PROCEEDINGS OF THE 31st European Symposium on Computer Aided Process Engineering (ESCAPE31), June 6-9, 2021, İstanbul, Turkey
M. Türkay, R. Gani (Editors).
http://dx.doi.org/10.1016/B978-0-323-88506-5.50142-X

AC-Optimal Power Flow Solutions with Security Constraints from Deep Neural Network Models

Zachary Kilwein[a], Fani Boukouvala[a], Carl Laird[b], Anya Castillo[b], Logan Blakely[b], Michael Eydenberg[b], Jordan Jalving[b], Lisa Batsch-Smith[b]

[a]*Georgia Institute of Technology, North Ave NW, Atlanta, GA, 30332, USA*

[b]*Sandia National Labs, 1515 Eubank Blvd SE, Albuquerque, NM, 87123, USA*

zkilwein3@gatech.edu

Abstract

In power grid operation, optimal power flow (OPF) problems are solved several times per day to find economically optimal generator setpoints that balance given load demands. Ideally, we seek an optimal solution that is also "*N-1* secure", meaning the system can absorb contingency events such as transmission line or generator failure without loss of service. Current practice is to solve the OPF problem and then check a subset of contingencies against heuristic values, resulting in, at best, suboptimal solutions. Unfortunately, online solution of the OPF problem including the full *N-1* contingencies (i.e., two-stage stochastic programming formulation) is intractable for even modest sized electrical grids. To address this challenge, this work presents an efficient method to embed N-1 security constraints into the solution of the OPF by using Neural Network (NN) models to represent the security boundary. Our approach introduces a novel sampling technique, as well as a tuneable parameter to allow operators to balance the conservativeness of the security model within the OPF problem. Our results show that we are able to solve contingency formulations of larger size grids than reported in literature using non-linear programming (NLP) formulations with embedded NN models to local optimality. Solutions found with the NN constraint have marginally increased computational time but are more secure to contingency events.
Keywords: Optimization, Machine Learning, Neural Networks, Power Grid Modeling

1. Introduction

1.1 Background

Power grids are very large, complex systems that require operation that is both financially optimal and secure to unforeseen outage events. The US Federal Energy Commission estimates that effective optimization of power grids can save tens of billions of dollars annually (Cain et al. 2012). Furthermore, the need for powerful optimization tools will only increase as more renewables and non-conventional energy sources are added to the grid. This is largely done via optimal power flow (OPF) programs which balance real-time load demands with generator outputs in the most economically optimal way. Some approaches to solving OPF include rigorous formulations with non-linear and non-convex functions (AC-OPF), convex relaxations of this problem (e.g. Second Order Cone OPF), and linear approximations (e.g. DC-OPF) that are standardly used in industrial applications. In the case of AC-OPF, further distinction can be made between local and global solution algorithms due to non-convexities. This work focuses on the optimization of the rigorous AC-OPF formulation with local solvers, since it can handle large problems without approximating underlying power flow physics and very often achieves solutions close to global optimality (O'Neill et al 2012).

As grids are operated close to their optimal points (i.e., less conservatively), security must be considered as they are more prone to system component failure that may result in serious financial and safety consequences. Operating at suboptimal set points will yield significant cost increases, whereas economically-optimal though insecure set points have potentially catastrophic consequences under a contingency event, as cascading failures can cause blackouts for millions of people. Therefore it is incumbent upon system operators to balance these interests effectively. One way to check system security is to simulate system failures and enforce the resulting security of the system as constraints. If this is done for every component in the system, it is said to be *N-1* secure. While the scalability and accuracy of AC-OPF algorithms have seen major improvements for both local and global techniques, ensuring *N-1* security remains challenging. The current practice is to solve the OPF problem without consideration of contingencies, and then check the solution for *N-1* feasibility. If the solution is not feasible, heuristics are used to add constraints to the OPF problem, and the process is repeated. This approach yields suboptimal results, and since only a subset of the contingencies is typically checked, this may also yield insecure results. While it is possible to formulate a two-stage stochastic programming problem that considers all contingencies, this large scale problem is not tractable for realistic networks. This work aims at solving security constrained (SC) AC-OPF problems with the aid of deep Neural Networks (NN) that can be trained offline to learn the input space of the original optimization problem that is *N-1* secure. The trained NN model can then be used as a single constraint to enforce *N-1* security of any resultant AC-OPF solution across the input space and contingency events.

1.2 Literature Review and Contribution

Capitanescu et al. (2011) provide a thorough review of previous research in the area of SC OPF and future methods that may address some of the challenges detailed above. Heuristic methods are frequently used in practice, but more robust relationships can be derived as algebraic constraints. However, due to the problem size and complexity, fully detailed formulations remain intractable and often reduced or simplified techniques are required. Gutierrez-Martinez et al. (2011) trained non-linear regression models and single layer NN models to find the security boundary of a given load profile for the worst case contingency. Our work builds on this by using deep NN models as a map of the secure and insecure space of all contingencies of interest, variable generation patterns, and higher dimensional representations of the security boundary. Velloso et al. 2020 replace the full SC AC-OPF with a NN model that includes physics-based constraints via a Lagrangian-dual. Venzke et al. (2020) trained a NN security classifier with ReLU activation functions and added the constraints to the AC-OPF problem as a set of linear constraints with binary variables that results in a mixed-integer non-linear program (MINLP). They linearize the power equations to convert this to a MILP.

This work instead maintains the non-linearity of the NN based security-constraints and formulates the overall problem instead as a nonlinear program (NLP) that can be solved efficiently via interior-point methods. Some key advantages of our approach include: (a) computationally challenging security simulations are performed offline, (b) security is represented by a single NN constraint without integer variables, (c) the fully detailed AC-OPF formulation is used, (d) the security constraint scales linearly with grid size, and (e) our method is implemented within the Pyomo platform that enables easy integration with Python-based algorithms for training ML models.

2. Overview of Methods

2.1. AC Optimal Power Flow Problem Formulation

Below, the AC-OPF formulation used in this work is given. Real power, reactive power, voltage magnitude and voltage angle are denoted by p, q, v and θ respectively. N and K denote the sets of all nodes and branches in the system.

$$Min\ C\left(p^{g}\right) \tag{1}$$

$$s.t.\ \sum_{k(n,m)\in K_n^{out}} p^{f}_{k(n,m)} + \sum_{k(n,m)\in K_n^{in}} p^{t}_{k(n,m)} - p^{g}_{n} + p^{d}_{n} = 0,\ \forall n \in N \tag{2}$$

$$\sum_{k(n,m)\in K_n^{out}} q^{f}_{k(n,m)} + \sum_{k(n,m)\in K_n^{in}} q^{t}_{k(n,m)} - q^{g}_{n} + q^{d}_{n} = 0,\ \forall n \in N \tag{3}$$

$$p^{f}_{k(n,m)} = g_k v_n^2 - v_n v_m\left(g_k cos\theta_{n,m} + b_k sin\theta_{n,m}\right),\ \forall k \in K \tag{4}$$

$$p^{t}_{k(n,m)} = g_k v_m^2 - v_n v_m\left(g_k cos\theta_{n,m} - b_k sin\theta_{n,m}\right),\ \forall k \in K \tag{5}$$

$$q^{f}_{k(n,m)} =- (b_k + b_k^{sh})v_n^2 - v_n v_m\left(g_k sin\theta_{n,m} - b_k cos\theta_{n,m}\right),\ \forall k \in K \tag{6}$$

$$q^{t}_{k(n,m)} =- \left(b_k + b_k^{sh}\right)v_m^2 + v_n v_m\left(g_k sin\theta_{n,m} + b_k cos\theta_{n,m}\right),\ \forall k \in K \tag{7}$$

$$v_n^{min} \le v_n \le v_n^{max},\ \forall n \in N \tag{8}$$

$$\theta_{nm}^{min} \le \theta_n - \theta_m \le \theta_{nm}^{max},\ \forall \{n, m\} \in K \tag{10}$$

$$p_n^{min} \le p_n \le p_n^{max},\ \forall n \in N \tag{11}$$

$$q_n^{min} \le q_n \le q_n^{max},\ \forall n \in N \tag{12}$$

$$(p_k^f + q_k^f)^2 \le\ (Thermal\ Limit)^2,\ \forall k \in K \tag{13}$$

$$(p_k^t + q_k^t)^2 \le\ (Thermal\ Limit)^2,\ \forall k \in K \tag{14}$$

Eq 1 minimizes the cost of real power at generator buses. Eqs 2-3 represent the nodal power balance derived from Kirchoff's Law. Power flow through each branch is given in Eqs 4-7. Finally, Eqs 8-14 ensure the AC-OPF solution is secure with no contingencies (*N-0*). To check static *N-1* security of an AC-OPF solution, (v_n, p^g) are set for all generators, (p^d, q^d) for all loads, and (v_n, θ) at the reference bus. Then under each contingency in a given set, the specific element is taken out of the system (e.g., broken line) and Eqs 2-7 are re-solved, without the objective function. If a solution is found that satisfies Eqs 8-14, the solution is secure to that contingency. If this is true for all contingencies, the system is said to be *N-1* secure. An extensive formulation guaranteeing *N-1* security requires a scenario for each contingency, leading to a very large non-linear two-step stochastic programming problem. In this work, we train a NN to capture the *N-1* security boundary and then embed this NN into the AC-OPF formulation.

2.2. Sampling and Data Set Construction

While one major advantage of this approach is to shift computational costs to offline contingency simulations, it is still necessary to have a methodology that samples data points in an intelligent manner. The input space for the problem is very large with a majority of points far from the security boundary. In order to train a NN model that effectively distinguishes these regions, many points must be generated on the security boundary itself in addition to secure/insecure points. Our sampling algorithm presented below.

Sampling Algorithm for Boundary Points

Load_Factors=[LF^{min},…,LF^{max}], Load_Dirs=[$\vec{LD}^{0}, \ldots, \vec{LD}^{j}$]
for LF **in** Load_Factors:
    sol=Solve_ACOPF($p^{d} * LF, Power\ Factor = p^{d}/(q^{d} + p^{d})^{2}$)
    y_{sec}=check_Nmin1(sol) (check N-1 compliance of baseline sol'n)
    for j **in** Load_Dirs:
        while y_{sec}=True: (continue until no longer N-1 secure)
            sol=solve_acopf($p^{d} * \vec{LD}^{j} * 1.1$)
            y_{sec}=check_Nmin1(sol)

A set of load factors is constructed between maximum and minimal values that give *N-0* security. For each of these load factors, an AC-OPF problem is solved given this load profile, and the resulting model is used to check *N-1* contingencies via the method described in Section 2.1. If the system is *N-1* secure, the load vector is ramped up by 10% increases until the security check fails. The points directly before and after failure are saved to enhance boundary classification. This is done for all LF's, several load ramping directions (normalized vectors generated via Latin Hypercube sampling), and variable generation patterns (i.e.. variable generator costing parameters) resulting in an increased set of boundary data points. Intermediate solutions that satisfy *N-1* contingency are saved as secure and points that are infeasible for AC-OPF are saved as insecure to complement the boundary data. This approach provides a set of points that straddle the security boundary, producing a more balanced and targeted training set.

2.3. Neural Network Model Training

Using the constructed data-set, the next stage is to train a NN model that can predict whether a given AC-OPF solution is N-1 secure. An important decision is the selection of input variables. The real power setpoints at generators (p^{g}) are included as they are the ultimate variables in the objective function of AC-OPF, allowing for a direct mapping. Power load (p^{d}) at all buses are also included since they have a strong correlation with grid security and improve NN accuracy. Thus, the input space of the NN model scales linearly with the number of buses and generators. More input variables may augment NN accuracy in future cases but didn't show strong independent correlation with system security. The Sequential model from Tensorflow's Keras is used as the basis for the feed-forward NN model. It is composed of two hidden layers of 20 nodes each and hyperbolic tangent activation function plus a softmax output layer. Output variable y_{sec} was set to 1 for insecure points and 0 for secure points. ADAM optimizer was used with a standard cross-entropy loss function. Training continued until validation error plateaued or 4000 epochs were reached.

2.4. Constrained Optimization Formulation

The trained NN can now be thought of as a large non-linear constraint that must be satisfied in order to return an AC-OPF solution that is secure under all contingencies. The NN constraint can be added to the original optimization problem (Eqs 1-14) to encode *N-1* security.

$$y_{sec} = NN(\{p^d\}, \{p^g\}) \tag{15}$$

$$y_{sec} \leq \alpha \tag{16}$$

Using the trained weights and biases of the NN, Eqs 15-16 are embedded within the AC-OPF model from Egret using Pyomo symbolic equation construction (Knuevan et al 2019). The resulting constraint is fully differentiable and able to be solved with IPOPT. Parameter α is a scalar that controls the constraint conservativeness.

3. Results

Two case studies are used to demonstrate the method: the first (Case 30) contains 30 buses, 6 generators, 21 loads, and 41 branches. The contingency set includes branches {1, 2, 3, 6, and 10}. The next (Case 118) contains 118 buses, 54 generators, 99 loads, and 186 branches. The contingency set includes branches {55,168,169,171,172,173,175}. The contingency set is selected during the sampling process by eliminating contingency events that always make the system insecure or never pose an issue to security. The unconstrained AC-OPF returns a solution that is not N-1 secure for 45% of the sampled points for Case 30 and 64% for Case 118. This proves that standard AC-OPF is inadequate for considering grid security. With the NN constraint, the vast majority of insecure operating conditions can be identified and avoided. The control of conservativeness by α can be illustrated through a receiver operator curve (ROC) which shows the false positive and true positive rate of the NN classifier. Figure 1 shows the ROC results for the NN classifier with the full training set and for a model with half the training data.

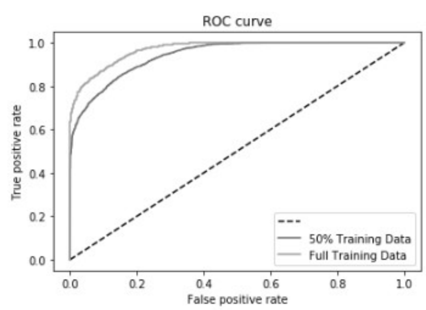

Figure 1. Receiver Operator Curve (ROC) Results (Case 118)

Again, there is an expected trade-off between conservativeness and accuracy. While a high true positive rate allows us the most confidence in grid security, the accuracy trade-off would be important to tailor to the specific grid application. For an α value of 0.5, model accuracy is 89%, much better than base case AC-OPF. The results for Case 30 show nearly identical behavior on the ROC. Another important factor to consider is computational time of the proposed approach, where it distinguishes itself from the extensive formulation. Literature values for Case 118 take ~400 sec to run SC AC-OPF (Kang et al 2015) on a single processor of Sandia's Red Mesa supercomputer. Our NNSC AC-OPF approach never exceeds **15 sec for all points, with an average of 7 sec**.

One more important thing to consider is the effect of the security constraint on the objective function. We can quantify the cost of incorporating the NNSC constraint by considering the change in objective function value compared to the base case AC-OPF. In order to distinguish meaningful cost increases (to preserve security) from an overly conservative NNSC, models are tested with points that are base case secure and insecure. Increases in the objective function are negligible for over 70% of secure test points and the majority of remaining points constitute increases of less than 5%. This shows that α of 0.5 does a good job of not being overly conservative with respect to security. For insecure inputs, the objective function increases up to 50% in order to meet the NNSC constraint. These solutions come with far greater confidence about security under contingency but also come at an economic cost. Further comparisons with the extensive formulation and cost/risk strategies would be needed to definitively set α in real-world applications.

4. Conclusions

Power grid optimization and security is a vital area of research that can help save money, avoid black-outs, and promote the integration of non-traditional energy sources onto the grid. Using detailed simulations, it is possible to optimize very accurate models of power grids using non-convex/non-linear formulations such as AC-OPF. Adding security constraints for all N-1 contingencies to this problem is computationally challenging and extensive formulations are intractable in large-scale grids. This work addresses this challenge by approximating security constraints with a NN model that can be trained offline and then embedded within a Pyomo-based model as a single non-linear constraint. Results show that this approach can accurately estimate the security boundary and provide a framework to balance optimality and security of system set points. The NLP formulation allows for computationally tractable solutions even for large grids.

Disclaimer: Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

References

B. Knueven et al.,2019, Egret v.01(beta), Sandia National Lab (SNL-NM), Python

M.B. Cain, R.P. Oneill, A. Castillo, 2012, History of Optimal Power Flow and Formulations, Federal Energy Regulatory Commision, p:1-36

F. Capitanescu, J.M. Ramos, P. Panciatici, D. Kirschen, A.M. Marcolini, L. Platbrood, L. Wehenkel, 2011, State-of-the-art, challenges, and future trends in security constrained optimal power flow, Electric Power Systems Research.

A. Gulli, S. Pal, 2017, Deep learning with Keras, Packt Publishing Ltd

V.J. Gutierrez-Martinez, C.A. Canizares, C.R. Fuerte-Esquivel, A. Pizano-Martinez, X. Gu, 2011, Neural-Network Security-Boundary Constrained Optimal Power Flow, IEEE Transactions on Power Systems.

W.E. Hart, C.D. Laird, J.P. Watson, D.L. Woodruff, G.A. Hackebeil, B.L. Nicholson, J.D. Siirola, 2017, Pyomo – Optimization Modeling in Python. Second Edition. Vol. 67. Springer.

J. Kang, A. Jayaraman, C. Laird,M. El-Halwagi, V. Sarin, 2015, An Efficient Interior-Point Decomposition for Parallel Solution of Large-Scale Nonlinear Problems with Significant Variable Coupling, Doctoral Thesis, Texas A&M

R. O'Neill, A. Castillo, M.B. Cain, 2012, The IV Formulation and Linearizations of the AC Optimal Power Flow Problem, FERC staff technical paper

X. Pan, M. Chen, T. Zhao, S.H. Low, 2020, DeepOPF: A Feasibility-Optimized Deep Neural Network Approach for AC Optimal Power Flow Problems, Under Review.

A. Velloso, P.Van Hentenryck, 2020, Combining Deep Learning and Optimization for Security-Constrained Optimal Power Flow, Under Review.

A. Venzke, D. Molzahn, S. Chatzivasileiadis, 2019, Efficient Creation of Datasets for Data-Driven Power System Applications, Electrical Engineering and Systems Science: Systems and Control, Under Review.

A. Venzke, D.T. Viola, J. Mermet-Guyennet, G.S. Misyris, S. Chatzivasileiadis, 2020, Neural Networks for Encoding Dynamic Security-Constrained Optimal Power Flow to Mixed-Integer Linear Programs, Under Review.