energies

MDPI

Article

Online Distribution Network Scheduling via Provably Robust Learning Approach

Naixiao Wang [1], Xinlei Cai [1], Linwei Sang [2,*], Tingxiang Zhang [2], Zhongkai Yi [2] and Ying Xu [2]

[1] Electric Power Dispatching and Control Center of Guangdong Power Grid Co., Ltd., Guangzhou 510220, China
[2] School of Electrical Engineering and Automation, Harbin Institute of Technology, Harbin 150001, China
* Correspondence: 1180610614@stu.hit.edu.cn

Abstract: Distribution network scheduling (DNS) is the basis for distribution network management, which is computed in a periodical way via solving the formulated mixed-integer programming (MIP). To achieve the online scheduling, a provably robust learn-to-optimize approach for online DNS is proposed in this paper, whose key lies in the transformation of the MIP-based DNS into the simple linear program problem with a much faster solving time. It formulates the parametric DNS model to construct the offline training dataset and then proposes the provably robust learning approach to learn the integer variables of MIP. The proposed learning approach is adversarial to minor perturbation of input scenario. After training, the learning model can predict the integer variables to achieve online scheduling. Case study verifies the acceleration effectiveness for online DNS.

Keywords: distribution network scheduling; learn-to-optimize; online scheduling; machine learning

check for updates

Citation: Wang, N.; Cai, X.; Sang, L.; Zhang, T.; Yi, Z.; Xu, Y. Online Distribution Network Scheduling via Provably Robust Learning Approach. *Energies* **2024**, *17*, 1361. https://doi.org/10.3390/en17061361

Academic Editor: Georgios Christoforidis

Received: 19 January 2024
Revised: 5 March 2024
Accepted: 6 March 2024
Published: 12 March 2024

1. Introduction

To mitigate climate change, the power system supply relies on more renewable generation. However, the volatility and uncertainty of renewable resources pose new challenges to the distribution system management. So the increasing penetration of distributed generation resources in the distribution networks leads to the focus on effective, efficient, and economic operation of a distribution network. Distribution network scheduling (DNS) is the basic scheduling technology of the distribution network management based on optimization research theory. The mathematical form of DNS can be formulated as the mixed-integer programming (MIP) problem and computed repeatedly daily to support the safe and economic operation for distribution system operators [1]. To keep the system safe and economic, DNS is solved for different input scenarios to reach the proper system decisions repeatedly, where the upcoming scenario could be similar to the historical scenario. However, repeatedly solving MIP for DNS could be time-consuming and energy-wasting, because solving similar scenarios of DNS requires solving the corresponding MIP, which is intractable and time-consuming [2–5]. For MIP, due to the complex features of integer variables, current popular algorithms leverage the idea of branch and bound or benders cut, and the corresponding solution time is unstable. Compared to MIP, linear programming (LP) is more easy to solve due to its continuous variables and convex features, which could be treated as tractable with a stable solution time. The online LP solution is easier than the online MIP solution. So, to achieve the online scheduling via LP, this paper proposes predicting the integer variables of DNS and leveraging the advanced machine learning (ML) methods to learn from the historical solution of DNS for accelerating the similar new DNS scenario solution, which can achieve the online DNS. The key idea behind online scheduling lies in reducing the MIP to LP via ML for the effective distribution network scheduling.

There is growing interest in leveraging ML to advancing the solution of optimization models from the ML area, which is called learn-to-optimize. This area is attracting the interest from both machine learning areas and operation research areas, which can be categorized into two types: (1) learning the values of variables in the optimization models via complex ML models, (2) and learning part of values of variables in the optimization models via complex ML models. The dividing and neural branching neural networks are proposed in Ref. [6] to learn how to branch-and-bound more efficiently by reinforcement learning approaches. The DeepOPF model is proposed in Ref. [7] to learn the ACOPF results more efficiently. End-to-end optimization proxies are proposed in Ref. [8] to achieve a fast economic dispatch. Confidence-Aware Graph Neural Networks are further proposed in Ref. [7] to learn the system reliability for online reliability assessments. However, the above work focuses on the learning of optimization results. We focus on combining learning and optimization for tractable online DNS.

To achieve online DNS in a provably robust way, we propose the provably robust learn-to-optimize (PR-L2O) approach, whose key lies in the transformation of the MIP-based DNS problem into the tractable corresponding convex form, where convex problems are easier than MIP. Concretely, the parametric DNS model is first proposed to map the input DNS scenarios to the values of MIP integer variables, and the DNS training/testing datasets are sampled offline from the above offline model. In the model training process, the provably robust learning approach is utilized to learn and predict the integer variable values in a provably robust way, which is adversarial to the input scenarios difference. The benefits of the provably robust learning method lies in the better generalization to address the vulnerability of machine learning models. After learning, we predict the integer variables of MIP-based DNS and solve the reduced convex problems. Traditional methods solve the complex mixed-integer program of DNS directly, and in comparison, the proposed method utilizes the learning models to identify the values of integer variables in MIP-based DNS and solve the reduced easy linear programming problem.

We summarize the main contributions of this paper into two parts as follows: (1) Based on the MIP-based DNS model, this paper investigates the similarity of integer values in MIP and leverages the historical solutions to facilitate the upcoming solution for online scheduling. (2) Considering the vulnerability of learning model, the provable robust learning approach is proposed to learn the mapping between the scenario and integer variables based on the convex outer bound formulation of the input scenario difference perturbation, the dual network design, and provably robust loss function design. In summary, the proposed PR-L2O approach can achieve the solution acceleration and enhance the benefits of L2O due to data drift between training and testing datasets. The primary objective of this research is leveraging the machine learning models to enhance the solution of distribution network scheduling in a provably robust way. Compared to our previous works [9,10], this paper leverages the provably robust learning [11,12] to achieve the online regulation [13] with better generalization performance. In contrast, [9] focuses on unit commitment, and [10] focuses on online voltage regulation without provably robustness consideration.

The remainder of this paper is organized as follows. Section 2 presents the parametric DNS formulation. Section 3 proposes the PR-L2O approach for online DNS. Section 4 verifies the effectiveness of this paper by numerical simulation. Section 5 summarizes and concludes the total paper.

2. Parametric Distribution Network Scheduling Formulation

This section extends the conventional distribution network scheduling to the parametric distribution network scheduling (P-DNS) problem [14] and discusses the P-DNS solving process [15,16], which achieves the mapping from input scenarios to output decision via MIP in an implicit way [17]. The key of P-DNS lies in achieving the mapping from the parameters (load) to the optimal decisions via solving the corresponding scheduling optimization problems implicitly [18–20].

2.1. DNS Model

Traditional DNS models schedule the distributed energy resources with continuous and discrete variables to minimize the system operating cost in an optimized way, which is based on the prediction of uncertain parameters in the distribution network. DNS is usually formulated as a mixed-integer convex program (MICP) mathematically. Furthermore, commercial solvers can solve the formulated MICP problems to obtain the optimal decisions under the various constraints embedded within the advanced algorithms, such as the branch and bound (B&B) algorithm and bender algorithm.

However, the solving of MICP in DNS is intractable due to the complex features of integer variables. To tackle this problem, firstly, we consider formulating a basis DNS model under basic distribution networks with a set G of distributed generators, a set of B buses, and a set of L distribution lines. DNS models optimize the integer variables $s_{i,t}$ and continuous integer power variables $p_{g,t}$ for the T load-ahead scheduling time-step.

$$\min \sum_{g\in G}\sum_{t\in T} f_g(p_{g,t}, s_{g,t}) \tag{1a}$$

$$s.t.(p_{g,t}, s_{g,t}) \in \mathcal{G}_g \quad g \in G \tag{1b}$$

$$\sum_{j\in\mathcal{J}(i)} p_{ij,t} - \sum_{h\in\mathcal{H}(i)} p_{hi,t} = p_{i,t}^{pv} - p_{i,t}^{d}, \forall i \in \mathcal{N_B} \tag{1c}$$

$$\sum_{j\in\mathcal{J}(i)} q_{ij,t} - \sum_{h\in\mathcal{H}(i)} q_{hi,t} = q_{i,t}^{svr} + q_{i,t}^{cb} + q_{i,t}^{pv} - q_{i,t}^{d}, \forall i \in \mathcal{N_B} \tag{1d}$$

$$V_{j,t} = V_{i,t} - \frac{r_{ij}p_{ij,t} + x_{ij,t}q_{ij,t}}{V_0}, \forall ij \in \mathcal{N_L} \tag{1e}$$

$$V_{min} \le V_i \le V_{max}, i \in \mathcal{N_B} \tag{1f}$$

$$V_{1,t} = V_{base,t}(r^{min} + \sum_s r_s \sigma_{s,t}^{oltc}) \tag{1g}$$

$$\sigma_{m-1,t}^{oltc} \ge \sigma_m^{oltc}, m \in [2, T_{max}^{oltc}] \tag{1h}$$

$$T_t^{oltc} = \sum_s \sigma_{s,t}^{oltc}, T_t^{oltc} \le T_{max}^{oltc} \tag{1i}$$

$$T_t^{oltc} - T_{t-1}^{oltc} \ge \sigma_t^{oltc,in} - \sigma_t^{oltc,de} T_{max}^{oltc} \tag{1j}$$

$$T_{t-1}^{oltc} - T_t^{oltc} \le \sigma_t^{oltc,in} T_{max}^{oltc} - \sigma_t^{oltc,de} \tag{1k}$$

$$\sum_{t\in\mathcal{T}} (\sigma_t^{oltc,in} + \sigma_t^{oltc,de}) \le N_{max}^{oltc} \tag{1l}$$

$$\sigma_t^{oltc,in} + \sigma_t^{oltc,de} \le 1 \tag{1m}$$

$$q_{i,t}^{cb} = q_{tap} T_{i,t}^{cb}, i \in \mathcal{N_{CB}} \tag{1n}$$

$$T_{i,t}^{cb} = \sum_s \sigma_{s,t}^{cb}, T_t^{cb} \le T_{max}^{cb}, i \in \mathcal{N_{CB}} \tag{1o}$$

$$T_{i,t}^{cb} - T_{i,t-1}^{cb} \ge \sigma_{i,t}^{cb,in} - \sigma_{i,t}^{cb,de} T_{max}^{cb}, i \in \mathcal{N_{CB}} \tag{1p}$$

$$T_{t-1}^{oltc} - T_t^{cb} \le \sigma_t^{cb,in} T_{max}^{cb} - \sigma_t^{cb,de}, i \in \mathcal{N_{CB}} \tag{1q}$$

$$\sum_{t\in\mathcal{T}} (\sigma_t^{cb,in} + \sigma_t^{cb,de}) \le N_{max}^{oltc}, i \in \mathcal{N_{CB}} \tag{1r}$$

$$\sigma_t^{cb,in} + \sigma_t^{cb,de} \le 1, i \in \mathcal{N_{CB}} \tag{1s}$$

$$(p_i^{pv})^2 + (q_i^{pv})^2 \le (S_i^{pv})^2, i \in \mathcal{N_{PV}} \tag{1t}$$

$$q_{i,min}^{svr} \le q_i^{svr} \le q_{i,max}^{svr}, i \in \mathcal{N_{SVR}}. \tag{1u}$$

The key variables are defined for clear delivery. $p_{g,t}$ and $s_{g,t}$ are the power output and status of generation i in time t. Equations (1c)–(1e) formulate the linearized distflow model; Equations (1h)–(1u) formulate the operation of OLTC, CB and PV. The other detailed variables definitions are referred to in Ref. [1]. We unify the integer variables of (1) as s_{dns} for simplicity.

2.2. P-DNS Model

Based on the DNS model, we formulate the parametric DNS (P-DNS) in a general form by (2). The P-DNS model achieves the mapping from the load demand d to the generator continuous power output varibles p_g and binary decision variables s_{dns} via the implicit MIP optimization model of DNS. We denote the system load as input parameters and replaces notation d with θ.

$$f(\theta) = \min_{p,s} \sum_{g \in G} f_g(p_g, s_{dns}; \theta) \tag{2a}$$

$$s.t. \quad f_i(p_g, s_{dns}; \theta) \leq 0, i = 1, \ldots, m_f \tag{2b}$$

$$h_j(p_g, s_{dns}; \theta) = 0, i = 0, \ldots, m_h \tag{2c}$$

$$p_g \in \mathbb{R}^T, s_{dns} \in \{0,1\}^T. \tag{2d}$$

Obviously, the binary variable s_{dns} are more "complicated" but less changing than the continuous variables. Because (2) is reduced to a tractable convex problem with fixed-integer variables. To achieve online regulation, we propose to predict the integer variables of s^*_{dns} based on the input parameters θ to achieve MIP-based DNS solution acceleration, as shown below in (3).

$$\hat{c}(\theta) = \min_{p_g} \sum_{g \in G} c_g(p_g; s^*_{dns}, \theta) \tag{3a}$$

$$s.t. \quad f_i(p_g; s^*_{dns}, \theta) \leq 0, i = 1, \ldots, m_f \tag{3b}$$

$$h_j(p_g; s^*_{dns}, \theta) = 0, i = 0, \ldots, m_h \tag{3c}$$

$$p_g \in \mathbb{R}^T \tag{3d}$$

$$s^*_{dns}(\theta) = f(\theta) \tag{3e}$$

where $s^*_{dns}(\theta)$ refers to values of integer variables in the MIP-based DNS under θ, denoted by policy; the $f(\cdot)$ in (3e) is the learning model, which is called the policy identifier and learned from the historical solution dataset.

3. Provably Robust Distribution Network Scheduling

Based on the P-DNS formulation, the provably robust learn-to-optimize (PR-L2O) approach is introduced to identify the integer values of P-DNS in MIP-based DNS model, which is composed of data processing, PR-L2O learning and online implementation, as shown in Figure 1.

As shown above, (1) in the data stage, we leverage the P-DNS model of (2) to generate solution instances for different scenarios and construct the corresponding P-DNS dataset; and (2) in the learning stage, we leverage the neural networks to construct the policy identifier and leverage the P-DNS dataset to train the identifier in a provably robust way. In online stage, for an upcoming scenario, the trained identifier predicts the integer values from different groups, and the optimal decisions are obtained from the solving reduced model of (3a). Compared to the solution of MICP, the reduced tractable convex problem is much easier to solve and can satisfy the online solution form. It should be noted that due to vulnerability of machine learning models, the prediction results of machine learning models could be affected by input noise, and learning models should be trained to be adversarially robust to these noises. In our online DNS case, the training set could be similar but different from the upcoming instances. The integer variables from the prediction models should be robust to these minor differences via the proposed provably robust training process, based on the intuition that minor differences between the training and testing instances do not change the values of integer variables. Compared to the solution of the tractable convex problem, the solution of the mixed-integer program is much more complex due to the existence of integer variables. So we propose to utilize the learning models to identify the values of integer variables in MIP-based DNS, which transforms it into a tractable convex problem.

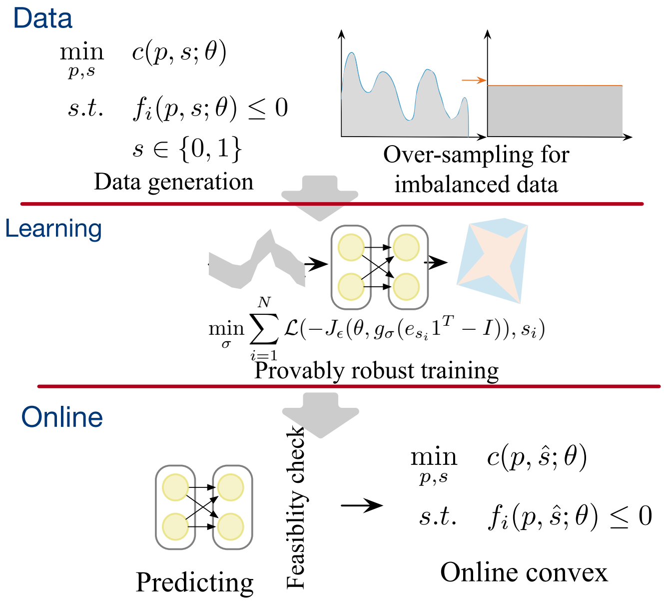

Figure 1. The online DNS framework via provably robust learning.

3.1. Provably Robust Learning Approach for Online DNS

Then, we go to the key of online DNS and leverage the provably robust learning approach, whose key lies in learning and predicting the integer variables in the model. The motivation behind the provable learning is that the upcoming scenarios may not be the same as the historical solutions, but they can feature similarity, which can be formulated as the input perturbation difference.

3.1.1. Motivation

Though the machine learning models feature a strong representational capacity, recent machine learning research demonstrates the prevalence of adversarial examples, where an adversarial example is a sample of input data which is modified very slightly in a way that is intended to cause a machine learning classifier to misclassify. It is called the "vulnerability" of machine learning models to the adversarial examples in real-life cases. To address the above vulnerability in learning the integer variables of P-DNS, a provably robust training approach for online DNS is proposed to learn the provably robust learning models, which achieve the provably robust feature. It means that the learning model can be adversarial to input scenario perturbation differences. The key of the perturbation robust model is the robust loss function, which can be derived from the convex outer bound formulation of the learning models under the input perturbation and dual network model.

3.1.2. Neural Networks for Online DNS

We leverage deep learning models to predict the integer variables of P-DNS of (2), where a k-layer neural networks then is formulated in (4) as follows:

$$\hat{z}_{i+1} = W_{i,c} z_i + b_{i,c}, \text{ for } i = 1, \ldots, k-1 \tag{4a}$$

$$z_i = \text{ReLU}(\hat{z}_i), 0, z_1 = \theta, \hat{s}_c = \hat{z}_k \text{ for } i = 2, \ldots, k-1 \tag{4b}$$

where $W_{i,c}$ is the linear transformation matrix of layer i; $b_{i,c}$ is the bias of layer i; z_i is the input neuron set of layer i; $\hat{z}_i$ is the output neuron set from the linear transformation; $\text{ReLU}(\cdot)$ is the activation function, which is defined as $\max(\cdot, 0)$; θ refers to the input scenario of P-DNS; and $\hat{s}_c$ refers to the predicted integer variables in P-DNS, compared to the true integer variables of s_c.

From a further model, in a function form, the multi-layer neural network of (4) can be formulated in the $\hat{s} = I_{\sigma_c}(\theta)$. So σ_c represents the set of $\{W_{i,c}, b_{i,c}\}_{i\in[k]}$ in (4).

3.1.3. Convex Outer Bound Formulation of the Neural Network

Based on (4), we further perturb the input scenario θ by Δ to construct the DNS input polytope of $||\Delta||$ and further construct the output polytope in (5) as follows:

$$\mathcal{B}_\epsilon(\theta) = \{I_\sigma(\theta + \Delta) : ||\Delta||_\infty < \epsilon\}, \tag{5}$$

where ϵ is defined as the minor perturbation scale value, and $\mathcal{B}_\epsilon(\theta)$ is the corresponding output adversarial polytope, which is highly non-convex as shown in the pink area of Figure 2.

It should be noted that the DNS input polytope $\mathcal{B}_\epsilon(\theta)$ is a non-convex set as shown in Figure 2, which is hard to be optimized over for the robust results.

So we construct a convex outer bound over the convex set to approximate $\mathcal{B}_\epsilon(\theta)$ by $\hat{\mathcal{B}}_\epsilon(x)$, as shown in the extended blue, in Figure 2. Compared to $\mathcal{B}_\epsilon(\theta)$, the relaxed $\hat{\mathcal{B}}_\epsilon(x)$ occupies more area, indicating higher conservativeness. Not only the pink area but the extended area is considered. The benefits from the above transformation leads to the convex area of the output area. Convex features lead to good optimization quality, which is the basis of provably robust training.

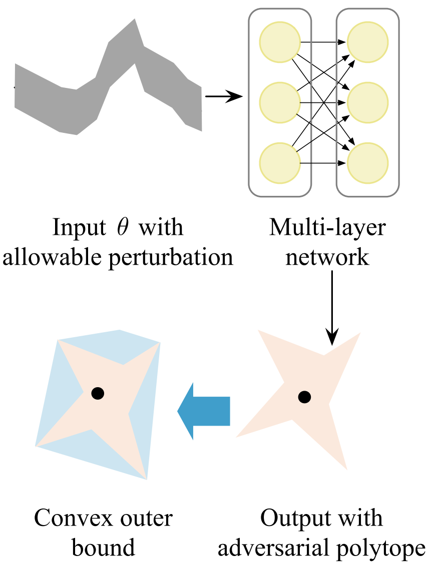

Figure 2. Input perturbation of networks.

The key of the above $\hat{\mathcal{B}}_\epsilon(x)$ lies in the relaxation of the ReLU function in (4), which is shown in Figure 3. The ReLU activation function of (4) can be bounded by its upper envelope in the red area of Figure 3.

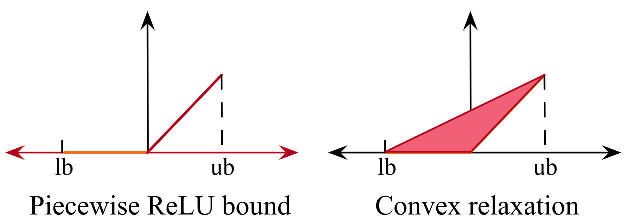

Figure 3. Convex upper envelope of ReLU activation function in multi-layer neural networks.

The above transformation formulation in Figure 3 can be further written mathematically in (6).

$$z_i \le \frac{u_i(u_i - \hat{z}_i)}{u_i - l_i}, \tag{6a}$$

$$z_i \ge \hat{z}_i, z_i \ge 0, \tag{6b}$$

where u_i is the set of upper bound in layer i, and l_i is the set of lower bound in layer i.

Based on the upper envelope of (6) and the neural network model of (4), the $\hat{\mathcal{B}}_\epsilon(x)$ of (5) can be further formulated via an explicit way in the following (7).

$$\hat{z}_{i+1} = W_i z_i + b_i, \text{for } i = 1, \ldots, k+1 \tag{7a}$$

$$z_i \leq \frac{u_i(u_i - \hat{z}_i)}{u_i - l_i}, \text{for } i = 1, \ldots, k+1 \tag{7b}$$

$$z_i \geq \hat{z}_i, z_i \geq 0, \text{for } i = 1, \ldots, k+1 \tag{7c}$$

$$z_1 = \theta + \Delta, \hat{s} = \hat{z}_k, ||\Delta||_\infty < \epsilon \tag{7d}$$

3.1.4. Robust Optimization in Online DNS via Dual Neural Network

Based on the upper envelope of $\hat{\mathcal{B}}_\epsilon(x)$, we leverage the idea of robust optimization to the prediction robustness of prediction models, which means that the worst prediction from the $\hat{\mathcal{B}}_\epsilon(x)$ will not lead to misclassification. The above intuition can be further written as the following linear programming in (8).

$$\min_{\hat{z}_k} \quad (\hat{z}_k)_{s^*} - (\hat{z}_k)_{s^{tar}} = c_{s^*}^T \hat{s} \tag{8a}$$

$$s.t. \quad \hat{z}_k \in \hat{\mathcal{B}}_\epsilon(\theta) \tag{8b}$$

$$c = e_{s^*} - e_{s^{tar}} \tag{8c}$$

where $e_{(\cdot)}$ represents the category of the strategies. If the objectives under all perturbed predictions are positive, the prediction model can achieve the provably robust under various scenario perturbation difference.

For all possible predictions in $\hat{\mathcal{B}}_\epsilon(x)$, the corresponding LPs are too many. So we consider the worst prediction by leveraging the dual problem of the original LP from the robust optimization perspective, where dual problems of LPs provide the lower bound of the original LP. The dual problems of multi-layer neural network are formulated as the maximum problem in (9).

$$\max_{\alpha} \quad J_\epsilon(\theta, g_\sigma(c, \alpha)) \tag{9a}$$

$$s.t. \quad \alpha_{i,j} \in [0, 1] \quad \forall i, j \tag{9b}$$

$$\text{where } J_\epsilon(\theta, v) = -\sum_{k=1}^{k-1} v_{i+1}^T b_i - \theta^T \hat{v}_1 - \epsilon ||\hat{v}_1||_1 \tag{9c}$$

$$+ \sum_{i=2}^{k-1} \sum_{j \in \mathcal{L}_i} l_{i,j} [v_{i,j}]_+. \tag{9d}$$

Ref. [11] considers the above dual problems via the dual neural network in $g_\theta(c, \alpha)$ and obtain the corresponding lower bound from the output of the dual neural network. So the corresponding dual neural network is formulated in (10).

$$v_k = -c \tag{10a}$$

$$\hat{v}_i = W_i^T v_{i+1}, \text{for } i = k-1, \ldots, 1 \tag{10b}$$

$$v_{i,j} = \begin{cases} 0 & j \in \mathcal{L}_i^- \\ \hat{v}_{i,j} & j \in \mathcal{L}_i^+ \\ \frac{u_{i,j}}{u_{i,j} - l_{i,j}} [\hat{v}_{i,j}]_+ - \alpha_{i,j} [\hat{v}_{i,j}]_- & j \in \mathcal{L}_i \end{cases} \tag{10c}$$

We note that $\mathcal{L}_i^-$, $\mathcal{L}_i^+$, and $\mathcal{L}_i^+$ refer to the set of activation functions with negative, both positive, or span zero bounds.

3.1.5. Provably Robust Training of Learning Models for Online DNS

Based on the dual neural network for worst prediction, the provably robust loss function is further designed to train neural networks in the provably robust way, which is called the provably robust neural network and adversarial to input perturbation difference. The key of the provably robust learning model lies in the provably robust learning. The key of provably robust learning lies in the provably robust learning loss function design. The key of the provably robust loss function lies in that it measures the difference between the worst prediction and the true prediction to achieve the robustness, as shown in (11).

$$\min_{\theta} \sum_{i=1}^{N} \max_{||\Delta||_\infty < \epsilon} \mathcal{L}(f_\theta(\theta_i + \Delta), s_i). \tag{11}$$

The above problem is a typical robust optimization in the min-max form, which is hard to optimize directly. So we consider the dual form of the inner optimization problem in the form of the minimization problem, leading to the minimization of the whole problem. Based on the above intuition, the corresponding dual minimizing problem of inner maximum problem is formulated in (12) as follows:

$$\max_{||\Delta||_\infty \leq \epsilon} \mathcal{L}(f_\sigma(\theta + \Delta), s) \leq \mathcal{L}(-J_\epsilon(\theta, g_\sigma(e_s 1^T - I)), s). \tag{12}$$

The dual problem in the minimization form formulates the upper bound of the primal maximization problems, leading to the provably robust training approach with conservative guarantee. So, further, the provably robust loss function is further transformed into (13) with a single layer.

$$\min_{\sigma} \sum_{i=1}^{N} \mathcal{L}(-J_\epsilon(\theta, g_\sigma(e_{s_i} 1^T - I)), s_i). \tag{13}$$

The corresponding dual problem results can be formulated by the dual neural networks of (10). So (13) can be minimized via the current stochastic gradient descent algorithms from the state-of-the-art autograd optimizer. We leverage the dual neural networks to achieve the minimization of provably robust loss by tuning the corresponding neural network parameters. The trained neural networks can predict the integer variable values for online DNS in a provably robust way. Based on the above, the total training process can be summarized by the following process: (1) generate proper dataset with mapping from input scenarios to optimal integer variables; (2) separate the total dataset into the training and testing part; (3) sample mini batches from the total dataset, calculate the robust loss function of (13), derive the corresponding gradients to update the parameters of neural networks; (4) training until convergence; and (5) evaluate the performance of learning models in testing to verify the generality of the proposed approach. The following case study verifies the effectiveness of the proposed approach for online DNS from acceleration and economy perspectives.

4. Case Study

We conduct a case study to verify the effectiveness of the online DNS by simulation on IEEE 123-bus systems from the perspectives of training analysis and online performance analysis. All models are written by Python 3.8; in particular, for the learning part, the neural networks are coded by the Pytorch package to formulate the machine learning models and robust loss function; and, for the optimization part, the corresponding P-DNS models are coded by the Cvxpy package to formulate the distribution network scheduling models. The total learning and optimization parts construct the total online distribution network scheduling models. The organization of this section is composed of the following parts: data sampling, simulation analysis, and discussion.

4.1. Data Sampling

Data are the basis of learning models, and the data sampling technique is the basic technique to learn the proper mapping relationship. For online DNS, we propose to learn the mapping relationship from the input scenarios to the values of integer variables in DNS models. So the uncertain temporal load profile is taken as the parameter of (2) as "X", and the corresponding MICP of DNS is solved to obtain the values of integer variables, as "Y". On top of that, we combine the "X" and "Y" to generate offline training dataset for learning models for the training process.

We first formulate the base load profile as $D_{base} \in R^{N_b \times T}$; then, we generate the corresponding whole dataset from the uniform distribution by (14) from the lower bound $0.6d_{base,ij}$ and higher bound $1.4d_{base,ij}$. The whole dataset can be further divided into training and testing datasets.

$$\{D|d_{ij} \sim U(0.6d_{base,ij}, 1.4d_{base,ij}), i \in [N_b]j \in [T]\}, \tag{14}$$

where $[N_b]$ is the set of power system buses; $[T]$ is the set of scheduling time, which could be set as different values for different scheduling types, such as intraday look-ahead scheduling and day-ahead scheduling.

4.2. Experiment Results

We verify the effectiveness of the PR-L2O for online DNS on the case 123-bus system from the training analysis and solution analysis perspectives in this part. The corresponding renewable data can refer to [21].

The proposed provably robust loss (13) is compared with the conventional cross-entropy loss in (15) to verify its robustness.

$$\mathcal{L}^{ce}(\theta, s) = -\frac{1}{M} \sum_{k=1}^{K} \sum_{m=1}^{M} s_m^k \log(f_\sigma(x_m)) \tag{15}$$

We note that M is the number of training samples, K is the number of possible classes, and s_m^k is the true policy for m instance for k class. We denote the model under (15) as the learn-to-optimize (L2O) model.

4.2.1. Training Analysis

After data processing, Figure 4 compares the changing of robust loss under PR-L2O and L2O; Figure 5 further compares the cross-entropy (CE) loss under PR-L2O and L2O.

Though with higher CE loss, the proposed robust-based learning features high CE. In contrast, though with lower CE, the conventional learning method feature higher robust loss especially with training number growth. This demonstrates the vulnerability of the conventional learning method, which means the prediction results from conventional learning could make the wrong prediction from the minor difference between the training instances and testing instances. This vulnerability could further lead to the suboptimality of reduced models, and the following part further verifies the deficiency.

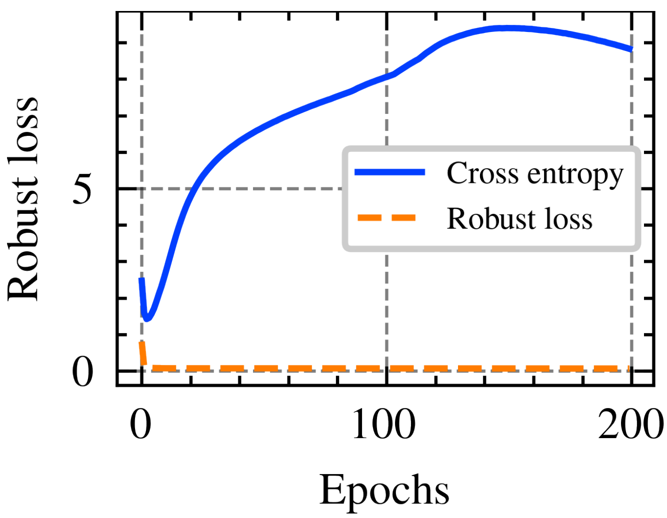

Figure 4. Solving time box-plot under different methods in the 123-bus system.

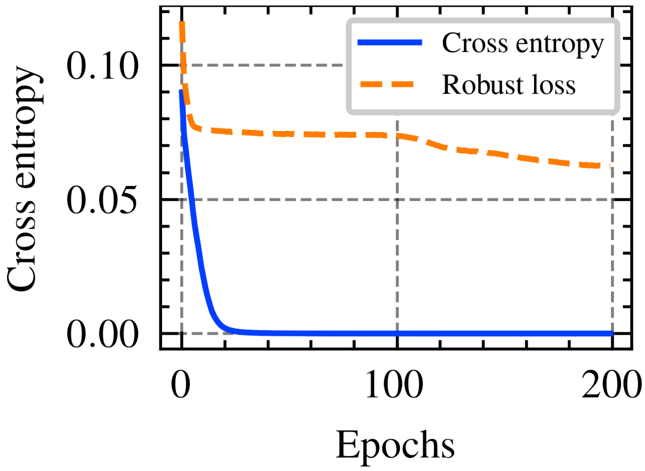

Figure 5. Solving time box-plot under different methods in the 123-bus system.

4.2.2. Online DNS Performance Analysis

After training, the learning models can predict the hard integer variables of P-DNS, and the optimal decisions can be obtained from the reduced tractable convex problems. For the solution time perspective, we compare the solution time of PR-L2O, L2O and B&B. B&B means the solution of complex mixed-integer programming via commercial solvers directly in the following Figure 6.

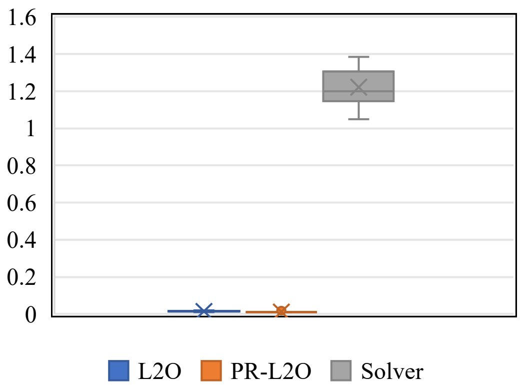

Figure 6. Box-plot of solution time for B&B, L2O and PR-L2O in the 123-bus system.

PR-L2O and L2O features less and more stable solution time, compared with B&B, because B&B is a search algorithm in essence.

After the figure analysis, the mean, variance and maximum solving time are further analyzed in Table 1 in the case 123-bus system.

Table 1. Solution time comparison of B&B, L2O and PR-L2O in the 123-bus system.

Models	B&B	L2O	PR-L2O
Mean	1.207 s	0.0149 s	0.0110 s
Maximum time	1.383 s	0.0182 s	0.0179 s

The learning-based methods achieve 77.3 times faster maximum solving time and 109.7 times faster mean solving time than B&B, which satisfies the requirement of online DNS.

The solution optimality of the above three approaches is analyzed in Table 2. This comparison takes the B&B as the benchmark with zero gaps, which leverages the metrics of the mean and maximum gap for numerical analysis. L2O achieves the mean gap of 0.141%, the maximum gap of 1.87% and average cost of $224.675; the proposed PR-L2O achieves the mean gap of 0.110%, the maximum gap of 0.72%, and average cost of $224.352. From Table 2, PR-L2O achieves a lower mean and maximum gap than L2O. Obviously, the prediction performance of PR-L2O features high benefits with the consideration of provable robustness. More robustness prediction leads to lower operation of online operation. The above demonstrate the generalization performance of the PR-L2O and proper policy identification by considering the input perturbation difference for online regulation. Traditional methods solve the complex mixed-integer program of DNS directly; and in comparison, the proposed method utilizes the learning models to identify the values of integer variables in MIP-based DNS and solve the reduced easy linear programming problem. The proposed approach achieves a high solution speed, verifying the effectiveness of the proposed learn-to-optimize approach.

Table 2. Performance comparison of B&B, L2O and PR-L2O in the 123-bus system.

	B&B	**L2O**	**PR-L2O**
Mean gap	0%	0.141%	0.110%
Maximum gap	0%	1.87%	0.72%
Operation cost	$225.610	$224.675	$224.352

4.3. Discussion

Based on the above analysis, we have two key results: (1) Predicting the integer variables can accelerate the solving process of the distribution network scheduling by reducing the complex MICP to simple CP, and the solution time is proper for online setting. The results conform to our assumptions. (2) Considering the vulnerability of machine learning models, leveraging the robust loss function in training learning models can enhance the economy of online DNS performances, because the learning models can make more proper decisions by considering the difference between the training scenarios and testing scenarios. The results also conform to our assumptions. In summary, the proposed PR-L2O approach can achieve the solution acceleration and enhance the benefits of L2O due to data drift between training and testing dataset.

5. Conclusions

To achieve the online DNS, this paper proposes the PR-L2O approach to learn and predict the integer variables of the MIP-based DNS problem. To address the vulnerability of the learning model, the PR-DNS training approach is leveraged to train the policy identifier, which is adversarial to the input difference perturbation in a provably robust way. The case study verifies the better performance in the test dataset and acceleration performance of PR-L2O. Future work may consider the further exploration and exploitation of history solution results for enhancing future model solutions in various ways.

Author Contributions: Conceptualization, Z.Y. and Y.X.; methodology, N.W. and X.C.; software, L.S. and T.Z.; investigation, X.C.; writing—review and editing, N.W. and X.C.; visualization, N.W.; supervision, Z.Y. and Y.X.; project administration, Y.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Science and Technology Project of China Southern Power Grid, grant number 036000KK52220032 (GDKJXM20220658).

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: Authors Naixiao Wang, Xinlei Cai were employed by the company Electric Power Dispatching and Control Center of Guangdong Power Grid Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Nomenclature

The main notations are provided below. Furthermore, other additional symbols are defined in the paper when needed.

A. Abbreviation

DNS	Distribution network scheduling;
MIP	Mixed-integer programming;
P-DNS	Parametric distribution network scheduling;
LP	Linear programming;
ACOPF	Alternate current optimal power flow;
PR-L2O	Provably robust learn-to-optimize;
L2O	Learn-to-optimize;
MICP	Mixed-integer convex program;

B. Sets

$t \in \mathcal{T}$	Set of the time periods;
$ij \in \mathcal{N_B}$	Set of the system branches;
$i \in \mathcal{N_{PV}}$	Set of the system PV;
$i \in \mathcal{N_{CB}}$	Set of the system CB;
$i \in \mathcal{N_{SVR}}$	Set of the system SVR;
$h \in \mathcal{H}(i)$	Set of the parent buses of bus i;
$j \in \mathcal{J}(i)$	Set of the child buses of bus i.

C. Parameters

S_i^{pv}	PV i capacity;
r_{ij}, x_{ij}	Resistance/reactance of branch ij;
V_0	Base voltage at the substation;
$T_{max}^{(\cdot)}$	Maximal OLTC/CB taps;
$N_{max}^{(\cdot)}$	Allowed maximum switching changing time for the OLTC/CB during the operation period;
q_{tap}	Reactive power supply of per unit CB;
$q_{i,max}^{svr}$, $q_{i,min}^{svr}$	Upper/lower bounds of SVR reactive power supply;
V_{max}, V_{min}	Upper/lower bounds of bus voltage.

D. Variables

$p_{ij,t}^{loss}$	Power loss of branch ij in time t;
$p_{ij,t}$, $q_{ij,t}$	Power and var of branch ij in time t;
$p_{i,t}^{pv}$, $p_{i,t}^{d}$, $q_{i,t}^{d}$	Active PV/load active power/load reactive power;
$V_{i,t}$	Bus voltage magnitude;
$q_{i,t}^{(\cdot)}$	Reactive load/power supply of PV/CB/SVR;
$\sigma_t^{(\cdot),in}$, $\sigma_t^{(\cdot),out}$	Auxiliary binary variables of OLTC/CB.

References

1. Sang, L.; Xu, Y.; Long, H.; Wu, W. Safety-Aware Semi-End-to-End Coordinated Decision Model for Voltage Regulation in Active Distribution Network. *IEEE Trans. Smart Grid* **2023**, *14*, 1814–1826. [CrossRef]
2. Gao, H.; Ma, W.; He, S.; Wang, L.; Liu, J. Time-Segmented Multi-Level Reconfiguration in Distribution Network: A Novel Cloud-Edge Collaboration Framework. *IEEE Trans. Smart Grid* **2022**, *13*, 3319–3322. [CrossRef]
3. Xie, L.; Zheng, X.; Sun, Y.; Huang, T.; Bruton, T. Massively Digitized Power Grid: Opportunities and Challenges of Use-Inspired AI. *Proc. IEEE* **2023**, *111*, 762–787. [CrossRef]
4. Sarma, D.S.; Cupelli, L.; Ponci, F.; Monti, A. Distributed Optimal Power Flow with Data-Driven Sensitivity Computation. In Proceedings of the 2021 IEEE Madrid PowerTech, Madrid, Spain, 28 June–2 July 2021; pp. 1–6.
5. Xavier, Á.S.; Qiu, F.; Wang, F.; Thimmapuram, P.R. Transmission constraint filtering in large-scale distribution network scheduling. *IEEE Trans. Power Syst.* **2019**, *34*, 2457–2460. [CrossRef]
6. Nair, V.; Bartunov, S.; Gimeno, F.; von Glehn, I.; Lichocki, P.; Lobov, I.; O'Donoghue, B.; Sonnerat, N.; Tjandraatmadja, C.; Wan, P.G.; et al. Solving mixed integer programs using neural networks. *arXiv* **2021**, arXiv:2012.13349.
7. Park, S.; Chen, W.; Han, D.; Tanneau, M.; Hentenryck, P.V. Confidence-Aware Graph Neural Networks for Learning Reliability Assessment Commitments. *IEEE Trans. Power Syst.* **2023**, *39*, 3839–3850. [CrossRef]
8. Chen, W.; Park, S.; Tanneau, M.; Hentenryck, P.V. Learning optimization proxies for large-scale Security-Constrained Economic Dispatch. *Electr. Power Syst. Res.* **2021**, *213*, 108566. [CrossRef]

9. Sang, L.; Xu, Y.; Wu, W.; Long, H. Online Voltage Regulation of Active Distribution Networks: A Deep Neural Encoding-Decoding Approach. *IEEE Trans. Power Syst.* **2024**, *39*, 4574–4586. [CrossRef]
10. Sang, L.; Xu, Y.; Sun, H. Ensemble Provably Robust Learn-to-Optimize Approach for Security-Constrained Unit Commitment. *IEEE Trans. Power Syst.* **2023**, *38*, 5073–5087. [CrossRef]
11. Wong, E.; Kolter, J.Z. Provable defenses against adversarial examples via the convex outer adversarial polytope. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018.
12. Wong, E.; Schmidt, F.R.; Metzen, J.H.; Kolter, J.Z. Scaling Provable Adversarial Defenses. In Proceedings of the 32nd International Conference on Neural Information Processing Systems, Red Hook, NY, USA, 3–8 December 2018; pp. 8410–8419.
13. Gasse, M.; Chételat, D.; Ferroni, N.; Charlin, L.; Lodi, A. Exact combinatorial optimization with graph convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, Vancouver, BC, Canada, 8–14 December 2019.
14. Kargarian, A.; Fu, Y.; Li, Z. Distributed distribution network scheduling for large-scale power systems. *IEEE Trans. Power Syst.* **2015**, *30*, 1925–1936. [CrossRef]
15. Cauligi, A.; Culbertson, P.; Stellato, B.; Bertsimas, D.; Schwager, M.; Pavone, M. Learning mixed-integer convex optimization strategies for robot planning and control. In Proceedings of the 2020 59th IEEE Conference on Decision and Control (CDC), Jeju, Republic of Korea, 14–18 December 2020; pp. 1698–1705.
16. Ding, J.Y.; Zhang, C.; Shen, L.; Li, S.; Wang, B.; Xu, Y.; Song, L. Accelerating primal solution findings for mixed integer programs based on solution prediction. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 1452–1459. [CrossRef]
17. Wang, J.; Botterud, A.; Bessa, R.; Keko, H.; Carvalho, L.; Issicaba, D.; Sumaili, J.; Miranda, V. Wind power forecasting uncertainty and integer commitment. *Appl. Energy* **2011**, *88*, 4014–4023. [CrossRef]
18. Chen, Y.; Casto, A.; Wang, F.; Wang, Q.; Wang, X.; Wan, J. Improving large scale day-ahead security constrained integer commitment performance. *IEEE Trans. Power Syst.* **2016**, *31*, 4732–4743. [CrossRef]
19. Palmintier, B.S.; Webster, M.D. Heterogeneous integer clustering for efficient operational flexibility modeling. *IEEE Trans. Power Syst.* **2014**, *29*, 1089–1098. [CrossRef]
20. Jabi, M.; Pedersoli, M.; Mitiche, A.; Ayed, I.B. Deep clustering: On the link between discriminative models and K-Means. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 1887–1896. [CrossRef] [PubMed]
21. Renewables. 2023. Available online: https://www.renewables.ninja/ (accessed on 20 September 2023).