Reduction of the Optimal Power Flow Problem through Meta-Optimization

Alex Robson [*] [1] **Mahdi Jamei** [*] [1] **Cozmin Ududec** [1] **Letif Mones** [1]

Abstract

We introduce a method for solving Optimal Power Flow (OPF) using meta-optimization, which can substantially reduce solution times. A pre-trained classifier that predicts the binding constraints of the system is used to generate an initial reduced OPF problem, defined by removing the predicted non-binding constraints. Through an iterative procedure, this initial set of constraints is then extended by those constraints that are violated but not represented in the reduced OPF, guaranteeing an optimal solution of the original OPF problem with the full set of constraints. The classifier is trained using a meta-loss objective, defined by the computational cost of the series of reduced OPF problems.

1. Introduction

Many complex systems, such as electricity grids, often require expensive computations for planning and operations. For example, a central task of electricity grid operators (Tong, 2004) is to periodically solve a constrained optimization problem, referred to as Optimal Power Flow (OPF). The goal of OPF is to dispatch generation in order to meet demand at minimal cost, while respecting reliability and security constraints. In general, this is a challenging problem for several reasons. First, OPF is a non-convex and non-linear constrained optimization problem that can be mixed integer in its full form. Second, it is computationally expensive due to the size of power grids, requiring a large number of diverse constraints to be satisfied. In order to minimize computational costs, various approximations are used, such as DC-OPF, which make the problem convex and reduce the number of variables and constraints. However, these approximations can lead to various inefficiencies in grid operations (Ilic et al., 2006).

Electricity production contributed to 27.5% of total U.S. greenhouse gas (GHG) emissions in 2017. An interesting research direction that may have large impacts on emissions is to take such effects into account directly in the formulation of OPF (Gholami et al., 2014), which makes the problem even more complex. Further, the integration of renewable energy sources such as wind and solar add other complications to OPF due to the volatility of these resources. Correspondingly, OPF needs to be solved as near to real-time as possible, further requiring improvements to OPF convergence times and robustness.

To solve the general OPF problem, interior-point methods (Wächter & Biegler, 2006) are typically used. These methods are robust but expensive as they require the computation of the second derivative of the Lagrangian of the system at each iteration. There are many approaches to overcome such problems by using modern machine learning techniques, which roughly fall into two distinct categories. One approach is based on directly inferring the optimal solution of OPF, using a regression in which the optimal solution is predicted based on the grid parameters. Although this is a straightforward end-to-end approach, there are two issues that make it impractical. First, since OPF is a constrained optimization problem, the solution is not a smooth function of the grid parameters. Therefore, properly training such regression models requires substantial training data (Guha et al., 2019). Second, although in theory this approach does not require solving OPF – as it provides a full set of grid state variables – there is no guarantee the solution satisfies all constraints. Such violations could lead to severe security issues for the grid.

Instead of using the outputs of a regression model directly, these can be used to warm-start by initializing an interior point method with the outputs of the regression model. This approach can significantly reduce the number of optimization iterations compared to the original problem. However, the computational gain is in practice marginal for several reasons. First, only primals are warm-started, then duals also need to converge (Jamei et al., 2019): interior-point methods set a minimum number of required iterations even if the primals are set to their optimal values. Second, if the initial value of primals is far from optimal, the optimization can lead to a different local minimum, or could potentially take even longer than the original problem. Finally, even if the predicted values are close to the optimal solution, they may begin in an infeasible region, which can result

[*] Equal contribution [1] Invenia Labs, Cambridge, United Kingdom. Correspondence to: Letif Mones <letif.mones@invenialabs.co.uk>.

in substantially longer solve times, or even convergence failure. The second approach is based on the observation that only a fraction of all constraints are actually binding at the optimum, so a reduced OPF can be formulated by keeping only the binding constraints. This suggests a classification formulation, in which the grid parameters are used to predict the binding status of each constraint. Unfortunately, this can also lead to security issues through false-negative predictions of the binding status of important constraints. However, by iteratively checking and adding violated constraints, and then solving the reduced OPF until all constraints of the full problem are satisfied, this issue can be avoided. Because the reduced OPF problems are much cheaper than the full problem, this approach (if converged in a few iterations) can be very efficient.

As discussed above, conventional methods do not guarantee security, and therefore additional steps are needed. This suggests the introduction of a new loss function for the classifier, which measures the total computational cost until a secure solution to the full OPF problem is provided by the above iterative procedure. In a previous work (Jamei et al., 2019) we combined the regression approach with such a meta-loss function and used meta-optimization to reduce the total number of OPF iterations by predicting an appropriate warm-start for the interior-point primal variables. Inspired by recent works in predicting active sets of constrains (Misra et al., 2018; Deka & Misra, 2019), in this paper we combine the classifier approach with meta-optimization to obtain a reduced OPF whose security iterations are computationally optimal. We demonstrate the capability of our method on several DC-OPF problems.

2. Methods

In order to explore a variety of distinct active sets, grid parameter samples with feasible solution were generated by varying these for 8 cases from Power Grid Lib (Babaeinejadsarookolaee et al., 2019). In the following, we restrict to DC-OPF problems and use the `PowerModels.jl` (Coffrin et al., 2018) OPF package. In order to encode the uncertainties that a grid operator might face, cases were generated by drawing samples for a combination of nodal loads, maximum output power of generators, line thermal ratings and line reactance values. In particular, the original case parameters were varied by a scaling factor drawn from uniform distributions of the form $\mathcal{U}(0.85, 1.15)$ for the load, and $\mathcal{U}(0.9, 1.1)$ for the other parameters.

The starting point of our method is to train a neural net based classifier using grid parameters as features to predict the binding status of the constraints of the full OPF problem. As the power flow equality constraints are always binding we limit considered constraints to branch flow and generator (upper) bound constraints (the latter restriction due to the potential for a divergent objective in the absence of lower limits of generator power). Each constraint is predicted to be binding or non-binding by a multi-label classifier. Correspondingly, we use a binary cross-entropy loss with sigmoid activation in the final layer. The reduced OPF problem consists of the same objective as the full problem, but only keeps those constraints that were predicted binding by the classifier.

As there may be violated constraints not included in the reduced model, to ensure convergence to a solution of the full problem we introduce a *security iteration procedure*. Here, a series of reduced OPF problems is solved and successively extended as needed. This has the following steps (Figure 1). 1) An initial candidate reduced set of constraints $\mathcal{A}_0$ is proposed by the classifier. A reduced OPF solution (p_0^*) is then found. 2) Then in each security iteration, $k \in 0 \ldots K$, the solution (p_k^*) of the reduced OPF is validated against the constraints $\mathcal{C}$ of the original full formulation. 3) Constraints ($\mathcal{N}_k$), which are violated in step k are added to the set of considered constraints, forming constraint set $\mathcal{A}_{k+1}$. 4) This procedure iterates until no violations are found ($\mathcal{N}_K = \emptyset$), and the solution ($p_K^*$) is feasible under the original constraints $\mathcal{C}$. We note that in general for convex OPF problems the procedure converges within a few iterations to the full solution.

A key aspect of our approach is to introduce a meta-loss function: we define the meta-loss objective as the total computation time of the security iteration procedure, and the meta-optimization as the optimization of the NN weights under the meta-loss objective over a training data set. To be specific, we refer to this step as meta-optimization as we are optimizing the performance of the OPF solver, by minimizing the solve-time. This optimization is through learning a reduced formulation of the OPF problem for the solver. The meta-loss is a non-differentiable function of the classifier weights, and therefore we use the gradient-free Particle Swarm Optimization (PSO) (Kennedy & Eberhart, 1995). The meta-optimization includes the solution of several reduced OPF problems for a given training example, which makes it more expensive than conventional neural net training: with N_t meta-training examples, N_p particles, and N_s meta-optimization steps, at least $N_\mathrm{t} \times N_\mathrm{p} \times N_\mathrm{s}$ reduced OPF calculations are performed.

To reduce the required number of iterations, the meta-optimization is started from a classifier trained by conventional stochastic gradient descent. This approach has the implicit assumption that the local optimum of the meta-loss is close to the local optimum of the cross-entropy loss. Although the optimal weights of a trained constraint classifier will be correlated with those trained under a meta-objective of reduced computation cost, each constraint will have variable importance in the convergence behaviour of the interior

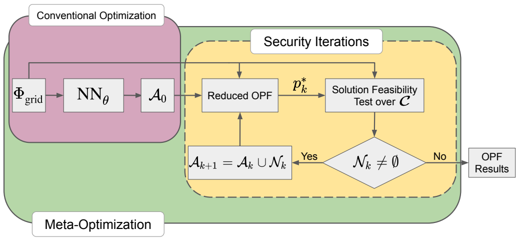
Conventional Optimization
Security Iterations
Φ_{grid}
NN_θ
$\mathcal{A}_0$
Reduced OPF
p_k^*
Solution Feasibility Test over $\mathcal{C}$
$\mathcal{A}_{k+1} = \mathcal{A}_k \cup \mathcal{N}_k$
Yes
$\mathcal{N}_k \neq \emptyset$
No
OPF Results
Meta-Optimization

Figure 1. Flowchart of meta-optimization under the security iteration procedure. Conventional optimization, that provides initial θ under optimal classification loss, is followed by meta-optimization of the meta-loss. Φ_{grid} is the vector of grid parameters, NN_θ is the classifier with weights θ. The meta-loss is computed within the security iteration, where $\mathcal{C}$ denotes the full set of constraints of the original OPF problem, $\mathcal{A}_k$ is the actual set used in the reduced OPF and $\mathcal{N}_k$ is the set of violated constraints. p_k^* is the solution of the reduced OPF, where $k = 1 \ldots K$ is the iteration index. The final solution p_K^* is at $k = K$ when $\mathcal{N}_K = \emptyset$.

point algorithm. Correspondingly, it is difficult to design a fully supervised and differentiable OPF-agnostic surrogate objective to reflect this. In practice, initial training through a cheap surrogate objective followed with training under the (expensive) meta-objective is a reasonable way to train.

3. Results

For scenario generation, Table 1 summarizes the number of distinct active sets along with the number of grid parameters and constraints as input and output sizes of the classifier, respectively. To highlight the complexity of the samples, we compared the number of active sets to those reported in (Ng et al., 2018), which were generated by scaling nodal load with a factor drawn from a normal distribution with $\mu = 1.0$ and $\sigma = 0.03$. The number of unique active sets in our samples is generally much higher, which can be attributed to varying more parameters (not just load), and selecting a wider deviation for the load values.

Table 1. Number of grid parameters, constraints, and unique active sets for different grids, using 1000 samples.

Case	# of grid params	# of constraints	# of active sets	
	(neural net input)	(neural net output)	(Ng et al., 2018)	This work
24-ieee-rts	125	140	5	15
30-ieee	105	86	1	1
39-epri	123	112	2	8
57-ieee	206	168	3	8
73-ieee-rts	387	432	21	8
118-ieee	490	410	2	66
162-ieee-dtc	693	592	9	188
300-ieee	1080	936	22	835

Based on the generated 1000 samples for each grid we also computed the upper limit of computational gain ($\frac{t_0 - t}{t_0} \times 100$) of the two machine learning approaches, i.e. using a perfect regressor (full DC-OPF with warm-start from the exact primals) and classifier (reduced DC-OPF using all and only binding constraints). The results are collected in Table 2.

The reduced OPF (classification) outperforms the warm-start approach (regression) even for small systems, and the gain becomes more significant for larger cases indicating a higher potential of the classification.

For the classifier, a 3-layer neural net with 50×50 hidden unit was used, with ReLu activations for the first two layers, sigmoid activation for the output layer, and dropout with a ratio of 0.4. ADAM with $\eta = 10^{-4}$ was used for optimization. We used a split of 900/100 for training and testing for each grid, terminating after convergence of the loss (20 epochs). For meta-optimization the training data consisted of 100 randomly drawn examples from those used for conventional training. We found that significant improvement can be achieved by using a relatively small number of 10 PSO iterations and 10 particles. For testing, 100 samples were used to compute the meta-loss before and after meta-optimization. For each grid, 40 independent experiments were performed by randomly selecting the corresponding conventional and meta-training data and test data.

Table 2. Average computation gain for perfect regression (i.e., full DC-OPF with warm-start) and classification (i.e., reduced DC-OPF) for 8 cases compared to conventional full DC-OPF.

Case	Gain	
	Regression	Classification
24-ieee-rts	45.6±0.6	52.1±0.6
30-ieee	16.3±0.9	45.2±0.7
39-epri	37.8±0.6	62.3±0.4
57-ieee	34.8±0.7	61.7±0.4
73-ieee-rts	45.0±0.6	59.6±0.4
118-ieee	38.8±0.6	74.8±0.2
162-ieee-dtc	60.9±0.4	84.8±0.1
300-ieee	54.7±0.3	84.2±0.1

For smaller grid sizes, marginal or no improvement was observed. This can be explained by the low number of distinct active sets (Table 1), which can be learned easily by a conventional classifier. However, as the number of distinct active sets increases, the meta-optimization begins improving the classifier more significantly (Table 3).

Table 3. Average computational gain computed from the meta-loss before and after meta-optimization.

Case	Gain
118-ieee	14.4±3.1
162-ieee-dtc	46.2±3.1
300-ieee	49.2±5.2

4. Conclusion

We introduced a meta-loss function that measures the computational cost of obtaining a guaranteed solution of an OPF problem, by iteratively solving a series of reduced problems. The initial reduced OPF problem consists of constraints predicted as binding by a neural network based classifier, and this set is then iteratively extended, resulting in a series of reduced OPFs. We further performed a meta-optimization of the classifier, minimizing the meta-loss. We found that training the classifier using a conventional loss function is appropriate for small grids, where the number of distinct active sets is limited, but as the grid size increases, the use of meta-optimization results in significant computational gains. Our preliminary results for even larger grids indicate an increasing gain with the system size.

Finally we note that the method can be easily extended to AC-OPF problems. In this case, although we expect a similar trend of increasing gain with the system size, in general this gain might be more moderate due to the high number of non-linear equality constraints providing the computationally most expensive part of the optimization using interior-point methods. We have begun preliminary investigations scaling to larger grids and AC-OPF and leave a full investigation to future work.

References

Babaeinejadsarookolaee, S., Birchfield, A., Christie, R. D., Coffrin, C., DeMarco, C., Diao, R., Ferris, M., Fliscounakis, S., Greene, S., Huang, R., Josz, C., Korab, R., Lesieutre, B., Maeght, J., Molzahn, D. K., Overbye, T. J., Panciatici, P., Park, B., Snodgrass, J., and Zimmerman, R. The Power Grid Library for Benchmarking AC Optimal Power Flow Algorithms. *arXiv e-prints*, art. arXiv:1908.02788, Aug 2019.

Coffrin, C., Bent, R., Sundar, K., Ng, Y., and Lubin, M. Powermodels.jl: An open-source framework for exploring power flow formulations. In *2018 Power Systems Computation Conference (PSCC)*, pp. 1–8. IEEE, 2018.

Deka, D. and Misra, S. Learning for DC-OPF: Classifying active sets using neural nets. *arXiv e-prints*, art. arXiv:1902.05607, Feb 2019.

Gholami, A., Ansari, J., Jamei, M., and Kazemi, A. Environmental/economic dispatch incorporating renewable energy sources and plug-in vehicles. *IET Generation, Transmission & Distribution*, 8(12):2183–2198, 2014.

Guha, N., Wang, Z., and Majumdar, A. Machine learning for ac optimal power flow. ICML, Climate Change: How Can AI Help? Workshop, 2019.

Ilic, M. D., Lang, J. H., Litvinov, E., and Luo, X. The critical role of computationally robust ac optimal power flow in reliable and efficient reactive power/voltage dispatch. In *2006 IEEE PES Power Systems Conference and Exposition*, pp. 689–698. IEEE, 2006.

Jamei, M., Mones, L., Robson, A., White, L., Requeima, J., and Ududec, C. Meta-optimization of optimal power flow. ICML, Climate Change: How Can AI Help? Workshop, 2019.

Kennedy, J. and Eberhart, R. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pp. 1942–1948 vol.4, Nov 1995. doi: 10.1109/ICNN.1995.488968.

Misra, S., Roald, L., and Ng, Y. Learning for Constrained Optimization: Identifying Optimal Active Constraint Sets. *arXiv e-prints*, art. arXiv:1802.09639, Feb 2018.

Ng, Y., Misra, S., Roald, L. A., and Backhaus, S. Statistical Learning For DC Optimal Power Flow. *arXiv e-prints*, art. arXiv:1801.07809, Jan 2018.

Tong, J. Overview of pjm energy market design, operation and experience. In *2004 IEEE International Conference on Electric Utility Deregulation, Restructuring and Power Technologies. Proceedings*, volume 1, pp. 24–27. IEEE, 2004.

Wächter, A. and Biegler, L. T. On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006.