

Topology-aware Learning Assisted Branch and Ramp Constraints Screening for Dynamic Economic Dispatch

Fouad Hasan, *Student Member, IEEE*, Amin Kargarian, *Senior Member, IEEE*

Abstract— Multi-interval or dynamic economic dispatch (D-ED) is the core of various power system management functions. This optimization problem contains many constraints, a small subset of which is sufficient to enclose the D-ED feasible region. This paper presents a topology-aware learning-aided iterative constraint screening algorithm to identify a feasibility outlining subset of network and generating units ramp up/down constraints and create a truncated D-ED problem. We create a colorful image from nodal demand, thermal unit generation cost, and network topology information. Convolutional neural networks are trained for constraint status identification using colorful images corresponding to system operating conditions and transfer learning. Filtering inactive line flow and ramp up/down constraints reduces the optimization problem’s size and computational burden, resulting in a reduction in solution time and memory usage. Dropping all inactive branch and ramp constraints may activate some of these originally inactive constraints upon solving the truncated D-ED. A loop is added to form a constraints coefficient matrix iteratively during training dataset preparation and algorithm utilization. This iterative loop guarantees truncated D-ED results feasibility and optimality. Numerical results show the proposed algorithm’s effectiveness in constraint status prediction and reducing D-ED size and solution time.

Keywords— Dynamic economic dispatch, branch and ramp constraints, topology change, machine learning, constraint classification.

I. INTRODUCTION

MULTI-TIME interval economic dispatch, also known as dynamic economic dispatch (D-ED), is solved daily and hourly for many energy management functions in power systems. D-ED has many constraints, including transmission network constraints and generating unit ramp limitations [1]. The curse of dimensionality and computational cost increase with the system size and scheduling time intervals. Despite improvements in solvers’ performance, processing technology, and computing memory, D-ED’s solution time and resource prerequisites continue to be crucial factors.

A. Background

Various approaches have been developed to reduce D-ED computational resource and time requirements and other energy

management functions, such as optimal power flow and unit commitment [2]. For instance, decomposition approaches are proposed to decompose economic dispatch into small, low computational subproblems and coordinate them using iterative approaches [3]. Another promising strategy is to identify and remove inactive and redundant constraints. Experimental and mathematical analyses show that most D-ED constraints are inactive or redundant, and only a small subset of constraints is needed to create the feasible region. Omitting inactive and redundant constraints relieves D-ED computational burden significantly.

Several studies are conducted for active and inactive constraints identification (inequality constraints that are satisfied with equality are called active or binding constraints). These studies focus only on thermal capacity limits of transmission lines, i.e., branch constraints. Reference [4] shows that more than 85% of branch constraints are inactive in unit commitment. The concept of umbrella constraints is presented in [5] to determine redundant constraints and provide the necessary and sufficient conditions to enclose the optimal power flow feasible space. Reference [6] presents an optimization-based bound tightening approach that solves multiple linear subproblems in parallel to identify redundant constraints. This study shows that roughly 99% of constraints are redundant for real-world systems. Reference [7] presents an iterative algorithm for unit commitment. All branch constraints are omitted at the first iteration and violated constraints are added in the optimization iteratively. It is reported that it is unnecessary to add all violated branch constraints to the original relaxed problem.

B. Motivation

Current mathematical model-based approaches either solve a series of optimization subproblems to find inactive constraints or perform a form of iterative constraint generation (ICG) where in each iteration, active line flow constraints identified in previous iterations are added to models [8]. These two approaches lead to computational time-saving. However, as the system size grows, this time-saving may fade away as i) solving subproblems to find inactive constraints might take more time, and ii) the number of ICG iterations increases. Moreover, the existing constraint screening methods only consider branch constraints. Intertemporal constraints, such as generators ramp up and down rates limiting power produced by thermal units in successive time intervals, increase D-ED memory usage and computational cost and constitute a considerable portion of

This work was supported by the National Science Foundation under Grant ECCS-1944752.

The authors are with the Electrical and Computer Engineering Department, Louisiana State University, Baton Rouge, LA 70803, (e-mail: fhasan1@lsu.edu, kargarian@lsu.edu).

inactive constraints. A simple solution may be relaxing ramp and branch constraints and applying the ICG technique. However, our observation shows that iteratively adding ramp constraints increases the number of ICG iterations considerably.

C. Contribution

This paper presents a topology-aware learning-aided iterative algorithm to predict the necessary and sufficient branch and ramp up/down constraints information needed for forming the D-ED problem's feasible design space under topology alteration. The goal is to formulate a truncated D-ED with less computational cost than the original optimization problem in terms of solution time and memory usage. Demand, thermal unit generation cost, and network topology information are three inputs to the proposed algorithm. The system admittance matrix is used as an input feature to account for topology alterations. Using these three features, each operating condition scenario is transformed into a colorful image whose red, green, and blues channels include, respectively, demand, thermal unit generation cost, and admittance matrix information. Pre-trained convolutional neural networks are adopted, and transfer learning is used to adjust them for power system constraints classification. The input to these classification learners is colorful images corresponding to system operating conditions, and the learners predict active and pseudo-active branch and ramp constraints. Pseudo-active constraints are inequalities that are not active at the optimal point but are required to ensure the truncated D-ED results optimality and feasibility. An iterative loop is added to the proposed algorithm to find pseudo-active constraints for the learners' training phase and ensure the feasibility of truncated D-ED results if misclassifications are observed in predicting active and pseudo-active branch and ramp constraints. Numerical results show the proposed algorithm's effectiveness in reducing the size and solution time of D-ED. We have posted our code on GitHub [9].

The main contributions of this paper are summarized as follows:

- A combined learning and model-based algorithm is developed to identify the status of branch and thermal unit ramp up/down constraints and formulate a reduced-size dynamic economic dispatch problem with respect to network topology, demand, and thermal unit generation cost information.
- Power system operating conditions are transformed into colorful images enabling users to take advantage of well-advanced computer vision learning techniques, such as EfficientNet-B7, with the help of transfer learning.
- An iterative loop is embedded to capture active (necessary information) and pseudo-active (sufficient information) constraints for training classifiers and ensuring the feasibility of truncated economic dispatch results.

D. Paper organization

The remainder of the paper is organized as follows. Relevant papers are reviewed in Section II. The problem formulation is given in Section III. The proposed algorithm is

presented in Section IV. Results are discussed in Section V, and concluding remarks are provided in Section VI.

II. REVIEW OF MACHINE LEARNING APPLICATIONS TO POWER SYSTEM OPTIMIZATION

Machine learning is a promising tool to provide a cost-effective solution to power system optimization problems and reduce solvers' computational burden. We review learning-based approaches to reduce the computational burden of power system optimization problems, e.g., optimal power flow (OPF), economic dispatch, and unit commitment. The existing approaches can be categorized into i) warm start prediction, ii) generator setpoint prediction (or black-boxing), and iii) hybrid learning-model-based approaches.

The benefit of a learning-based warm start to solve AC OPF is discussed in [10]. A learner's prediction is a warm start for the optimization solver. However, this approach does not reduce the OPF size as the complete set of constraints is still used, which may not yield considerable speed improvement.

Feasibility enforced deep learning is presented in [11-13] to predict generator voltage and power setpoints from demand. Lagrangian duals are combined with deep learning to enforce constraint satisfaction. In [14], deep neural networks are used to predict active power and bus voltages. A penalty function is used to ensure the feasibility of operational constraints. References [15-17] use black-box strategies to predict generation setpoints directly. However, learning a continuous-valued multi-dimensional variable (e.g., generation setpoints) is a demanding machine learning task. Moreover, a slight mismatch between predicted and actual generation setpoints may yield suboptimality or power balance equations infeasibility. This makes system operators reluctant to deploy black-boxing learning-based approaches. Recent studies combine learning and power system models to develop hybrid learning and physics-based approaches [18]. One of these approaches uses machine learning to predict inactive/redundant optimization constraints instead of directly predicting the optimal output. Reference [19] presents experimentation on predicting transmission constraints, warm start, and affine subspace to improve mixed-integer solvers' computational performance for unit commitment. A data-driven method is developed in [20] to identify inactive and redundant constraints for single period unit commitment. A statistical learning-based ensemble control policy is presented in [21] to track real-time DC OPF. In [22], a simple neural network classifier is trained to predict DC OPF active line flow constraints. In [23], an umbrella constraint prediction algorithm is developed instead of predicting binding constraints. A constraint is an umbrella constraint if its removal changes feasible solutions of the original optimization problem. A learning-based approach is presented in [24] to classify zero probability events, which are inactive constraints, and reduce joint chance constraints' computational burden for solving OPF. In [25], a two-step prescreening approach is presented to identify and remove non-dominating constraints.

These hybrid constraint classification approaches do not consider thermal unit generation cost and power network

topology alterations. Also, only spatial line flow constraints are considered, not intertemporal generating unit ramp constraints whose contribution to computational burden cannot be ignored. We aim to address these knowledge gaps in this paper.

III. DYNAMIC ECONOMIC DISPATCH

The considered problem is a multi-interval economic dispatch with generating unit ramp up and down constraints. The objective function is to minimize generation costs subject to power balance (1b), generation limits (1c), generating units ramp up and down limitations (1d) and (1e) denoted by $\mathcal{h}_{RU}(x)$ and $\mathcal{h}_{RD}(x)$, and transmission line flow limits (1f) denoted by $\mathcal{h}_L(x)$.

$$\min_p \sum_t \sum_u \gamma_{ut} p_{ut} \quad (1a)$$

s.t.

$$\sum_u p_{ut} = \sum_n d_{nt} \quad \forall t \quad (1b)$$

$$P_u^{\min} \leq p_{ut} \leq P_u^{\max} \quad \forall u, \forall t \quad (1c)$$

$$\mathcal{h}_{RU}(x): p_{ut} - p_{ut-1} \leq \mathcal{R}\mathcal{U}_u \quad \forall u, \forall t \quad (1d)$$

$$\mathcal{h}_{RD}(x): p_{ut-1} - p_{ut} \leq \mathcal{R}\mathcal{D}_u \quad \forall u, \forall t \quad (1e)$$

$$\mathcal{h}_L(x): -P_l^{\max} \leq \mathbf{SF}_l(\mathbf{p}_t^{\text{inj}} - \mathbf{D}_t) \leq P_l^{\max} \quad \forall l, \forall t \quad (1f)$$

where t , u , n , and l are indices for time, units, buses, and lines, respectively. Variable p_{ut} is power produced by unit u at time t . Parameter γ_{ut} denotes generation cost. Parameters $\mathcal{R}\mathcal{U}_u$ and $\mathcal{R}\mathcal{D}_u$ are ramp up and down limits of generating unit u . \mathbf{SF} is the generation shift factor matrix. Parameter d_{nt} is demand at bus n at time t . Bus injection and demand matrices are $\mathbf{p}_t^{\text{inj}}$ and \mathbf{D}_t .

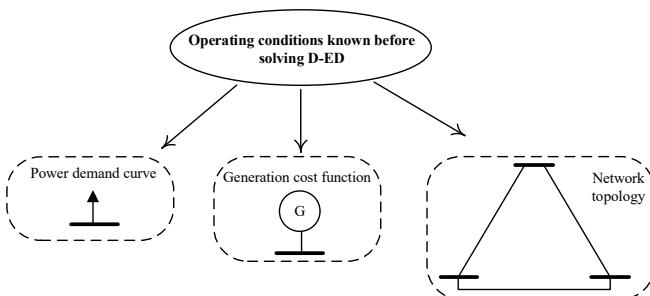


Fig. 1. Operating conditions known before solving the D-ED problem.

IV. PROPOSED LEARNING AIDED ITERATIVE APPROACH

The proposed learning-aided truncated economic dispatch approach is presented in this section.

A. Truncated D-ED

Branch flow and generating unit ramp up and down constraints constitute a large portion of the D-ED constraint set. Branch flows make constraints geographically dependent, and ramp up/down limitations introduce intertemporal dependency. These constraints, particularly branch constraints, contribute significantly to computational burden and memory usage, yet

most have no impact on the optimization feasible design space. The status of these constraints depends on system operating conditions shown in Fig. 1, including power demand, thermal unit generation cost, and network topology that are known before solving D-ED. The demand varies more significantly than the other two features. However, depending on thermal unit generation costs and network topology, active constraints may differ for a given demand value. Also, while demand and electricity market prices are correlated, demand might not be correlated to generation costs. Thus, these three features are selected to identify the status constraints.

Figures 2a and 2b show, respectively, an overview of the proposed constraint classification training algorithm and its utilization. The objective of the proposed algorithm is to reformulate (1) by the following truncated D-ED.

$$\min_p \sum_t \sum_u \gamma_{ut} p_{ut} \quad (2a)$$

s.t.

$$(1b) \& (1c) \quad (2b)$$

$$\tilde{\mathcal{A}}(\mathcal{h}_{RU}(x)) \leq 0 \quad (2c)$$

$$\tilde{\mathcal{A}}(\mathcal{h}_{RD}(x)) \leq 0 \quad (2d)$$

$$\tilde{\mathcal{A}}(\mathcal{h}_L(x)) \leq 0 \quad (2e)$$

where $\tilde{\mathcal{A}}(\mathcal{h}_{RU}(x))$, $\tilde{\mathcal{A}}(\mathcal{h}_{RD}(x))$, and $\tilde{\mathcal{A}}(\mathcal{h}_L(x))$ denote sets of ramp up, ramp down, and line flow constraints required to ensure that the truncated optimization problem (2) is equivalent to (1). The proposed learning-aided algorithm is designed based on the following two remarks.

Remark 1: For a given network topology, demand, and generation cost scenario, if constraints (1d) – (1f) are satisfied with equality, they must be included in the optimization problem.

$$p_{u,t} - p_{u,t-1} = \mathcal{R}\mathcal{U}_u \quad \forall u, t \in \Omega_{RU}^t \quad (3)$$

$$p_{u,t-1} - p_{u,t} = \mathcal{R}\mathcal{D}_u \quad \forall u, t \in \Omega_{RD}^t \quad (4)$$

$$|\mathbf{SF}_l(\mathbf{p}_t^{\text{inj}} - \mathbf{D}_t)| = P_l^{\max} \quad \forall l, t \in \Omega_L^t \quad (5)$$

where Ω_{RU}^t denote the set of generators with active ramp up limits at time t , Ω_{RD}^t indicates the set of generators with active ramp down at time t , and Ω_L^t is the set of active line constraints at time t .

Remark 2: If generation cost coefficients of several thermal units are the same, multiple non-unique optimal solutions with different generation schedules but the same objective value might exist. Thus, dropping all inactive line flow, ramp up, and ramp down constraints and keeping only active constraints obtained at an optimal solution and resolving the truncated problem may trigger several originally inactive constraints to be activated. Removing these particular non-binding constraints, named *pseudo-active constraints* in this paper, may move the optimal solution to a point with the same objective value as the original problem but with a different generation schedule that is infeasible from the original problem's perspective. Thus, (3)–(5) are necessary but not sufficient to form $\tilde{\mathcal{A}}(\mathcal{h}_{RU}(x))$, $\tilde{\mathcal{A}}(\mathcal{h}_{RD}(x))$, and $\tilde{\mathcal{A}}(\mathcal{h}_L(x))$.

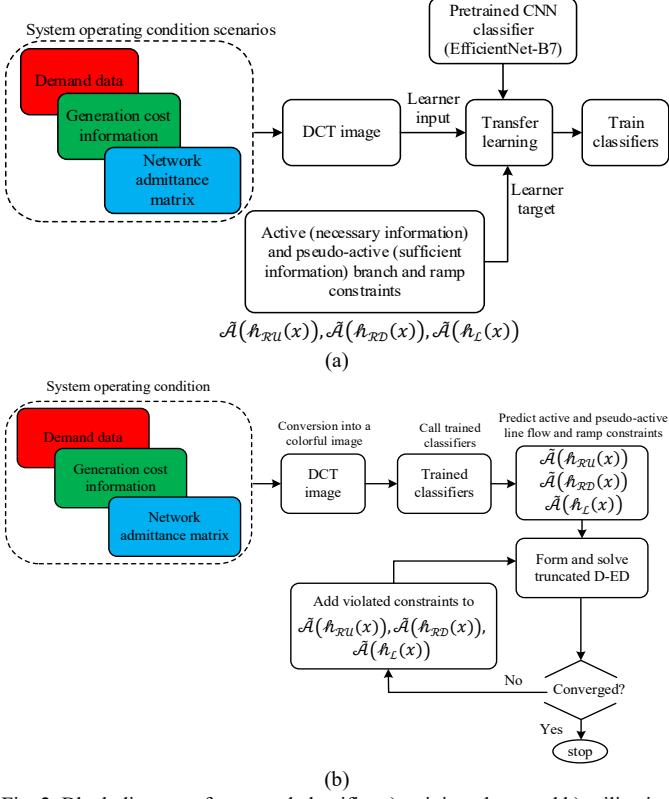


Fig. 2. Block diagram of proposed classifier a) training phase and b) utilization with an embedded loop.

Remark 2 is a critical observation. It indicates that for a constraint filtering algorithm to be effective, not only active line flow and ramp up/down constraints must be identified but also several other inactive constraints may need to be included in the D-ED problem. If only active constraints are used to formulate a truncated optimization problem, the resultant optimal generation schedule may differ from the original optimal schedule. Although the objective function value does not change as generation exchange occurs among generators with the same cost coefficient, the new generation setting may activate several pseudo-active constraints and thus yields infeasibility from the original D-ED's perspective even if there are a few pseudo-active constraints. Active constraints are sufficient to formulate a truncated problem if no generators have identical cost coefficients.

Quasi-active constraints, which appear due to integer variables, are introduced in [20]. Although the concept of quasi-active constraints (i.e., not active in the optimal point but if removed change the optimal point) is similar to what we call pseudo-active constraints, pseudo-active constraints appear mainly because of non-uniqueness of thermal unit generation cost functions. Thus, we have used the term pseudo-active to avoid confusion with the term quasi-active constraints in [20].

B. Illustrative Examples

Example 1: Consider a simple economic dispatch problem (6) with three generators and a 2-hour horizon. The objective is to minimize generation costs with respect to power balance,

generator limits, and ramp up/down constraints (branch flow constraints are ignored for simplicity).

$$\min_{p_{u,t}} 10p_{1,1} + 10p_{2,1} + 15p_{3,1} + 10p_{1,2} + 10p_{2,2} + 15p_{3,2} \quad (6a)$$

s.t.

$$0 \leq p_{1,1}, p_{1,2}, p_{2,1}, p_{2,2}, p_{3,1}, p_{3,2} \leq 200 \quad (6b)$$

$$p_{1,1} + p_{2,1} + p_{3,1} = 100 \quad (6c)$$

$$p_{1,2} + p_{2,2} + p_{3,2} = 200 \quad (6d)$$

$$p_{1,2} - p_{1,1} \leq 50 \quad (6e)$$

$$p_{1,1} - p_{1,2} \leq 50 \quad (6f)$$

$$p_{2,2} - p_{2,1} \leq 60 \quad (6g)$$

$$p_{2,1} - p_{2,2} \leq 60 \quad (6h)$$

$$p_{3,2} - p_{3,1} \leq 100 \quad (6i)$$

$$p_{3,1} - p_{3,2} \leq 100 \quad (6j)$$

This problem has multiple solutions as two generators have the same cost. An optimal solution is $p_{1,1} = 100$, $p_{2,1} = 0$, $p_{3,1} = 0$ for hour 1 and $p_{1,2} = 150$, $p_{2,2} = 50$, $p_{3,2} = 0$ for hour 2 with an objective value of \$3,000. The only active inequality constraint is (6e). If we remove all inactive inequality constraints (6f) – (6j), the solver may provide $p_{1,1} = 0$, $p_{2,1} = 100$, $p_{3,1} = 0$, $p_{1,2} = 0$, $p_{2,2} = 200$, and $p_{3,2} = 0$ with the objective value of \$3,000 as a solution of truncated problem (6a) – (6e). This solution is not feasible from the original problem perspective as constraint (6g), called pseudo-active constraint, is violated. This mathematical example shows the necessity of pseudo-active constraints to recover the optimal solution if generator cost functions are not unique.

Example 2: A system with ten interconnected areas representing the IEEE 118-bus system is used to illustrate remark 2. The original D-ED is solved with all branch and ramp constraints. The number of active line flow, ramp up, and ramp down constraints are 279, 365, and 330, respectively. A truncated D-ED is formed using these active constraints and dropping other inactive constraints. The truncated D-ED is solved, and the number of newly activated line and ramp up/down constraints is 63, 155, and 134. Three hundred fifty-two newly activated (pseudo-active) constraints should also be included in D-ED. We carry out this iterative procedure. The iterative loop converges after 33 iterations. Figure 3 shows the number of accumulated constraints (active plus pseudo-active) over iterations. The number of newly activated constraints usually reduces as more iterations are carried out. Also, multiple pseudo-active constraints are identified at each iteration. The total number of active (necessary) and pseudo-active (sufficient) line flow, ramp up, and ramp down constraints required to enclose the feasible set of D-ED is 518, 1024, and 941 obtained upon the loop convergence.

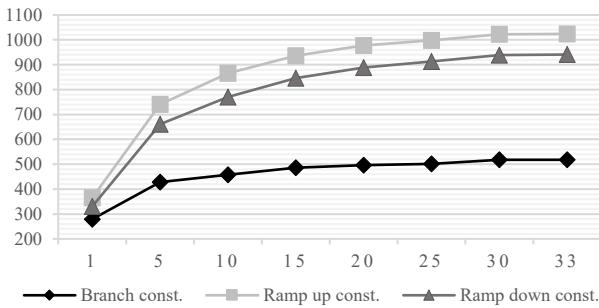


Fig. 3. Number of accumulated active constraints over iterations.

Example 3: Consider the 118-bus system with the baseload and base topology. Generators 4, 5, 10, 29, 36, 43, 44, and 45 have the same production cost coefficient. The D-ED subproblem is formed using only active constraints, with no iterative loop. Table I shows power generation values. While the summation of power produced by these units and operational costs are the same for the original and truncated D-ED models, their generation schedules are different. This generation difference results in the violation of some originally inactive constraints that are not included in the truncated D-ED and makes its solution infeasible. For instance, the ramp-up constraint of unit 5 for the transition from interval 1 to interval 2, which is inactive in the original SCED, is activated in the truncated D-ED.

TABLE I
POWER GENERATION VALUES IN MW

Gen. no.	4	5	10	29	36	40	43	44	45
original D-ED	150	300	300	256	150	50	100	100	100
Truncated D-ED	226	100	100	80	300	20	100	300	100

C. Proposed Training Dataset Generation Algorithm

We present the iterative algorithm shown in the following pseudocode to capture necessary (active constraints) and sufficient (pseudo-active constraints) information of $\tilde{\mathcal{A}}(\mathcal{h}_{RU}(x))$, $\tilde{\mathcal{A}}(\mathcal{h}_{RD}(x))$, and $\tilde{\mathcal{A}}(\mathcal{h}_L(x))$ for forming the truncated D-ED. Our experimental results on many cases show that when merely active constraints from the original D-ED solution are used to form the truncated problem, multiple iterations are required to find pseudo-active constraints. Our goal is to minimize the number of iterations using machine learning. We use information in the last iteration of Algorithm I to train constraint classifiers.

Algorithm I Pseudocode to capture necessary and sufficient information

1. For each operating condition scenario, set iteration index $k = 1$ and $flag^k = \emptyset$
2. Set $\mathcal{A}^k(\mathcal{h}_L(x)) = \mathcal{A}^k(\mathcal{h}_{RU}(x)) = \mathcal{A}^k(\mathcal{h}_{RD}(x)) = \emptyset$
3. Solve D-ED problem (1)
4. Identify active constraints $\mathcal{A}^k(\mathcal{h}_L(x))$, $\mathcal{A}^k(\mathcal{h}_{RU}(x))$ and $\mathcal{A}^k(\mathcal{h}_{RD}(x))$
5. **if** $\mathcal{A}^k(\mathcal{h}_L(x)) = \mathcal{A}^k(\mathcal{h}_{RU}(x)) = \mathcal{A}^k(\mathcal{h}_{RD}(x)) = \emptyset$
6. $flag^k \leftarrow \emptyset$

```

7. else
8.    $flag^k \leftarrow 1$ 
9. end if
10. Drop (1d) – (1f) from (1) and form a D-ED subproblem
11. while  $flag^k \neq \emptyset$ 
12.    $\tilde{\mathcal{A}}^k(\mathcal{h}_L(x)) = \cup_1^k \mathcal{A}^k(\mathcal{h}_L(x))$ 
13.    $\tilde{\mathcal{A}}^k(\mathcal{h}_{RU}(x)) = \cup_1^k \mathcal{A}^k(\mathcal{h}_{RU}(x))$ 
14.    $\tilde{\mathcal{A}}^k(\mathcal{h}_{RD}(x)) = \cup_1^k \mathcal{A}^k(\mathcal{h}_{RD}(x))$ 
15.   Add  $\tilde{\mathcal{A}}^k(\mathcal{h}_L(x))$ ,  $\tilde{\mathcal{A}}^k(\mathcal{h}_{RU}(x))$ , and  $\tilde{\mathcal{A}}^k(\mathcal{h}_{RD}(x))$  to
        the D-ED subproblem and solve it
16.    $k = k + 1$ 
17.   Identify pseudo-active constraints  $\mathcal{A}^k(\mathcal{h}_L(x))$ ,
         $\mathcal{A}^k(\mathcal{h}_{RU}(x))$  and  $\mathcal{A}^k(\mathcal{h}_{RD}(x))$ 
18.   If  $\mathcal{A}^k(\mathcal{h}_L(x)) = \mathcal{A}^k(\mathcal{h}_{RU}(x)) = \mathcal{A}^k(\mathcal{h}_{RD}(x)) = \emptyset$ 
19.      $flag^k \leftarrow \emptyset$ 
20.   else
21.      $flag^k \leftarrow 1$ 
22.   end if
23. end while
24. Store  $\tilde{\mathcal{A}}^k(\mathcal{h}_L(x))$ ,  $\tilde{\mathcal{A}}^k(\mathcal{h}_{RU}(x))$ , and  $\tilde{\mathcal{A}}^k(\mathcal{h}_{RD}(x))$ 

```

Algorithm I is carried out for every operating condition scenario and accumulated active plus pseudo-active constraints obtained from iteration one to the last iteration are labeled. For a given network topology, demand and generation cost scenarios are generated using the following equations.

$$d_{nt}^m = \omega \times [d_{base,n}(1 - \Delta_d^L) + \eta_{p,n} \times (\Delta_d^U - \Delta_d^L)] \quad (7)$$

$$\gamma_{nt} = [\gamma_{base,n}(1 - \Delta_\gamma^L) + \eta_{a,n} \times (\Delta_\gamma^U - \Delta_\gamma^L)] \quad (8)$$

Parameters Δ_d^L and Δ_d^U are the upper and lower bounds of demand variation for each load curve, and Δ_a^U and Δ_a^L are upper and lower bounds of generation cost variation. Random parameter $\eta_{a,n}$ follows a uniform distribution between 0 and 1. It adds randomness to the base case generation cost γ_{base} at bus n . Two randomness factors are introduced for demand to capture various plausible operating conditions. Random parameter ω shifts the load curve to model its daily and seasonal variation. It can be varied in a range using historical data and load growth predictions or such that any further increment/decrement makes economic dispatch infeasible. Random parameter $\eta_{p,n}$, which follows a uniform distribution between 0 and 1, models the uncertain geographic load distribution. Parameter $\eta_{p,n}$ is generated for every load point, allowing load at different buses to fluctuate independently. Consider the base case load curve on the left side of Fig. 4. It would become similar to right side curves after adding load curve shifting randomness ω and nodal load distribution randomness $\eta_{p,n}$.

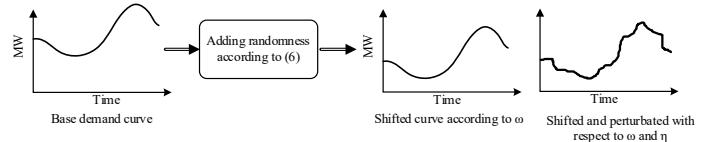


Fig 4: Demand scenario generation.

A major bottleneck of existing learning-based constraint classification approaches is that learners are trained for a fixed topology. But in real-word, the network topology changes frequently. A new learner would be required every time the network topology alters. We use the admittance matrix information to address this problem. A topology alteration changes some buses' self-impedance that can be detected by observing diagonal elements of the admittance matrix. We use (7) and (8) to generate a set of demand and generation cost scenarios for every topology configuration. This leads to the following operating condition matrix (\mathcal{OC}_t) at time period t . The first and second columns of \mathcal{OC}_t contain nodal demand and generation costs, and its third column is the admittance matrix diagonal elements.

$$\mathcal{OC}_t = \begin{bmatrix} d_{1t} & \gamma_{1t} & Y_{11,t} \\ d_{2t} & \gamma_{2t} & Y_{22,t} \\ \vdots & \vdots & \vdots \\ d_{nt} & \gamma_{nt} & Y_{nn,t} \end{bmatrix} \quad (9)$$

where n is the number of buses. An operating condition scenario is formed by combining \mathcal{OC}_t for all scheduling periods $t = 1, \dots, T$. The D-ED problem (1) is solved for each operating condition scenario. Inequalities (1d)–(1f) are dropped from (1), and a D-ED subproblem is formed using active generator ramping limitations and line flow constraints (3)–(5). This D-ED subproblem is solved. If new active ramp up/down or line flow constraints are observed, $flag^k \leftarrow 1$, and an iterative loop is started. At each iteration k , all active and pseudo-active constraints from iteration 1 to $k-1$ are added to the D-ED subproblem. The loop is carried out, and active and pseudo-active constraints are accumulated. If no newly activated constraint is detected and $flag^k = \emptyset$, the accumulated constraints $\tilde{\mathcal{A}}^k(\mathcal{h}_L(x))$, $\tilde{\mathcal{A}}^k(\mathcal{h}_{RU}(x))$, and $\tilde{\mathcal{A}}^k(\mathcal{h}_{RD}(x))$ are stored for training classification learners.

D. Operating Condition Conversion into DCT Colorful Image

Knowing that i) a node/line in a power system interacts with its neighboring nodes and transmission lines and is loosely coupled with distant nodes and lines [26] and ii) a pixel of an image is highly correlated to its neighboring pixels, we obtain the intuition to convert the constraint screening classification into a computer vision type problem. We use a 3-D tensor to convert a power system operating condition scenario into a colorful image. The matrices of this tensor corresponding to red, green, and blue color channels contain, respectively, demand, thermal units' generation cost, and network topology information. We call this image a DCT image (D: demand, C: cost, and T: topology).

Demand and generation cost terms are extended to every bus to have the same sized matrices. The demand/generation cost input for every time period is a vector with n elements that are set to zero for buses with no generators or no load. Hence, demand matrix \mathcal{D} and cost coefficient matrix Γ are $n \times T$, where T is the considered scheduling horizon.

$$\mathcal{D} = \begin{bmatrix} d_{11} & \dots & d_{1T} \\ \vdots & \ddots & \vdots \\ d_{n1} & \dots & d_{nT} \end{bmatrix} \quad (10)$$

$$\Gamma = \begin{bmatrix} \gamma_{11} & \dots & \gamma_{1T} \\ \vdots & \ddots & \vdots \\ \gamma_{n1} & \dots & \gamma_{nT} \end{bmatrix} \quad (11)$$

We use diagonal elements of the admittance matrix at every time period and form the following $n \times T$ matrix.

$$Y = \begin{bmatrix} Y_{11,1} & \dots & Y_{11,T} \\ \vdots & \ddots & \vdots \\ Y_{nn,1} & \dots & Y_{nn,T} \end{bmatrix} \quad (12)$$

Consider an operating condition for the IEEE 118-bus system with a scheduling horizon of 24 periods. The red channel of the DCT image, i.e., \mathcal{D} , is a 118×24 matrix with each column having 91 nonzero elements and 27 zeros. The green channel matrix Γ has 54 nonzero elements and 64 zeros in each column. As all diagonal elements of the Y-bus matrix are nonzero, all elements of blue channel Y , whose size is 118×24 , are nonzero. A DCT image for an operating condition scenario is shown in Fig. 5.

The locations of zeros added to \mathcal{D} and Γ matrices are fixed. This zero padding does not affect the training time and learners' performance. As explained in the next section, the output after convolution and pooling operations of the zero-padded pixels is zero. As a result, neuron weights corresponding to zero pixels are not updated after backpropagation.

Any alteration in one or a combination of demand, cost, and topology related tensors changes the DCT image. A well-trained convolutional neural network (CNN) can capture even slight pattern changes in an image. Furthermore, system features and active constraints do not vary drastically after, for instance, a topology alteration. Thus, features learned by CNN before and after outage of a line l can help the learner predict constraints status if a line near line l is out.

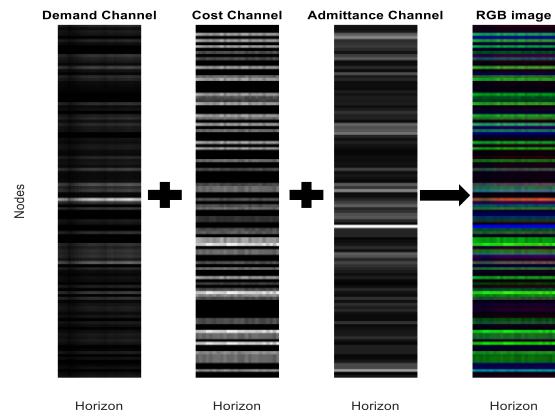


Fig. 5. Red, green, and blue channels of DCT image corresponding to an operating condition scenario for the 118-bus system.

E. Learning Strategy

CNN has shown promising performance in image analysis and computer vision problems. It consistently performs better

and has become the state-of-the-art image classification, object detection, and segmentation technique. We have tailored the considered constraint classification problem as a computer vision problem and have selected CNN to tackle it.

CNN Classifier: CNN extracts meaningful local features through repeated convolution/pooling operations. CNN exploits the shift invariance, local connectivity, and compositionality. By carefully organizing the input shape of CNN, it is possible to exploit the power system temporal and geographical dependency information. The weights of learnable neurons are updated by interacting with three dimensions (i.e., demand, cost, and admittance) of a DCT image. Every neuron in a layer is correlated to a small region of the preceding layer instead of all neurons. Batches of images of a particular shape are feed to CNN to extract feature vectors through convolution. After every convolution, the number of extracted features depends on the number of filters. The trainable layer parameters are optimized using a loss function.

Learner Architecture: Instead of training a CNN learner from scratch, we have selected EfficientNet-B7, a CNN-based pre-trained model developed by the Google Brain Team. EfficientNet-B7 is one of the latest state-of-the-art developments in the image classification domain [27]. EfficientNet-B7 attains 84.3% top-1 and 97.1% top-5 accuracy with 66M parameters and 37B FLOPS (floating point operations per second), whereas the earlier best GPipe achieves similar performance with 557M parameters while being 8.4 times larger than EfficientNet-B7 [27].

Hyperparameters play a significant role in CNN efficiency and accuracy. Effective scaling/hyperparameter tuning is still an open question [27]. EfficientNet-B7 tuning follows a compound scaling method. It provides a compound coefficient to uniformly scale network width (number of channels), depth (number of layers), and image resolution together instead of independently scaling each parameter. EfficientNet-B7 developers have already set these parameters through extensive experimentation. This pre-trained model reduces the need for setting many hyperparameters. Also, features learned by this pre-trained model can help enhance the accuracy and efficiency of the branch and ramp constraint classification problem. This architecture can serve as a foundation for power system optimization problems that can be converted into computer vision problems.

Transfer Learning: Transfer learning refers to utilizing features learned from a problem and leveraging them for a new problem to improve learning performance and accuracy. We propose exploiting pre-trained EfficientNet-B7 and using transfer learning to adapt it with the considered constraint classification problem. Such pre-trained models contain important features preserved in a feature space and transferable to other tasks. Three alternatives exist: 1) reusing the trained weights of one or more layers of a pre-trained network. 2) Fine-tuning all layers entirely for a new dataset (a weight initialization scheme using pre-trained weights). 3) Keeping pre-trained weights fixed and adding new layers on top of the pre-trained network.

Figure 6 shows the concept of transfer learning that follows several steps, as shown in Algorithm II. We remove the output layer of EfficientNet-B7 and add a customized output layer whose size depends on the number of branch and ramp constraints. We also add a hidden layer before the output layer. The added hidden and output layers will transform the old features into predictions on a new dataset. We fine-tune the weights of the last few pre-trained layers (i.e., layers before the added new layers) of EfficientNet-B7, as these final layers capture more data specific features. One can unfreeze some (e.g., three) last hidden layers before the output layer or unfreeze the last hidden layer and increase the number of unfrozen last hidden layers until a desirable accuracy is obtained. These layers are fine-tuned at a low learning rate with the new DCT images representing power system operating conditions.

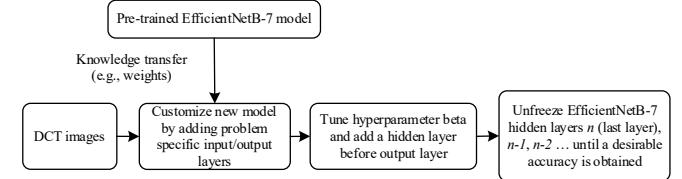


Fig. 6. Block diagram of the proposed transfer learning procedure.

Algorithm II Pseudocode to transfer learning

1. Import weights of a pre-trained EfficientNet-B7 model
2. Remove the top output layer
3. Freeze layers to avoid destroying learned features
4. Add a new trainable output layer on top of frozen layers
5. Add a new trainable hidden layer before the output layer
6. Train only new layers on using DCT image datasets representing power system operating conditions (a few epochs)
7. Unfreeze the last hidden layer and train the model at a very low learning rate (several epochs)
8. Repeat Step 7 by unfreezing the last hidden layers one by one until desirable accuracy is obtained

Loss Function: The considered constraints screening problem is a binary classification. A common loss function for binary classification problems is binary cross-entropy. However, this function may not be suitable for the line and ramp constraints classification problem as the dataset is unbalanced. The numbers of active and inactive constraints are not in the same order. Most constraints are inactive for the power system to comply with North American Electric Reliability Corporation (NERC) standards. On the other hand, the conventional accuracy metric is interpretable but not robust against uneven data and can yield misleading evaluation.

We have used the $F\beta$ score loss function with a customized β to reduce the impact of unbalanced data. A loss function should be continuous and differentiable for learning optimization problems. $F\beta$ score, which is a discrete value, is modified to make it differentiable.

$$F\beta\text{score} = (1 + \beta^2) \frac{Precision * Recall}{\beta^2 * Precision + Recall}$$

$$= \frac{(1 + \beta^2) * TP}{(1 + \beta^2) * TP + FP + \beta^2 * FN} \quad (13)$$

where precision, recall, TP, TN, FP, and FN metrics are:

- *True positives* (TP): Actual and predicted status is ACTIVE
- *True negatives* (TN): Actual and predicted status is INACTIVE.
- *False positives* (FP): Actual status is INACTIVE, and predicted status is ACTIVE (type I error).
- *False negatives* (FN): Actual status is ACTIVE and predicted status is INACTIVE (type II error).

$$Precision = \frac{TP}{TP + FP} \quad (14)$$

$$Recall = \frac{TP}{TP + FN} \quad (15)$$

Hyperparameter Tuning: Many important hyperparameters are set by EfficientNet-B7 developers. We only need to set β , the number of neurons in the newly added hidden layer before the output layer, and the number of EfficientNet's hidden layers that should be unfrozen. Hyperparameter β in (13) controls the importance of precision and recall and is usually tuned through experiment. $\beta < 1$ (e.g., 0.5) assigns more weight to precision and less weight to recall. $\beta = 1$ assigns the same weight to both precision and recall. $\beta > 1$ (e.g., 2) gives less weight to precision and more weight to recall. This approach is suitable when both precision and recall carry similar significance, but more attention is needed on false negatives. The number of neurons in the newly added hidden layer can be set as a 2^n number closest to the number of neurons in the output layer.

F. Line flow, Ramp Up, and Ramp Down Classifiers

We train three classifiers, one for each constraint type, instead of training a single classifier for all constraints. The first classifier is dedicated to line flow constraint status identification. The second classifier is devoted to generating unit ramp up limitations, and the third classifier is dedicated to ramp down constraints. This strategy can enhance the constraint classification accuracy and speed up the training process by parallel training.

V. NUMERICAL SIMULATION AND RESULTS ANALYSIS

The proposed algorithm is tested on the EEE 24-bus system, the IEEE 118-bus system, and the 6515-bus French system. The considered scheduling horizon has 24 time periods. The YALMIP toolbox and IBM-ILLOG-CPLEX are used to model and solve D-ED [28, 29]. The Python-based Keras framework is used for machine learning. Simulations are carried out on a computer with Intel(R) Xeon(R) 2.10 GHz CPU and 512 GB of RAM. We have posted our code on GitHub [9].

A. Active and Pseudo-active Constraint Statistics

Many operating condition scenarios are generated for each test system. The average percentage of active and pseudo-active constraints required to form the D-ED feasible region is reported in Table II. For instance, for the 118-bus system, the original D-ED problem has $186 \times 24 = 4463$ branch

constraints, $54 \times 23 = 1242$ ramp up constraints, and 1242 ramp down constraints. On average, the number of active branch, ramp up, and ramp down constraints are 4, 47, and 45, respectively. Sets $\mathcal{A}^k(\mathcal{H}_L(x))$, $\mathcal{A}^k(\mathcal{H}_{RU}(x))$, and $\mathcal{A}^k(\mathcal{H}_{RD}(x))$ include 10, 76, and 71 active and pseudo-active constraints. Pseudo-active constraints form a large percentage of $\mathcal{A}^k(\mathcal{H}_L(x))$, $\mathcal{A}^k(\mathcal{H}_{RU}(x))$, and $\mathcal{A}^k(\mathcal{H}_{RD}(x))$, without which the feasible design region of the truncated D-ED is not the same as that of the original D-ED. Table II shows that a larger percentage of generating unit ramp constraints is required than branch constraints to formulate the truncated D-ED. A similar trend is observed for the 6515-bus French system, where 19 generating units do not have a unique generation cost function.

TABLE II
AVERAGE PERCENTAGE OF ACTIVE AND PSEUDO-ACTIVE CONSTRAINTS

System	Active constraints			Active + pseudo-active constraints		
	Branch	RU	RD	Branch	RU	RD
Case24	0.47	3.5	2.9	0.53	3.6	3.0
Case118	0.12	3.9	3.75	0.26	6.4	5.9
Case 6515	0.3	18.3	9.7	0.06	20.7	10.7

B. Dataset Preparation and Learners Architecture

Algorithm I is implemented to generate training datasets. Nodal demand and cost are varied within a range to generate scenarios for each plausible network topology. Table III shows the perturbation range for each system as compared to the base case values. D-ED is solved for each scenario. Active and pseudo-active constraints are labeled as 1, and the rest are labeled as 0. The operating condition scenarios are converted into the DCT image format. We assign a branch label set, a ramping up label set, and a ramping down label set for each DCT image. Three classifiers are trained whose input is DCT images. The target of classifiers 1, 2, and 3 are the branch label set, the ramping up label set, and the ramping down label set. Classifier parameters are given in Table IV. To form each classifier, the EfficientNet-B7 architecture is imported along with its weights, excluding the top layer. This truncated architecture becomes the base model. A new model is created using transfer learning by adding a customized hidden layer and output layer to the base model to comply with the new learning tasks (e.g., learning branch classification).

Historical data should be collected to utilize the proposed algorithm for large real-world systems. Redundant unique samples should be dropped. If the dataset is large, various scenario clustering and reduction techniques (e.g., K-means) can be implemented. Similar scenarios can be grouped in the same cluster, and one or multiple representatives from each class can be selected. Although the training might take time for large systems, it is an offline procedure carried out once.

TABLE III
VARIATION RANGE

System	Load		Cost coefficient	No. of scenario	
	ω	Δ_d^L to Δ_d^U		Train	Test
Case24	70%-130%	97%-103%	$\pm 15\%$	4000	1000
Case118	90%-119%	97%-103%	$\pm 15\%$	4000	1000
Case6515	80%-119%	97%-103%	$\pm 15\%$	3010	700

TABLE IV

HYPERPARAMETERS OF MODIFIED EFFICIENTNET-B7 ARCHITECTURE: ADDED LAYERS AFTER FLATTENING

Final FC layers and Training parameters			
Classifiers (Branch, RU, RD)	Added hidden layer=1, Batch size=500, Validation split=10%, Early stopping with Patience 10 (min. no. of epochs)	Activation =ReLU & Sigmoid	Loss & metric= F2 Optimizer= Adam

C. Prediction Analysis

The size of test datasets is given in Table III. The constraint statuses predicted by classifiers are compared with ground truth data obtained by solving D-ED. As the power system is safety-critical, we analyze false negatives and false positives performance statistics. The classification accuracy depends on hyperparameter values, such as β and the number of training epochs. The percentage of FPs and FNs can be controlled by tuning hyperparameters. Since having constraints that are active but classified as inactive is undesirable, we set $\beta = 2$, meaning that recall is twice as important as precision. This reduces the number of FNs. Table V shows the average FP and FN percentages. In general, more ramping constraints are misclassified in the FP category than line constraints. This might be because intertemporal connectivity makes classifying ramp constraints more complex than line flow constraints. Some unnecessary constraints are added to the truncated D-ED due to FPs. This increases the size of the truncated D-ED. Having fewer FNs is more crucial as they include the necessary information to form a feasible design space. A few FNs are observed that will be added to the optimization constraints using the iterative loop. The prediction error is inevitable, and thus removing this loop would make the truncated economic dispatch solution infeasible for test scenarios with nonzero FNs.

TABLE V

AVERAGE FPs AND FNs PER SCENARIO

System	False negative (FN)			False positive (FP)		
	branch	RU	RD	branch	RU	RD
Case24 (Eff)	0.31	7.0	2.9	17.5	4.4	40
Case24 (NN)	1.1	6.1	6.3	1.4	10.3	9.0
Case24 (CNN)	0.2	6.4	6.6	22.7	20.0	3.2
Case118 (Eff)	1.3	2.3	5.6	151	370	278
Case118 (NN)	0	0	0	12.4	78.3	72.3
Case118 (CNN)	1.7	60	54	278	841	405
Case6515 (Eff)	2.8	8.4	4.5	133	4156	4399
Case6515 (NN)	0.2	0.03	0.03	256	7636	7524

One can select a smaller β to reduce FPs. But it would increase FNs, and thus the number of iterations and overall solution time. In the worst-case scenario, the number of iterations would be equal to the number of branch and ramp constraints minus FNs. However, it would not happen as not all

branch and ramp constraints are active. Also, our observations (see Table VI) show that several pseudo-active constraints are added to optimization after carrying out each iteration.

We have also trained a CNN and a neural network (NN). Generally speaking, these two learners have more FP misclassifications. Although we have used the pre-trained EfficientNet-B7, one can use NN and CNN. One of the advantages of EfficientNet-B7 is its pre-trained known structure. A user does not need to make significant changes in the learner structure, such as the number of hidden layers and neurons. Unlike CNN and NN, for which the best learner structure should be found based on many trials, a user needs only to change the size of output and input layers of the pre-trained EfficientNet-B7 based on the considered power system size.

D. Truncated D-ED Runtime Analysis and Solution Quality

We use an integrality gap index to show how close are the solutions of the truncated and original D-EDs [8]. f^{T-DED} and f^{D-ED} are, respectively, objective values obtained from the proposed algorithm and the original D-ED problem.

$$\text{Integrality Gap\%} = \frac{|f^{T-DED} - f^{D-ED}|}{f^{D-ED}} \times 100 \quad (16)$$

The average integrality gap for all test scenarios is negligible (less than 10^{-7}) for all three test systems, showing that the proposed truncated D-ED algorithm provides the same solution as the original D-ED.

The computation time saving obtained by the proposed D-ED algorithm is reported in Table VI. The average number of iterations and time over test scenarios are reported. For the 118-bus system, for instance, the iterative loop converges after 2.07 iterations on average. This is due to the FN misclassifications. Without the iterative loop, the truncated D-ED solution may become infeasible as a few necessary active or pseudo-active constraints are missed in the truncated constraint set. The time saving becomes promising as the system size increases. While no time saving is observed for the 24-bus system, the optimization solution time is reduced by 99% for the 6515-bus system.

TABLE VI
PERFORMANCE ANALYSIS: AVERAGE ITERATION NUMBERS AND TIME-SAVING

Systems	Original D-ED time	Truncated D-ED		Time saving
		No. iter	Total time	
Case24	11 ms	1.82	14 ms	No save
Case118	220 ms	2.07	140 ms	32%
Case6515	238 sec	1.69	< 1 sec	99%

E. Comparison with ICG

We compare the proposed approach with ICG using the 6515-bus system. ICG is a popular method in which constraints are relaxed, a master problem is formulated and solved, and violated constraints are added to the master problem iteratively. Simulations are run for all test operating condition scenarios, and average values are reported in Table VII.

TABLE VII
COMPARISON WITH ICG

Test system	Average solver time (sec.)	Average number of iterations
Original D-ED (benchmark)	238	-
ICG	64	50.4
Proposed approach	< 1	1.69

The original D-ED problem with all constraints is solved to obtain benchmark results. It takes 238 seconds. The proposed approach takes much fewer iterations and less time than ICG to find the optimal D-ED solution. The proposed approach is 98% faster than ICG.

F. Memory Usage Analysis and Runtime Comparison for Combined Branch and Ramp Constraints Screening

The average memory requirement in megabytes (MB) for building the constraint set and solving time are reported in Table VIII for only branch constraints screening and branch and ramp constraints screening. The least memory usage is observed after screening both branch and ramp constraints and dropping inactive constraints from the model. For instance, for the 6515-bus system, the original D-ED problem occupies 5489 MB of memory. It reduces almost 37 times by dropping inactive branch constraints and 211 times if both inactive branch and ramp constraints are dropped. While screening only branch constraints leads to better time saving for smaller systems, screening both branch and ramp constraints saves more time for larger systems. For the 6515-bus system, branch and ramp constraints screening achieve 30% more time-saving. Since screening only branch constraints would lead to a good time saving even for large systems, one may ignore ramp constraints screening. However, we suggest branch and ramp constraints screening for larger systems to reduce memory usage significantly. For instance, for the 6515-bus system, screening both branch and ramp constraints results in a better time saving and a significant RAM requirement reduction. It thus makes solving large systems possible even without the need for supercomputers with large memory.

TABLE VIII

AVERAGE MEMORY REQUIREMENT (MB) TO BUILD CONSTRAINTS AND SOLVER TIME FOR TWO CONSTRAINT SCREENING SCHEMES

System	Original problem	Screening branch constraints		Screening branch & ramp constraints	
		RAM	RAM	Time	RAM
Case24	7	2.5	10 ms	0.35	14 ms
Case118	44	12.4	20 ms	1.2	140 ms
Case6515	5489	145	1 sec	26	0.7 sec

G. Demand vs. DCT as Learner Input

Under a given demand value, thermal units' generation cost coefficients and grid topology may differ, resulting in different active/inactive constraint sets. If demand is used as the only input feature, the learner may face difficulty predicting the status of constraints. More misclassifications may increase the problem size and number of loop iterations and thus the solver time. Table IX shows the number of loop iterations and solver

time. We suggest using DCT as the learner input to reduce the solver time and required memory usage. However, one can use only demand since the embedded iterative loop can eventually capture all required constraints to form a truncated D-ED.

TABLE IX
TIME GAIN ANALYSIS USING DEMAND AND DCT AS LEARNER INPUT

Systems	Truncated D-ED (only demand)		Truncated D-ED (DCT image)	
	No. iter	Total time	No. iter	Total time
Case24	1.8	14 ms	1.82	14 ms
Case118	3.8	192 ms	2.07	140 ms
Case6515	1.71	< 1 sec	1.69	< 1 sec

E. Hamming Distance Analysis

Hamming distance measures the difference between two binary strings. It is an indicator of output feature sensitivity to input features and the robustness of the proposed algorithm. We have perturbed the demand, identified active constraints for two consecutive demand scenarios, and calculated Hamming distance between the active constraint status indicators, which are 0/1 strings. Table X shows the average Hamming distance for branch and ramp constraints. The sensitivity of output features to input features is not high. Hamming distances corresponding to branch constraints are lower than those of ramp constraints. As shown in Table V, this could justify observing more ramp constraints misclassifications than branch constraints.

TABLE X
HAMMING DISTANCE ANALYSIS

System	Branch	Ramp up	Ramp down
Case24	0.34%	5.19%	4.8%
Case118	0.21%	6.21%	5.64%
Case6515	0.013%	2.02%	1.69%

VI. CONCLUSION

A small subset of inequality constraints contains enough information to form the dynamic economic dispatch feasible region. This paper presents a learning-aided iterative algorithm to identify active and pseudo-active branch flow and thermal unit ramp up/down constraints required to form the D-ED feasible space for each operating condition scenario. Three classifiers are trained, one for each type of constraints, taking into consideration network topology. Using these classifiers' predictions, a truncated D-ED is formed that is smaller and less computationally expensive than the original D-ED problem.

The number of iterations of the learning-aided approach is much less than the classical ICG. Also, filtering active and pseudo-active constraints reduces iterations much more than filtering only active constraints. The benefit of constraint filtering is more significant for larger systems. The average runtime saving for the 6515-bus system is 99%. We have observed that filtering both branch and ramp constraints would lead to better time saving and memory usage than filtering only branch constraints. However, the learning-aided approach can filter out only branch constraints. This would result in a good enough time and memory usage saving for large systems.

REFERENCES

[1] D. W. Ross and S. Kim, "Dynamic economic dispatch of generation," *IEEE transactions on power apparatus and systems*, no. 6, pp. 2060-2068, 1980.

[2] G. B. Sheble and G. N. Fahd, "Unit commitment literature synopsis," *IEEE Transactions on Power Systems*, vol. 9, no. 1, pp. 128-135, 1994.

[3] A. Kargarian *et al.*, "Toward distributed/decentralized DC optimal power flow implementation in future electric power systems," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 2574-2594, 2016.

[4] Q. Zhai, X. Guan, J. Cheng, and H. Wu, "Fast identification of inactive security constraints in SCUC problems," *IEEE Transactions on Power Systems*, vol. 25, no. 4, pp. 1946-1954, 2010.

[5] A. J. Ardakani and F. Bouffard, "Identification of umbrella constraints in DC-based security-constrained optimal power flow," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 3924-3934, 2013.

[6] R. Madani, J. Lavaei, and R. Baldick, "Constraint screening for security analysis of power networks," *IEEE Transactions on Power Systems*, vol. 32, no. 3, pp. 1828-1838, 2016.

[7] A. S. Xavier, F. Qiu, F. Wang, and P. R. Thimmapuram, "Transmission Constraint Filtering in Large-Scale Security-Constrained Unit Commitment," *IEEE Transactions on Power Systems*, vol. 34, no. 3, pp. 2457-2460, 2019.

[8] D. A. Tejada-Arango, P. Sánchez-Martín, and A. Ramos, "Security constrained unit commitment using line outage distribution factors," *IEEE Transactions on power systems*, vol. 33, no. 1, pp. 329-337, 2017.

[9] *Code*. Available: <https://github.com/FouadHasan/Topology-Aware-Learning-Assisted-Branch-and-Ramp-Constraints-Screening-for-Dynamic-Economic-Dispatch>

[10] K. Baker, "Learning warm-start points for AC optimal power flow," in *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*, 2019, pp. 1-6: IEEE.

[11] F. Fioretto, T. W. Mak, and P. Van Hentenryck, "Predicting AC optimal power flows: Combining deep learning and lagrangian dual methods," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, vol. 34, no. 01, pp. 630-637.

[12] M. Chatzos, F. Fioretto, T. W. Mak, and P. Van Hentenryck, "High-Fidelity Machine Learning Approximations of Large-Scale Optimal Power Flow," *arXiv preprint arXiv:2006.16356*, 2020.

[13] A. Veloso and P. Van Hentenryck, "Combining Deep Learning and Optimization for Security-Constrained Optimal Power Flow," *arXiv preprint arXiv:2007.07002*, 2020.

[14] X. Pan, M. Chen, T. Zhao, and S. H. Low, "DeepOPF: A Feasibility-Optimized Deep Neural Network Approach for AC Optimal Power Flow Problems," *arXiv preprint arXiv:2007.01002*, 2020.

[15] A. S. Zamzam and K. Baker, "Learning optimal solutions for extremely fast AC optimal power flow," in *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, 2020, pp. 1-6: IEEE.

[16] Y. Yang, Z. Yang, J. Yu, K. Xie, and L. Jin, "Fast Economic Dispatch in Smart Grids Using Deep Learning: An Active Constraint Screening Approach," *IEEE Internet of Things Journal*, 2020.

[17] A. Venzke, G. Qu, S. Low, and S. Chatzivasileiadis, "Learning optimal power flow: Worst-case guarantees for neural networks," in *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, 2020, pp. 1-7: IEEE.

[18] K. Baker, "A Learning-boosted Quasi-Newton Method for AC Optimal Power Flow," *arXiv preprint arXiv:2007.06074*, 2020.

[19] Á. S. Xavier, F. Qiu, and S. Ahmed, "Learning to solve large-scale security-constrained unit commitment problems," *INFORMS Journal on Computing*, 2020.

[20] S. Pineda, J. M. Morales, and A. Jimenez-Cordero, "Data-Driven Screening of Network Constraints for Unit Commitment," *IEEE Transactions on Power Systems*, 2020.

[21] Y. Ng, S. Misra, L. A. Roald, and S. Backhaus, "Statistical learning for DC optimal power flow," in *2018 Power Systems Computation Conference (PSCC)*, 2018, pp. 1-7: IEEE.

[22] D. Deka and S. Misra, "Learning for DC-OPF: Classifying active sets using neural nets," in *2019 IEEE Milan PowerTech*, 2019, pp. 1-6: IEEE.

[23] A. J. Ardakani and F. Bouffard, "Prediction of Umbrella Constraints," in *IEEE Power Systems Computation Conference (PSCC)*, 2018, pp. 1-7.

[24] K. Baker and A. Bernstein, "Joint Chance Constraints in AC Optimal Power Flow: Improving Bounds through Learning," *IEEE Transactions on Smart Grid*, 2019.

[25] S. Zhang, H. Ye, F. Wang, Y. Chen, S. Rose, and Y. Ma, "A Data-aided Security Constraint Prescreening Technique and Application to Real-world System," in *2019 North American Power Symposium (NAPS)*, 2019, pp. 1-6: IEEE.

[26] A. J. Wood, B. F. Wollenberg, and G. B. Sheble, *Power generation, operation, and control*. John Wiley & Sons, 2013.

[27] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning*, 2019, pp. 6105-6114: PMLR.

[28] J. Lofberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *2004 IEEE international conference on robotics and automation (IEEE Cat. No. 04CH37508)*, 2004, pp. 284-289: IEEE.

[29] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on power systems*, vol. 26, no. 1, pp. 12-19, 2011.

Fouad Hasan (S'18) received his B.Sc. degree in electrical engineering from Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh, in 2015. He is currently pursuing his Ph.D. degree with the Department of Electrical and Computer Engineering, Louisiana State University, Baton Rouge, LA, USA. His research interests include power systems optimization, electricity market, and machine learning.

Amin Kargarian (SM'20) is an Associate Professor with the Electrical and Computer Engineering Department, Louisiana State University, Baton Rouge, LA, USA. His research interests include optimization, machine learning, quantum computing, and their applications to power systems.