

Article

Two-Stage Optimization Model Based on Neo4j-Dueling Deep Q Network

Tie Chen ^{1,2} , Pingping Yang ^{1,2,*}, Hongxin Li ^{1,2}, Jiaqi Gao ^{1,2} and Yimin Yuan ^{1,2}

¹ College of Electrical and New Energy, China Three Gorges University, Yichang 443002, China; chent@ctgu.edu.cn (T.C.); 202208580121063@ctgu.edu.cn (H.L.); 202208580121052@ctgu.edu.cn (J.G.); 202208580121302@ctgu.edu.cn (Y.Y.)

² Hubei Provincial Key Laboratory for Operation and Control of Cascaded Hydropower Station, China Three Gorges University, Yichang 443002, China

* Correspondence: 202208580121282@ctgu.edu.cn

Abstract: To alleviate the power flow congestion in active distribution networks (ADNs), this paper proposes a two-stage load transfer optimization model based on Neo4j-Dueling DQN. First, the Neo4j graph model was established as the training environment for Dueling DQN. Meanwhile, the power supply paths from the congestion point to the power source point were obtained using the Cypher language built into Neo4j, forming a load transfer space that served as the action space. Secondly, based on various constraints in the load transfer process, a reward and penalty function was formulated to establish the Dueling DQN training model. Finally, according to the ϵ – greedy action selection strategy, actions were selected from the action space and interacted with the Neo4j environment, resulting in the optimal load transfer operation sequence. In this paper, Python was used as the programming language, TensorFlow open-source software library was used to form a deep reinforcement network, and Py2neo toolkit was used to complete the linkage between the python platform and Neo4j. We conducted experiments on a real 79-node system, using three power flow congestion scenarios for validation. Under the three power flow congestion scenarios, the time required to obtain the results was 2.87 s, 4.37 s and 3.45 s, respectively. For scenario 1 before and after load transfer, the line loss, voltage deviation and line load rate were reduced by about 56.0%, 76.0% and 55.7%, respectively. For scenario 2 before and after load transfer, the line loss, voltage deviation and line load rate were reduced by 41.7%, 72.9% and 56.7%, respectively. For scenario 3 before and after load transfer, the line loss, voltage deviation and line load rate were reduced by 13.6%, 47.1% and 37.7%, respectively. The experimental results show that the trained model can quickly and accurately derive the optimal load transfer operation sequence under different power flow congestion conditions, thereby validating the effectiveness of the proposed model.



Citation: Chen, T.; Yang, P.; Li, H.; Gao, J.; Yuan, Y. Two-Stage Optimization Model Based on Neo4j-Dueling Deep Q Network. *Energies* **2024**, *17*, 4998. <https://doi.org/10.3390/en17194998>

Academic Editor: Claus Leth Bak

Received: 5 September 2024

Revised: 29 September 2024

Accepted: 2 October 2024

Published: 8 October 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

1.1. Background

The widespread integration of distributed generation (DG) and electric vehicles [1–3] has led to frequent power flow congestion [4–7] in the active distribution network (ADN). In power grids at 110 kV and above, grid topology is generally adjusted through load transfer [8–10], while in grids at 35 kV and below, network reconfiguration is employed to alleviate power flow congestion [11–15]. On this basis, several studies [16–18] further incorporated energy storage devices, enhancing the flexibility of load transfer and network reconfiguration. Various studies [19,20] achieved load balancing in ADNs through multiple load transfers and coordinated load transfers between multiple grid levels.

Load transfer can be regarded as an optimization problem of switches by controlling the line switch and feeder switch to adjust the network topology transfer flow. Current

research mainly focuses on voltage levels above 35 kV [21–26]; however, power flow congestion frequently occurs at 10 kV voltage levels. Although the large number of DG access points causes risks of congestion, it also provides additional power support; if the support capacity of the DG can be fully tapped, the congestion can be eliminated by load transfer at 10 kV.

Load transfer requires consideration of a large number of nonlinear constraints. For example, after the load transfer is completed, the ADN must maintain a radial operation mode and satisfy safety constraints [8]. During the load transfer, factors such as operational safety [27], operational costs and the impact of loop closing caused by transfer without power interruption [26] must be carefully considered. It is difficult to solve this problem using conventional methods. For example, the multi-stage optimization method, nonlinear programming method and other mathematical optimization methods [28–30] will produce dimensional issues when the scale of the ADN is too large. The heuristic algorithm [31,32], while useful in some cases, can be difficult to perform large-scale nonlinear calculations, resulting in slow searches of load transfer operation sequences and difficulty in global searches. They may not be well suited for handling a significant number of nonlinear constraints efficiently. Meta-heuristic algorithms [33–35], such as the particle swarm optimization algorithm, gray Wolf algorithm and simulated annealing algorithm, do not converge when there are too many optimization targets and constraints. In addition, DG and load fluctuations will lead to complex scenes, and switch combinations will show explosive growth.

1.2. Contributions

As mentioned above, when the traditional algorithms solve the load transfer problem in the case of power flow congestion, there will be a series of problems such as non-convergence of the algorithm and slow optimization efficiency. Therefore, we urgently need a method to solve such problems quickly and efficiently. Deep reinforcement learning (DRL) optimizes action strategies through trial-and-error interactions between agents and the environment. It has achieved good results in solving sequential decision problems such as power equipment maintenance [36], power supply path optimization [37] and power flow control [38]. It is suitable to solve the problem of load transfer.

DRL needs to build action space. Due to the large number of ADN switch combinations, if all switches are included in the solution space, the solution efficiency will be affected [21,23]. However, load transfer involves only local switches. If the potential switch space is searched first and the number of decision switches is reduced, the action space can be reduced and the computing efficiency can be improved.

In summary, this paper proposes a two-stage optimized load transfer model of Neo4j-Dueling DQN. Firstly, the graph model of ADN is established by using Neo4j, where the power flow data and topology structure of ADN are mapped to the graph data. Subsequently, the interactive environment of Dueling DQN is constructed based on this graph model, and the state space, action space and reward function are designed to complete the training and testing of Dueling DQN. Finally, within the Neo4j-Dueling DQN framework, the two-stage load transfer path optimization is realized. Specific contributions are as follows:

- (1). The diagram model of the ADN was established with Neo4j. The elements, topology and power flow data of ADN were transformed into nodes, relationships and attributes of the graph model. As a result, the graph model accurately reflected the variation rate between the power flow and the topology. The graph model was used to calculate the security constraints of load transfer.
- (2). A search for potential load transfer space was carried out. The load transfer space contains all the load transfer paths that exist under the current operating condition. Regardless of which path is chosen for the load transfer, the ADN can meet the constraints of safe operation.

- (3). The process of load transfer was fully considered. The reward function of Dueling DQN was established based on the safety constraint of load transfer operation. The Dueling DQN agent selects the action switch from the load transfer space, realizes interaction with the Neo4j environment, and obtains the optimal switch operation sequence of load transfer.

2. Model Structure and Framework

As shown in Figure 1, the model includes the Neo4j graph model and Dueling DQN DRL model to realize dynamic interaction between the two models. The Neo4j graph model forms the action space, receives the actions of Dueling DQN, forms and updates the state space. The agent of Dueling DQN selects and executes an action in the action space according to the current state, passes the action to the Neo4j graph model, updates the reward, and optimizes the operation steps. This section introduces the Neo4j graph model and Dueling DQN model in detail.

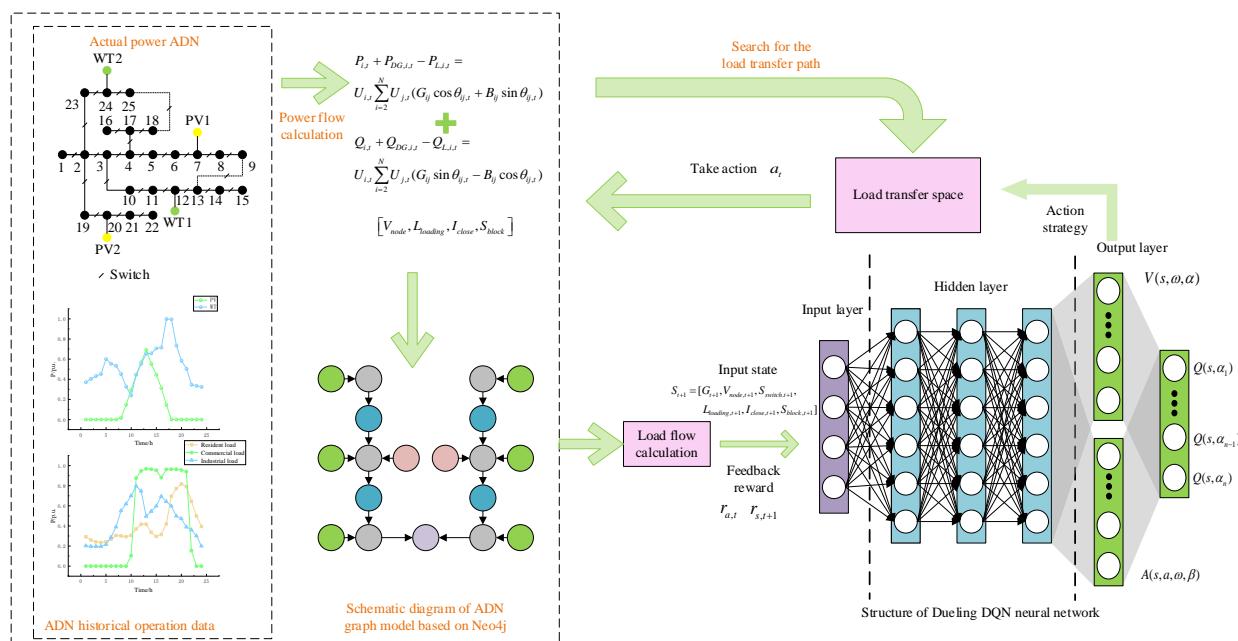


Figure 1. Model structure and framework of Neo4j–Dueling DQN.

2.1. Graph Structure Model Based on Neo4j

E represents the equipment of the ADN, R represents the connection relationship between the devices, and S represents the evaluation index of the ADN. The ADN model $G = (E, R, S)$ was established by using Neo4j [39].

Device $E = \{E_1, E_2, E_3, E_4\}$ refers to four types of nodes: the load node, bus node, switch node and DG node.

The connection relation $R = \{R_1, R_2\}$ represents the edges of the Neo4j model and describes the connection or disconnection between nodes. The R value of connecting switch nodes is determined by the on–off state of the switch, and the Dueling DQN model controls the on–off state of the switch.

The evaluation index $S = \{S_1, S_2, S_3, S_4\}$ represents the attributes of nodes and edges, including the node voltage deviation rate, line load rate, line loss and closing current. Attributes are stored as key-value pairs in nodes and edges.

A power flow calculation model is built into the graph model, and the historical operation data of wind power and photovoltaic and different loads are added to calculate the power flow distribution dynamically.

The power flow data are used to calculate the node attributes. When the switch status changes, the node attributes also change. The mapping is shown in Figure 2.

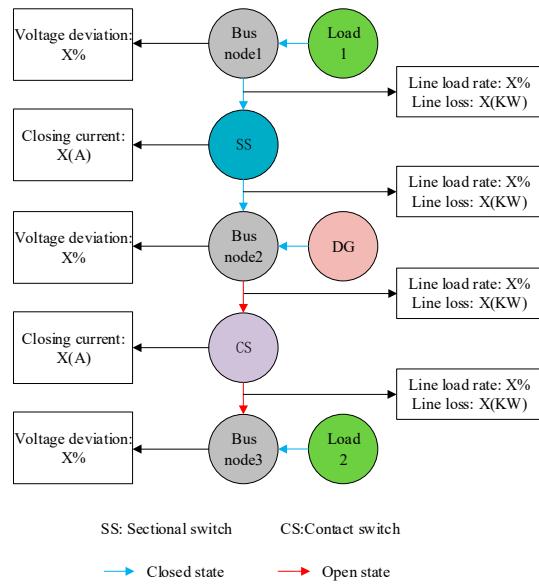


Figure 2. Mapping of Neo4j node attributes.

2.2. Constraints of Load Transfer Space Search

The load transfer space S_A is the set of all load transfer paths that meet the operating conditions of the ADN. The purpose is to form the action space of Dueling DQN and narrow the scope of optimization.

After load transfer, the ADN shall meet the following requirements: ① ensure the radial distribution of the ADN topology to avoid ring network operation; ② ensure that the voltage deviation and line load are within the acceptable level to meet the power flow constraints. Various constraints are as follows:

- (1). Power flow balance constraints:

$$P_{i,t} + P_{DG,i,t} - P_{L,i,t} = U_{i,t} \sum_{i=2}^N U_{j,t} (G_{ij} \cos \theta_{ij,t} + B_{ij} \sin \theta_{ij,t}), \quad (1)$$

$$Q_{i,t} + Q_{DG,i,t} - Q_{L,i,t} = U_{i,t} \sum_{i=2}^N U_{j,t} (G_{ij} \sin \theta_{ij,t} - B_{ij} \cos \theta_{ij,t}) \quad (2)$$

where $P_{i,t}$ and $Q_{i,t}$ are the active and reactive power injected by node i at time t , respectively; $P_{DG,i,t}$ and $Q_{DG,i,t}$ are the active and reactive power injected by DG into node i , respectively. $P_{L,i,t}$ and $Q_{L,i,t}$ are the active and reactive power of node i load, respectively. $U_{i,t}$ and $U_{j,t}$ are the voltage amplitudes of nodes i and j at time t , respectively. G_{ij} , B_{ij} , and $\theta_{ij,t}$ are the conductance, susceptance, and voltage phase angle differences at time t between adjacent nodes, respectively.

- (2). Nodal voltage constraint:

$$U_{i,\min} \leq U_{i,t} \leq U_{i,\max} \quad (3)$$

where $U_{i,\min}$ and $U_{i,\max}$ are the upper and lower limits of node i voltage at time t .

- (3). Line load constraints:

$$S_{k,t} \leq 100\%. \quad (4)$$

where $S_{k,t}$ is the load of line k at time t .

- (4). Topology constraint:

$$g \in G_r \quad (5)$$

where g is the network structure after the load transfer; G_r is the radial network structure.

2.3. Search for Load Transfer Space

The goal of load transfer is to form a new supply path to the power flow congestion point. The Cypher statement built into Neo4j can search all power supply paths from the congestion point to the power supply based on the principle of bi-directional breadth and depth search [39]. It can also calculate whether the load transfer meets the constraints according to the constraints in Section 2.2. If the constraints are met, it is the potential power supply path. Part of the code for the search process is shown in Table 1, and the search principle is shown in Figure 3.

Table 1. Load transfer space search process part of the code.

Algorithm: Power Supply Path Search Algorithm

Input: Position of the congestion node and power node

Output: Power supply path

1 Find the congestion node: match ($m: bus \{name: '%s'\}$) return m , %s Represents the name of the congestion node

2 Find the power node: match ($n: transformer \{name: '%s'\}$) return n , %s Represents the name of the power node

3 Search for the path between the congestion node and the power node:

match path = (m) - [r*..] - (n)

where not (n)-[: connect]->()-[: disconnect]->(j)-[: disconnect]->()-[: connect]->(m), j is the contact switch node

return path

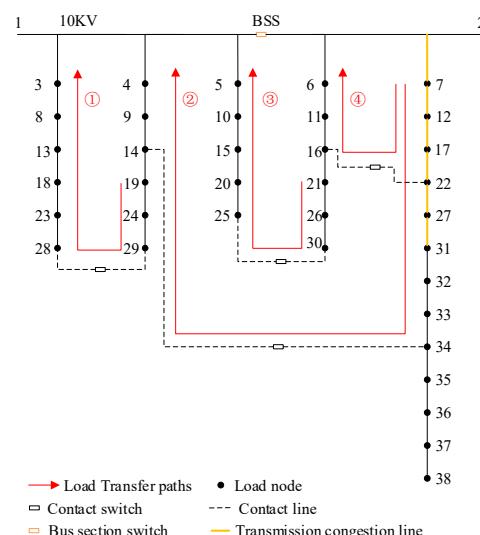


Figure 3. Results of potential power supply path search.

The qualified switch action combination is stored in S_A , and the elements in S_A are de-processed to obtain the final load transfer space S_A .

3. Load Transfer Model Based on Dueling DQN

Reinforcement learning consists of three parts: an agent, environment and reward. The agent interacts with the environment by performing actions, and receives feedback from the environment, which is a reward. By learning the action strategy, the agent maximizes the cumulative reward obtained in the process of interacting with the environment and learns the optimal strategy to maximize the reward in the way of continuous exploration. The exploration process of reinforcement learning can be described by Markov decision processes (MDPs), which can be described by (S, A, R, γ, P) , where S is the set of all environmental states, A is the set of executable actions; R is the set of rewards obtained by the agent after action; γ is the discount factor of future rewards; P is the state transition probability [40].

3.1. Dueling DQN Algorithm

DQN [41] is a classic DRL algorithm. The optimal solution is obtained by calculating the maximum reward of the Q function. Because in different ADN states, the benefits brought by the same switch may be completely different, the state reward and the action reward of the ADN should be calculated separately. Therefore, Dueling DQN is adopted as a solution algorithm. The Q function of Dueling DQN is divided into value function $V(s; \omega, \alpha)$ and advantage function $A(s, a; \omega, \beta)$ [42]. $V(s; \omega, \alpha)$ describes the state reward value after the model state changes, and $A(s, a; \omega, \beta)$ describes the action reward value. The formula is as follows:

$$Q(s, a, \omega, \alpha, \beta) = V(s, \omega, \alpha) + \left(A(s, a, \omega, \beta) - \frac{1}{|A|} \sum_{a' \in A} A(s, a', \omega, \beta) \right) \quad (6)$$

where, ω , α and β are common hidden layer parameters, value function layer parameters and advantage function layer parameters, respectively. A is the set of all actions; a' is the action with the maximum Q value under the state s' ; s' is the next state of the state s . A carries out centralized processing on vector A , highlighting the differences in actions, and reflecting the advantages and disadvantages of each action in a specific state. The Dueling DQN neural network structure is shown in Figure 4. The interaction model between Dueling DQN and Neo4j is shown in Figure 5.

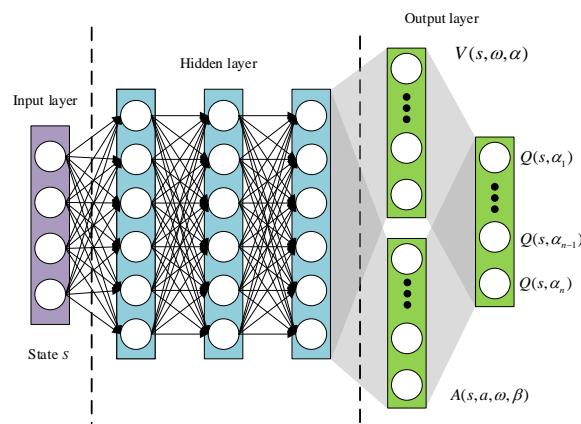


Figure 4. Structure of Dueling DQN neural network.

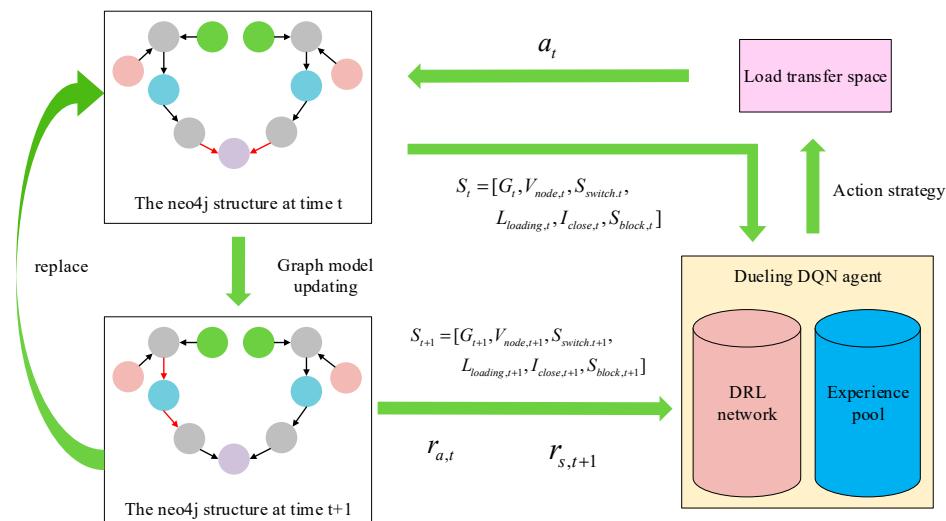


Figure 5. The interaction model between Dueling DQN and Neo4j.

In Figure 5, $r_{a,t}$ represents the reward value obtained by action a occurring at time t ; $r_{s,t+1}$ represents the reward value of the state at time $t + 1$. S_t and S_{t+1} represent the operating state at time t and time $t + 1$, both of which are explained in detail in Section 3.2.

3.2. State Space

As far as possible, the state space should take into account the factors that affect the decision. For the load transfer problem, from a numerical point of view, the line loading rate $L_{loading}$, node voltage V_{node} and closing current I_{close} are the key analysis data. From a spatial point of view, topological structure G , switch state S_{switch} and power flow congestion state S_{block} can be used as the basis for selecting a suitable load transfer path. Therefore, these data are selected to construct state space S , as shown below:

$$S = [G, V_{node}, S_{switch}, L_{loading}, I_{close}, S_{block}]. \quad (7)$$

Based on the description of state space, this paper defines that there are target states, end states and transition states in load transfer. The ‘target state’ indicates that no flow congestion occurs and all constraints are met. The ‘end state’ is a violation of the loop current constraint, or repeated action. ‘Transition state’ is any state other than the above.

$$S_s = \begin{cases} \text{end state} \\ \text{target state} \\ \text{transition state} \end{cases} \quad (8)$$

3.3. Action Space

The action space of this paper is the S_A . Because most of the irrelevant switches are removed when the S_A is constructed, the exploration time of the agent will be shortened, the generation of invalid actions will be reduced, and the convergence speed of the Dueling DQN algorithm will be accelerated.

3.4. Reward Function

In this paper, the ADN operating constraints and economic benefits are considered comprehensively, and the reward function R is set up. R guides the neural network to mine the state information of the ADN and form an action sequence, including the reward part and punishment part.

3.4.1. Reward Part

The main goal is to eliminate power flow congestion. The secondary objectives are to reduce line loss, reduce voltage deviation, balance line load and reduce the number of switch operations. Therefore, the reward part is based on the above five goals, which are as follows:

- (1). The main target reward $R_{state}(r)$:

$$R_{state}(r) = \begin{cases} -2, & S_s = \text{end state} \\ 0.5, & S_s = \text{Transition state} \\ 10, & S_s = \text{Target state} \end{cases} \quad (9)$$

The reason why the reward value of the target state is set to 10 is that the agent will not fall into the small reward of obtaining the transition state while ignoring the big reward of the target state when making action decisions.

- (2). The line loss reward $R_{loss}(r)$: the smaller the line loss, the higher the reward.

$$R_{loss}(r) = m \sum_{i=1}^{N_l} \frac{I_i^2 R_i}{P_i}. \quad (10)$$

where N_l is the total number of lines; $I_i^2 R_i$ is the line loss of line i ; P_i is the active power of line i . m is the line loss coefficient, which keeps the value of R_{loss} at approximately $-1\text{--}0$.

- (3). The voltage deviation reward $R_{volt}(r)$: generally, the voltage offset allowed by ADN nodes is $\pm 5\%$, and the smaller the offset, the higher the reward.

$$V_{i,d} = \frac{V_i^* - V_i}{V_i^*} \quad (11)$$

$$R_{volt}(r) = h \frac{\sum_{i=1}^{N_n} (V_{i,d})^2}{N_n} \quad (12)$$

where $V_{i,d}$ is the voltage deviation of node i ; N_n is the number of nodes; V_i^* is the standard voltage of i node; V_i is the actual voltage of the i node. h is the voltage deviation coefficient, so that the value of $R_{volt}(r)$ can be maintained at approximately $-1\text{--}0$.

- (4). The line loading rate reward $R_{line}(r)$: the more balanced the load of the line, the higher the reward.

$$R_{line}(r) = \frac{\sum_{i=1}^{N_l} (L_i - L_i^{ave})^2}{N_l}, \quad L_i = \frac{I_i}{I_i^*}, \quad L_i^{ave} = \frac{\sum_{i=1}^{N_l} L_i}{N_l} \quad (13)$$

where L_i is the actual load rate of line i , I_i is the actual current of line i , and I_i^* is the rated current of line i . L_i^{ave} indicates the average line loading rate of the ADN.

- (5). The switch operation time reward $R_{sw}(r)$: the lower the number of switches, the higher the reward.

$$R_{sw}(r) = 1 - 2 \frac{A_{sw}}{N_{sw}} \quad (14)$$

where A_{sw} is the total number of switches that operate in this load transfer; N_{sw} is the total number of switches in the load transfer space. The calculated R_{sw} value is $-1\text{--}1$.

The reward part is as follows:

$$R_r = R_{state}(r) + R_{loss}(r) + R_{volt}(r) + R_{line}(r) + R_{sw}(r) \quad (15)$$

3.4.2. Punishment Part

There are many constraints in the process of load transfer. This paper mainly adopts the node voltage constraint, line power flow constraint, topology constraint, closing current constraints, and repetitive action constraint as the basis for setting a penalty function. The details are as follows:

- (1). The voltage deviation penalty $R_{volt}(p)$: a penalty is given if the node voltage exceeds $\pm 5\%$.

$$R_{volt}(p) = \begin{cases} -2, & |V_{i,d}| > 5\% \text{ and } S_s = \text{end state} \\ -1, & |V_{i,d}| > 5\% \text{ and } S_s = \text{Transition state} \\ 0, & |V_{i,d}| \leq 5\% \end{cases} \quad (16)$$

- (2). The line power flow over limit penalty item $R_{line}(p)$: the upper limit of line power flow is 100%, resulting in a penalty if exceeded.

$$R_{line}(p) = \begin{cases} -2, & L_i > 100\% \text{ and } S_s = \text{end state} \\ -0.5, & L_i > 100\% \text{ and } S_s = \text{Transition} \\ 0, & L_i \leq 100\% \end{cases} \quad (17)$$

(3). The topology constraint penalty $R_{loop}(p)$:

$$R_{loop}(p) = \begin{cases} -2, & g \notin G_r \text{ and } S_s = \text{end state} \\ 0, & \text{other} \end{cases} \quad (18)$$

(4). The closing current penalty $R_{close}(p)$: the closing steady-state current must not exceed a fixed-time overcurrent protection setting value and the closing impulse current must not exceed the setting value of the current velocity break protection to avoid a penalty.

$$R_{close}(p) = \begin{cases} 0, & I_M \leq I_{act.I} \\ 0, & I_m \leq I_{act.III} \\ -2, & I_M > I_{act.I} \\ -2, & I_m > I_{act.III} \end{cases} \quad (19)$$

where I_M is the closing impulse current; $I_{act.I}$ is the setting value of the current velocity break protection; I_m is the closing steady-state current; $I_{act.III}$ is the fixed time overcurrent protection setting value.

(5). The switch repeated action penalty $R_{act}(p)$: penalties are given for repeated actions.

$$R_{act}(p) = \begin{cases} -2, & \text{exist} \\ 0, & \text{inexistence} \end{cases} \quad (20)$$

The punishment part is as follows:

$$R_p = R_{volt}(p) + R_{line}(p) + R_{loop}(p) + R_{close}(p) + R_{act}(p) \quad (21)$$

3.5. Action Selection Strategy

Agents of Dueling DQN follow the ϵ – *greedy* strategy when selecting actions [42]. ϵ represents the exploration rate. That is, an action is randomly selected with a certain probability to explore in order to find more states and action combinations, and gradually reduces the exploration rate and increases the utilization rate. This paper divides agent action modes into exploration mode and non-exploration mode. In the non-exploration mode, the action with the highest value is directly selected, and the ϵ – *greedy* strategy is used in the exploration mode. The rules for action selection are shown in Figure 6. The update formula of ϵ is as follows:

$$\epsilon = \epsilon_{start} - \frac{\epsilon_{start} - \epsilon_{min}}{n} \quad (22)$$

where ϵ_{start} is the probability of the agent's initial random choice of action, ϵ_{min} is the minimum probability that the agent randomly chooses an action and n is the number of times the algorithm performs load transfer operations.

As shown in Figure 6, when the action is selected, the first thing to determine is whether the current mode is exploration mode or non-exploration mode. For exploration mode, the ϵ – *greedy* strategy is adopted, which generates a random number p between 0 and 1, and determines the size relationship between p and ϵ . If p is less than or equal to ϵ , an action is selected at random. If p is greater than ϵ , the best action for the current state is selected.

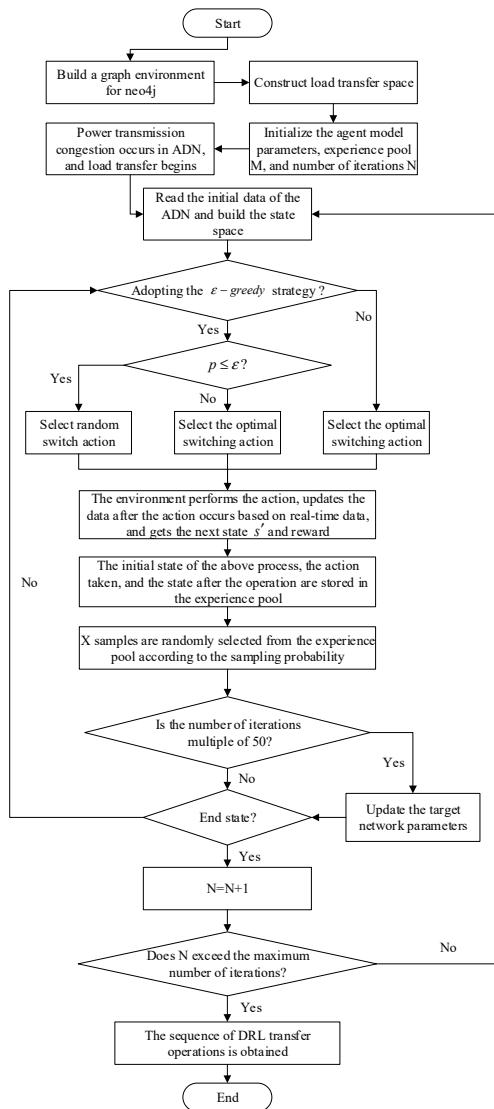


Figure 6. The algorithm process of Neo4j-Dueling DQN.

4. Example Analysis

4.1. Case Preparation

The improved 79-node real system is used for simulations. The specific structure is shown in Figure 7. The ADN consists of two substations, four transformers T1–T4 and nine feeder lines L1–L9. The entire network consists of 75 sectional switches and 10 contact switches, 9 photovoltaic nodes, and 4 wind power nodes. The grid-connected information of DG is shown in Table 2. The load type of each node in the ADN is shown in Table 3.

Table 2. DG grid-connected situation.

DG Type	Grid-Connected Node	Single Node Capacity /MW
Photovoltaic	35, 38, 47, 54, 69	5
	16, 44, 60, 52	8
Wind power	19, 22, 37	5
	50	10

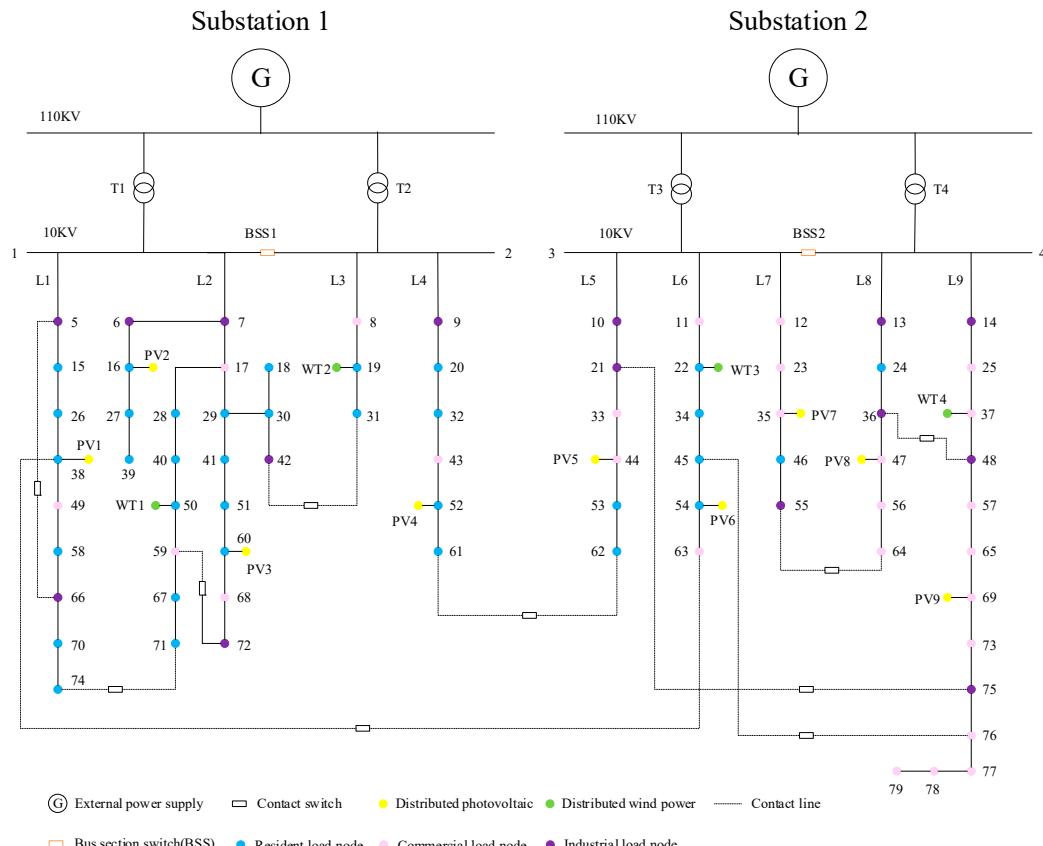


Figure 7. ADN topology diagram.

Table 3. Load type of each node.

Node Type	Node Position
Resident load	15, 16, 18, 19, 20, 22, 24, 26, 27, 28, 29, 30, 31, 32, 34, 38, 39, 40, 41, 45, 46, 50, 51, 52, 53, 54, 58, 60, 61, 62, 67, 70, 71, 74
Commercial load	8, 11, 12, 17, 23, 25, 33, 35, 37, 43, 44, 47, 49, 56, 57, 59, 63, 64, 65, 68, 69, 73, 76, 77, 78, 79
Industrial load	5, 6, 7, 9, 10, 13, 14, 21, 36, 42, 48, 55, 66, 72, 75

4.2. Training Process

The actual 79-node ADN operation data are used as sample data. When the sample data are selected, random data are sampled at time points during the period of power flow congestion, and each sampling time point has the same probability of being selected.

The setting of hyperparameters of the Dueling DQN algorithm is shown in Table 4. The accumulation of reward value for model training is shown in Figure 8.

Table 4. Hyperparameter settings.

Hyperparameter	Value
Learning rate	0.0005
Discount factor	0.95
Exploration rate	1.0
Minimum exploration rate	0.01
Batch quantity/number	128
Experience pool capacity/number	10,000
Target network update frequency/round	50

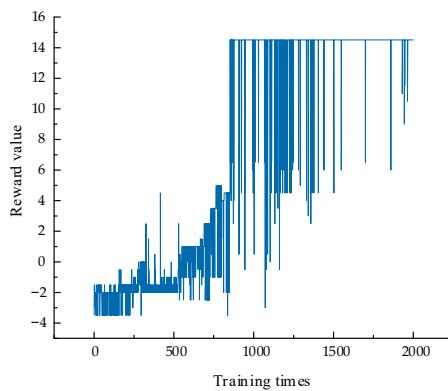


Figure 8. Cumulative reward value results of model training.

Figure 8 shows the result diagram of reward value of Neo4j-Dueling DQN model training. It can be seen from the figure that the maximum reward value is reached after 850 rounds of training. The reward value oscillates because the agent is constantly trying new options to avoid falling into local optimality. However, from the perspective of the overall trend of reward value, the fluctuation becomes smaller and smaller, and the average reward value tends to be stable at last. The effectiveness of the load transfer model proposed in this paper is proved.

4.3. Analysis of Load Transfer Results

In the actual ADN, the operating conditions of one single day were selected to analyze the load transfer results. The output curve of DG is shown in Figure 9. Different types of load demand are shown in Figure 10. The power flow congestion of feeder L1–L9 is shown in Figure 11.

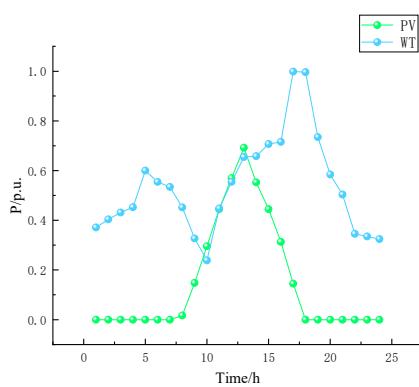


Figure 9. The output curve of DG.

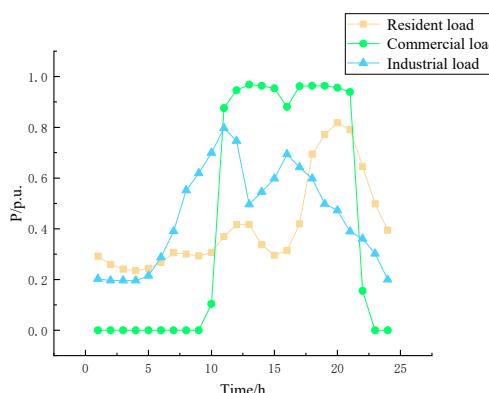


Figure 10. Three types of load demand curves.

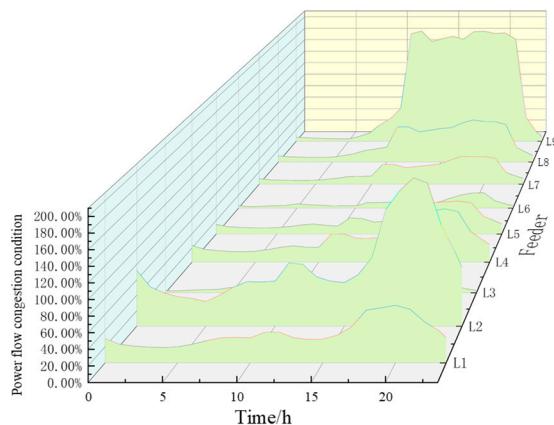


Figure 11. L1-L9 power flow congestion condition.

It can be seen from Figure 11 that power flow congestion mainly occurs on feeder L2 and L9 and occurs between 10:30 and 23:00. Therefore, this paper divides it into three scenarios for analysis.

Scenario 1 (10:30–17:00): Power flow congestion occurs on feeder L9.

Scenario 2 (17:00–21:30): Power flow congestion occurs on feeder L2 and L9.

Scenario 3 (21:30–23:00): Power flow congestion occurs on feeder L2.

When power flow congestion occurs, the model exploration value after training is set to 0, and the action output of the highest value is directly carried out. Therefore, the decision time mainly depends on the number of actions. The load transfer path, search time and load transfer results of the model in different scenarios are shown in Table 5. The comparison of evaluation indicators before and after the load transfer is shown in Table 6.

Table 5. Load transfer path, search time and load transfer results.

Scenario	Load Transfer Space	Search Time/s	Load Transfer Operation Sequence	Decision Time/s	Total Time/s
1	21–33, 33–44, 44–53, 53–62, 36–47, 47–56, 56–64, 25–37, 37–48, 48–57, 57–65, 65–69, 69–73, 73–75, 75–76, 21–75, 45–76, 61–62, 55–64, 36–48	11.51	close45–76; open75–76; close21–75; open69–73; close36–48; open37–48; close55–64; open36–47	2.87	14.38
2	21–33, 33–44, 36–47, 47–56, 56–64, 25–37, 37–48, 48–57, 57–65, 65–69, 69–73, 73–75, 75–76, 21–75, 61–62, 55–64, 36–48, 5–15, 15–26, 26–38, 38–49, 49–58, 58–66, 40–50, 50–59, 59–67, 67–71, 17–29, 41–51, 51–60, 60–68, 68–72, 31–42, 59–72, 5–66, 38–63, 71–74	70.33	close31–42; open17–29; close71–74; open50–59; close45–76; open75–76; close21–75; open69–73; close36–48; open37–48; close55–64; open36–47	4.37	74.70
3	5–15, 15–26, 26–38, 38–49, 49–58, 58–66, 40–50, 50–59, 59–67, 67–71, 7–17, 17–29, 41–51, 51–60, 60–68, 68–72, 31–42, 59–72, 5–66, 38–63, 71–74	51.74	close31–42; open17–29; close71–74; open50–59; close5–66; open15–26; close38–63; open49–58; close59–72; open51–60	3.45	55.19

As can be seen from Table 5, scenario 2 contains the congestion of scenario 1 and scenario 3. The congestion is serious and there are many load transfer paths. Therefore, the load transfer space is larger than scenario 1 and scenario 3, and the corresponding search

time is the longest. The search time of load transfer space in scenario 3 is longer than that in scenario 1, which is caused by complex feeder structures and excessive load transfer paths.

Table 6. Comparison of evaluation indicators before and after the load transfer.

Scenario	Load Transfer Situation	Line Loss (KW)	Voltage Deviation Evaluation	Line Load Rate Evaluation
1	before	1965.4	1.357	0.443
	after	864.4	0.325	0.196
2	before	3444.6	2.782	0.380
	after	2009.2	0.754	0.165
3	before	1605.2	1.400	0.235
	after	1387.4	0.741	0.146

From the load transfer results, it can be observed that scenario 1 takes the shortest decision time, while Scenario 2 takes the longest decision time. However, the time used is seconds, which meets the requirements of online applications.

As shown in Tables 5 and 6, in scenario 1, in order to solve the power flow congestion problem, eight switch operations are required. After load transfer, the line loss, voltage deviation and line load rate are reduced to about 56.0%, 76.0% and 55.7%, respectively, compared with before load transfer. In scenario 2, in order to solve the power flow congestion problem, 12 switch operations are required. Line loss, voltage deviation and line load rate are reduced by 41.7%, 72.9% and 56.7% respectively. In scenario 3, in order to solve the power flow congestion problem, 10 switch operations are required. Line loss, voltage deviation and line load rate are reduced by 13.6%, 47.1% and 37.7%, respectively. The above data show that the power flow congestion problem is solved after the load transfer, and some optimization is carried out compared with that before the load transfer.

Figure 12 shows the changes in some reward indexes in the load transfer process of the three scenarios. It can be seen from the figure that voltage deviation, line load rate, and line loss all show a downward trend in the three scenarios. This means that each step of the load transfer process has been optimized.

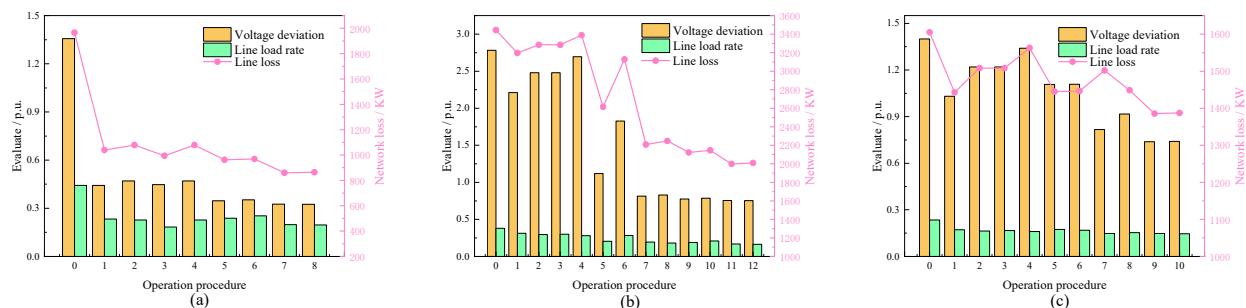


Figure 12. Changes in evaluation indexes in the process of load transfer under three scenarios. (a) Changes in evaluation indicators in scenario 1, (b) changes in evaluation indicators in scenario 2, and (c) changes in evaluation indicators in scenario 3.

Figure 13 shows the closing steady-state current and closing impulse current in the closing process of load transfer under three scenarios. The setting value of current break speed protection is 1460 A. The setting value of fixed-time overcurrent protection is 850 A. It can be seen from the figure that the loop can be closed smoothly in the process of load transfer in the three scenarios, and there is no failure to meet the loop closing conditions. The rationality of the result operation sequence is verified.

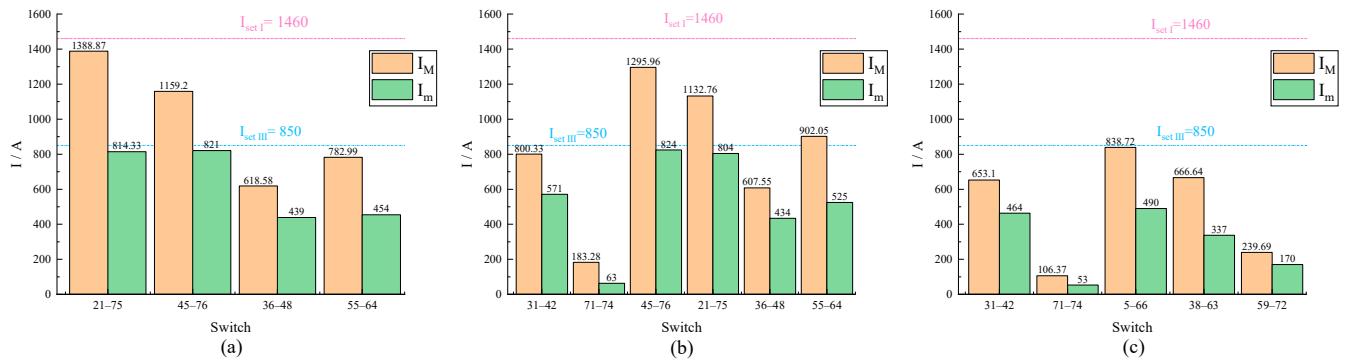


Figure 13. Changes in closing current during load transfer in three scenarios. (a) Change in closing current in scenario 1, (b) change in closing current in scenario 2; (c) change in closing current in scenario 3.

Figure 14 shows the constraint changes in penalty items in the load transfer process, whose values have been normalized. It can be seen from the figure that there is no violation of the large penalty value constraint of loop closing current constraint or the repeated action constraint in the load transfer process of the three scenarios. As for the penalty of voltage deviation, it is mainly reflected in scenario 2 and scenario 3. In both cases, the voltage deviation is guaranteed to be within 5% in the seventh step, and all constraints can be met in the subsequent load transfer operations. The effectiveness of the load transfer strategy in this paper is verified.

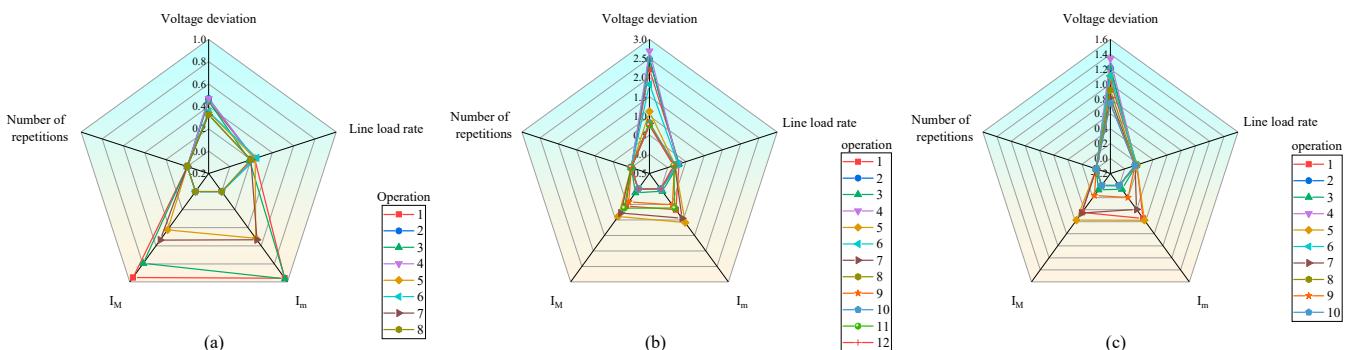


Figure 14. Constraint changes in penalty item during load transfer operation. (a) The change in penalty constraints in scenario 1, (b) the change in penalty constraints in scenario 2, and (c) the change in penalty constraints in scenario 3.

4.4. Comparative Analysis of Training Effect

To verify the advantages of the load transfer method proposed in this paper based on Neo4j-Dueling DQN, we compared it with Double DQN and DQN algorithms. All three algorithms set the same reward function and hyperparameter. The comparison of average reward values in the training process is shown in Figure 15.

Figure 15 shows the comparison results of the training effects of Dueling DQN, Double DQN and DQN algorithms. From the point of view of the maximum reward value acquisition speed, the proposed algorithm is faster than the other two algorithms. From the perspective of the maximum reward value, both the proposed algorithm and Double DQN algorithm can obtain the maximum reward value within limited training times. However, the DQN algorithm cannot obtain the maximum reward value. In summary, the Dueling DQN algorithm used in this paper is superior to the other two algorithms.

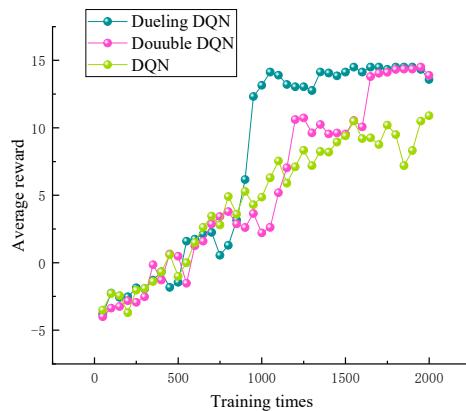


Figure 15. Comparison of the average reward value of the three algorithms.

5. Conclusions and Future Work

5.1. Conclusions

This paper presents a load transfer optimization method based on Neo4j-Dueling DQN. The real-time decision and optimal operation of load transfer in the case of power flow congestion are realized by considering various constraints in the process of load transfer in the ADN. The conclusions of this research are as follows:

- (1). We built ADN diagram models using Neo4j, which reflect the change rate between the power flow of the ADN and the topology. Linkage with Dueling DQN can obtain the action space and state space of Dueling DQN.
- (2). We searched for all the load transfer paths that met the operating constraints of the ADN and formed the load transfer space. This reduced the action space of Dueling DQN and improved the operation efficiency.
- (3). The reward function of Dueling DQN was established based on the safety constraint of load transfer operation. Through the linkage with the Neo4j graph model, the operation steps satisfying both the operation constraint and the state constraint can be obtained, and an online real-time decision can be made.

5.2. Future Work

However, the study still has limitations that warrant further investigation. Specifically, this paper does not discuss the influence of interruptible loads on load transfer operation under power market conditions. Future work should focus on building load transfer models with interruptible loads in order to exert the load regulation effect under power market conditions. In addition, the influence of uncertain factors such as policy changes and environmental changes on the training effect was not considered in the process of model training, and no specific robustness analysis was carried out. Combining these external factors can improve the accuracy and robustness of model training by capturing additional sources of uncertainty.

Author Contributions: Conceptualization, T.C.; Data curation, Y.Y.; Methodology, T.C. and P.Y.; Resources, J.G. and Y.Y.; Software, P.Y. and H.L.; Supervision, J.G. and Y.Y.; Validation, P.Y. and H.L.; Visualization, J.G.; Writing—original draft, P.Y.; Writing—review and editing, T.C., P.Y. and H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Natural Science Foundation of China (51907104), The Opening Fund of Hubei Province Key Laboratory of Operation, and Control of Cascade Hydropower Station (2019KJX08).

Data Availability Statement: The original contributions presented in the study are included in the article and further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ADN	Active distribution network
DG	Distributed generation
DRL	Deep reinforcement learning
DQN	Deep Q network
MDP	Markov decision processes
KW	Kilowatt

References

1. Sarvesh Babu, R.G.; Mithra Vinda Reddy, K.; Shwetha, S.; Sivasankari, G.S.; Narayanan, K.; Sharma, A.; Tellez, A.A. Techno-economic assessment of distribution system considering different types of electric vehicles and distributed generators. *IET Gener. Transm. Distrib.* **2024**, *18*, 1815–1829. [[CrossRef](#)]
2. Ghofrani, M. Synergistic Integration of EVs and Renewable DGs in Distribution Micro-Grids. *Sustainability* **2024**, *16*, 3939. [[CrossRef](#)]
3. Mehroliya, S.; Arya, A. Optimal planning of power distribution system employing electric vehicle charging stations and distributed generators using metaheuristic algorithm. *Electr. Eng.* **2024**, *106*, 1373–1389. [[CrossRef](#)]
4. Sun, H.; Jin, T.; Gao, Z.; Hu, S.; Dou, Y.; Lu, X. A Transmission and Distribution Cooperative Congestion Scheduling Strategy Based on Flexible Load Dynamic Compensation Prices. *Energies* **2024**, *17*, 1232. [[CrossRef](#)]
5. Zhao, Y.; Zhang, Y.; Li, Y.; Chen, Y.; Huo, W.; Zhao, H. Optimal configuration of energy storage for alleviating transmission congestion in renewable energy enrichment region. *J. Energy Storage* **2024**, *82*, 110398. [[CrossRef](#)]
6. Dehnavi, E.; Akmal, A.A.S.; Moeini-Aghaie, M. A novel day-ahead and real-time model of transmission congestion management using uncertainties prioritizing. *Electr. Eng.* **2024**, *106*, 4031–4044. [[CrossRef](#)]
7. Ullah, K.; Ullah, Z.; Shaker, B.; Ibrar, M.; Ahsan, M.; Saeed, S.; Wadood, H.; Lu, S. Line Congestion Management in Modern Power Systems: A Case Study of Pakistan. *Int. Trans. Electr. Energy Syst.* **2024**, *2024*, 6893428. [[CrossRef](#)]
8. Li, H.; Yan, J.; Liu, Y. A Link-Path Model-Based Load-Transfer Optimization Strategy for Urban High-Voltage Distribution Power System. *IEEE Access* **2020**, *8*, 3728–3737. [[CrossRef](#)]
9. Ma, J.; Ma, W.; Qiu, Y.; Yan, X.; Wang, Z. Load transfer strategy based on power transfer capability for main-transformer fault. *Int. Trans. Electr. Energy Syst.* **2015**, *25*, 3439–3448. [[CrossRef](#)]
10. Liu, Z.; Xiong, R.; Tian, Z.; Liang, X.; Yan, F. Evaluation of maximum power supply carrying capacity of medium-voltage distribution network considering feeder segment transfer. *Electr. Eng.* **2024**. [[CrossRef](#)]
11. Luo, F.; Wu, X.; Wang, Z.; Duan, J. A dynamic reconfiguration model and method for load balancing in the snow-shaped distribution network. *Front. Energy Res.* **2024**, *12*, 1361559. [[CrossRef](#)]
12. Gao, H.; Ma, W.; Xiang, Y.; Tang, Z.; Xu, X.; Pan, H.; Zhang, F.; Liu, J. Multi-objective Dynamic Reconfiguration for Urban Distribution Network Considering Multi-level Switching Modes. *J. Mod. Power Syst. Clean Energy* **2022**, *10*, 1241–1255. [[CrossRef](#)]
13. Morsy, B.; Hinneck, A.; Pozo, D.; Bialek, J. Security constrained OPF utilizing substation reconfiguration and busbar splitting. *Electr. Power Syst. Res.* **2022**, *212*, 108507. [[CrossRef](#)]
14. Hrgović, I.; Pavić, I. Substation reconfiguration selection algorithm based on PTDFs for congestion management and RL approach. *Expert Syst. Appl.* **2024**, *257*, 125017. [[CrossRef](#)]
15. El-Azab, M.; Omran, W.A.; Mekhamer, S.F.; Talaat, H.E.A. Congestion management of power systems by optimizing grid topology and using dynamic thermal rating. *Electr. Power Syst. Res.* **2021**, *199*, 107433. [[CrossRef](#)]
16. Cai, Z.; Yang, K.; Chen, Y.; Yang, R.; Gu, Y.; Zeng, Y.; Zhang, X.; Sun, S.; Pan, S.; Liu, Y.; et al. Multistage Bilevel Planning Model of Energy Storage System in Urban Power Grid Considering Network Reconfiguration. *Front. Energy Res.* **2022**, *10*, 952684. [[CrossRef](#)]
17. Liu, Y.; Zeng, Y.; Zhang, X.; Liu, C.; Jin, Y.; Liu, J.; Yang, X.; Xie, Z. Key technology and system for auxiliary decision-making of load transfer in urban high voltage distribution network. *Electr. Power Autom. Equip.* **2023**, *43*, 192–199.
18. Zeng, Y.; Liu, Y.; Gao, H.; Zhang, X.; Zhao, L.; Liu, C.; Wei, W.; Liu, J. Load Transfer Capability of HV Distribution Network and Coordinated Operation With Energy Storage Power Station Based on Model Predictive Control. *Power Syst. Technol.* **2021**, *45*, 1902–1911.
19. Zhu, J.; Dong, S.; Xu, C.; Zhu, B.; Ni, Q.; Xu, Q. Evaluation Model of Total Supply Capability of Distribution Network Considering Multiple Transfers. *Power Syst. Technol.* **2019**, *43*, 2275–2281.
20. Zhou, N.; Mo, F.; Xiao, S.; Gu, F.; Lei, C.; Wang, Q. Coordinated Power Transfer Optimization of Multi-voltage-level Distribution Network Considering Topology Constraints. *Proc. Chin. Soc. Electr. Eng.* **2021**, *41*, 3106–3119.
21. Yu, W.; Liu, D.; Huang, Y. Load transfer and islanding analysis of active distribution network. *Int. Trans. Electr. Energy Syst.* **2015**, *25*, 1420–1435. [[CrossRef](#)]
22. Duan, Q.; Zhao, Y.; Yan, L.; Lu, Z.; Ma, C.; Wang, Y.; Ai, X. Load transfer optimization methods for distribution network including distribution generation. *Power Syst. Technol.* **2016**, *40*, 3155–3162.

23. Jiang, W.; Wu, L.; Zhang, L.; Jiang, Z. Research on load transfer strategy optimisation with considering the operation of distributed generations and secondary dispatch. *IET Gener. Transm. Distrib.* **2020**, *14*, 5526–5535. [[CrossRef](#)]
24. Yang, Q.; Li, G.; Bie, Z.; Wu, J.; Lin, C.; Liu, D. Coordinated Power Supply Restoration Method of Resilient Urban Transmission and Distribution Networks Considering Intermittent New Energy. *High Volt. Eng.* **2023**, *49*, 2764–2779.
25. Guan, Z.; Tang, P.; Mao, C.; Wang, D.; Wang, L.; Liu, W.; Du, M.; Li, J.; Wang, X. Control Strategy and Implementation of Seamless Closed-Loop Load Transfer Mobile Prototype for 400 V Distribution Network. *IEEE Access* **2024**, *12*, 12279–12294. [[CrossRef](#)]
26. Zhou, N.; Gu, F.; Lei, C.; Yao, Y.; Wang, Q. A Power Transfer Optimization Model of Active Distribution Networks in Consideration of Loop Closing Current Constraints. *Trans. China Electrotech. Soc.* **2020**, *35*, 3281–3291.
27. Chen, L.; Li, Z.; Deng, C.; Liu, H.; Weng, Y.; Xu, Q.; Wu, Z.; Tang, Y. Effects of a flux-coupling type superconducting fault current limiter on the surge current caused by closed-loop operation in a 10 kV distribution network. *Int. J. Electr. Power Energy Syst.* **2015**, *69*, 160–166. [[CrossRef](#)]
28. Li, Z.; Xu, Y.; Wang, P.; Xiao, G. Restoration of a Multi-Energy Distribution System With Joint District Network Reconfiguration via Distributed Stochastic Programming. *IEEE Trans. Smart Grid* **2024**, *15*, 2667–2680. [[CrossRef](#)]
29. Xing, H.; Hong, S.; Sun, X. Active Distribution Network Expansion Planning Considering Distributed Generation Integration and Network Reconfiguration. *J. Electr. Eng. Technol.* **2018**, *13*, 540–549.
30. Fu, Y.-Y.; Chiang, H.-D. Toward Optimal Multiperiod Network Reconfiguration for Increasing the Hosting Capacity of Distribution Networks. *IEEE Trans. Power Deliv.* **2018**, *33*, 2294–2304. [[CrossRef](#)]
31. Pereira, E.C.; Barbosa, C.H.N.R.; Vasconcelos, J.A. Distribution Network Reconfiguration Using Iterative Branch Exchange and Clustering Technique. *Energies* **2023**, *16*, 2395. [[CrossRef](#)]
32. Harsh, P.; Das, D. A Simple and Fast Heuristic Approach for the Reconfiguration of Radial Distribution Networks. *IEEE Trans. Power Syst.* **2023**, *38*, 2939–2942. [[CrossRef](#)]
33. Mojaradi, Z.; Tavakkoli-Moghaddam, R.; Bozorgi-Amiri, A.; Heydari, J. A two-stage risk-based framework for dynamic configuration of a renewable-based distribution system considering demand response programs and hydrogen storage systems. *Int. J. Hydrogen Energy* **2024**, *62*, 256–271. [[CrossRef](#)]
34. Dey, I.; Roy, P.K. Simultaneous network reconfiguration and DG allocation in radial distribution networks using arithmetic optimization algorithm. *Int. J. Numer. Model. Electron. Devices Fields* **2023**, *36*, e3105. [[CrossRef](#)]
35. Li, Q.; Huang, S.; Zhang, X.; Li, W.; Wang, R.; Zhang, T. Topology Design and Operation of Distribution Network Based on Multi-Objective Framework and Heuristic Strategies. *Mathematics* **2024**, *12*, 1998. [[CrossRef](#)]
36. Chen, T.; Li, H.; Cao, Y.; Zhang, Z. Substation Operation Sequence Inference Model Based on Deep Reinforcement Learning. *Appl. Sci.* **2023**, *13*, 7360. [[CrossRef](#)]
37. Damjanović, I.; Pavić, I.; Puljiz, M.; Brcic, M. Deep Reinforcement Learning-Based Approach for Autonomous Power Flow Control Using Only Topology Changes. *Energies* **2022**, *15*, 6920. [[CrossRef](#)]
38. Kim, S.; Yoon, S.; Lim, H. Deep Reinforcement Learning-Based Traffic Sampling for Multiple Traffic Analyzers on Software-Defined Networks. *IEEE Access* **2021**, *9*, 47815–47827. [[CrossRef](#)]
39. Besta, M.; Gerstenberger, R.; Peter, E.; Fischer, M.; Podstawski, M.; Barthels, C.; Alonso, G.; Hoefer, T. Demystifying Graph Databases: Analysis and Taxonomy of Data Organization, System Designs, and Graph Queries. *ACM Comput. Surv.* **2023**, *56*, 31. [[CrossRef](#)]
40. Jalali Khalil Abadi, Z.; Mansouri, N.; Javidi, M.M. Deep reinforcement learning-based scheduling in distributed systems: A critical review. *Knowl. Inf. Syst.* **2024**, *66*, 5709–5782. [[CrossRef](#)]
41. Gholizadeh, N.; Kazemi, N.; Musilek, P. A Comparative Study of Reinforcement Learning Algorithms for Distribution Network Reconfiguration With Deep Q-Learning-Based Action Sampling. *IEEE Access* **2023**, *11*, 13714–13723. [[CrossRef](#)]
42. Wang, Z.; Zhang, S.; Luo, W.; Xu, S. Deep reinforcement learning with deep-Q-network based energy management for fuel cell hybrid electric truck. *Energy* **2024**, *306*, 132531. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.