

Model-Based Reinforcement Learning Method for Microgrid Optimization Scheduling

Yao, Jinke; Xu, Jiachen; Zhang, Ning; Guan, Yajuan

Published in:
Sustainability (Switzerland)

DOI (link to publication from Publisher):
[10.3390/su15129235](https://doi.org/10.3390/su15129235)

Creative Commons License
CC BY 4.0

Publication date:
2023

Document Version
Publisher's PDF, also known as Version of record

[Link to publication from Aalborg University](#)

Citation for published version (APA):
Yao, J., Xu, J., Zhang, N., & Guan, Y. (2023). Model-Based Reinforcement Learning Method for Microgrid Optimization Scheduling. *Sustainability (Switzerland)*, 15(12), Article 9235. <https://doi.org/10.3390/su15129235>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.



- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal -

Take down policy

If you believe that this document breaches copyright please contact us at vbn@aub.aau.dk providing details, and we will remove access to the work immediately and investigate your claim.

Article

Model-Based Reinforcement Learning Method for Microgrid Optimization Scheduling

Jinke Yao ¹ , Jiachen Xu ¹, Ning Zhang ^{2,*} and Yajuan Guan ³ 

¹ Department of Information Science and Engineering, Northeastern University, Shenyang 110819, China; 20205359@stu.neu.edu.cn (J.Y.)

² School of Electrical Engineering and Automation, Anhui University, Hefei 230601, China

³ Department of Energy, Aalborg University, 266100 Elburg, Denmark

* Correspondence: zhangning@ahu.edu.cn

Abstract: Due to the uncertainty and randomness of clean energy, microgrid operation is often prone to instability, which requires the implementation of a robust and adaptive optimization scheduling method. In this paper, a model-based reinforcement learning algorithm is applied to the optimal scheduling problem of microgrids. During the training process, the current learned networks are used to assist Monte Carlo Tree Search (MCTS) in completing game history accumulation, and updating the learning network parameters to obtain optimal microgrid scheduling strategies and a simulated environmental dynamics model. We establish a microgrid environment simulator that includes Heating Ventilation Air Conditioning (HVAC) systems, Photovoltaic (PV) systems, and Energy Storage (ES) systems for simulation. The simulation results show that the operation of microgrids in both islanded and connected modes does not affect the training effectiveness of the algorithm. After 200 training steps, the algorithm can avoid the punishment of exceeding the red line of the bus voltage, and after 800 training steps, the training result converges and the loss values of the value and reward network converge to 0, showing good effectiveness. This proves that the algorithm proposed in this paper can be applied to the optimization scheduling problem of microgrids.

Keywords: microgrid scheduling method; model-based reinforcement learning; Monte Carlo Tree Search; environment simulator



Citation: Yao, J.; Xu, J.; Zhang, N.; Guan, Y. Model-Based Reinforcement Learning Method for Microgrid Optimization Scheduling. *Sustainability* **2023**, *15*, 9235. <https://doi.org/10.3390/su15129235>

Academic Editor: Pablo García Triviño

Received: 4 April 2023

Revised: 1 June 2023

Accepted: 5 June 2023

Published: 7 June 2023



Copyright: © 2023 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

With the growth of the global economy, the pressing issues of rising energy demand and environmental protection have become major concerns for researchers. Distributed energy system (DES) [1–4] and multi-energy system (MES) integrate advanced communication and control technologies with the power system and have become the mainstream solution [5,6]. Microgrids, due to their ability to improve the flexibility, reliability, and power quality of the grid, have become an important component of these systems. Microgrids provide an alternative to centralized power generation and long-distance transmission that address their inherent limitations. Despite the benefits, the variability and intermittent nature of clean energy production pose challenges to the safe and stable operation of microgrids. As a result, optimizing scheduling for microgrids has garnered considerable attention to ensure the secure and stable grid connection of clean energy production capacity.

Currently, there have been relatively mature technological developments in the field of economic scheduling for distributed energy and multi-energy systems [7–10]. However, the problem of scheduling and management of microgrids as energy vehicles relatively relies on traditional optimization models and algorithms. Heuristic algorithms are an extremely important part of traditional optimization algorithms, such as the particle swarm optimization algorithm for microgrid management proposed by Eseye et al. [11] and Zeng et al. [12], are the primary approaches utilized to resolve microgrid optimal issues. In 2018, Elsayed et al. [13] proposed a microgrid energy management method based on the

random drift particle swarm optimization algorithm. Moreover, Zare et al. [14] applied the mixed-integer linearization method (MILP) to microgrid optimization problems in 2016. Other researchers have combined energy security issues with economic optimization issues, proposing a two-level energy scheduling approach [15] and an alternating direction method of multipliers (ADMM) [16]. Nevertheless, traditional optimal scheduling algorithms necessitate the accurate establishment of environmental dynamics models, which cannot be adaptively applied to various microgrid structures. Heuristic algorithms, such as particle swarm optimization (PSO), tend to converge to local optimal solutions due to gradient-based parameter updates. Moreover, the algorithm parameters cannot be optimized and need to be adjusted based on experience. In summary, due to the constantly changing scenario of microgrids and the complexity and diversity of energy coupling, heuristic algorithms have defects such as high dependence on experience selection, poor generalization, and poor robustness.

Fuzzy optimization algorithms are efficient in handling uncertainties and ambiguity, which makes them suitable for solving complex coupling scenarios of multiple energy sources in microgrids. Fossati et al. [17] and Banaì et al. [18] have demonstrated the feasibility of applying fuzzy optimization to the optimization scheduling problem of microgrids. However, fuzzy optimization methods suffer from the uncertainties and ambiguity of decision variables and constraints, as well as the complexity of solving the problem, and the poor interpretability of the results.

The advancement of big data technology and the emergence of neural networks have facilitated the application of reinforcement learning (RL) in various domains [19]. Reinforcement learning combines deep neural networks with Markov decision process (MDP) optimization procedures and is a learning mechanism that maps environmental observations to selected action values. Subsequently, the reinforcement learning process updates neural network parameters through the pursuit of maximum rewards from environmental feedback. Reinforcement learning explains the decision-making process by establishing a state-action-reward model. It continuously learns through trial or error and feedback to obtain the optimal strategy. This makes it more efficient, interpretable, generalizable, and flexible which is compared to heuristic algorithms. As such, reinforcement learning has the potential to meet the requirements for optimizing microgrid scheduling problems.

Currently, the prevailing algorithm utilized in integrated energy systems is model-free reinforcement learning, which has shown remarkable performance and learning outcomes in microgrid optimization scheduling. By optimizing the loss/reward function, this approach only updates neural network parameters between states and actions and does not require specific environmental dynamics modeling. The application of Q-learning to microgrid optimization was the first proposal [20–22], but the limited effectiveness of Q-learning in the field of microgrid scheduling was attributed to the large Q-table and overestimated Q-value. Double Q-learning (DQN) [23] and Deep Deterministic Policy Gradient (DDPG) [24,25] have achieved more promising results than Q-learning with the emergence of deep neural networks and strategy gradient optimization. To address the overestimation and high square error drawbacks of the DDPG algorithm, Garrido et al. [26] applied the TD3 algorithm to microgrids. Furthermore, Schulman et al. [27] proposed the PPO algorithm in the year 2017, which OpenAI has adopted as the baseline for reinforcement learning algorithms. PPO has been extended to the microgrid management domain [28].

Nonetheless, model-free reinforcement learning algorithms have many limitations as they only establish a connection between observation values and action values through the optimization object, disregarding the effectiveness of the environmental dynamic foundation. Consequently, modifying the optimization task may not result in the anticipated outcome using the original value path. In the face of this obstacle, the proposed approach entails the need for neural network reconstruction and parameter optimization, which ultimately results in longer training duration and compromised model resilience.

Model-based reinforcement learning [29] is intended to address these issues by starting with learning the environmental dynamic model and then planning based on the acquired model. Typically, these models focus on either reconstructing the accurate state of the environment [30–32] or completing observational sequences [33]. In 2016, Google’s DEEP-MIND produced the AlphaGo algorithm, which triumphed over human chess players in the field of Go. Energy scheduling techniques with the AlphaGo algorithm as the core notion have also been adopted in the microgrid arena. For instance, Li et al. [34] established a distribution system outage repair system utilizing tree search as the primary strategy. Nonetheless, AlphaGo is a model-based reinforcement learning algorithm that requires prior knowledge from experts in the domain of Go before training, such as the rules for playing chess and winning or losing. Therefore, it is challenging to implement AlphaGo in general optimization control problems. Another limitation of AlphaGo is the high dimensional observation space, which requires large space and increased training time, resulting in excessive costs for training. The comparison of advantages and disadvantages of relevant research on microgrid optimization scheduling problems is shown in Table 1.

Table 1. Frequency used notation.

Reference	Explicability of Results	Require Prior Knowledge or not	Computational Complexity	Generalization for Optimization Task	Feature of the Paper
[13]	Low	Yes	Low	Low	Used random drift particle swarm optimization to Solving the energy management problem of residential microgrids
[17]	Low	Yes	Low	Low	Built a fuzzy expert system to control the power output of the storage system and determined the day-ahead microgrid scheduling
[28]	High	No	High	Low	Proposed a safe implementation of Proximal Policy Optimization for EV scheduling problem
[34]	High	Yes	High	Low	Applies the tree search algorithms to solve problem of restoring load after a fault in a power system.

The generalization, flexibility, and low dependence on professional knowledge of model-based reinforcement learning algorithms make them suitable for constantly changing scenarios, complex energy structures, and diverse optimization tasks in microgrids. The established environment model and policy model provide interpretability to the results. The utilization of deep neural networks enhances the accuracy and reliability of the results. Therefore, this paper borrows the algorithm ideas of model-based reinforcement learning to optimize the scheduling of results.

To address the problems identified in the above analysis, this paper introduces a microgrid energy scheduling method that combines environmental modeling with optimal strategy learning. The key contributions of this research are summarized as follows:

1. This study converts the microgrid optimization scheduling issue into reinforcement learning tasks by identifying the observation space, state space, and reward functions within the microgrid model.
2. This paper introduces model-based reinforcement learning to address the optimal scheduling problem in microgrid systems, which adopts the MuZero algorithm’s excellent performance in discrete action spaces into the application of Monte Carlo Tree Search for the determination of the optimal strategy.
3. This paper builds a joint training model for three major networks to enable the algorithm to learn and optimize simultaneously for environmental dynamic and optimal strategy determination without prior professional knowledge.

The following organization is adopted in this paper: in Section 2, the microgrid scheduling model is developed. The relevant principles and processes of the algorithm are provided in Section 3. Subsequently, in Section 4, the simulation results are presented, while Section 5 summarizes the research study.

2. Microgrid Scheduling Model

Inspired by the microgrid model proposed by Biagion et al. [35], known as the Power Grid World, we have established a new microgrid model that incorporates HVAC, PV, and ES. The microgrid structure is presented in Figure 1.

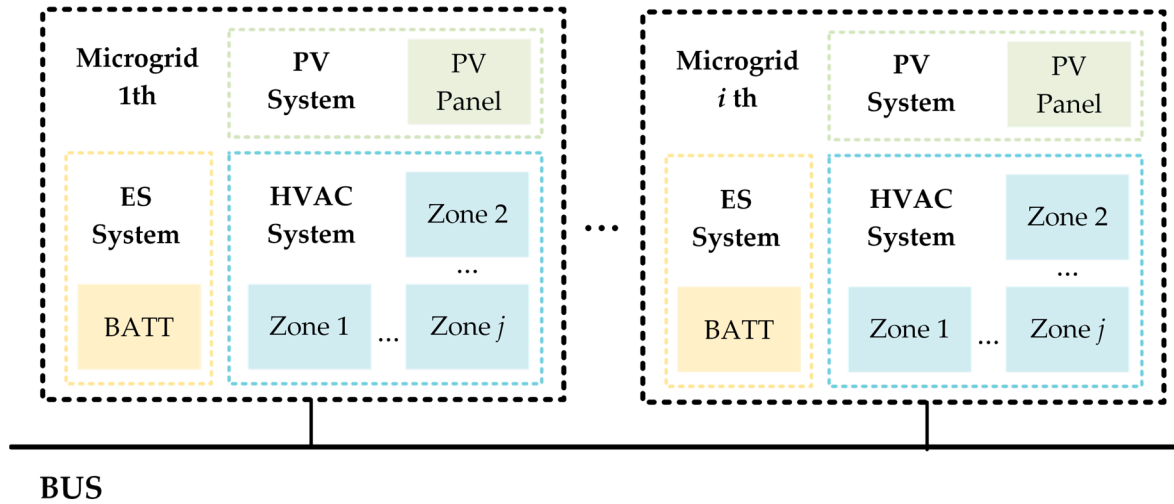


Figure 1. The Structure of Microgrid Scheduling Model.

The following section provides a detailed description of the model used by the i th microgrid.

2.1. HVAC System

The system mainly refers to the load within the microgrid model, which aims to regulate the output of HVAC power and discharge temperature in order to balance the internal space temperature. The state space, action space, and reward function of the HVAC system are described below.

The observation space of the HVAC system is:

$$o_t^{i,HVAC} = (T_t^i, Tc_t^i, T_t^{i,out}, p_t^{i,consumed}, U_t^{i,bus}), \quad (1)$$

where $T_t^{i,out}$ represents the outdoor temperature. $p_t^{i,consumed}$ represents the power consumption. And $U_t^{i,bus}$ represents the current voltage of the bus where the i th microgrid is located. Where $T_t^i = (T_t^{i,j})$, $j \in HVAC$, represents the temperature collection of all spaces in the i th microgrid and $T_t^{i,j}$ represents the temperatures in the j th space of the HVAC system. And $Tc_t^i = (Tc_t^{i,high}, Tc_t^{i,low})$ represents the upper and lower limits of the comfort temperature at the current moment.

The action space of the HVAC system is:

$$\begin{aligned} a_t^{i,HVAC} &= (f_t^i, Td_t^i), \\ f_t^i &= (f_t^{i,j}) \quad j \in HVAC, \end{aligned} \quad (2)$$

where $f_t^{i,j}$ represents the flow rate of HVAC in j th space of the system, with the range of $(f_{\min}^{i,j}, f_{\max}^{i,j})$. And Td_t^i represents the discharge temperature of i th microgrid, with a range of $(Td_{\min}^i, Td_{\max}^i)$.

To achieve the objectives of temperature comfort maintenance comfort and minimizing power consumption in the system, this paper proposes the design of separate reward functions.

2.1.1. Discomfort Reward

To ensure that the temperature in each space falls within a range that would make people inside feel comfortable, a discomfort reward is established to penalize observations that exceed the maximum comfortable temperature or fall below the minimum comfortable temperature. Firstly, the difference is obtained by comparing the comfort temperature limits, and the temperature of each space at the given moment t :

$$\Delta T_t^{i,j} = (T_t^{i,j} - Tc_t^{i,high}, Tc_t^{i,low} - T_t^{i,j}). \quad (3)$$

After obtaining the temperature difference, the uncomfortable temperature value $\tau_t^{i,j}$ of the j th space is calculated as follows:

$$\tau_t^{i,j} = \max(\Delta T_t^{i,j}, 0). \quad (4)$$

Compute the discomfort reward by taking the sum of squared discomfort temperature values for each space:

$$r_t^{i,temp} = \sum_{j \in HVAC} -(\tau_t^{i,j})^2. \quad (5)$$

2.1.2. Energy Consumption Reward

To attain the objective of maximizing energy conservation and minimizing emission, energy consumption incentives are implemented as a measure against high-power consumption. The energy consumption incentives are exclusively linked to power consumption, and their implementation aims to reduce energy consumption:

$$r_t^{i,consumed} = -P_t^{i,consumed}. \quad (6)$$

2.1.3. HVAC System Reward

With regards to the discomfort reward and energy consumption reward, this article formulates the reward intended for the HVAC system design:

$$r_t^{i,HVAC} = \alpha \times \frac{r_t^{i,consumed}}{\beta} + (1 - \alpha) \times r_t^{i,temp}, \quad (7)$$

where α, β are the balance coefficients used to balance discomfort rewards and energy consumption rewards to satisfy the different control requirements.

2.2. PV System

Photovoltaic power generation is considered a clean and sustainable energy source. However, due to the inherent uncertainty and randomness in its production capacity, there exist potential safety hazards in connecting it to the electrical grid. According to the research data of Biagion et al. [35], the photovoltaic power generation during a day mainly exhibits a parabolic shape, similar to the curve of solar radiation energy. The normalized daily photovoltaic power generation data is shown in Figure 2.

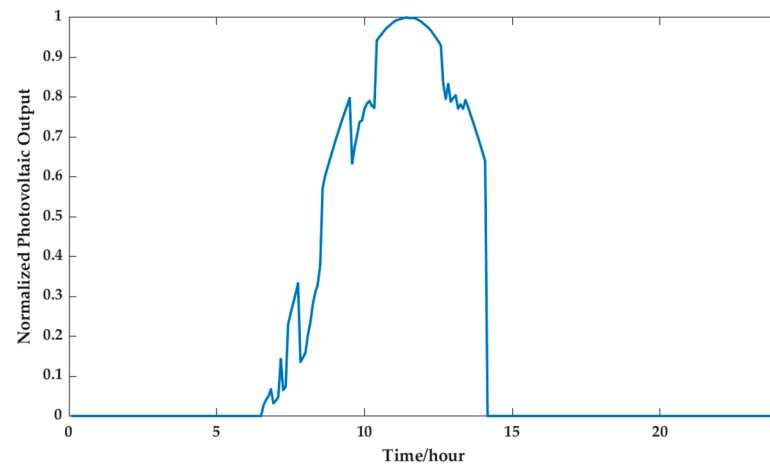


Figure 2. Variations in Normalized Photovoltaic Power Generation throughout a Day.

As productivity equipment, this paper integrates the photovoltaic (PV) system into the Microgrid Scheduling Model. In this model, the PV system serves as the production capacity equipment that generates power for other parts of the system. The subsequent sections highlight the state space, action space, operation model and reward function associated with a PV system.

The observation space of the PV system is:

$$o_t^{i,PV} = (P_t^{i,EV}), \quad (8)$$

where $P_t^{i,EV}$ represents the actual production capacity at time t , and this part of the observation value is provided by the actual observation data of a day.

The action space of the PV system is:

$$a_t^{i,PV} = (\xi_t^i) \quad \xi_t^i \in [0, 1], \quad (9)$$

where ξ_t^i is the ratio between the production capacity at time t and the local consumption, which is defined as the consumption rate of clean energy.

Based on the aforementioned physical definitions of observation and action space, it can be inferred that the effective electrical energy input from the PV system to the microgrid is:

$$P_t^{i,PVreal} = o_t^{i,PV} \times a_t^{i,PV}, \quad (10)$$

where $P_t^{i,PVreal}$ represents the actual amount of electrical energy that is being consumed in the j th microgrid at a given time t .

To ensure consistent and sustainable clean energy absorption by the microgrid, the paper designs the reward functions of photovoltaic systems in the form of a defined light rejection rate as follows:

$$r_t^{i,PV} = -\frac{o_t^{i,PV} - P_t^{i,PVreal} - p^{bus}}{o_t^{i,PV}}, \quad (11)$$

where p^{bus} represents the maximum energy that the electronic bus can transport.

2.3. ES System

The Microgrid Scheduling Model integrates an Energy Storage (ES) system that primarily employs a battery pack to store electrical energy. Its main function is to stabilize the power demand and supply within the system. The ensuing sections explicate the state space, action space, transition model, and reward function associated with the ES system.

The observation space of the ES system is:

$$o_t^{i,ES} = (E_t^{i,storage}), \quad (12)$$

where $E_t^{i,storage}$ represents the standard of energy which the pack stored at time t .

The action space of the ES system is:

$$a_t^{i,ES} = (p_t^i) \quad p_t^i \in [-1, 1], \quad (13)$$

where p_t^i represents the ratio of the energy charged or discharged at time t to the maximum energy charged or discharged during each cycle. When $p_t^i \in (0, 1]$, it represents the discharge of the battery, otherwise, it is charged.

From the above definitions of observed values and action values, it can be concluded that the theoretical state transformation of the ES system is as follows:

$$E_t^{theoretical} = \begin{cases} o_{t-1}^{i,ES} - \frac{p_{t-1}^{max} \times a_{t-1}^{i,ES} \times \sigma}{v_{discharge}} & a_t^{i,ES} > 0 \\ o_{t-1}^{i,ES} - p_{t-1}^{max} \times a_{t-1}^{i,ES} \times \sigma \times v_{charge} & a_t^{i,ES} < 0 \end{cases} \quad (14)$$

where σ represents the control interval of the system. $v_{charge}, v_{discharge}$ represent the efficiency of the discharging or charging of the energy.

Due to the limitation of energy storage in the ES system, the theoretical value at the current time may differ from the actual stored energy. Therefore, the observation of the ES system is as follows:

$$o_t^{i,ES} = \begin{cases} \min(E_t^{theoretical}, E_{max}^{storage}) & a_t^{i,ES} > 0 \\ \max(E_t^{theoretical}, E_{min}^{storage}) & a_t^{i,ES} < 0 \end{cases} \quad (15)$$

where $E_{max}^{storage}, E_{min}^{storage}$ represent the upper and lower limit of the ES system.

Based on the above analysis, the reward function is designed for behaviors with charging that exceeds the upper limit and discharging that falls below the lower limit as follows:

$$r_t^{i,ES} = \mu \times \max(E_t^{theoretical} - E_{max}^{storage}, E_{min}^{storage} - E_t^{theoretical}, 0), \quad (16)$$

where μ is the balance coefficient used to balance the relationship between ES system rewards and other system rewards.

2.4. System Reward

In grid-connected mode, a microgrid will interact with other microgrids on the same bus. Therefore, the optimal scheduling of multiple microgrids should not only focus on the development of each microgrid individually but also on the overall system optimization.

The paper presents a designed reward function that guarantees the safe and stable operation of the bus voltage system. Each microgrid shall bear the overall system rewards on an average basis.

$$r_t^{i,sys} = -\frac{U_t^{i,viol} \times \rho}{N_{agent}}, \quad (17)$$

$$U_t^{i,viol} = \max(U_t^{i,bus} - U_{max}^{bus}, U_{min}^{bus} - U_t^{i,bus}, 0),$$

where $U_{max}^{bus}, U_{min}^{bus}$ represents the upper and lower limits of bus voltage under safe operation. ρ represents a very high penalty coefficient. And N_{agent} represents the number of microgrids on the same bus.

2.5. Model Reward

Following the aforementioned analysis, the reward for individual microgrid can be obtained as:

$$r_t^i = r_t^{i,sys} + r_t^{i,HVAC} + r_t^{i,PV} + r_t^{i,ES}. \quad (18)$$

The paper defines four rewards that hold equal significance in optimizing the system. The reward for the whole model should comprise the summation of the rewards of each microgrid:

$$r_t = \sum_{i \in sys} r_t^i. \quad (19)$$

3. Algorithm Design

To ensure that the optimal scheduling method of microgrid systems is closely tied to environmental dynamics, this paper utilizes a model-based reinforcement learning method for optimization. The Muzero algorithm framework, proposed by Schrittwieser et al. [36], in 2020, has demonstrated superior performance in studying discrete action spaces. Accordingly, this paper employs it to resolve the optimal scheduling challenge in microgrids.

As MuZero leverages Monte Carlo Tree Search (MCTS) [37] to identify optimal strategy selection and requires discrete action values. Sinclair et al. [38] investigated the feasibility of solving problems in continuous action spaces by discretizing the action space. This paper discretizes the action space of the model:

$$x_t^{i,j} = (a_1, a_2, \dots, a_j) \quad a_j \in a_t^{i,j}. \quad (20)$$

The paper conducts the average discretization of motion space selection to ensure the comprehensiveness and universality of motion selection.

The learning and training process of optimal scheduling strategies is grounded in the MuZero training process, as demonstrated in Figure 3.

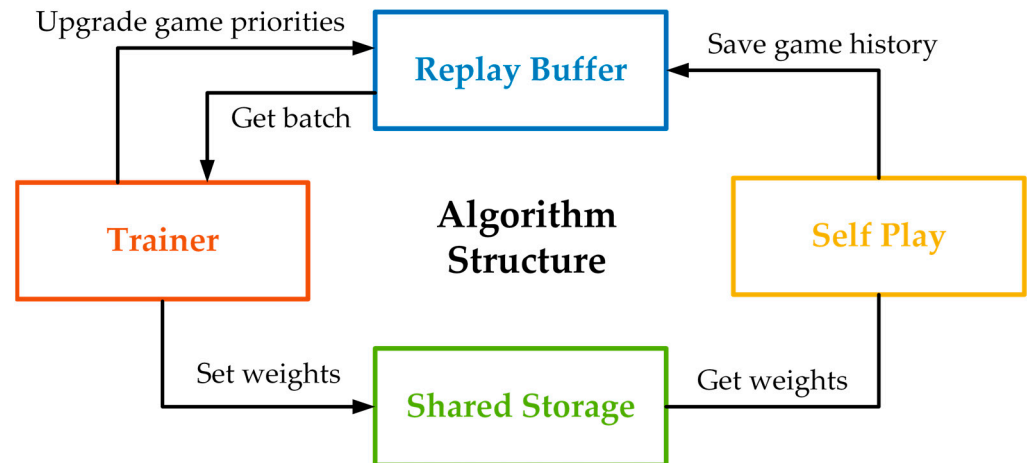


Figure 3. Flowchart of the Algorithm Architecture.

Figure 3 showcases that the algorithm's training process extensively comprises four facets: Self-play, Replay Buffer, Trainer, and Shared Storage. The Replay Buffer functions to hoard the game history data collated by the Self-play phase, while the Shared Storage preserves the Trainer training model parameters. The upcoming sections of this chapter expound upon the Self-play and Trainer components in detail.

3.1. Self-Play

The Self-play module's importance lies in generating game data through the algorithm's interaction with the real environment, supplying data for the following model training. Crucial to Self-play is the utilization of MCTS [37] as a strategy optimization

technique, coupled with Representation, Prediction, and Dynamic learning function for auxiliary simulation. The primary flow of MCTS exploration in Self-play from time t is illustrated in Figure 4.

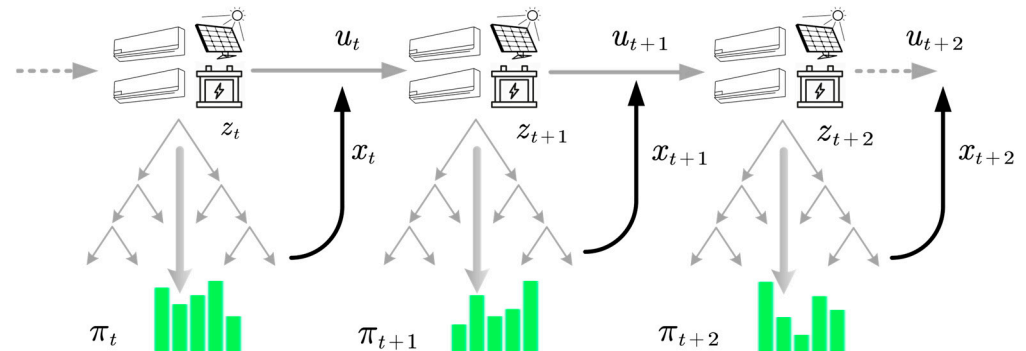


Figure 4. The Structure of MCTS.

The primary role of Monte Carlo Tree Search (MCTS) is to determine the most effective approach, acquire timely rewards, and evaluate the projected worth. The particular course of action involved in identifying the optimal strategy at a given time t is depicted in Figure 5.

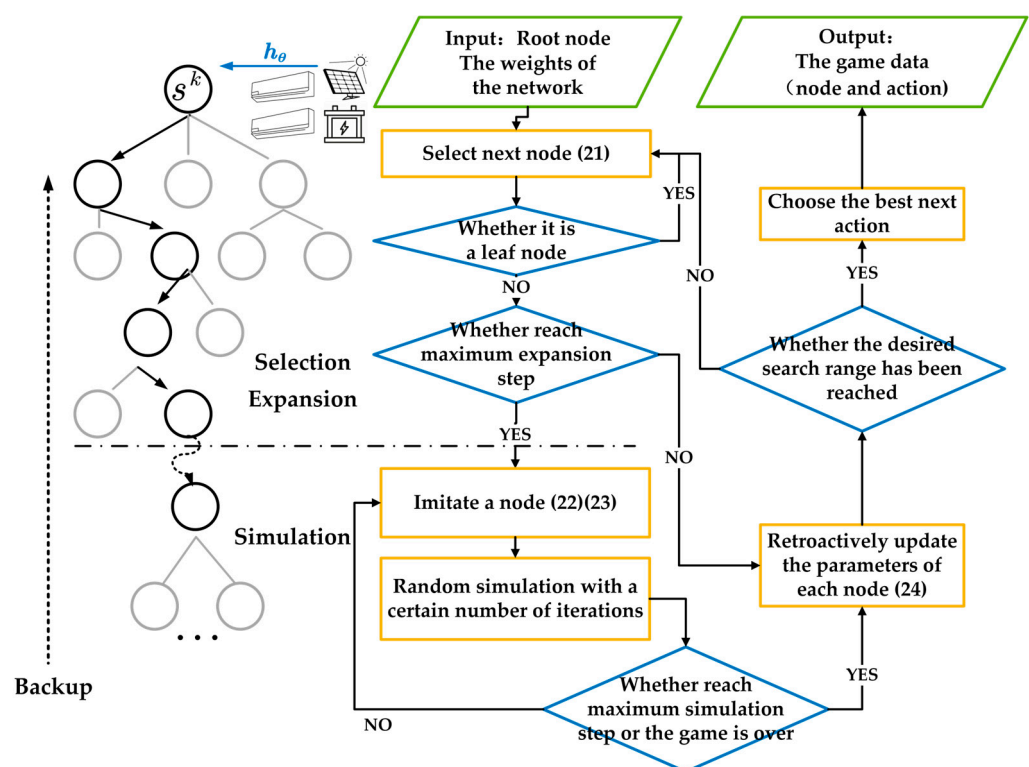


Figure 5. The Search Process of MCTS with the Learned Model.

MCTS utilizes the three fundamental steps of Selection, Expansion, and Backup to screen the optimal next node. The process will be explained in detail below.

3.1.1. Selection

The selection of an action for the next node at the current node primarily employs Upper Confidence Bound (UCB). UCB's action selection mainly relies on the estimated

reward garnered by selecting the action and the number of previous times the action has been chosen:

$$x^k = \arg \max \left\{ Q(s, x) + P(s, x) \frac{\sqrt{\sum N(s, y)}}{1 + N(s, x)} \times \left(c_1 + \log \left(\frac{\sum N(s, y) + c_2 + 1}{c_2} \right) \right) \right\}, \quad (21)$$

where $Q(s, x)$ represents the mean predicted value that results from performing action x in the current state. This value is primarily used to assist Monte Carlo Trees (MCTs) in selecting actions that are expected to yield greater future rewards. The addition of further content following the plus sign serves to expand the feasible options available for MCTs to consider. Meanwhile, $P(s, x)$ denotes the prior probability of selecting action x , while c_1, c_2 is leveraged to ensure a balance of search tree efficiency and breadth. Consequently, the search tree is more inclined to seek out and explore high-value nodes that have yet to be discovered.

3.1.2. Expansion

When the simulation reaches the maximum number of steps before the game ends, a new sub-node is expanded. The information pertaining to the new sub-node is calculated by the presently trained policy and dynamic network. Subsequently, the reward and value of the new node are obtained.

$$r^{m+1}, s^{m+1} = g_p(s^m, x^{m+1}), \quad (22)$$

$$p^{m+1}, v^{m+1} = f_\theta(s^{m+1}), \quad (23)$$

where m is the step of selecting a leaf node, and $r^{m+1}, s^{m+1}, p^{m+1}, v^{m+1}$ represent the reward, state, policy and value that are calculated or estimated by the three networks. The data is stored and subsequently subjected to a random simulation consisting of 1 step once expansion has been completed.

3.1.3. Backup

The primary function of this process entails backtracking from the leaf node to the root node of the whole process, updating the values that correspond to each node along the way.

$$\begin{aligned} Q(s^{k-1}, a^k) &:= \frac{N(s^{k-1}, a^k) \times Q(s^{k-1}, a^k) + G^k}{N(s^{k-1}, a^k) + 1}, \\ N(s^{k-1}, a^k) &:= N(s^{k-1}, a^k) + 1, \\ G^k &= \sum_{\Phi=0}^{m-k-1} \gamma^\Phi r_{k+1+\Phi} + \gamma^{m-k} v^m, \end{aligned} \quad (24)$$

where G^k represents the cumulative median reward in MCTS and γ is the time loss constant. $N(s^{k-1}, a^k)$ represents the times that the node was selected and after passing through a node, $N(s^{k-1}, a^k)$ increases by 1.

The state, action strategy, reward, and value corresponding to the optimal selection obtained from the combined search of MCTS and learned models in the actual environment simulator will be recorded in the Replay Buffer for subsequent learning of the neural network model.

3.2. Trainer

Given the extended duration of MCTS, it is not recommended for utilization in scenarios involving rapid changes within the microgrid. For this reason, a “Trainer” link has been incorporated to facilitate reinforcement learning through deep neural network training aimed at learning the environmental dynamics, rewards, and punishment mecha-

nisms present in the current milieu. In total, three functions have been established, mainly consisting of three neural networks, which are described in detail below.

3.2.1. Representation Function

The purpose of this function is to transform the observation space into a hidden state within MuZero. This step is necessary to address the challenge posed by complex observation spaces, such as Go, which has the potential to generate cumbersome computing tasks:

$$s^0 = h_{\theta}(o_t), \quad (25)$$

where s^0 is the hidden state of the root node o_t , θ is the weights of the representation model, and the included neural network is an encoder network that encodes the observations of the current state as the input into a hidden state. This model solves the problem of increasing computational tasks due to excessive observation space.

3.2.2. Prediction Function

The model comprises two networks; the Policy network and the Value network. The input to this model is the hidden state, which was highlighted in Section 3.2.1. The objective of this model is to acquire the optimal strategy and expected average reward for the current moment corresponding to the aforementioned hidden state:

$$p^k, v^k = f_{\vartheta}(s^k), \quad (26)$$

where ϑ represents the weight of the prediction model, while p^k refers to the optimal strategy derived from the model's policy network, and v^k represents the expected value derived from the value network within the model.

3.2.3. Dynamic Function

This model presents an environmental dynamics simulation for the reinforcement learning method. It calculates the hidden state at the current moment by utilizing the previous moment's hidden state and the current moment's action:

$$r^k, s^k = g_{\rho}(s^{k-1}, x^k). \quad (27)$$

Training this model allows the method to acquire knowledge of the dynamic model within the real environment, as the model serves as a transition model between hidden states.

3.2.4. Training Process

The training concept involves training three models to simulate MCTS for optimal strategy selection, as well as simulation prediction of timely and expected rewards. Consequently, this article presents the following loss function for updating parameters:

$$l_t(\theta, \vartheta, \rho) = \sum_{k=0}^K l^P(\pi_{t+k}, p_t^k) + \sum_{k=0}^K l^V(z_{t+k}, v_t^k) + \sum_{k=0}^K l^R(u_{t+k}, r_t^k) + c(\|\theta\|^2 + \|\vartheta\|^2 + \|\rho\|^2), \quad (28)$$

where l^P, l^V, l^R represents the loss function of the policy, value, and reward networks in the microgrid model presented in this article. Specifically, the cross-entropy loss serves as the loss function for the policy network, while the square error loss serves as the loss function for the value network and reward network.

The pseudocode for parallel training of three major networks is shown in Algorithm 1.

Algorithm 1: The Pseudocode of Trainer

Input: Game data from the self-play process, the current parameters θ, ϑ, ρ of the Representation, Prediction and Dynamic model.

Output: The updated weight parameters of the model.

```

1  start
2  (25) representation function to transform the observation  $o_t$  into hidden state  $s_0$ 
3  repeat
4      (26) prediction function to choose the action  $p_k$  and estimate the value  $v_k$ 
5      (27) dynamic function to use the environmental dynamics and reward function to
        calculate the next state  $s_k$  and corresponding reward  $r_k$ 
6      (28) to update  $\theta, \vartheta, \rho$ 
7  until complete a game training batch
4  end

```

The corresponding network structure and training process for joint parallel training of three networks are shown in Figure 6.

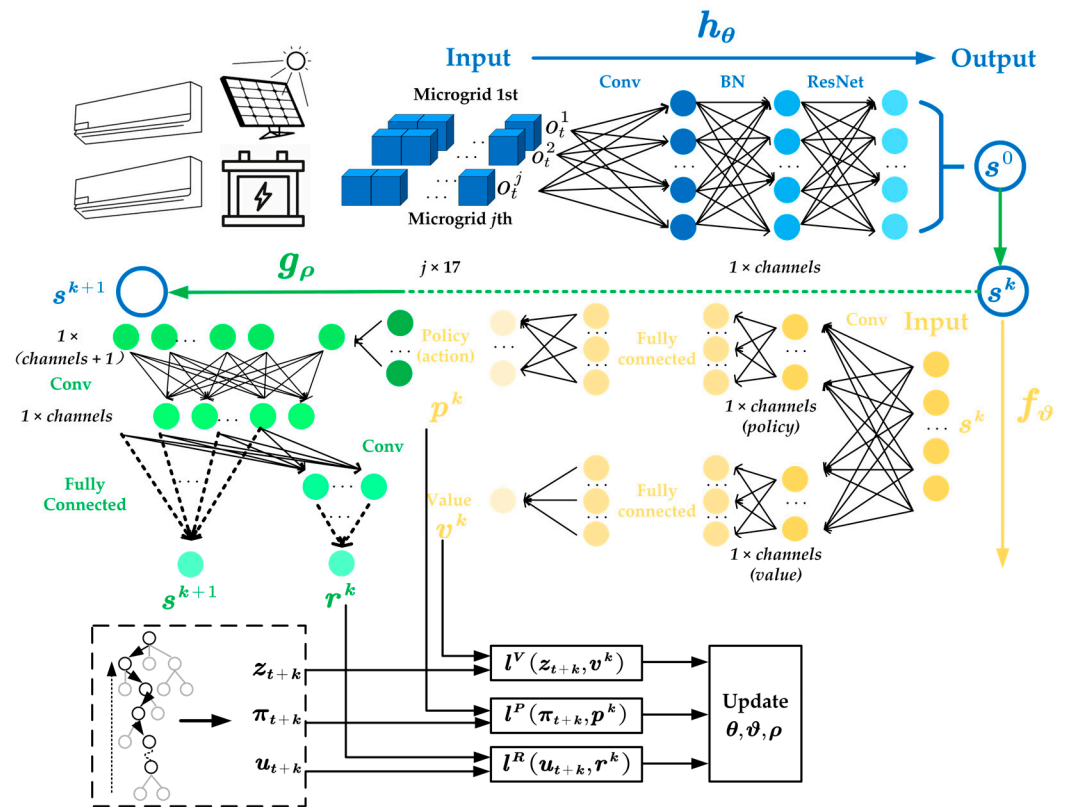


Figure 6. Schematic Diagram of Joint Training of Three Functions.

4. Simulation Results

The simulation verification implementation can be bifurcated into two vital stages, namely parameter setting and results presentation. In Section 3.1 of this article, the article will lay down the model's parameters, whereas Section 3.2 will highlight the simulation results.

4.1. Model and Neural Network Parameter Design

Based on the aforementioned analysis, the parameters may be applied to the Microgrid Scheduling Model, the Reinforcement Learning Algorithm, and the neural network structure.

4.1.1. Microgrid Scheduling Model Parameters

To begin with, an environmental simulator should be constructed to simulate the operation model of microgrids as outlined in Section 2. The pertinent parameters regarding the model discussed in Section 2 are listed in Table 2.

Table 2. Parameters of the Environmental Simulator.

Parameters	Value	Parameters	Value
$Tc_t^{i,low}$	22	α	0.2
$Tc_t^{i,high}$	28	β	2
$f_{min}^{i,j} (j = 1, 2, 3, 4)$	0.22	p^{max}	15
$f_{max}^{i,j} (j = 1, 2, 3, 4)$	2.2	σ	0.083
$f_{min}^{i,j} (j = 5)$	0.32	ρ	10,000
$f_{max}^{i,j} (j = 5)$	3.2	$v_{discharge}$	0.95
Td_{min}^i	10	v_{charge}	0.9
Td_{max}^i	16	$E_{storage}^{min}$	3
U_{min}^{bus}	0.95	$E_{storage}^{max}$	50
U_{max}^{bus}	1.05		

Where $Tc_t^{i,low}$, $Tc_t^{i,high}$ represent the default value of the comfort temperature set, which may vary over time, and U_{min}^{bus} , U_{max}^{bus} denote the upper and lower limits of the bus voltage following normalization.

4.1.2. Reinforcement Learning Algorithm Parameters

The reinforcement learning algorithm based on the MuZero framework, as discussed in Section 3, is devised, and the parameters related to training as specified in the same section can be found in Table 3.

Table 3. Parameters of the Reinforcement Learning Algorithm.

Parameters	Value	Parameters	Value
c_1	1.25	Policy channels	32
c_2	19,625	Value channels	2
m	300	Reward channels	2
l	20	Dynamic layers	16
Learning-rate	0.02	Policy layers	128
Encoding size	8	Value layers	16
Learning-decay	0.9	Reward layers	16
Training steps	1000	Optimizer	Adam
Channels	32	Batch size	128

4.1.3. Neural Network Parameters

As per the learning model structure laid out in Section 3, the Representation, Prediction, and Dynamic learning networks are entirely interconnected internally. The detailed structure of the fully-connected network can be found in Table 4.

4.2. Result

The operation of microgrids can be bifurcated into two modes, namely islanded operation and grid-connected operation. The fundamental difference between these two modes lies in their connection to other microgrids. This paper comprehensively analyzes and simulates the results of both modes under the purview of the reinforcement learning method. To facilitate the operation of the reinforcement learning algorithm, we correct the reward to the absolute value of the reciprocal of the original reward.

Table 4. Structure of the Fully-connected Layer in Learning Neural Network.

Model	Network	Layer	Information
Representation	Encode	Linear Identity	Input: observation space Output: encoding size /
Prediction	Policy	Linear	Input: encoding size
		ELU	Output: policy channels $\alpha = 1.0$
		Linear	Input: policy channels
	Value	Identity	Output: action space + encoding size /
		Linear	Input: encoding size
		ELU	Output: 16 $\alpha = 1.0$
Dynamic	Encode state	Linear	Input: 16
		ELU	Output: 9 /
		Linear	Input: action space + channels
	Reward	Identity	Output: 16 $\alpha = 1.0$
		Linear	Input: 16
		Identity	Output: 8 /

4.2.1. Islanded Operation

The “Islanded operation” pertains to a singular microgrid located on a bus, which can function independently and sustain itself with its electrical energy. This paper sets the number of microgrids in electrical models $i = 1$ in the islanded operation situation. In the islanded operation mode, the bus’s rewards are exclusively allocated to only the microgrid. To begin with, we scrutinize the performance of the method in the given environment, which is highlighted in Figure 7.

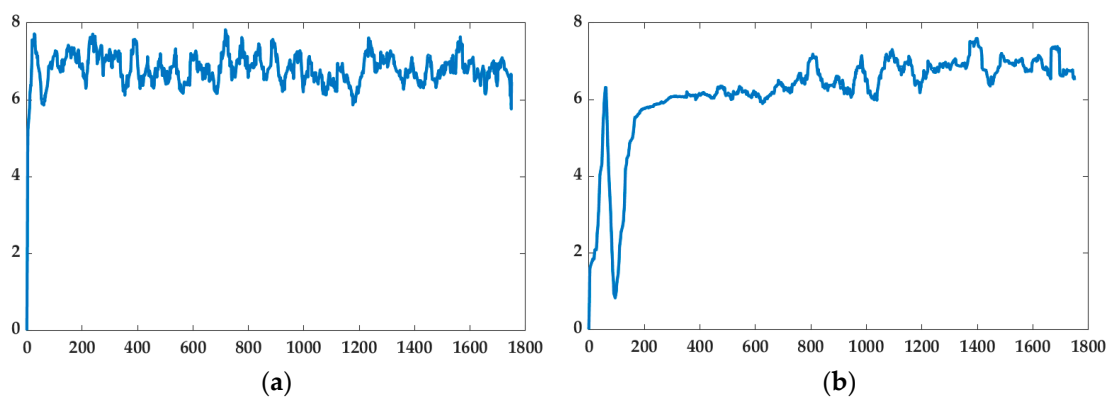


Figure 7. Performance of the Algorithm in Microgrid Island Operation Mode. (a) Optimization curve for timely rewards during training; (b) Optimization curve of expected average value during training.

It can be seen in the figure that (a) is represented by the optimization of the algorithm for the timely reward of the model, because the specific situation of each period in the energy environment is different, so it is mainly based on the long-term stable operation of the

microgrid, rather than the acquisition of timely reward (b) indicates that the algorithm for the average expected reward, which can be seen from the figure has converged, representing that the value path optimization of the algorithm for the microgrid optimization scheduling problem has been completed.

Subsequently, the training for environmental cognition primarily encompasses learning across the three networks, namely policy, reward, and value. The optimization of the loss values can be observed in Figure 8.

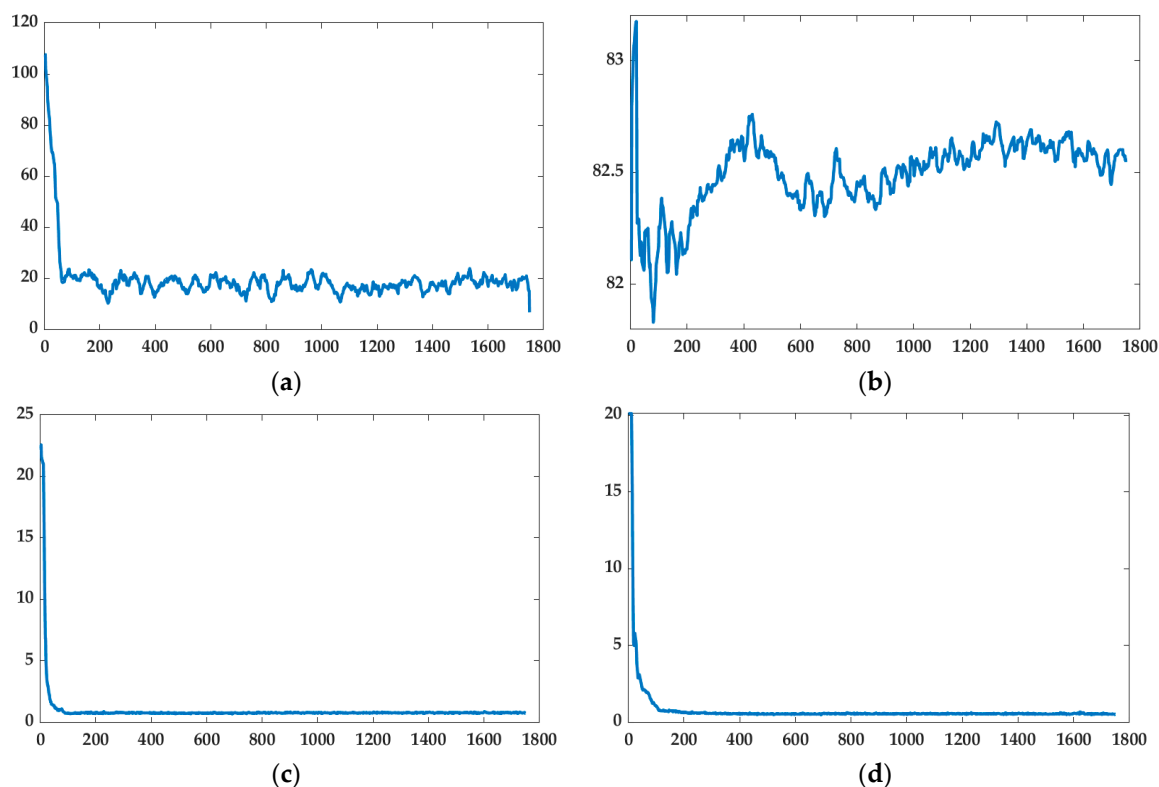


Figure 8. Performance of the MESA Recognizing Environment (a) Total Weighted Loss; (b) Policy Weighted Loss; (c) Value Weighted Loss; (d) Reward Weighted Loss.

Based on the observations illustrated in Figure 8c,d, the loss values for both the Value and Reward networks have decreased to a value within 1. This development suggests that MESA has gained a precise comprehension of the environmental timely rewards and expected average values. However, the loss values for the policy network are gradually converging but are still comparatively high. This outcome could be due to the present level of discretization, which fails to converge to the minimum value. At the same time, due to the improvement in discretization level, the action space has increased, and the probability of encountering two actions with expected values equal to the immediate reward has increased. Additionally, the loss of the policy network has tended to converge, indicating that the results are acceptably good.

4.2.2. Grid-Connected Operation

In contrast to islanded microgrids, grid-connected microgrids necessitate comprehensive consideration of not just their self-sufficiency but the optimal functioning of multi-microgrids. In this study, is designated, and the cumulative reward is determined by averaging the system reward of each microgrid. The obtained outcome is then juxtaposed with the results obtained in the case of islanded operation, as manifested in Figure 9.

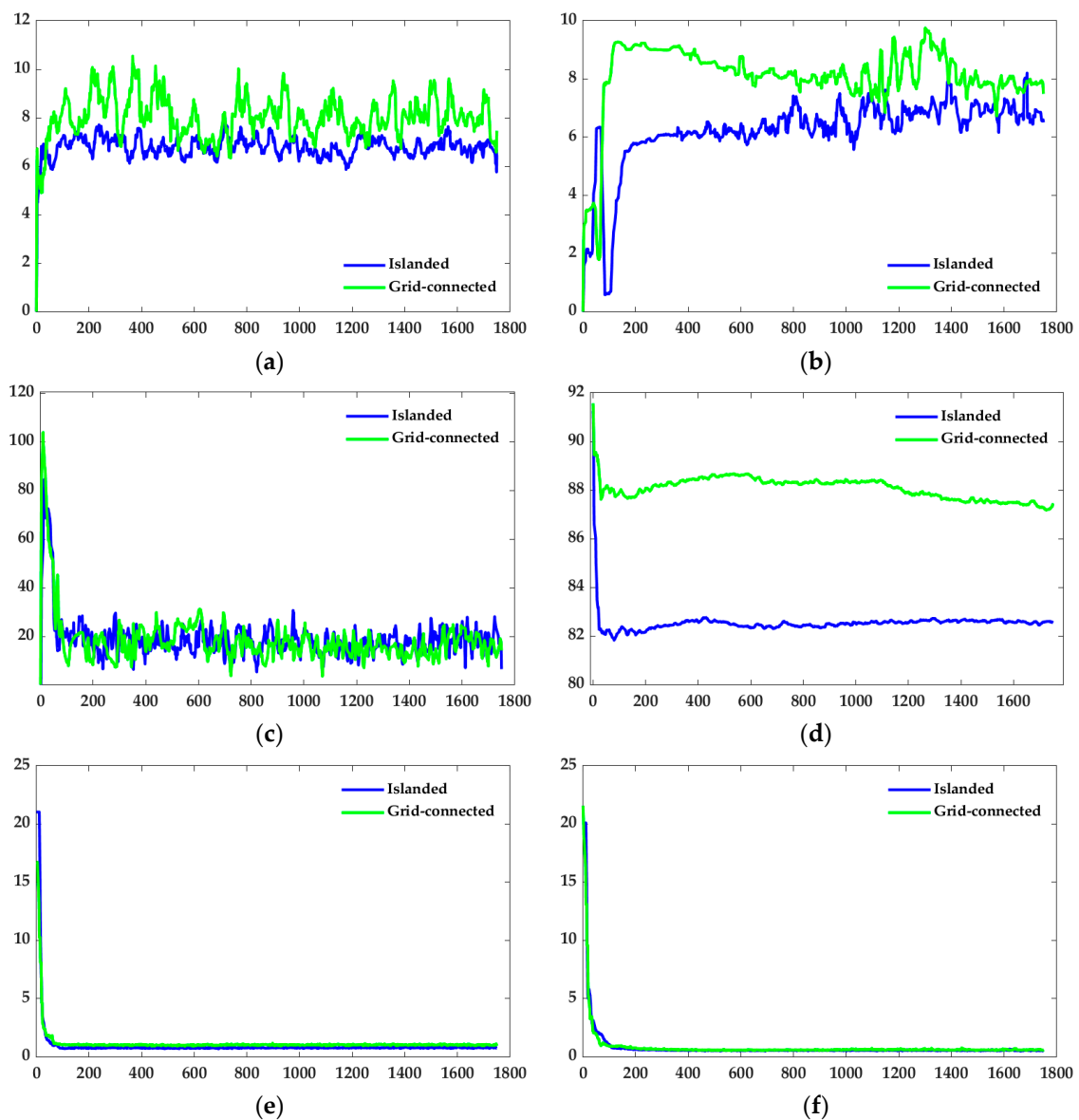


Figure 9. Comparison of Training Results of MESA in Isolated and Grid-connected Modes (a) Total Reward; (b) Mean Value; (c) Total Weights loss; (d) Policy Weights Loss; (e) Value Weights Loss; (f) Reward Weights Loss.

Figure 9a,b demonstrate that the timely rewards and expected average optimization of grid-connected operations surpass those of isolated operations according to IEEE standards. The interaction between microgrids after grid connection enhances their stability, thereby confirming the efficacy of the algorithm in the optimal scheduling of grid-connected operations. It is important to note, however, that the increased complexity of the model leads to slower updates of parameter loss values for the three main strategy networks during grid-connected operations, as seen in Figure 9c–f.

5. Conclusions

In this paper, we thoroughly examined the optimal scheduling problem of microgrids, inclusive of clean energy, and this paper utilizes a model-based reinforcement learning algorithm to resolve the optimal scheduling dilemma of microgrids. The proposed method can effectively acquire the microgrid's operational model through the self-play process of the three learning functions, namely MCTs, Representation, Prediction, and Dynamic. Moreover, it optimizes the operation strategy by updating the learning network's parameters during the self-play process

via game data accumulation and reward function optimization, thereby reducing computational tasks and augmenting the strategy's robustness and adaptability. By establishing a microgrid environment simulator encompassing HVAC, PV, and ES systems, we substantiated that the proposed model-based reinforcement learning method can be successfully adopted for both microgrid island and grid-connected operation modes. The simulation result demonstrates the algorithm proposed is effective for the optimization scheduling of the microgrid system. The convergence of the loss of the value and reward network demonstrates that the algorithm is highly efficient in learning high-value actions in the environment. The reason why the policy network converges to a higher value is that the action space is large, and there are approximate high-value actions that can be chosen.

Author Contributions: Conceptualization, J.Y.; Methodology, J.Y.; Software, J.X.; Validation, Y.G.; Formal analysis, J.X.; Resources, N.Z.; Writing—original draft, J.Y.; Writing—review & editing, N.Z. and Y.G.; Visualization, N.Z. and Y.G. All authors have read and agreed to the published version of the manuscript.

Funding: National Natural Science Foundation of China (62203004).

Data Availability Statement: Data is unavailable due to privacy.

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

References

1. Alanne, K.; Saari, A. Distributed energy generation and sustainable development. *Renew. Sustain. Energy Rev.* **2006**, *10*, 539–558. [\[CrossRef\]](#)
2. Song, Z.; Wang, X.; Wei, B.; Shan, Z.; Guan, P. Distributed Finite-Time Cooperative Economic Dispatch Strategy for Smart Grid under DOS Attack. *Mathematics* **2023**, *11*, 2103. [\[CrossRef\]](#)
3. Li, Y.; Gao, D.W.; Gao, W.; Zhang, H.; Zhou, J. A Distributed Double-Newton Descent Algorithm for Cooperative Energy Management of Multiple Energy Bodies in Energy Internet. *IEEE Trans. Ind. Inform.* **2021**, *17*, 5993–6003. [\[CrossRef\]](#)
4. Zhang, H.; Li, Y.; Gao, D.W.; Zhou, J. Distributed Optimal Energy Management for Energy Internet. *IEEE Trans. Ind. Inform.* **2017**, *13*, 3081–3097. [\[CrossRef\]](#)
5. Parhizi, S.; Lotfi, H.; Khodaei, A.; Bahramirad, S. State of the art in research on microgrids: A review. *IEEE Access* **2015**, *3*, 890–925. [\[CrossRef\]](#)
6. Li, T.; Huang, R.; Chen, L.; Jensen, C.S.; Pedersen, T.B. Compression of Uncertain Trajectories in Road Networks. In Proceedings of the 46th International Conference on Very Large Data Bases, Online, 31 August–4 September 2020; Volume 13, pp. 1050–1063.
7. Rezvani, A.; Gandomkar, M.; Izadbakhsh, M.; Ahmadi, A. Environmental/economic scheduling of a micro-grid with renewable energy resources. *J. Clean. Prod.* **2015**, *87*, 216–226. [\[CrossRef\]](#)
8. Li, Y.; Zhang, H.; Liang, X.; Huang, B. Event-triggered based distributed cooperative energy management for multienergy systems. *IEEE Trans. Ind. Inf.* **2019**, *15*, 2008–2022. [\[CrossRef\]](#)
9. Liu, C.; Wang, D.; Yin, Y. Two-Stage Optimal Economic Scheduling for Commercial Building Multi-Energy System through Internet of Things. *IEEE Access* **2019**, *7*, 174562–174572. [\[CrossRef\]](#)
10. Li, Y.; Gao, D.W.; Gao, W.; Zhang, H.; Zhou, J. Double-Mode Energy Management for Multi-Energy System via Distributed Dynamic Event-Triggered Newton-Raphson Algorithm. *IEEE Trans. Smart Grid* **2020**, *11*, 5339–5356. [\[CrossRef\]](#)
11. Eseye, A.T.; Zheng, D.; Zhang, J.; Wei, D. Optimal energy management strategy for an isolated industrial microgrid using a modified particle swarm optimization. In Proceedings of the 2016 IEEE International Conference on Power and Renewable Energy (ICPRE), Shanghai, China, 21–23 October 2016; pp. 494–498.
12. Zeng, Y.; Zhao, H.; Liu, C.; Chen, S.; Hao, X.; Sun, X.; Zhang, J. Multi objective optimization of microgrid based on Improved Multi-objective Particle Swarm Optimization. In Proceedings of the 2022 International Seminar on Computer Science and Engineering Technology (SCSET), Indianapolis, IN, USA, 8–9 January 2022; pp. 80–83.
13. Elsayed, W.T.; Hegazy, Y.G.; Bendary, F.M.; El-Bages, M.S. Energy management of residential microgrids using random drift particle swarm optimization. In Proceedings of the 2018 19th IEEE Mediterranean Electrotechnical Conference (MELECON), Marrakech, Morocco, 2–7 May 2018; pp. 166–171.
14. Zaree, N.; Vahidinasab, V. An MILP formulation for centralized energy management strategy of microgrids. In Proceedings of the 2016 Smart Grids Conference (SGC), Kerman, Iran, 20–21 December 2016; pp. 1–8. [\[CrossRef\]](#)
15. Picioroaga, I.I.; Tudose, A.; Sidea, D.O.; Bulac, C.; Eremia, M. Two-level scheduling optimization of multi-microgrids operation in smart distribution networks. In Proceedings of the 2020 International Conference and Exposition on Electrical and Power Engineering (EPE), Iasi, Romania, 22–23 October 2020; pp. 407–412.
16. Ma, W.J.; Wang, J.; Gupta, V.; Chen, C. Distributed energy management for networked microgrids using online ADMM with regret. *IEEE Trans. Smart Grid* **2016**, *9*, 847–856. [\[CrossRef\]](#)

17. Fossati, J.P.; Galarza, A.; Martín-Villate, A.; Echeverría, J.M.; Fontán, L. Optimal scheduling of a microgrid with a fuzzy logic controlled storage system. *Int. J. Electr. Power Energy Syst.* **2015**, *68*, 61–70. [\[CrossRef\]](#)
18. Banaei, M.; Rezaee, B. Fuzzy scheduling of a non-isolated micro-grid with renewable resources. *Renew. Energy* **2018**, *123*, 67–78. [\[CrossRef\]](#)
19. Lyu, L.; Shen, Y.; Zhang, S. The Advance of Reinforcement Learning and Deep Reinforcement Learning. In Proceedings of the 2022 IEEE International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA), Changchun, China, 25–27 February 2022; pp. 644–648. [\[CrossRef\]](#)
20. Leo, R.; Milton, R.S.; Kaviya, A. Multi agent reinforcement learning based distributed optimization of solar microgrid. In Proceedings of the 2014 IEEE International Conference on Computational Intelligence and Computing Research, Coimbatore, India, 18–20 December 2014; pp. 1–7.
21. Li, T.; Chen, L.; Jensen, C.S.; Pedersen, T.B. TRACE: Real-time Compression of Streaming Trajectories in Road Networks. In Proceedings of the 47th International Conference on Very Large Data Bases, Copenhagen, Denmark, 16–20 August 2021; Volume 13, pp. 1175–1187.
22. Singh, N.; Elamvazuthi, I.; Nallagownden, P.; Badruddin, N.; Ousta, F.; Jangra, A. Smart Microgrid QoS and Network Reliability Performance Improvement using Reinforcement Learning. In Proceedings of the 2020 8th International Conference on Intelligent and Advanced Systems (ICIAS), Kuching, Malaysia, 13–15 July 2021; pp. 1–6.
23. Shu, Y.; Bi, W.; Dong, W.; Yang, Q. Dueling double q-learning based real-time energy dispatch in grid-connected microgrids. In Proceedings of the 2020 19th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES), Xuzhou, China, 16–19 October 2020; pp. 42–45.
24. Xie, L.; Li, Y.; Xiao, J.; Yang, J.; Xu, B.; Ye, Y. Research on Autonomous Operation Control of Microgrid Based on Deep Reinforcement Learning. In Proceedings of the 2021 IEEE 5th Conference on Energy Internet and Energy System Integration (EI2), Taiyuan, China, 22–24 October 2021; pp. 2503–2507.
25. Skiparev, V.; Belikov, J.; Petlenkov, E. Reinforcement learning based approach for virtual inertia control in microgrids with renewable energy sources. In Proceedings of the 2020 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe), The Hague, The Netherlands, 26–28 October 2020; pp. 1020–1024.
26. Garrido, C.; Marín, L.G.; Jiménez-Estévez, G.; Lozano, F.; Higuera, C. Energy Management System for Microgrids based on Deep Reinforcement Learning. In Proceedings of the 2021 IEEE CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON), Valparaíso, Chile, 6–9 December 2021; pp. 1–7.
27. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. *arXiv* **2017**, arXiv:1707.06347.
28. Weiss, X.; Xu, Q.; Nordström, L. Energy Management of Smart Homes with Electric Vehicles Using Deep Reinforcement Learning. In Proceedings of the 2022 24th European Conference on Power Electronics and Applications (EPE'22 ECCE Europe), Hanover, Germany, 5–9 September 2022; pp. 1–9.
29. Sutton, R.S.; Barto, A.G. Adaptive Computation and Machine Learning. In *Reinforcement Learning: An Introduction*, 2nd ed.; MIT Press: Cambridge, MA, USA, 2018.
30. Deisenroth, M.; Rasmussen, C.E. PILCO: A model-based and data-efficient approach to policy search. In Proceedings of the 28th International Conference on Machine Learning (ICML-11), Bellevue, WA, USA, 28 June–2 July 2011; pp. 465–472.
31. Li, T.; Chen, L.; Jensen, C.S.; Pedersen, T.B.; Gao, Y.; Hu, J. Evolutionary Clustering of Moving Objects. In Proceedings of the IEEE 38th International Conference on Data Engineering, Virtual, 9–12 May 2022; pp. 2399–2411.
32. Heess, N.; Wayne, G.; Silver, D.; Lillicrap, T.; Erez, T.; Tassa, Y. Learning continuous control policies by stochastic value gradients. *Adv. Neural Inf. Process. Syst.* **2015**, *28*.
33. Graepel, T. AlphaGo-Mastering the game of go with deep neural networks and tree search. *Lect. Notes Comput. Sci.* **2016**, *9852*.
34. JLi, J.; Ma, X.-Y.; Liu, C.-C.; Schneider, K.P. Distribution System Restoration with Microgrids Using Spanning Tree Search. *IEEE Trans. Power Syst.* **2014**, *29*, 3021–3029. [\[CrossRef\]](#)
35. Biagioni, D.; Zhang, X.; Wald, D.; Vaidhyanathan, D.; Chintala, R.; King, J.; Zamzam, A.S. Powergridworld: A framework for multi-agent reinforcement learning in power systems. In Proceedings of the Thirteenth ACM International Conference on Future Energy Systems, Online, 28 June–2 July 2022; pp. 565–570.
36. Schrittwieser, J.; Antonoglou, I.; Hubert, T.; Simonyan, K.; Sifre, L.; Schmitt, S.; Guez, A.; Lockhart, E.; Hassabis, D.; Graepel, T.; et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature* **2020**, *588*, 604–609. [\[CrossRef\]](#) [\[PubMed\]](#)
37. Coulom, R. Efficient selectivity and backup operators in Monte-Carlo tree search. In Proceedings of the Computers and Games: 5th International Conference, CG 2006, Turin, Italy, 29–31 May 2006; Revised Papers 5. Springer: Berlin/Heidelberg, Germany, 2007; pp. 72–83.
38. Sinclair, S.; Wang, T.; Jain, G.; Banerjee, S.; Yu, C. Adaptive discretization for model-based reinforcement learning. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 3858–3871.

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.