Is learning for the unit commitment problem a low-hanging fruit?

S. Pineda [*], J.M. Morales

University of Malaga, Spain

A B S T R A C T

The blast wave of machine learning and artificial intelligence has also reached the power systems community, and amid the frenzy of methods and black-box tools that have been left in its wake, it is sometimes difficult to perceive a glimmer of Occam's razor principle. In this letter, we use the unit commitment problem (UCP), an NP-hard mathematical program that is fundamental to power system operations, to show that *simplicity* must guide any strategy to solve it, in particular those that are based on *learning* from past UCP instances. To this end, we apply a naive algorithm to produce candidate solutions to the UCP and show, using a variety of realistically sized power systems, that we are able to find optimal or quasi-optimal solutions with remarkable speedups. To the best of our knowledge, this is the first work in the technical literature that quantifies how challenging learning the solution of the UCP actually is for real-size power systems. Our claim is thus that any sophistication of the learning method must be backed up with a statistically significant improvement of the results in this letter.

1. Introduction

The unit commitment problem (UCP) is currently one of the most fundamental mathematical tools to operate power systems. The UCP determines the on/off commitment status and power dispatch of generating units to satisfy electricity demand at a minimum cost while complying with the technical limits of generation and transmission assets [1]. The UCP is usually formulated as a mixed-integer optimization problem that is proven to be NP-hard [2] and therefore, the technical literature includes several methods to trim down its computational burden [3,4]. Existing strategies comprise formulation tightening [5,6], decomposition techniques [7] and constraint screening [8]. These methods, however, overlook the fact that slight variations of the same UCP are to be solved everyday and therefore, learning from the past may be a powerful weapon to tackle future UCP instances.

In the same vein, some learning-based methodologies have been recently proposed to reduce the computational burden of the UCP using information about previously solved instances. References [9,10] review current learning-based methods for power system problems. In particular, the authors of [11] learn a proxy of the UCP's solution to be used in long-term planning assessments. Despite being simple and fast (with speedups up to $260\times$), the proposed strategy involves an average optimality error of 3.5% for the IEEE-96RTS test system, which precludes its use in short-term operation. Using Classification and Regression Trees (CART) or Random Forest (RF), reference [12] presents a

methodology to find the relationships between the solutions of the original and relaxed versions of the UCP and reports average optimality errors of between 0.14% and 0.23%. A supervised classification procedure is proposed in [13] to learn the transmission capacity constraints that can be screened out of the UCP. Using a 2000-bus system, the authors report a speedup of $19\times$ if retrieving the original solutions must be guaranteed, and a speedup of $43\times$ for an average suboptimality error of 0.04%. The authors of [14] describe a sophisticated methodology to cluster decision variables depending on the difficulty of the UCP instance to be solved. They achieve speedups of between $1.5\times$ and $2\times$, an average optimality error of 0.1%, and a maximum optimality error of 1% for the IEEE 118-bus system. The authors of [15] use the UCP solution of the previous day UC as a warm start strategy and report speedups of $2\times$ for the MISO power system. This speedup can be increased to $2\times-12\times$ if additional historical information is considered, as shown in [16]. In [17], the authors use machine learning to determine unnecessary constraints of the stochastic unit commitment and attain speedups of $14\times$ for a 500-bus power system. Finally, reference [18] proposes different strategies to learn initial solutions with speedups of $4.3\times$ while retrieving the optimal solution of the UCP. The authors also discuss a learning-based procedure to learn the relationships among binary variables, which yields speedups of $10.2\times$ but no optimality guarantees. These methodologies are tested on a set of large-sized networks.

It is apparent that when it comes to learning for the UCP, one may

---

easily get lost in a myriad of methods and approaches, all of which promise reasonable computational savings. The takeaway message is that, despite the high theoretical complexity of the UCP, having access to previous instances may significantly reduce the solution task in practice, since today's commitment decisions are probably very similar to those made yesterday, one week ago, or even one year ago. Given the high potential of using historical data to reduce the computational burden of the UCP, the questions that naturally arise are: Are existing learning methods an actual breakthrough in the solution of the UCP or are they just picking the low-hanging fruit? Do sophisticated learning methods to solve the UCP substantially outperform painless naive learning strategies? How should we benchmark the performance of learning methods to solve the UCP? In this letter, we aim to answer these questions by learning the UCP's solution through a straightforward K-nearest neighbor procedure. The purpose of this letter is, by no means, to propose a learning-based methodology that outperforms existing ones under certain conditions. What we claim is that the performance of existing and upcoming learning-based methods to solve the UCP should be thoroughly compared with some painless naive methods such as the one we suggest, in order to justify the hassle from the increased complexity and sophistication, and the loss of transparency. In this context, we refer to a naive method as a learning procedure that is simple, intuitive, easy to implement and interpret, and that requires a very reduced number of hyper-parameters.

2. A naive learning method

The unit commitment problem can be generally formulated as the following optimization program:

$$\min_{\mathbf{u},\mathbf{y}} f(\mathbf{u},\mathbf{y}) \tag{1a}$$

$$\text{s.t.} \, g_j(\mathbf{u},\mathbf{y},\mathbf{d}) \leq 0, \forall j \tag{1b}$$

where $\mathbf{u}$ and $\mathbf{y}$ denote, respectively, the binary and continuous variables, $\mathbf{d}$ represents the input parameters such as net demand throughout the network, and $f(\cdot), g_j(\cdot)$ are the objective function and the technical constraints, in that order. Under some mild assumptions, model (1) becomes a mixed-integer quadratic programming problem that can be solved using optimization solvers at a usually high computational cost. With some abuse of notation, we express the solution of (1) as a function of the input parameters, namely, $\mathbf{u}^{\text{UC}}(\mathbf{d}), \mathbf{y}^{\text{UC}}(\mathbf{d})$. If binary variables $\mathbf{u}$ are fixed to given values $\widetilde{\mathbf{u}}$, model (1) becomes an optimal power flow (OPF) problem, which is easier to solve and whose optimal solution is denoted as $\mathbf{y}^{\text{OPF}}(\widetilde{\mathbf{u}}, \mathbf{d})$.

Suppose we have access to a sample set of optimal solutions of problem (1) for different input parameters, referred to as $S = \{(\mathbf{d}_i, \mathbf{u}_i^{\text{UC}}, C_i^{\text{UC}}, T_i^{\text{UC}})\}_{i \in \mathscr{I}}$, where $\mathbf{u}_i^{\text{UC}} = \mathbf{u}^{\text{UC}}(\mathbf{d}_i)$, and $C_i^{\text{UC}}, T_i^{\text{UC}}$ are, respectively, the objective function and the computational time of problem (1) for instance $i \in \mathscr{I}$, where $\mathscr{I}$ is the set of available samples and each instance i represents a 24-hour generation scheduling. Intuitively, the naive learning methodology we propose as a benchmark consists in fixing the binary variables to those of close past instances to solve several OPFs in parallel and select the one with the lowest cost. We use a leave-one-out cross-validation methodology and therefore, the learning method is run for each instance $i \in \mathscr{I}$ with $\mathscr{I} \setminus \{i\}$ as the training set as follows:

1) Compute the distance between the input parameters $\mathbf{d}_i$ and those of the remaining instances using a norm, for example, $\| \mathbf{d}_i - \mathbf{d}_{\widetilde{i}} \|_2, \forall \widetilde{i} \in \mathscr{I}, \widetilde{i} \neq i$.
2) Find the set of the K-nearest neighbors to i with the lowest distances computed in step 1), denoted as $\mathscr{I}_i^K$.
3) Solve the optimal power flow for input parameters $\mathbf{d}_i$ and binary variables fixed to $\mathbf{u}_{\widetilde{i}}^{\text{UC}}$. That is, compute $\mathbf{y}^{\text{OPF}}(\mathbf{u}_{\widetilde{i}}^{\text{UC}}, \mathbf{d}_i), \forall \widetilde{i} \in \mathscr{I}_i^K$ and

denote the objective function and solving time as $C_{\widetilde{ii}}^{\text{OPF}}$ and $T_{\widetilde{ii}}^{\text{OPF}}$, respectively.
4) Among all feasible problems solved in step 3), approximate the cost of the UCP for instance i as $\widehat{C}_i^{\text{UC}} = \min_{\widetilde{i}} C_{\widetilde{ii}}^{\text{OPF}}$. The lowest sub-optimality gap is thus computed as $\Delta_i = (\widehat{C}_i^{\text{UC}} - C_i^{\text{UC}})/C_i^{\text{UC}}$.
5) Problems in step 3) are solved in parallel and the speedup factor is thus computed as $S_i = T_i^{\text{UC}}/(\max_{\widetilde{i}}(T_{\widetilde{ii}}^{\text{OPF}}) + T_i^L)$, where T_i^L is the learning time of steps 1) and 2).

Once steps 1)-5) are run for each instance, we determine the average suboptimality gap $\overline{\Delta}$, the maximum suboptimality gap $\Delta^{\max}$, the average speedup $\overline{S}$, and the number of instances for which the K problems solved in step 3) are infeasible N^{IN}.

3. Numerical results

In this section, we provide the results obtained by the learning method described in Section 2 for nine large-scale European test systems used in [18] and available for download at [19].

The performance of the naive learning method is illustrated using the solution of 500 UCP instances that differ on the 24-hour load profile. That is, we assume that a year and a half of data is available. Note that considering less than one year of data is not appropriate since seasonal effects may not be captured. On the other hand, considering a very large data set of 10 years, for example, can also lead to inaccurate results since topology changes are expected to occur during such a long period of time. According to the procedure proposed in [18], each 24-hour load profile is randomly generated as follows:

1) Let $\overline{P}_g$ denote the capacity of each unit g. The peak demand $\overline{D}$ is randomly generated as $\overline{D} = 0.6 \times (\sum_g \overline{P}_g) \times \text{unif}(0.925, 1.075)$, where $\text{unif}(a, b)$ represents a random sample from a uniform distribution in the interval $[a, b]$.
2) Let $\overline{\beta}_b$ denote the nominal load distribution factor of bus b. The distribution factor β_b is randomly generated as $\beta_b = \overline{\beta}_b \times \text{unif}(0.9, 1.1)$, which is subsequently normalized as $\beta_b^{\text{N}} = \beta_b / \sum_b \beta_b$ to satisfy that $\sum \beta_b^{\text{N}} = 1$.
3) Let $t \in \mathscr{T}$ denote the index for the hourly time periods of the UCP, and let μ_t and σ_t represent, respectively, the average and standard deviation of the ratio between the aggregated load level for two consecutive hours D_t/D_{t-1}. The hourly variation factors ν_t are randomly generated as $\nu_t = \text{normal}(\mu_t, \sigma_t)$, where $\text{normal}(a, b)$ represents a sample from a normal distribution with a mean equal to a and a standard deviation equal to b. The temporal factor γ_t is computed as $\gamma_t = \prod_{\tau=1}^t \nu_\tau$, which is normalized as $\gamma_t^{\text{N}} = \gamma_t/\max\{\gamma_t\}_{t \in \mathscr{T}}$ to satisfy that $\max\{\gamma_t^{\text{N}}\}_{t \in \mathscr{T}} = 1$.
4) The load demand at each bus b and time period t is computed as $D_{bt} = \overline{D} \times \beta_b^{\text{N}} \times \gamma_t^{\text{N}}$.

For each load profile, we solve the specific unit commitment formulation provided in the Appendix of [18]. For the sake of simplicity, we consider neither reserves nor security constraints. Due to its high computational burden, all UCP instances are solved using the constraint generation approach proposed in [20] using Gurobi 9.1.2 [21] with a MIP gap set to 0.01%. Afterwards, the learning method is used for each instance and test system assuming that the number of neighbors is set to 50, that is, 10% of the total number of instances. A summary of the most relevant results is provided in Table 1. More specifically, columns 2 and 3 include the average and maximum suboptimality gap for each test system. Columns 4-8 contain the number of instances whose sub-optimality gap belongs to given intervals. For instance, column 6 provides the number of instances with a suboptimality gap of between

Table 1

Numerical results.

System	$\overline{\Delta}$	$\Delta^{\max}$	<0.01%	0.01–0.02%	0.02–0.05%	0.05–0.1%	>0.1%	N^{IN}	$\overline{S}$
1888rte	0.0174	0.2394	230	131	109	20	9	1	116.5 ×
1951rte	0.0382	0.3759	47	116	217	85	27	8	150.4 ×
2848rte	0.0186	0.1332	179	138	159	14	8	2	132.6 ×
3012wp	0.0485	0.4864	37	78	212	132	36	5	188.8 ×
3375wp	0.1256	0.8073	9	14	102	164	198	13	215.9 ×
6468rte	− 0.0001	0.0175	498	2	0	0	0	0	41.2 ×
6470rte	− 0.0016	0.0187	496	4	0	0	0	0	171.9 ×
6495rte	− 0.0001	0.0481	496	3	1	0	0	0	41.0 ×
6515rte	− 0.0009	0.0133	497	3	0	0	0	0	101.7 ×

0.02% and 0.05%. The number of instances for which all OPF problems are infeasible is included in column 9. Finally, column 10 provides the average speedup of the naive learning method in relation to solving the original UCP instances using constraint generation.

The results in Table 1 lead to several interesting observations. Firstly, the average suboptimality error for the four systems with more than 6000 buses is negative. This means that, on average, fixing the binary variables to those of the neighbors yields an objective function that is actually lower than that of the original UCP. Indeed, the suboptimality error is below the MIP gap in more than 99% of the instances. Finally, these systems do not include any infeasible instances and report speedups between $41\times$ and $172\times$. We conjecture that the good performance of the naive approach for these power systems is due to their low network congestion level. According to these results, we can conclude that using complicated learning-based techniques for these four systems seems unnecessary since naive alternatives significantly reduce the UCP computational burden with negligible suboptimality errors. Secondly, for systems 1888rte and 2848rte, the 50 OPF solved become infeasible for 1 and 2 instances, respectively. For these two systems, the average suboptimality errors are slightly higher than the predefined MIP gap, while the computational times are reduced by two orders of magnitude. Therefore, the naive learning method is a competitive approach for these two systems. Finally, the number of infeasible instances for systems 1951rte, 3012wp and 3375wp are 8, 5 and 13, respectively. The average optimality errors exceed the MIP gap and amount to 0.0382%, 0.0485%, and 0.1256%, in the same order, whereas the speedup factor ranges between $150\times$ and $216\times$ for these three systems. We believe that, given the simplicity of the learning-based method we have used to obtain all these results, any other alternative method demanding further sophistication is only worth consideration if it succeeds in beating these numbers. In other words, using complicated learning approaches for these three systems would only be justified when the computational time, the suboptimality error or the infeasible cases are drastically lower than those reported by the naive learning method described in this letter.

4. Conclusion

A highly relevant topic within the scientific community working on power systems is the reduction of the computational burden of the unit commitment problem. However, assessing the improvements of sophisticated learning-based approaches is sometimes tricky due to the lack of benchmark methods. This letter describes a painless and naive learning method and shows that, for some power systems, learning the solution of the unit commitment problem is indeed a low-hanging fruit that can be effortlessly picked. For other systems, the results of the naive learning approach are less impressive but, at the very least, they should be used to benchmark the performance of existing and upcoming learning methods to solve the unit commitment problem. This letter also opens a debate about the complexity/performance trade-off of learning-based procedures to operate current power systems since, in our

opinion, their implementation would only happen if these methods are simple, transparent, interpretable and reliable.

CRediT authorship contribution statement

S. Pineda: Conceptualization, Methodology, Investigation, Formal analysis, Validation, Writing – original draft. **J.M. Morales:** Conceptualization, Methodology, Investigation, Formal analysis, Validation, Writing – original draft.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

References

[1] S. Sen, D. Kothari, Optimal thermal generating unit commitment: a review, Int. J. Electr. Power Energy Syst. 20 (7) (1998) 443–451.
[2] P. Bendotti, P. Fouilhoux, C. Rottner, On the complexity of the unit commitment problem, Ann. Oper. Res. 274 (1-2) (2019) 119–130.
[3] B. Saravanan, S. Das, S. Sikri, D. Kothari, A solution to the unit commitment problem-a review, Front. Energy 7 (2) (2013) 223–236.
[4] Y. Chen, A. Casto, F. Wang, Q. Wang, X. Wang, J. Wan, Improving large scale day-ahead security constrained unit commitment performance, IEEE Trans. Power Syst. 31 (6) (2016) 4732–4743, https://doi.org/10.1109/TPWRS.2016.2530811.
[5] H. Pandžić, T. Qiu, D.S. Kirschen, Comparison of state-of-the-art transmission constrained unit commitment formulations. 2013 IEEE PES General Meeting, 2013, pp. 1–5, https://doi.org/10.1109/PESMG.2013.6672719.
[6] D.A. Tejada-Arango, S. Lumbreras, P. Sánchez-Martín, A. Ramos, Which unit-commitment formulation is best? A comparison framework, IEEE Trans. Power Syst. 35 (4) (2020) 2926–2936, https://doi.org/10.1109/TPWRS.2019.2962024.
[7] Y. Fu, M. Shahidehpour, Z. Li, Security-constrained unit commitment with ac constraints, IEEE Trans. Power Syst. 20 (2) (2005) 1001–1013.
[8] Q. Zhai, X. Guan, J. Cheng, H. Wu, Fast identification of inactive security constraints in SCUC problems, IEEE Trans. Power Syst. 25 (4) (2010) 1946–1954, https://doi.org/10.1109/TPWRS.2010.2045161.
[9] Y. Yang, L. Wu, Machine learning approaches to the unit commitment problem: current trends, emerging challenges, and new strategies, Electr. J. 34 (1) (2021) 106889.

[10] G. Ruan, H. Zhong, G. Zhang, Y. He, X. Wang, T. Pu, Review of learning-assisted power system optimization, CSEE J. Power Energy Syst. 7 (2) (2021) 221–231, https://doi.org/10.17775/CSEEJPES.2020.03070.

[11] G. Dalal, E. Gilboa, S. Mannor, L. Wehenkel, Unit commitment using nearest neighbor as a short-term proxy. 2018 Power Systems Computation Conference (PSCC), 2018, pp. 1–7.

[12] X. Lin, Z.J. Hou, H. Ren, F. Pan, Approximate mixed-integer programming solution with machine learning technique and linear programming relaxation. 2019 3rd International Conference on Smart Grid and Smart Cities (ICSGSC), 2019, pp. 101–107, https://doi.org/10.1109/ICSGSC.2019.00-11.

[13] S. Pineda, J.M. Morales, A. Jiménez-Cordero, Data-driven screening of network constraints for unit commitment, IEEE Trans. Power Syst. 35 (5) (2020) 3695–3705, https://doi.org/10.1109/TPWRS.2020.2980212.

[14] Y. Yang, Integrated data-driven framework for fast SCUC calculation, IET Gener. Transm. & Dis. 14 (10) (2020) 5728–5738.

[15] Y. Chen, F. Wang, Y. Ma, Y. Yao, A distributed framework for solving and benchmarking security constrained unit commitment with warm start, IEEE Trans. Power Syst. 35 (1) (2020) 711–720, https://doi.org/10.1109/TPWRS.2019.2930706.

[16] Y. Chen, F. Pan, J. Holzer, E. Rothberg, Y. Ma, A. Veeramany, A high performance computing based market economics driven neighborhood search and polishing algorithm for security constrained unit commitment, IEEE Trans. Power Syst. 36 (1) (2021) 292–302, https://doi.org/10.1109/TPWRS.2020.3005407.

[17] F. Mohammadi, M. Sahraei-Ardakani, D. Trakas, N.D. Hatziargyriou, Machine learning assisted stochastic unit commitment during hurricanes with predictable line outages, IEEE Trans. Power Syst. (2021) 1, https://doi.org/10.1109/TPWRS.2021.3069443.

[18] A.S. Xavier, F. Qiu, S. Ahmed, Learning to solve large-scale security-constrained unit commitment problems, INFORMS J. Comput. 33 (2) (2021) 739–756.

[19] A.S. Xavier, F. Qiu, S. Ahmed, Dataset: learning to solve large-scale security-constrained unit commitment problems, 2020, https://doi.org/10.5281/ZENODO.3648686.

[20] Y. Fu, Z. Li, L. Wu, Modeling and solution of the large-scale security-constrained unit commitment, IEEE Trans. Power Syst. 28 (4) (2013) 3524–3533, https://doi.org/10.1109/TPWRS.2013.2272518.

[21] G. Optimization, Gurobi optimizer reference manual, 2021.