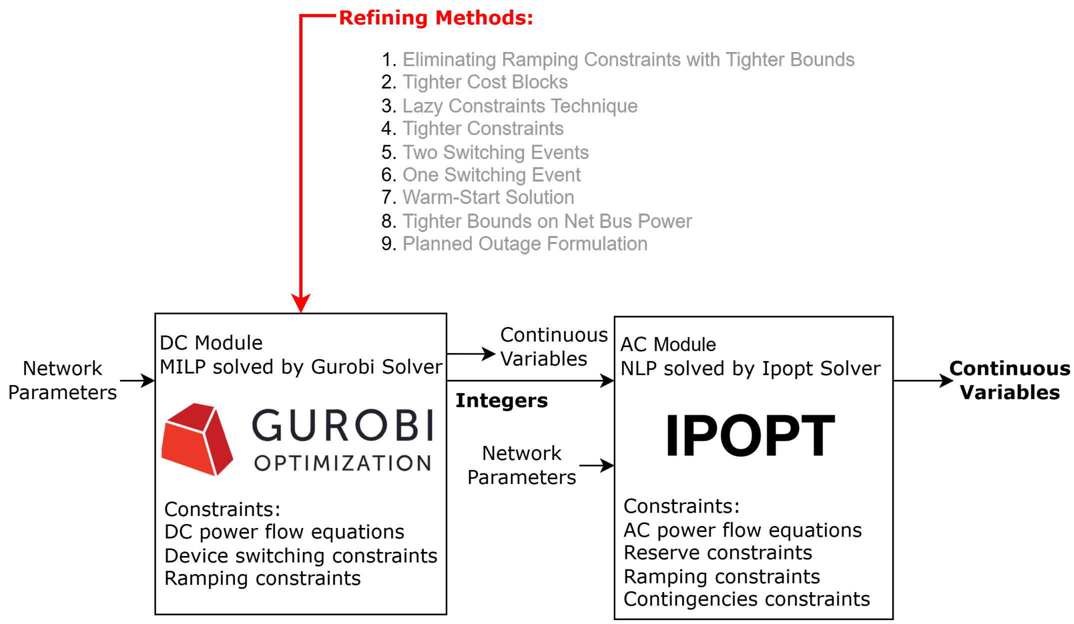


Fig. 1: Our two-stage solution strategy incorporates DC and AC modules from [2], [3]. In this study, we enhance the convergence speed of the DC module (first block) through the proposed refinement methods, as illustrated in the figure.

$$0 \le p_{jtm} \le p_{jtm}^{\max} u_{jt}^{\text{on}} \quad \forall m \in M_{jt}; \forall j \in J_i^{\text{cs,pr}} \tag{7}$$

This can also be reformulated to reduce the number of optimization constraints associated with cost blocks from 208,352 to 74,376 as follows:

$$0 \le \sum_{\forall m \in M_{jt}} p_{jtm} \le \sum_{\forall m \in M_{jt}} p_{jtm}^{\max} u_{jt}^{\text{on}} \quad \forall j \in J_i^{\text{cs,pr}} \tag{8}$$

However, according to the Gurobi log, the number of constraints remains the same between Eqs. (7) and (8), indicating that Gurobi automatically reformulates them internally.Therefore, Eq. (7) is adapted here.

C. Lazy Constraints Technique

The lazy constraints are proposed to remove the minimum uptime, minimum downtime, and maximum startup of the devices. The Gurobi callback is used to verify if these constraints are met. If they are not, they are added to the model (i.e., lazy constraints). The number of constraints was reduced by 45,161 for 4224-bus network (4%). However, in some cases, there was no noticeable improvement in the speedup. The time consumed by Gurobi during the callback was low (less than 2 seconds). The time during callback and adding the lazy constraint does not increase the total time significantly.

In other cases, lazy constraints do not improve the model because removing them increases the feasible region, making it harder for Gurobi to find the optimal solution, which in turn increases the solving time. Explicitly adding these constraints usually helps Gurobi find a solution more efficiently.

D. Tighter Constraints

The minimum uptime constraint is given as follows [4]:

$$u_{jt}^{sd} \le 1 - \sum_{t' \in T_{jt}^{up,min}} u_{jt'}^{su} \quad \forall j \in j_i^{cs,pr} \tag{9}$$

A tighter formulation is proposed below:

$$u_{jt}^{sd} \le 1 - u_{jt'}^{su} \quad \forall t' \in T_{jt}^{up,min}; \forall j \in j_i^{cs,pr} \tag{10}$$

The same approach can be applied to the minimum downtime constraints:

$$u_{jt}^{su} \le 1 - u_{jt'}^{sd} \quad \forall t' \in T_{jt}^{dn,min}; \forall j \in j_i^{cs,pr} \tag{11}$$

E. Two Switching Events

The first switching event occurs at the beginning of the control horizon ($t = 0$). The minimum uptime constraint mandates that the device remain operational for a specified duration before it can be switched off. Consequently, the second switching event takes place once the minimum uptime requirement is satisfied. This approach reduces the number of binary variables by a factor of nine. Specifically, for a control horizon of 18, instead of assigning a binary variable to each time step, only two are required. However, this approach does not make Gurobi faster for the networks tested in this study; in fact, Gurobi takes more time to solve, which can be attributed to the structure of the Gurobi workflow or the heuristic methods it uses. This approach might be helpful for larger networks or longer control horizons.

F. One Switching Event

Devices are more likely to switch only once, either after the minimum uptime or downtime is met. Adding a constraint to enforce a single switching event reduces the search space for Gurobi, thus leading to faster convergence.

$$\sum_{t \in T} u^{\text{su}}_{jt} + u^{\text{sd}}_{jt} \leq 1, \quad \forall j \in J^{\text{cs,pr}} \tag{12}$$

This single switching event can occur at any point within the control horizon as long as it satisfies other constraints.

G. Warm-Start Solution

A warm-start in Gurobi does not always improve the model. Even with a feasible initial solution, it can sometimes slow down the solving process due to factors like changes in the search path, lower initial bounds, reduced effectiveness of presolve and cutting-plane techniques, and additional node processing overhead. Moreover, it may limit parallelism, reducing overall efficiency, and consume excessive time to process the provided solution. In this case, solving without a warm-start takes 1.5 minutes to complete the problem. However, using a warm-start takes about 1 minute to process the given solution, which is added to the total time. As a result, solving with the warm-start consumes a total of 3.5 minutes.

H. Startup-Shutdown Using a Single Variable

The startup and shutdown decision variables can be replaced by a single variable (W), where

$$W_{jt} = u^{on}_{jt} - u^{on}_{jt-1} \tag{13}$$

W is 0 if there is no change, 1 in the case of a startup, and -1 in the case of a shutdown. The startup case can be identified using $\max(W, 0)$, and the shutdown case using $\max(-W, 0)$. However, Gurobi requires replacing the max constraint by introducing new binary variables. Therefore, this approach is not effective if startup and shutdown costs are included.

I. Planned Outage Formulation

In the planned outage period the device is set to be off:

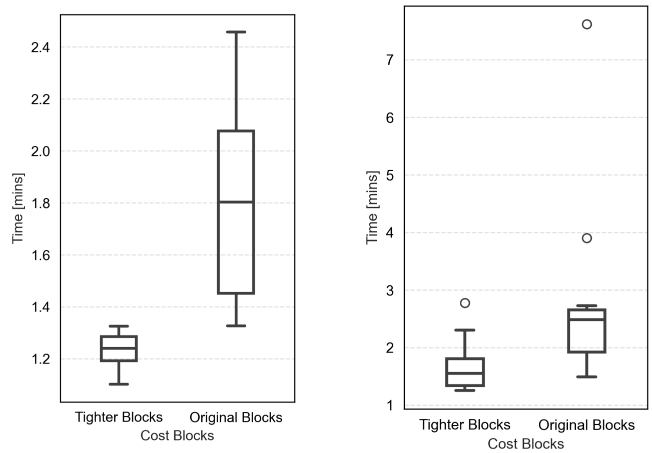
$$u^{on}_{jt'} = 0 \quad \forall t' \in T^{Outage}_{jt} \tag{14}$$

A compact version is proposed as follows to reduce the number of constraints:

$$\sum_{\forall t' \in T^{Outage}_{jt}} u^{on}_{jt'} = 0 \tag{15}$$

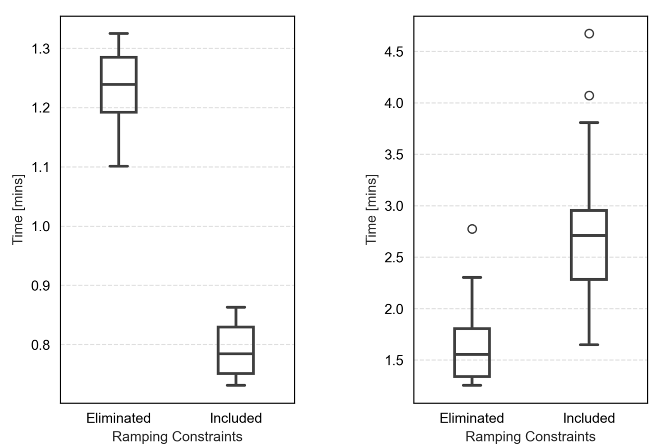
J. Tighter Bounds on Net Bus Power

Since the maximum power generated by each device is known, and the devices associated with each bus are given, the maximum power flowing into or out of a bus can be approximated. These tighter bounds can improve computational speed by a factor of five.



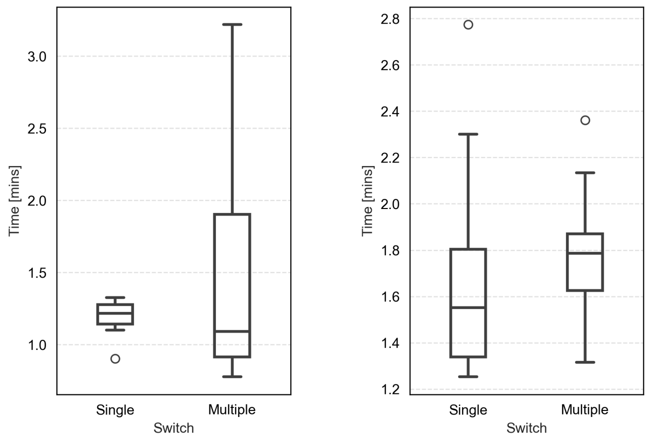
(a) (b)

Fig. 2: Computation time distribution across different scenarios: a comparison between tighter cost blocks and the original cost blocks. (a) 1576-bus network and (b) 4224-bus network. The simulations were performed with one-switching-event technique, tighter bounds on net bus power, and with replacing ramping constraints with tightened power bounds. The results indicate that tighter cost blocks improve the model's convergence speed.



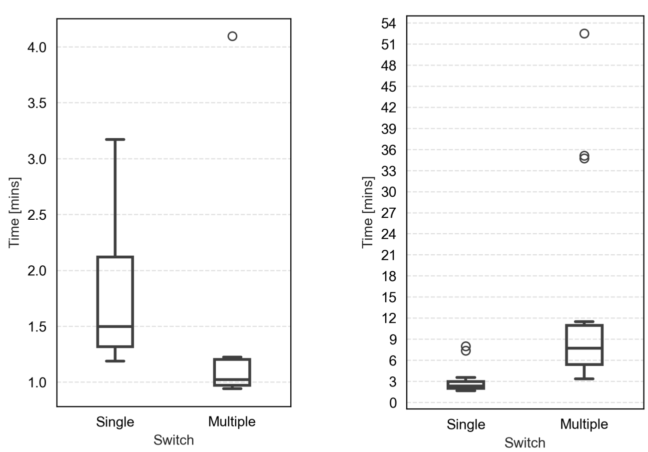
(a) (b)

Fig. 3: Computation time distribution across different scenarios: a comparison between the model with ramping constraints eliminated by tightening the power bounds and the original model, which includes the ramping constraints. (a) 1576-bus network and (b) 4224-bus network. The simulations were performed with the one-switching-event technique, tighter bounds on net bus power, and tightened cost blocks. Eliminating the ramping constraints improves the model's convergence speed for the 4224-bus network but not for the 1576-bus network.



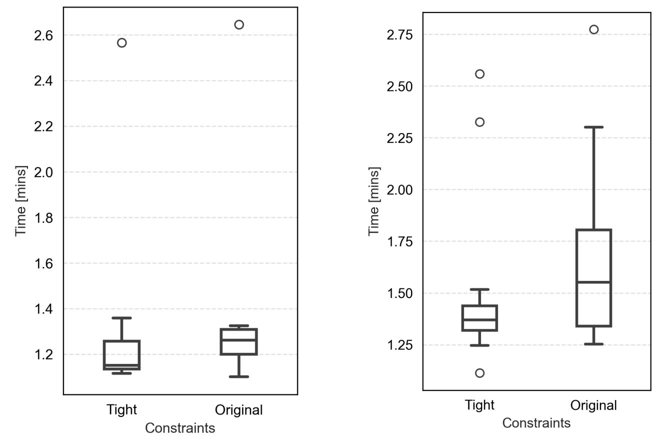
(a) (b)

Fig. 4: Computation time distribution across different scenarios: A comparison between the model with a one-switch event and the original model, which allows multiple switches. (a) 1576-bus network and (b) 4224-bus network. The simulations were performed with tighter bounds on net bus power, tightened cost blocks, and by replacing ramping constraints with tightened power bounds. On average, the one-switch event improves the model's convergence speed for both the 1576-bus and 4224-bus networks.



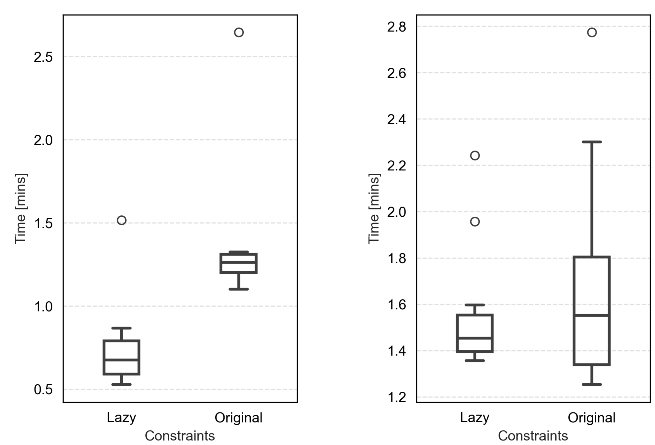
(a) (b)

Fig. 5: Computation time distribution across different scenarios: A comparison between the model with a one-switch event and the original model, which allows multiple switches. (a) 1576-bus network and (b) 4224-bus network. The simulations were performed with tighter bounds on net bus power. The one-switch-event technique improves the model's convergence speed for the 4224-bus networks.



(a) (b)

Fig. 6: Computation time distribution across different scenarios: A comparison between the model with tighter minimum uptime and downtime constraints and the original model with the original constraints. (a) 1576-bus network and (b) 4224-bus network. The simulations were performed with tighter bounds on net bus power, tightened cost blocks, replacing ramping constraints with tightened power bounds, and the one-switch event technique. The improvement in the model's convergence speed is more noticeable with the 4224-bus network.



(a) (b)

Fig. 7: Computation time distribution across different scenarios: A comparison between the model with lazy constraints and the original model with the original constraints. (a) 1576-bus network and (b) 4224-bus network. The simulations were performed with tighter bounds on net bus power, tightened cost blocks, replacing ramping constraints with tightened power bounds, and the one-switch event technique. The lazy-constraint technique improves the model's convergence for both the 1576-bus and 4224-bus networks.

TABLE II: Speed-Up Achieved by Employing Different Techniques. 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**
A	Eliminating Ramping Constraints by Tighter Bounds	12.25x
B	Tighter Cost Blocks	4.12x
C	Lazy Constraints	1.75x
D	Tighter minimum uptime & downtime Constraints	3.75x
E	Two Switching Events	1.78x
F	One Switching Event	1.95x
G	Warm-Start Solution	4.32x
H	Tighter Bounds on Net Bus Power	9.89x

VI. Results

A. Summary

VII. Conclusion

In this study, we explored various relaxation and tightening techniques to improve the computational efficiency of unit commitment optimization for large-scale power grids. The proposed methods resulted in significant solver performance improvements, achieving substantial speedups without compromising solution quality, especially for large networks such as the 4,224-bus system. These findings highlight the potential for enhancing the scalability and practical applicability of optimization models in power grids, especially for large, complex systems with millions of variables and constraints.

Acknowledgment

During the preparation of this work, the authors used ChatGPT in order to rephrase some statements to enhance the readability. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

This work is financially supported by the US ARPA-E (#DE-AR0001646).

References

[1] ARPA-E, "Grid optimization challenge 3 datasets," https://gocompetition.energy.gov/challenges/600650/datasets; last accessed July 28, 2025.

[2] 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.

[3] H. Sharadga, J. Mohammadi, C. Crozier and K. Baker, "Scalable Solutions for Security-Constrained Optimal Power Flow with Multiple Time Steps," *IEEE Transactions on Industry Applications*,2025 , doi: 10.1109/TIA.2025.3532927.

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

[5] Gurobi Optimization, Documentation. https://www.gurobi.com/. Accessed: July 22, 2025.