Fast Multi-Period Security-Constrained Economic Dispatch Based on Deep Neural Networks

Yan Yang, Sijun Du, Zhengchun Zhu, Mingxu Xiang and Juan Yu*
State Key Laboratory of Power Transmission Equipment & System Security and New Technology, College of Electrical Engineering, Chongqing University, Chongqing 400044, China
*Corresponding author’s e-mail: yujuancqu@cqu.edu.cn

Abstract. Security-constrained economic dispatch (SCED) is one of the most important daily tasks for operators. The scale of security constraints is huge for practically-sized power systems, which makes the SCED difficult or even impossible to be solved. Whereby, the number of active security constraints is relatively small. By eliminating the inactive security constraints, the complexity of SCED can be significantly reduced. Focusing on it, this paper proposes an intelligent framework to accelerate the calculation of SCED without any loss of accuracy. The proposed framework uses a deep neural network (DNN) to reduce the online computational cost significantly by shifting the heavy computation into offline training. More specifically, a DNN is used to extract the feature of SCED, which can effectively pre-identify the active constraints of SCED model. Moreover, an efficient lightweight learning strategy is presented to improve the learning efficiency of the DNN by the feature selection and feature decomposition. The effectiveness of the proposed method is demonstrated in modified IEEE benchmark systems.

1. Introduction

Security-constrained economic dispatch (SCED) guarantees the secure and economic operation of power grids, which is performed in everyday operation in power industries[1]. An enormous number of transmission constraints in N-1 contingencies make the SCED model large and computationally challenging. For practically-sized power grids, it is inefficient and not robust to directly optimize the SCED model with all N-1 security constraints. Therefore, a series of work has been carried out in the industry and academia on how to accelerate the calculation efficiency of the SCED problem.

In the industry, operators firstly solve the SCED model only with generation ramp constraints and power balance constraints. Then, active constraints are iteratively added into the SCED model by N-1 analysis until no new active constraints are identified. This method could achieve the optimal solution and mitigate the computational pressure. However, the computational time directly rises with the number of iterations. According to the experience of operators, the most likely active constraints will be added to the SCED model in the first iteration to accelerate the calculation. It is simple and efficient but lack of the scientific guidance.

In recent years, some advances based on analytical derivation for the SCED problem have been made along two avenues: 1) decomposition techniques to solve some relatively small-scale optimization problems instead of the large-scale SCED problem[2], [3], and 2) contingency constraints filtering approaches to effectively reduce the problem size[4] -[6]. The decomposition techniques are not efficient enough because several optimal problems still need to be solved. Besides, it may only obtain the sub-optimal solutions. Essentially, the difficulty of identifying the active constraints is comparable to that of...
obtaining the global optimal solution of the SCED problem. Therefore, it is hard to efficiently identify all the active constraints by above analytical identification methods before optimizing the SCED. Especially, some data-driven approaches have been proposed to identify the constraints. For instance, violation ranking is used in [7] to select the candidate constraints, which is simple and efficient but not robust. Statistical learning is used in [8] to establish a series of candidate active sets for the DC optimal power flow problem. An optimal model is proposed to find the best one from the candidate active sets. The computational performance of reference [8] will be compromised when the number of candidate active sets is large.

Studies have demonstrated that the deep neural network (DNN) has the ability to approximate any function with high accuracy in theory[9]. With the fast development of deep learning, it helps DNNs bring significant improvements to many areas. It also has ignited a boom of DNN-based applications in power systems [10], [11]. In fact, the fast-developing deep learning techniques have also provided a promising way to effectively capture the complex nonlinear relationship between the optimal solution and system operating condition for the SECD problem. Besides, it allows the DNN directly map the solutions of all the unsolved samples, which can dramatically speed up the calculation of SCED. Therefore, taking advantage of the numerous historical operation data and correct SCED model, this paper utilizes a DNN to quickly compute the solutions of SCED.

It is not practical to design a DNN with universal approximation capacity for all the SCED problems. Therefore, the crux of the DNN-based SCED is how to guarantee the calculation accuracy. Besides, an effective learning strategy for the SCED is the key technology in this study.

Regarding the problem mentioned above, a DNN-based approach is proposed to fast calculate SCED. The contributions are as follows:

I) An intelligent framework based on DNN is proposed to speed up the calculation of SCED. An efficient data-driven method based on DNN is proposed to calculate the SCED by reducing the size of the N-1 constraints. The proposed method can greatly improve the efficiency of SCED calculation without compromising its accuracy.

II) A lightweight learning strategy is designed to improve the training efficiency. The generation power output instead of the active constraints is used as the output feature vector to reduce the computational complexity of the DNN. Besides, multi-period generation power is decomposed into single-period generation power to be learned for both reducing training samples and improving learning accuracy.

This paper is organized as follows. The intelligent framework SCED model based on a DNN is proposed in Section 2. Lightweight learning strategy for SCED model is presented in Section 3. Numerical test is shown in Section 4, followed by conclusions in Section 5.

2. Intelligent Framework Bases on DNNs to Speed UP SCED

2.1. Basic model of SCED

The objective function of the SCED model is shown as follows:

$$\min_{P_G} P_G a P_G^T + b P_G + c $$

(1)

Where $P_G^T$ represents the generator output; $a$, $b$, and $c$ represent the operational costs.

According to the current industrial practice, a DC power flow model is used to model the power flow [3], [6]. The constraints in the SCED model are shown as follows:

- System load balance

$$e_G P_G = e_D P_D$$

(2)

Where $e_G$ and $e_D$ represent the “all-one” vector, and $P_D$ is the load demand.

- Generator output limits

$$P_G \leq P_G \leq P_G$$

(3)

where $P_G$ and $P_G$ are the upper and lower limits of generator output, respectively.

- Transmission line capacity constraints
Where
\[
P_{\text{line}}^{i,j,c} \leq P_{\text{line}}^{i,j,c} \leq P_{\text{line}}^{i,j,c}, (i, j) \in K, c \in C
\]

(4)

In this model, \(P_{\text{line}}^{i,j,c}\) is the power flow on the branch \((i, j)\) in condition \(c\) \((c = 0\) is the normal condition); \(S_{i,j,c}\) is to the power transfer distribution factor matrix.

2.2. Data-driven framework for SCED model

In industry entries, the SCED model is generally solved iteratively to reduce the high computational burden, illustrated in Figure 1 (the red dotted box). 1. It first solves an economic dispatch model that only contains constraints (1)-(3); 2. Then the solution result is analysed for N-1 security constraints, and the active security constraint sets is added to the economic dispatch model to solve again; 3. Finally, the second process is repeated for iterative solution until there is no new set of active security constraints, and then the optimal solution of the SCED model is obtained.

The above method solves the SCED model iteratively, which can effectively obtain the optimal result and realize economic dispatch. However, as the scale of the system increases, the N-1 security constraints also increase exponentially. Therefore, the number of iterations of the iterative method for solving SCED may increase rapidly, resulting in a longer solution time. Therefore, the pre-identification of N-1 security constraints becomes significant important.

The intelligent calculation framework of SCED is shown as Figure 1 (the black solid line). This framework tries to avoid the iterative calculation (the blue dotted box), by pre-identifying the set of the active N-1 security constraints. Hence, the computational burden of the SCED model can be significantly reduced. Besides, DNNs are embedded in the calculation flowchart of SCED, which won’t affect the solution accuracy of the SCED model. The key to achieve it is how to utilize the strong feature-extracted ability of DNNs to accurately pre-identify all the active N-1 security constrains. Directly utilizing DNNs is not attractive, because the SCED is complex and requires a huge neural network structure must be needed to extract the complex features of SCED, and the huge structure generally suffers from enormous training data and heavy computational burden, especially for large-scale power systems. Therefore, an efficient lightweight learning strategy is presented to improve the learning efficiency of the DNN by the feature selection and feature decomposition.

Figure 1. Intelligent calculation framework of SCED
3. Lightweight learning strategy for SCED model based on deep learning

3.1. Feature vector selection
Essentially, deep neural network fits or learns functions and problems by quantifying the impact of input changes on output. In the pre-identification of active security constraints of the SCED model, it is to analyse the influence of changes in system operating conditions on the identification of active security constraints.

Therefore, the input feature vector is designed as the node injection power and the power plant quote.

\[ X = [ P_i, Q_i, a, b, c ] \] (6)

Correspondingly, the input feature vector can be intuitively designed as the status that whether the N-1 security constraint is active. However, this will increase the output dimension of the neural network, resulting in a more neural network structure, higher training costs, and higher training difficulty.

Fortunately, the active constraint sets can be directly determined by the generator output. Therefore, the output feature vector is designed as generation power output in this paper.

\[ Y = P_G \] (7)

3.2. Feature decomposition
Generally, the basic learning strategy is to train the unique deep neural network by using the full-period system operating conditions and the corresponding generation power output. However, such a training strategy will result in an enormous neural network structure, and require more training samples and longer training time, which is also a huge challenge for computer hardware performance. In order to reduce the computational burden and improve the efficiency of training, this paper decomposes the feature vector into 24 sub-feature vectors according to the dispatch period by equation (8), which are used to train 24 deep neural networks. This reduces the complexity of each network, and the required training costs are also reduced accordingly.

\[ V = [ V_1, V_2, ... , V_{24} ] \] (8)

As for other components of learning strategy, Similar to reference [12], in the lightweight learning strategy of this paper, Z-score is still used as the normalization method and ReLU is used as the activation function except for the last layer which uses the linear activation function. The square difference function is used as loss function in this paper. In view of the Adam [13] optimization algorithm combines the ability of AdaGrad to handle sparse gradients and the ability of RMSProp to handle non-stationary targets and it shows excellent performance in optimization problems with large data or network parameters, the Adam is chosen as optimization algorithm in the training process.

Above all, the proposed fast calculation of SCED model based on deep learning is summarized in Figure 2.
4. Numerical test

4.1. Case information

The system data of the IEEE 30-bus and IEEE 118-bus systems are given in [14]. The power demand is sampled randomly by normal distribution to generate samples with a standard deviation (10% of the expected value). The load demand in [14] is chosen as the expected value.

This paper will compare the following SCED model solving methods (M0-M2):

M0: Solution method of SCED model based on deep learning with lightweight (period-period) learning strategy (The proposed method).

M1: Solution method of SCED model based on deep learning with full-period learning strategy.

M2: Industrial iterative solution method of SCED model.

When training the above-mentioned neural network using the deep learning method, the training end criterion is that the neural network meets the early stopping method or the number of iterations reaches the threshold 100.

All the cases in this paper are simulated and tested in the hardware environment of Intel Core(TM) i8-8700K CPU @ 3.70 GHz and 32 GB RAM, and Yalmip makes MATLAB call the Gurobi solver.

4.2. Effectiveness comparison with different learning strategies

In order to verify the effectiveness of the proposed lightweight learning strategy, this section will compare the prediction accuracy and the required computing resources of M1 and M0 in the optimal generation power output prediction.

Table 1 shows that the computational resources required for M1 and M0 to achieve the same prediction accuracy. As can be seen from the table, the training samples required to use the full-period learning strategy is 100,000 and the training time is 10,500s in case 30, however, the training samples required to use the proposed light weight learning strategy are less than one-third of the former, only 30,000, and the training speed is also nearly four times faster than the former; in the testing of case 118, the full-period learning strategy has exceeded the computing power of the hardware due to the heavy computational burden and cannot be calculated, but the proposed lightweight learning strategy can still complete training and calculation quickly.

Table 2 further shows the prediction accuracy that M1 and M0 can achieve under the same number of training samples. It can be seen from Table 2 that when the training sample is fixed at 30,000, the probability that the absolute error of prediction generation power output $> 5$MW is 12.8% using the full-
period learning strategy. In the same situation, the probability of absolute error > 5 MW is less than 1% with the proposed lightweight learning strategy and the training speed is faster.

Overall, the lightweight learning strategy proposed in this paper can effectively reduce the training and calculation burden of deep neural networks, and achieve accurate and fast generation power output prediction.

Table 1. Computational resources required for full-by-period learning and period-period learning strategy (the probability of absolute error > 5 MW is less than 1%)

| Case   | Method | Required samples | Network structure | The number of epochs | Time for each epoch (s) | Training time (s) |
|--------|--------|------------------|-------------------|----------------------|------------------------|------------------|
| Case30 | M1     | 100,000          | 800 × 4           | 500                  | 21.1                   | 10,500           |
|        | M0     | 30,000           | 2 × 100(× 24)     | 100(× 24)            | 1.1                    | 2,640            |
| Case118| M1     |                  |                   |                      |                        |                  |
|        | M0     | 20,000           | 2 × 100(× 24)     | 100(× 24)            | 1.1                    | 2,640            |

Note: "/" represents that the current required computing resources exceed the hardware support and cannot be calculated.

Table 2. Comparison of prediction accuracy between full-by-period learning and period-period learning strategy under Case30 (the number of training samples is fixed at 30,000)

| Method | Training samples | Network structure | Probability of absolute error > 5 MW | The number of epochs | Time for each epoch (s) | Training time (s) |
|--------|------------------|-------------------|--------------------------------------|----------------------|------------------------|------------------|
| M1     | 30,000           | 3 × 500           | 12.8%                                | 500                  | 11.4                   | 5,700            |
| M0     | 30,000           | 2 × 100(× 24)     | <1%                                  | 50(× 24)             | 2.6                    | 3,120            |

4.3. Validity comparison of solving SCED model with different method

This section will verify the validity of the proposed method in solving the SCED model by comparing M2 and M0.

Table 3 shows the proportion of iterations required for M2 and M0 to solve the SCED model. As can be seen from the table, in the SCED solution test of case 118, the industrial method requires at least 3 iterations to converge, and in most cases 4-5 iterations are required; while the proposed method only requires convergence in one iteration, two iterations are required in rare cases (0.9%). In case 30, the industrial method needs at least 4 iteration to converge, and in most cases 5-6 iterations are required, in some extreme cases (0.2%), it takes 7 iterations to converge; However, the proposed method can still be solved in at most two iterations.

Table 4 show that comparison of M2 and M0 in SCED solving speed. In case 30, the average calculation time of the industrial method is 9.55s in 1,000 test samples, and the proposed method is only 2.34s. Therefore, the average speedup ratio of the method proposed in this article is 4.08 compared to the industrial method, among them, the maximum speedup ratio reached 7.33; the situation is similar in case 118, the average calculation speed of the method proposed in this paper is 4.91 times that of the industrial method, and the fastest can reach 5.74 times of the industrial method.

In conclusion, the method proposed in this paper can effectively reduce the number of iterations of the SCED model, thereby speeding up the solution of the SCED model.
Table 4. The calculation time of the industrial iterative solution method and the proposed method to solve the SCED model

| Case   | Method | Average calculation time (s) | Average speedup | Maximum speedup |
|--------|--------|------------------------------|-----------------|-----------------|
| Case 30| M2     | 9.55                         | 4.08            | 7.33            |
|        | M0     | 2.34                         |                 |                 |
| Case 118| M2    | 42.50                        | 4.91            | 5.74            |
|        | M0     | 8.66                         |                 |                 |

Note: “-” represents that the number of iterations did not appear

5. Conclusion

This paper proposes a fast solution method for multi-period SCED model. 1) This method designs an intelligent calculation framework of SCED, the framework embeds the deep learning method into the existing iterative solution method, and pre-identifies the active constraints through the deep neural network. Without affecting the accuracy of the iterative solution method, the number of iterations is greatly reduced. 2) This method also proposes a lightweight learning strategy which redesigns and decomposes the feature vector and the network structure of the deep neural network to reduce the training cost of DNN, thereby improving the training efficiency and accuracy of DNN. 3) Simulation experiments on IEEE 30-bus and IEEE 118-bus systems show that the proposed lightweight learning strategy can effectively reduce the training costs and computational burden of DNN. The proposed method can quickly solve the multi period SCED model in at most two iterations, and the average solution speed is about 4 and 5 times faster than that of the iterative method in case 30 and case 118, respectively.

Acknowledgments

The authors are grateful for the support from Science and Technology Project of State Grid Corporation of China (5100-201999333A-0-0-00).

References

[1] Z. Yang, A. Bose, H. Zhong, N. Zhang, J. Lin, Q. Xia, and C. Kang. (2017) LMP revisited: A linear model for the loss-embedded LMP. IEEE Trans. Power Syst., vol. 32, no. 5, pp. 4080–4090.
[2] Y. Liu, M. C. Ferris and F. Zhao. (2015) Computational study of security constrained economic dispatch with multi-stage rescheduling. IEEE Trans. Power Syst., vol. 30, no. 2, pp. 920-929.
[3] F. Capitanescu, M. Glavic, D. Ernst, and L. Wehenkel. (2007) Contingency filtering techniques for preventive security-constrained optimal power flow. IEEE Trans. Power Syst., vol. 22, no. 4, pp. 1690–1697.
[4] R. Madani, J. Lacaie, and Ross Baldick. (2017) Constraint screening for security analysis of power networks. IEEE Trans. Power Syst., vol. 32, no. 2, pp. 1828-1838.
[5] Y. Yang, X. Duan, Q. Zhai. (2017) Fast grid security assessment with N-k contingencies. IEEE Trans. Power Syst., vol. 32, no. 3, pp. 2193-2203.
[6] A. J. Ardakani and F. Bouffard. (2013) Identification of umbrella constraints in DC-based security-constrained optimal power flow. IEEE Trans. Power Syst., vol. 28, no. 4, pp. 3924–3934.
[7] A. Santos Xavier, F. Qiu, F. Wang and P. R. Thimmapuram. (2019) Transmission constraint filtering in large-scale security-constrained unit commitment. IEEE Trans. Power Syst., vol. 34, no. 3, pp. 2457–2460.
[8] Y. Ng, S. Misra, L. Roald, S. Backhaus. (2018) Statistical learning for DC optimal power flow.
IEEE Power Syst. Comput. Conf. (PSCC), Dublin, Ireland.

[9] U. Shaham, A. Cloninger, R.R. Ronald. (2018) Provable approximation properties for deep neural networks. Appl. & Comput. Harmonic Analysis, vol. 44, no. 3, pp. 537-557.

[10] M. Dong and L. S. Grumbach. (2019) A Hybrid Distribution Feeder Long-Term Load Forecasting Method Based on Sequence Prediction. in IEEE Trans. Smart Grid. Doi: 10.1109/TSG.2019.2924183.

[11] K. L. López, C. Gagné, M. Gardener. (2019) Demand-side management using deep learning for smart charging of electric vehicles. IEEE Trans. Smart Grid., vol. 10, no. 3, pp. 3265–3275.

[12] Y. Yang, Z. Yang, J. Yu, K. Xie and L. Jin. (2020) Fast Economic Dispatch in Smart Grids Using Deep Learning: An Active Constraint Screening Approach. IEEE Internet of Things Journal, doi: 10.1109/JIOT.2020.2993567.

[13] Kingma, Diederik & Ba, Jimmy. (2014) Adam: A Method for Stochastic Optimization. International Conference on Learning Representations.

[14] Power Systems Test Case Archive. [Online]. Available: http://www.ee.washington.edu/research/pstca/pf118/pg_tca118bus.htm.