

Transfer Reinforcement Learning-Based Optimal Scheduling for Distribution Networks Considering Topological Changes[#]

Jiankai Ling¹, Yuguang Song¹, Zekuan Yu¹, Haiwang Zhong^{1,2}

1 Department of Electrical Engineering, Tsinghua University

2 Sichuan Energy Internet Research Institute
(Corresponding Author: zhonghw@tsinghua.edu.cn)

ABSTRACT

With the large-scale integration of a high proportion of renewable energy into distribution networks, its inherent volatility and randomness introduce challenges such as voltage limit violations and power flow limit violations to distribution network operation. To address these challenges, topology changes are required to optimize power flow distribution, improve voltage quality, and enhance renewable absorption capacity. However, traditional methods that rely on precise physical models struggle to meet real-time scheduling demands, while conventional data-driven approaches lack generalization capability. This paper, based on transfer reinforcement learning, conducts corresponding research on the optimal scheduling problem under distribution network topology changes. Experiments on a modified IEEE 123-bus system demonstrate that compared to traditional methods, the proposed algorithm significantly improves decision-making efficiency across two typical topological change scenarios: network reconfiguration and change in the number of power sources.

Keywords: distribution network, deep transfer reinforcement learning, optimal scheduling, topology changes, renewable energy

NOMENCLATURE

Abbreviations

DDPG	Deep Deterministic Policy Gradient
DQN	Deep Q-Network
DG	Distributed Generator
WT	Wind Turbine
PV	Photovoltaic
ESS	Energy Storage System

Symbols

A_t	The decision of the agent at time t
-------	-------------------------------------

B_{ij}	Susceptance of line (i,j)
C_{DG}	DG output cost
$C_{balance}$	Balancing unit output cost
C_{ESS}	ESS output cost
C_{loss}	Network loss cost
C_t	Electricity price at the balancing node at time t
$E_{ESS,t-1}$	State of charge of ESS at time t-1
G_{ij}	Conductance of line (i,j)
\dot{I}_{ij}	Current flow from node i to node j
$\Delta I_{up,i}$	Current limit relaxation variable
n_{DG}	Number of DG
$P_{i,ESS}$	Output of ESS
P_{ch}	ESS charging power
P_{dis}	ESS discharging power
P_{ij}	Active power from node i to node j
Q_{ij}	Reactive power from node i to node j
R_{ij}	Resistance of line (i,j)
\dot{S}_{ij}	Complex power flow from node i to node j
$\Delta U_{down,i}$	Lower voltage limit relaxation variable at node i
W_i	Node voltage relaxation variable
\dot{W}_{ij}	Line voltage coupling relaxation variable
η_{ch}	ESS charging efficiency
η_{dis}	ESS discharging efficiency

1. INTRODUCTION

With the depletion of fossil fuels and the advancement of a new round of technological revolution, China has proposed the "Dual Carbon" goals. Under these goals, distribution networks are transforming from traditional passive systems for distributing electricity into new systems with active distribution networks. This transformation is accompanied by the large-scale integration of a high proportion of renewable energy and distributed energy devices, leading to a flexible and variable grid topology and significantly increased source-load uncertainty. Traditional methods based on mechanistic models, such as an implicit Z-bus power flow calculation method proposed by reference [1] and the distflow model proposed by reference [2], rely on precise parametric modeling and struggle to cope with real-time fluctuations. In contrast, artificial intelligence algorithms like reinforcement learning do not require precise model formulation, offer fast online decision-making, and can handle large-scale controllable resources. Therefore, reinforcement learning can serve as one of the solutions. Some scholars have demonstrated the engineering potential of this method in improving renewable energy integration capacity and alleviating peak-shaving pressure through the DDPG algorithm[3]. Reference [4] proposed a federated learning based deep reinforcement learning framework to address the coupling problems in distribution networks. Reference [5] also proposed a multi-agent reinforcement learning approach to solve the optimal scheduling problem coordinated with source-grid-load-storage resources. Some scholars have also made improvements to reinforcement learning to adapt to topology changes. Reference [6] addressed the voltage regulation problem under multiple topologies by integrating clustering with multi-agent reinforcement learning. Reference [7] integrated graph learning with reinforcement learning to address open-loop tertiary voltage control and proposed an H2MGNODE framework for handling topological variations. However, as the types of topological changes continue to increase, relying solely on reinforcement learning exhibits poor adaptability when facing scenarios with topology changes in distribution networks. To overcome this limitation, transfer learning has been introduced as a promising technique domain to enhance learning performance in a related but different target domain. Reference [8] effectively addressed the adaptability issue of online dynamic assessment models to unlearned faults through transfer learning. Reference [9] solved the power forecasting problem for

photovoltaic power stations lacking historical data by combining transfer learning with Mixup. Based on this, this paper proposed a topology-adaptive Scheduling Framework based on transfer reinforcement learning. There are three contributions.

The first one is that we build a relaxation-based simulation environment. Constructing a multi-period power flow model for distribution networks, quantifying decision infeasibility through relaxation variables to solve the sparse reward problem in the early training stage.

The second one is that we give an adaptive scaling transfer mechanism. Designing a shared layer and adaptive layer, adopting an adaptive scaling network to automatically match the dimensions of topological changes.

The third one is that we give an offline-online collaborative architecture. Achieving rapid response to topological change scenarios through offline pre-training of a base model and online fine-tuning.

The structure of the paper is as follows: Section 2 provides the simulation environment model, the transfer reinforcement learning model, and the offline-online collaborative architecture; Section 3 validates the performance of the method in two different scenarios; and Section 4 offers conclusions.

2. SIMULATION ENVIRONMENT AND DEEP TRANSFER REINFORCEMENT LEARNING MODEL

2.1 Simulation Environment

There are two primary requirements for the simulation environment of transfer reinforcement learning. First, the simulation environment needs to be able to quantify the quality of decisions. Traditional distribution network models determine power flow convergence and provide the complete power flow solution upon successful convergence. However, in cases of non-convergence, they typically fail to provide a detailed power flow distribution or quantify the degree of divergence. For convergent cases, they can provide the corresponding power flow distribution; however, for non-convergent cases, they cannot quantify the degree of non-convergence. Second, the environment needs to be generalizable. Transfer reinforcement learning deals with multi-topology environments; therefore, the simulation environment must be able to compute power flow under different topologies and needs to feedback the current grid topology to the agent to facilitate the learning and selection of the transfer part of the agent. Consequently, this paper builds upon the traditional

optimal power flow calculation model for distribution networks provided in reference [10] and reference [11], makes corresponding improvements, and presents a simulation environment adapted for transfer reinforcement learning.

The constraints considered in this model primarily include power flow constraints, voltage and current constraints, and balancing unit output constraints.

$$\begin{cases} \dot{S}_{ij} = \dot{V}_i \dot{I}_{ij}^H \\ \sum \dot{S}_{ij} = (P_{Gi} - P_{Di}) + i(Q_{Gi} - Q_{Di}) \end{cases} \quad (1)$$

Constraints (1) are power flow constraint. Since constraint (1) is nonlinear, to facilitate solving, voltage relaxation variables and second-order cone relaxation are introduced.

$$\begin{cases} \sum P_{ij} = P_{Gi} - P_{Di} \\ \sum Q_{ij} = Q_{Gi} - Q_{Di} \\ P_{ij} = G_{ij}(W_i - a_{ij}) + B_{ij}b_{ij} \\ P_{ji} = G_{ij}(W_j - a_{ij}) + B_{ij}b_{ij} \\ Q_{ij} = -G_{ij}b_{ij} + B_{ij}(W_i - a_{ij}) \\ Q_{ji} = G_{ij}b_{ij} + B_{ij}(W_j - a_{ij}) \\ \dot{W}_{ij} = a_{ij} + ib_{ij} \\ a_{ij}^2 + b_{ij}^2 = W_i W_j \end{cases} \quad (2)$$

The relaxed constraints become (2). The relationship between the relaxation variables is shown in (3).

$$\begin{cases} (U_i^{\min})^2 - \Delta U_{down,i} \leq W_i \leq (U_i^{\max})^2 + \Delta U_{up,i} \\ R_{ij}(I_{ij}^{\min})^2 \leq P_{ij} + P_{ji} \leq R_{ij}(I_{ij}^{\max})^2 + \Delta I_{up,i} \end{cases} \quad (4)$$

Constraints (4) are voltage and current constraints. For the upper and lower voltage limits, relaxation variables $\Delta U_{down,i}$ and $\Delta U_{up,i}$ are added. For the upper current limit, relaxation variable $\Delta I_{up,i}$ is added.

This is because, during the initial learning phase of the agent, the decisions made are often infeasible. Using only convergence/non-convergence as feedback to the agent would lead to sparse rewards, making it difficult for the agent to determine the learning direction. Therefore, introducing relaxation variables allows the agent to receive sufficient environmental feedback even when decisions lead to non-convergence.

$$\begin{cases} P_{Gi}^{\min} \leq P_{Gi} \leq P_{Gi}^{\max} \\ Q_{Gi}^{\min} \leq Q_{Gi} \leq Q_{Gi}^{\max} \end{cases} \quad i \in \mathbb{G}_0 \quad (5)$$

Constraints (5) are balancing unit output constraints. Where \mathbb{G}_0 is the set of nodes where the

balancing units are located. The output constraints of other units will be ensured in subsequent sections.

2.2 Reinforcement Learning Model

The deep reinforcement learning algorithm employed in this paper is the Deep Deterministic Policy Gradient (DDPG) algorithm. Three main advantages are possessed by DDPG algorithm when compared to algorithms such as DQN. First, DDPG can directly output continuous action values. Second, the use of target networks and soft updates of network parameters reduces Q-value fluctuations and stabilizes the learning process. Finally, the experience replay mechanism significantly improves data utilization efficiency.

2.2.1 Action Space

In the optimal scheduling problem for distribution networks studied here, power sources include balancing units, distributed generators (DG), renewable energy units (specifically Wind Turbines - WT, and Photovoltaics - PV), and energy storage systems (ESS). Therefore, the action space can be represented as (6).

$$A_t = [A_{DG,t}, A_{WT,t}, A_{PV,t}, A_{ESS,t}] \quad (6)$$

Regarding the balancing unit, which is utilized to balance the mismatch between total generation and load pursuant to common engineering practices, and the agent does not make extra decisions for it; the learning of its decision optimality will be further designed within the reward function. Furthermore, to improve learning efficiency and reduce the complexity of the reward function, the activation function between the last two layers of the Actor network is designed as hyperbolic tangent function, embedding the decision output limits as hard constraints within the network. Thus, the relationship between the network decision value and the physical value is (7).

$$A_{i,ph,t} = \frac{A_{i,t}(A_{i,\max} - A_{i,\min}) + (A_{i,\max} + A_{i,\min})}{2} \quad (7)$$

$$i \in \{DG, WT, PV, ESS\}$$

Where $A_{i,ph,t}$ is the physical power output decision value, $A_{i,t}$ is the neural network decision value. Two specific clarifications are requisite. For renewable energy sources, $A_{i,\max}$ and $A_{i,\min}$ correspond to the maximum and minimum values of their forecasted power output. For energy storage, adopting separate decision variables for both charging and discharging power would necessitate the constraints shown in (8).

$$\begin{cases} P_{ch} \geq \frac{1}{\eta_{ch}} (E_{ESS,\min} - E_{ESS,t-1} + \frac{P_{dis}}{\eta_{dis}}) \\ P_{ch} \leq \frac{1}{\eta_{ch}} (E_{ESS,\max} - E_{ESS,t-1} + \frac{P_{dis}}{\eta_{dis}}) \end{cases} \quad (8)$$

It can be seen that the upper and lower limits for charging and discharging power are coupled, making it impossible to determine their respective limits at any given time. Additionally, if charging and discharging are decided separately for the ESS, a penalty related to the current state of charge (SOC) of the ESS would need to be introduced in the subsequent penalty function, complicating the design of the reward function. Therefore, these two decision variables are merged (hereinafter referred to as the ESS output). Its upper and lower limit expressions are (9).

$$\begin{cases} P_{t,ESS\min} = -\min(P_{ESS\min}, E_{t-1,ESS\min}) \\ P_{t,ESS\max} = \min(P_{ESS\max}, E_{t,ESS\max} - E_{t-1,ESS}) \end{cases} \quad (9)$$

2.2.2 State Space

The agent needs to make decisions based on environmental information, so the state space should include relevant information beneficial for the agent's decision-making. The state space designed in this paper is as follows.

$$S_t = [P_{Dt}, Q_{Dt}, P_{re,t}, E_{ESS,t}, C_t] \quad (10)$$

It includes node load, forecasted output of renewable energy, energy storage capacity and electricity price at the balancing node, respectively. Note that the dimensions and magnitudes of these state variables are not entirely consistent; therefore, normalization is required. The normalization range refers to the output range of the action space $[-1, 1]$, which is beneficial for training stability.

2.2.3 Reward Function

The reward function guides the learning process of the agent. It mainly consists of two parts: power generation cost and constraint violation penalty.

The cost function (11) represent the costs of distributed generators, balancing units, energy storage, and network loss respectively. Since the training objective of DDPG is to maximize the cumulative discounted reward, a negative sign must be added in front of the cost in the final reward function.

$$\begin{cases} C_{DG} = \sum_{i=1}^{n_{DG}} a_{i,DG} (P_{i,DG})^2 + b_{i,DG} P_{i,DG} + c_{i,DG} \\ C_{balance} = \sum_{i=1} C_i P_{Gi} \quad i \in \mathbb{G}_0 \\ C_{ESS} = \sum_{i=1}^{n_{ESS}} c_{i,ESS} |P_{i,ESS}| \\ C_{loss} = \sum_{i,j} c_{i,loss} P_{ij} \end{cases} \quad (11)$$

The penalty function (12) is composed of power flow violation penalty, node voltage penalty, renewable energy curtailment penalty, and energy storage terminal state penalty.

$$\begin{cases} r_{t,pen1} = w_1 \sum_{i=1}^{n_l} \max(P_{i,t,line} - P_{i,line,max}, 0) \\ r_{t,pen2} = w_2 \sum_{i=1}^{n_B} \max(U_{i,t} - U_{i,max}, 0) + \max(U_{i,min} - U_{i,t}, 0) \\ r_{t,pen3} = w_3 (\sum_{i=1}^{n_{PV}} P_{i,PV,max} - P_{i,PV,t} + \sum_{i=1}^{n_{WT}} P_{i,WT,max} - P_{i,WT,t}) \\ r_{T,pen4} = w_4 \sum_{i=1}^{n_{ESS}} \min(P_{i,ESS,T} - P_{i,ESS,1}, 0) \end{cases} \quad (12)$$

2.3 Transfer Learning Model

Based on the transfer method, transfer learning can be mainly categorized into four types: instance-based, feature-based, relation-based, and model-based transfer [12]. For the problem studied in this paper, the source domain corresponds to the base-case operating scenario, and the target domain corresponds to operating scenarios with topological changes. The task for both is to provide optimal scheduling for the distribution network under the current topology. Based on this, the transfer method adopted in this paper is model-based transfer learning.

2.3.1 Shared Parameter Model

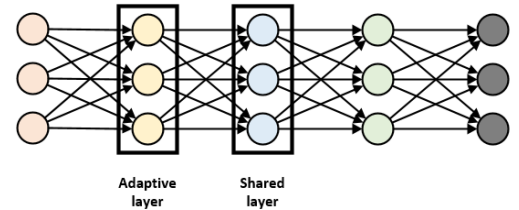


Fig. 1 Neural Network

The core of model-based transfer learning lies in the sharing of network parameters. In the problem studied here, since the task objective remains the same and the main changes involve the input and output dimensions,

we divide the neural network into two types of layers: adaptive layers and shared layers, as shown in Fig. 1. For the shared layers, which inherit the knowledge learned during source domain training, their parameters will be frozen in subsequent training to preserve their original knowledge to the greatest extent. The design of the adaptive layers is to adapt to the new scenarios after transfer, requiring a certain degree of initialization and fine-tuning training.

2.3.2 Adaptive Layers Based on Flexible Scaling Network

When faults in the distribution network cause node disconnection, or when distributed renewable energy sources are temporarily connected or disconnected (topological changes), the dimensions of the input and output of the corresponding agent's neural network will change.

According to the calculation formula of the neural network's forward propagation algorithm, the connection relationship between neurons can be represented as (13).

$$y = wx + b \quad (13)$$

Where y is the output of the neuron in the input layer; x is the input to the neuron in the input layer; w and b are the weight and bias. If represented using matrices, after the dimensions of the input and output layers change, the connection relationship can be expressed as (14).

$$\begin{bmatrix} y_{src} & y_{tgt} \end{bmatrix} = \begin{bmatrix} w_{src} & w_{tgt} \end{bmatrix} \begin{bmatrix} x_{src} \\ x_{tgt} \end{bmatrix} + \begin{bmatrix} b_{src} & b_{tgt} \end{bmatrix} \quad (14)$$

Where parameters with the subscript src represent the common parameters before and after the change; parameters with the subscript tgt represent the parameters of the changed part.

2.4 Deep Transfer Reinforcement Learning Architecture

The deep transfer reinforcement learning architecture adopts an offline-online collaborative. In offline operation, for each specific operating scenario within the multi-scenario set, the transfer reinforcement learning algorithm is applied for transfer. Then, a small amount of training is required until convergence is achieved. After convergence, the new network parameters obtained are the parameters for the transferred scheduling agent.

In online operation, when a new scenario arrives, it is first checked whether an agent for this scenario exists in the transferred agent set. If it exists, it can be directly

invoked for decision-making. If it does not exist, the transfer reinforcement learning method can be used online to perform the transfer, and this new agent is added to the agent set.

3. CASE STUDIES

This paper utilizes a modified IEEE 123-bus system case for analysis, the topology of which is shown in Fig.2. We consider two typical types of topological changes and their specific manifestations in the test case. The first one is network reconfiguration. The switch between Node 62 and Node 60 is opened, while the switch between Node 62 and Node 68 is closed. The Second one is the change in the number of power sources. A distributed wind turbine is integrated at Node 93.

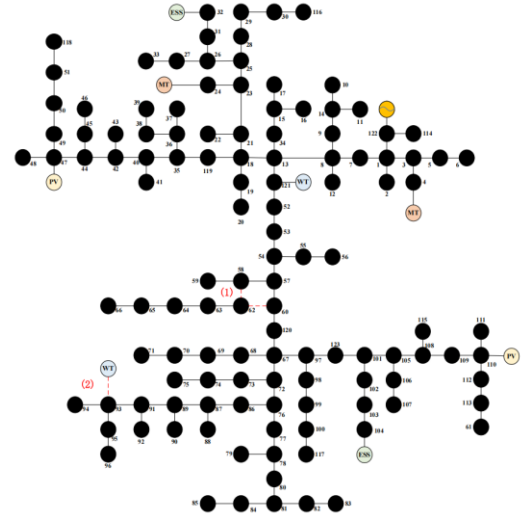


Fig. 2 Topology of a modified IEEE 123-bus system

In this comparative training, the performance of the transferred agent was compared against an agent without transfer learning, which used the original parameters directly. First, for scenario (1), Fig. 3 shows that the reward values of the two agents were essentially the same after convergence. However, the agent without transfer learning required an additional 250 episodes to achieve convergence compared to the transferred agent.

Second, for scenario (2), as is shown in Fig. 4, the results indicate that as the degree of topological change increased, the decisions made by the transferred agent remained highly stable. In contrast, the agent undergoing retraining from scratch could not guarantee convergence within the limited number of training episodes. We further compared the decided output of PV at Node 47 in the 800th episode in scenario (2), as shown in Fig. 5. The results demonstrate that retraining

within a limited timeframe failed to yield satisfactory dispatch decisions.

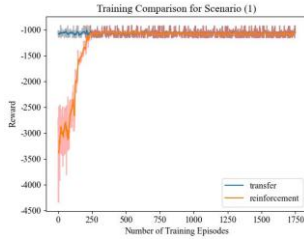


Fig. 3 Training Comparison for Scenario (1)

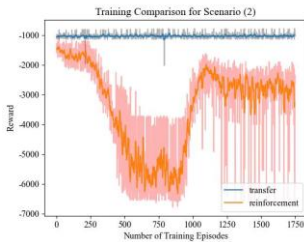


Fig. 4 Training Comparison for Scenario (2)

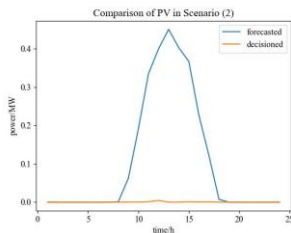


Fig. 5 the output of PV in Scenario (2)

4. CONCLUSION

This paper investigates the optimal scheduling problem in distribution networks based on transfer reinforcement learning. Regarding the simulation environment, relaxation variables were introduced upon the traditional model to quantify the degree of decision infeasibility. Building upon the traditional DDPG algorithm, a shared parameter model was constructed to solve the optimal scheduling problem for distribution networks under multiple scenarios. The proposed algorithm was validated on a modified IEEE 123-bus system. The results demonstrate that, compared to conventional data-driven methods, the proposed algorithm can effectively enhance the decision-making efficiency of the agent in new scenarios while ensuring the same final convergence performance, and make the scheduling decisions more reasonable.

ACKNOWLEDGEMENT

The work was supported by the Natural Science Foundation of China (U24B2077)

REFERENCE

- [1] Tian, P., Jin, Y., Zhang, G., Peng, S., Huang, C., Wang, C., & Xie, N. (2024). An implicit Z-bus-based sequential power flow algorithm for VSC AC/DC systems. *International Journal of Electrical Power & Energy Systems*, 155, 109648.
- [2] Baran, M., & Wu, F. F. (1989). Optimal sizing of capacitors placed on a radial distribution system. *IEEE Transactions on Power Delivery*, 4(1), 735-743
- [3] Li, P., Zhong, H. M., Ma, H. W., Li, J. F., Liu, Y., & Wang, J. H. (2025). Multi-Time Scale Source-Load Storage Cooperative Optimal Control of Active Distribution Network Based on Deep Reinforcement Learning. *Transactions of China Electrotechnical Society*, 40(5), 1487-1502.
- [4] Bahrami, S., Chen, Y. C., & Wong, V. W. S. (2021). Deep Reinforcement Learning for Demand Response in Distribution Networks. *IEEE Transactions on Smart Grid*, 12(2), 1496-1506
- [5] Xu, Y. Y., Yao, L. Z., Liao, S. Y., Cheng, F., Xu, J., Pu, T. J., & Wang, X. Y. (2025). A Real-Time Optimal Scheduling Method for Source-Grid-Load-Storage Based on Multi-Agent Actor-Double-Critic Deep Reinforcement Learning. *Proceedings of the CSEE*, 45(2), 513-527
- [6] Xiang, Y., Lu, Y., & Liu, J. (2023). Deep reinforcement learning based topology-aware voltage regulation of distribution networks with distributed energy storage. *Applied Energy*, 332, 120510.
- [7] Donon, B., Cubelier, F., Karangelos, E., Wehenkel, L., Crochepierre, L., Pache, C., Saludjian, L., & Panciatici, P. (2024). Topology-aware reinforcement learning for tertiary voltage control. *Electric Power Systems Research*, 234, 110658.
- [8] Ren, C., & Xu, Y. (2020). Transfer Learning-Based Power System Online Dynamic Security Assessment: Using One Model to Assess Many Unlearned Faults. *IEEE Transactions on Power Systems*, 35(1), 821-824
- [9] Lu, Y., Wang, G., & Huang, S. (2022). A short-term load forecasting model based on mixup and transfer learning. *Electric Power Systems Research*, 207, 107837
- [10] Farivar, M., & Low, S. H. (2013). Branch Flow Model: Relaxations and Convexification—Part I. *IEEE Transactions on Power Systems*, 28(3), 2554-2564.
- [11] Chen, Y., Yang, W., Chen, Q., Li, H., Xu, H., & Yin, G. (2022, 15-17 Aug. 2022). The Convex-Relaxation-Based Method for Power Flow Analysis. 2022 34th Chinese Control and Decision Conference (CCDC).
- [12] Pan, S. J., & Yang, Q. (2010). A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10), 1345-1359