applied sciences

MDPI

Article

Energy-Efficient Power Scheduling Policy for Cloud-Assisted Distributed PV System: A TD3 Approach

Hao Zhang [1], Fuhao Liu [2], Baichao Ma [3], Shengfang Zhang [4], Yuxin Peng [2], Rentao Gu [2] and Jiansong Miao [2,*]

[1] Economic & Technology Research Institute, State Grid Shandong Electric Power Company, Jinan 250000, China; zhanghao-jyy@sd.sgcc.com.cn
[2] School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China; liufuhao@bupt.edu.cn (F.L.); pyx_110@bupt.edu.cn (Y.P.); rentaogu@bupt.edu.cn (R.G.)
[3] State Grid Heze Power Supply Company, Heze 274000, China; mabaichao@hz.sd.sgcc.com.cn
[4] State Grid Liaocheng Power Supply Company, Liaocheng 252000, China; zhangshengfang@lc.sd.sgcc.com.cn
* Correspondence: miaojiansong@bupt.edu.cn

Abstract: To cope with climate change and other environmental problems, countries and regions around the world have begun to pay attention to the development of renewable energy under the drive of achieving the global carbon emission peak and carbon neutrality goal. The distributed photovoltaic (PV) power grid is an effective solution that can utilize solar energy resources to provide clean a energy supply. However, with the continuous grid connection of distributed energy, it poses great challenges to the power supply stability and security of the grid. Therefore, it is particularly important to promote the local consumption of distributed energy and the construction of the energy internet. This paper aims to study the cooperative operation and energy optimization scheduling problem among distributed PV power grids, and proposes a new scheme to reduce the electricity cost under the constraint of power supply and demand balance. The optimization problem is modeled as a Markov decision process (MDP), and a Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm is used to solve the MDP problem. Simulation results show that the proposed algorithm outperforms other benchmark algorithms in terms of reducing electricity cost, convergence and stability, and verifies its effectiveness.

Keywords: distributed PV system; cloud-assisted network; power scheduling; TD3 algorithm

check for updates

Citation: Zhang, H.; Liu, F.; Ma, B.; Zhang, S.; Peng, Y.; Gu, R.; Miao, J. Energy-Efficient Power Scheduling Policy for Cloud-Assisted Distributed PV System: A TD3 Approach. *Appl. Sci.* **2023**, *13*, 11611. https://doi.org/10.3390/app132111611

Academic Editor: Filipe Soares

Received: 7 September 2023
Revised: 11 October 2023
Accepted: 19 October 2023
Published: 24 October 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

Nowadays, with the continuous aggravation of environmental problems, driven by the carbon peaking and carbon neutralization target [1], it has become an inevitable trend to build a new type of power system, vigorously develop new energy sources and improve energy utilization [2].

More attention is paid to solar energy, with its advantages of environmental protection and convenient availability, by various countries due to the increased demand for clean energy that reduces greenhouse gas emissions in the context of low-carbon development [3]. In particular, the development of distributed photovoltaic (PV) power generation systems, which are decentralized and installed in multiple locations, has effectively improved the efficiency of solar energy utilization [4]. Distributed PV systems offer a number of advantages, firstly they help to reduce greenhouse gas emissions, minimize the impact on the environment and promote sustainable development and environmental protection. Secondly, they can enable regions to reduce their dependence on traditional power supplies and increase energy independence. It can also be interconnected with the grid, injecting excess energy into the grid and reducing the cost of electricity to the users [5].

Despite the many advantages of distributed PV power systems mentioned above, they also face a series of challenges. First, the access of large-scale distributed energy devices increases the complexity of the grid system operation, and it is difficult to characterize the

system through an accurate physical model [6]. Secondly, the intermittent nature of PV power generation and the stochastic nature of user-side behavior lead to a large number of uncertainties contained in distributed PV systems [7]. Finally, flexible loads such as electric vehicles increase the dynamic characteristics of the system, requiring an efficient real-time online energy dispatch strategy [8].

Microgrid as a kind of small-scale distribution system for managing distributed energy can effectively solve the above problems [9]. On the power side, the local consumption of distributed energy can reduce the transmission loss and the impact on the power grid [10]. On the load side, the optimized scheduling of energy can significantly improve the efficiency of energy utilization [11]. Among them, the control strategy for energy storage is a key focus in microgrid energy scheduling research.

Nguyen et al. in [12] consider uncertain outputs of photovoltaic power and charge or discharge efficiencies for the batteries, minimizing the operational cost. In [13], an energy cooperation strategy of two microgrids is developed, in which each microgrid has a separate renewable energy generator and energy storage system, which enables the microgrid to exchange energy and minimize the cost of the energy extracted from the external grid. The methods above for energy scheduling are mostly traditional optimization algorithms, which are difficult to apply when the actual scenarios are more complex and the amount of data is huge in the case of multi-parks [14–17].

In recent years, deep reinforcement learning has become a research hotspot in the field of artificial intelligence, which has the advantage of self-learning through interactive trial-and-error in dynamic environments, and provides new solutions for dealing with complex microgrid control problems [18,19]. Among them, Yang Z et al. propose an energy sharing strategy in [20] based on energy trading to coordinate a multi-energy microgrid with integrated regional energy interconnections, and a distributed optimization of energy allocation using the Alternating Direction Method of Multipliers (ADMM) algorithm. Hua H et al. in [21] give the application of deep reinforcement learning algorithm based on asynchronous dominance in solving the energy scheduling and control problem of model-free, multi-objective, multi-microgrids, and illustrate the basis for the construction of the deep neural network through specific numerical examples. In [22], Lu et al. present a deep RL-based scheme for microgrids to trade energy with each other and with the main power grid to increase the utility of microgrids and reduce the dependence on the main grid. Ref. [23] addresses the impact of renewable energy on the frequency stability of the power system, and proposes a load frequency coordinated control scheme based on the deep deterministic policy gradient (DDPG) algorithm for the distributed PV power network. The scheme considers the nonlinear constraints of the system, the load fluctuations, and the randomness of PV and the energy storage system (ESS), and learns the optimal control strategy through deep reinforcement learning methods, the experimental results in both the time domain and frequency domain demonstrate the superiority of the scheme in uncertain environments.

However, existing studies mainly focus on applying reinforcement learning to single microgrid scenarios, and neglect the inter-microgrid energy scheduling problem. To fill this gap, we propose an efficient power scheduling policy for the multi-parks distributed PV system in this paper. Our main contributions are summarized as follows:

- We design a cloud-assisted distributed PV system for multi-park, which enables an internal trading market to coordinate the power allocation among different parks and achieve the supply-demand balance in the system.
- The optimization problem is formulated as a Markov decision process (MDP), and solved by using the TD3 algorithm, which is a state-of-the-art reinforcement learning method that can handle continuous action spaces.
- We conduct extensive simulations with various parameters to assess the performance of the proposed TD3-based optimization scheme. The simulation results and numerical analysis show that our scheme can significantly reduce the electricity cost of all parks compared with other baseline schemes.

The rest of this paper is organized as follows. The model of the PV system in each park and the power trading system are written in Section 2. The problem formulation and the solution with TD3 are placed in Sections 3 and 4, respectively. Then, the simulation results are discussed in Section 5. Finally, the whole paper is summarized in Section 6.

2. System Model

2.1. Network Model

As shown in Figure 1, we consider a cloud-assisted multi-park distributed PV system, where an internal trading market in the cloud facilitates the communication and coordination among the parks. The electric energy can be transferred between different parks via an energy bus. This market enables the energy-rich parks to sell their surplus power to other parks at a favorable price, rather than to the external grid at a low price, and the energy-deficient parks to buy the power at a low price. The internal market settles the total cost of each park based on their energy transactions. Therefore, the multi-park cooperative operation based on the internal market can exploit the energy complementarity and increase the self-supply and self-consumption ratio of the distributed PV system.

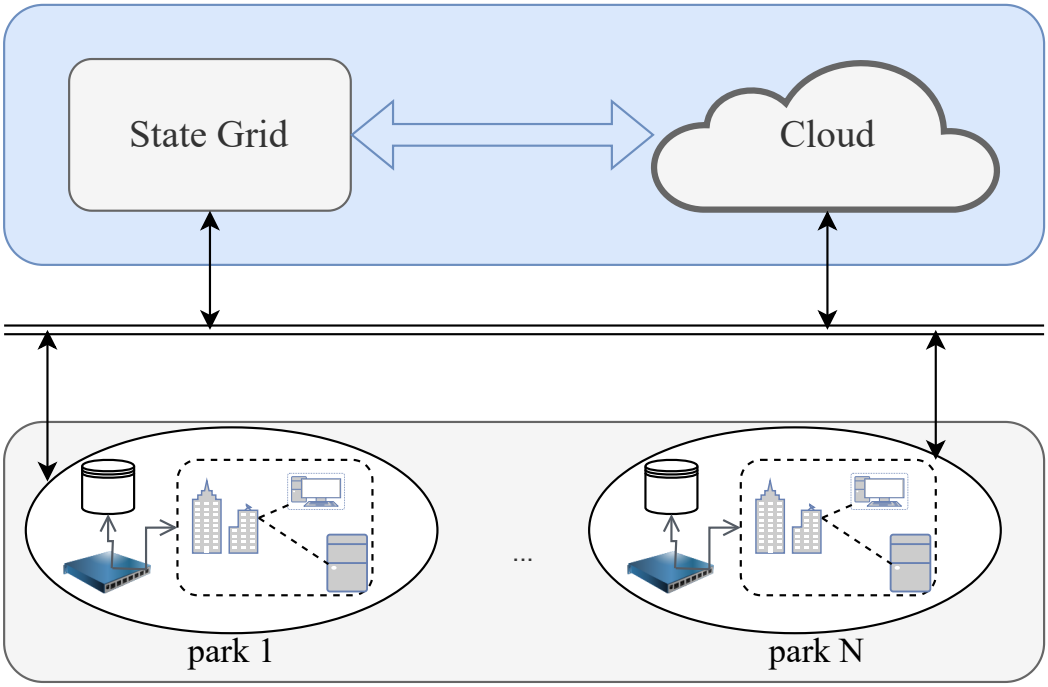


Figure 1. Cloud-assisted multi-park distributed PV system.

In the proposed cloud-assisted multi-park distributed PV system, each park communicates its planned energy purchase or sale information to an internal trading market. The internal trading market then allocates the electric power among the parks to achieve the balance of supply and demand in the system.

To enable efficient and robust energy exchange and optimization, we employ a distributed coordination mechanism based on an intelligent agent. Each park has a local intelligent agent, which optimizes its charging and discharging amount of the energy storage device according to the current electricity supply and demand, electricity price, and other information of all parks. Instead of using a fixed priority rule, we schedule the priority of the energy-demanding and supply-given parks based on an intelligent mechanism, which allows each agent to adapt its strategy and behavior dynamically according to its objective function (i.e., minimizing its electricity cost). Our system can automatically balance the energy flow among the parks according to the real-time market situation, so as to minimize the total electricity cost, while ensuring that each park can meet its own demand.

2.2. Single Park Operation Model

The set of park is denoted as $\mathcal{P} = \{1, 2, \ldots, n\}$. Each single park consists mainly of a PV device, a power storage device and users. The park manager can get the operation schedule policy from the cloud and then deals with the external grid or other parks for electricity.

We use net metering to calculate the total electricity cost of each park, which means that we record the electricity delivered to and obtained from the external grid or other parks by the distributed PV park through a bidirectional meter, and settle according to certain pricing rules. Therefore, the operating cost of electricity of the park i in the time slot t is

$$C_{i,t} = \sum_{t=1}^{T} \left(C_{i,t}^{grid} + C_{i,t}^{in} \right) \tag{1}$$

where $C_{i,t}^{grid}$ and $C_{i,t}^{in}$ are total transaction costs with the external grid and other parks, respectively, which can be given as

$$C_{i,t}^{grid} = p_{t,b}^{grid} E_{i,t,b}^{grid} - p_{t,s}^{grid} E_{i,t,s}^{grid} \tag{2}$$

$$C_{i,t}^{in} = p_{t,b}^{in} E_{i,t,b}^{in} - p_{t,s}^{in} E_{i,t,s}^{in} \tag{3}$$

where $p_{t,b}^{grid}$, $p_{t,s}^{grid}$ denote the price purchased or sold of the external grid, $p_{t,b}^{in}$, $p_{t,s}^{in}$ denote the price in the internal trading market, analogously. $E_{i,t,b}^{grid}$, $E_{i,t,s}^{grid}$ denote bought or sold electricity from the external grid, and $E_{i,t,b}^{in}$, $E_{i,t,s}^{in}$ denote the bought or sold electricity from the internal trading market, respectively.

2.3. Market Cleaning Mechanism

In this paper, an internal trading market established in the cloud is used to realize the settlement of energy transactions between parks, and each park only needs to provide its own power buying or selling information to the cloud.

Referring to the micro-grid internal electricity market settlement scheme [24], the cleaning price of the electricity trading market can be expressed as

$$p_{t,s} = \begin{cases} \frac{p_t^{in} E_{t,b} + p_{t,s}^{grid} \left(E_{t,s} - E_{t,b} \right)}{E_{t,s}}, & \text{if } E_{t,s} \geq E_{t,b} \\ p_t^{in}, & \text{if } E_{t,s} < E_{t,b} \end{cases} \tag{4}$$

$$p_{t,b} = \begin{cases} p_t^{in}, & \text{if } E_{t,s} \geq E_{t,b} \\ \frac{p_t^{in} E_{t,s} + p_{t,b}^{grid} \left(E_{t,b} - E_{t,s} \right)}{E_{t,b}}, & \text{if } E_{t,s} < E_{t,b} \end{cases} \tag{5}$$

where $p_{t,s}$ and $p_{t,b}$ are the current selling and buying price in the trading market. $E_{t,s}$ and $E_{t,b}$ denote the total amount of selling and buying electricity by the all parks, respectively. p_t^{in} denotes internal cleaning price threshold, i.e., the maximum price of energy sold and the minimum price of energy purchased, which can be given by

$$p_t^{in} = \frac{p_{t,s}^{grid} + p_{t,b}^{grid}}{2} \tag{6}$$

From the cleaning Formulas (4) and (5), we can know that the cleaning price is jointly determined by all parks. When electricity supply exceeds demand among the parks, the internal energy purchase price will reach the lower limit, which is much lower than the price of energy purchased from the external grid; otherwise, the internal energy sale price will reach the upper limit, which is much higher than the price of energy sold to the external grid.

2.4. Energy Storage Device Operation Model

The cloud-assisted distributed photovoltaic (PV) system is a novel architecture that integrates PV generation, energy storage devices, and cloud computing. In this system, the information of PV energy $E_{i,t}^{pv}$, electric loads $E_{i,t}^{load}$ and energy storage $SOC_{i,t}$ in each park i for each time slot t will be collected and uploaded to the cloud server. The cloud server acts as a park manager that can optimize the operation of the energy storage devices and the internal market transactions among the parks. The park manager can calculate the optimal buying or selling information for each park, which would be uploaded to the internal market, after selecting the optimal charging or discharging power of the energy storage device. Let $\varphi_i (i \in \{1, 2, ...n\})$ denote the charging or discharging power of park i in the time slot t, where $\varphi_i > 0$ means that the energy storage device chooses to charge from the grid or PV generation, and $\varphi_i < 0$ means that the energy storage device chooses to discharge to the grid or load demand. Then the energy storage action vector of all parks in the time slot t can be defined as $\phi = [\varphi_1, \varphi_2, \ldots, \varphi_n]$. The objective of the park manager is to maximize the total profit of all parks by determining ϕ for each time slot, subject to some constraints such as power balance, energy storage capacity, and market clearing.

3. Problem Formulation

In this section, we formulate the electricity scheduling policy for cloud-assisted multi-park distributed PV system as an optimization problem, and the objective is to minimize the total cost of all parks in the system. The problem can be formulated as follows:

$$\min_{\phi} \sum_{i=1}^{n} C_{i,t} \tag{7}$$

$$s.t. \quad C1: E_{i,t}^{load} = E_{i,t,s} + E_{i,t,b} + E_{i,t}^{pv} + E_{i,t}^{ess} \tag{8}$$

$$C2: -E_{i,t,max}^{ess} \leq E_{i,t}^{ess} \leq E_{i,t,max}^{ess} \tag{9}$$

$$C3: SOC_{i,t+1} = SOC_{i,t} + E_{i,t}^{ess} \tag{10}$$

$$C4: 0 \leq SOC_{i,t} \leq SOC_{max} \tag{11}$$

where $E_{i,t,s}, E_{i,t,b}$ denote the amount of buying or selling power of park i, $E_{i,t,max}^{ess}$ indicates the maximum power of charging or discharging, $SOC_{i,t}$ denotes the power status of the energy storage system at the current moment, SOC_{max} is the upper boundaries of the electric storage charge state, respectively.

$C1$ denotes that the internal operation of the park i meets the balance of power supply and demand. $C2$ and $C4$ indicate that the action and the power amount of the storage device are limited. $C3$ indicates that the power of the energy storage device at the next moment is equal to the current moment plus the charging and discharging strategy at the current moment.

The original problem (7) can be addressed by working out the optimal solution of the energy storage action vector ϕ, which is a continuous variable. Instead of solving the problem by conventional optimization methods, we propose Deep Reinforcement Learning (DRL) algorithm to find the optimal ϕ.

4. Problem Solution

4.1. Constrained Markov Decision Process

In this section, the above optimization problem (7) is modeled as a constrained MDP, which is a tuple $\langle \mathcal{S}, \phi, r \rangle$, and the details of each element are described as follows.

- State: The state consists of two components $\mathcal{S} = [SOC_t, C_t^{all}]$, where SOC_t is the power state of the energy storage system and C_t^{all} represents the total cost of all parks at the current moment.

- Action: ϕ is the action space that contains the charging or discharging power policy $\varphi_i \in [\varphi_1, \varphi_2, \ldots, \varphi_n]$, which satisfies $\varphi_i > 0$ if the energy storage device chooses to charge, otherwise, $\varphi_i < 0$. And φ_i is a continuous variable.
- Reward: For each step in the iteration process, the agent will get a reward in a certain state $\mathcal{S}$ after every possible action in ϕ, which should be related to the objective function. Since the objective of our optimization problem is to obtain the minimal sum cost and the purpose of the RL is to obtain the maximum reward, the value of the reward function should be negatively correlated to the total cost. At the same time, to unify the order of magnitude, the immediate reward can be given by

$$reward = -\sum_{i=1}^{n} C_{i,t} - punish \tag{12}$$

$$punish = \alpha E_{i,t}^{\text{ess}-\text{overc}} + \beta E_{i,t}^{\text{ess}-\text{overd}} \tag{13}$$

where α and β are the penalty factors, $E_{i,t}^{\text{ess}-\text{overc}}$ and $E_{i,t}^{\text{ess}-\text{overd}}$ are the overcharge and overdischarge of energy storage device, respectively.

4.2. The Solution Based on TD3

This section proposes an energy scheduling strategy based on the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm, and theoretically demonstrates the superior performance of the TD3 algorithm by comparing it with the Deep Deterministic Policy Gradient (DDPG) algorithm. First, the DDPG algorithm is introduced in detail, then the shortcomings of the DDPG algorithm are analyzed, and finally, the TD3 algorithm is presented as an improvement of the DDPG algorithm by contrast.

4.2.1. DDPG Algorithm

DDPG algorithm is a kind of reinforcement learning algorithm that can handle continuous action spaces, and it combines the advantages of Deep Q-Network (DQN) algorithm and Deterministic Policy Gradient (DPG) algorithm, and can effectively learn the optimal policy in high-dimensional and complex environments.

DDPG algorithm is based on the Actor-Critic framework, which uses two deep neural networks to approximate the value function and the policy function, respectively. One is the Critic network, which evaluates the action value of the current state, and the other is the Actor network, which generates the optimal action of the current state. The objective of the Critic network is to minimize the mean squared error between the current action value and the target action value, while the objective of the Actor network is to maximize the action value given by the Critic network. To improve the stability and convergence of learning, DDPG algorithm also uses the techniques of Experience Replay (ER) and Target Network (TN), that is, in each iteration, a mini-batch of state-action-reward-next state tuples is randomly drawn from the replay buffer as training data, and a fixed target network is used to calculate the target action value, instead of using the real-time updated network. The procedure of DDPG algorithm is shown in Algorithm 1 and the architecture is shown in Figure 2.

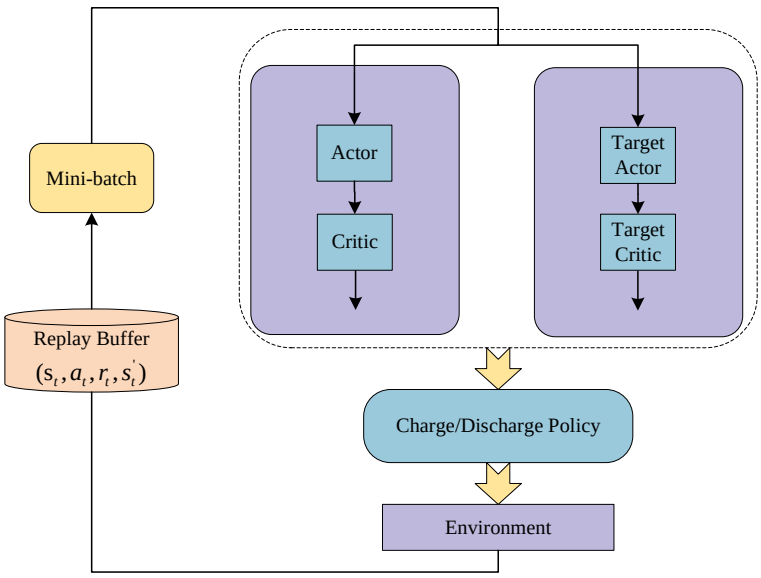


Figure 2. The structure of DDPG algorithm.

Algorithm 1 DDPG

1: Initialize the Actor network π_ϕ of and the Critic network Q_θ with random weights ϕ and θ, as well as the target Actor network $\pi_{\phi'}$ and the target Critic network $Q_{\theta'}$ with the same weights of the original network parameters, i.e., $\phi^{'} \leftarrow \phi, \theta^{'} \leftarrow \theta$.
2: Initialize a replay buffer D, which is used to store the sampled tuples (s_t, a_t, r_t, s_t'), where s_t is the state, a_t is the action, r_t is the reward, s_t' is the next state in the MDP.
3: **for** each episode $t = 1$ to T **do**
4: Initialize a random process N, which is used to add exploration noise to the Actor network.
5: Observe the initial state s_1.
6: **for** each time step t **do**
7: Select an action $a_t = \pi_\phi(s) + \mathcal{N}_t$ according to the current policy and exploration noise.
8: Execute the action and observe the reward r_t and the next state $s^{'}$.
9: Store the transition tuple $(s_t, a_t, r_t, s^{'})$ in the replay buffer D.
10: Randomly sample a mini-batch of tuples from the replay buffer D, denoted as (s_i, a_i, r_i, s_i'), where $i = 1, 2, ..., N$ and N is the batch size.
11: For each sampled transition tuple, calculate the target action value $y_i = r_i + \gamma Q_{\theta'}\left(s_i', \pi_{\phi'}\left(s_i'\right)\right)$, where γ is the discount factor.
12: Update the critic network by minimizing the mean squared error loss, i.e., $\theta \leftarrow \theta - \alpha \bigtriangledown_\theta \frac{1}{N}\sum_{i=1}^{N}\left(Q_\theta\left(s_i, \pi_\phi(s_i)\right) - y_i\right)^2$, where α is the learning rate.
13: Update the actor network by using the sampled policy gradient, i.e., $\phi \leftarrow \phi - \alpha \bigtriangledown_\phi \frac{1}{N}\sum_{i=1}^{N} Q_\theta\left(s_i, \pi_\phi(s_i)\right)$.
14: Update the target networks by using soft updates, i.e., $\theta^{'} \leftarrow \tau\theta + (1-\tau)\theta^{'}$, $\phi^{'} \leftarrow \tau\phi + (1-\tau)\phi^{'}$, where τ is the update rate.
15: If s_t' is a terminal state, end the episode, otherwise, let $t \leftarrow t + 1$, and continue the next time step.
16: **end for**

However, DDPG algorithm also has some drawbacks and problems, mainly in the following two aspects:

- Overestimation bias Since DDPG algorithm uses the same network to select and evaluate actions, it tends to overestimate the Q-values, which affects the quality of the policy. This is because when calculating the target action value, DDPG algorithm uses a maximization operation. The problem with this is that if the target Critic network has some errors in the Q-values of some actions, then the maximization operation will amplify these errors, leading to overestimation of the target action value, which affects the update of the current Critic network and Actor network.
- Policy instability: Since DDPG algorithm updates the Actor network and the target network in each iteration, this may lead to policy instability, which affects the convergence of the policy. This is because in DDPG algorithm, the Actor network and the Critic network are interdependent, that is, the update of the Actor network depends on the action value given by the Critic network, and the update of the Critic network depends on the action generated by the Actor network. If the update frequency of the Actor network and the target network is too high, then the Critic network may not be able to adapt to the changes of the Actor network in time, resulting in inaccurate action values, which affects the update of the Actor network.

4.2.2. TD3 Algorithm

Twin Delayed Deep Deterministic Policy Gradient (TD3) is an improved version of the DDPG algorithm, which addresses the issues of overestimation bias and policy instability in DDPG algorithm, and proposes three key techniques:

- Twin critics: TD3 algorithm uses two Critic networks to evaluate the action value of the current state, denoted as Q_{θ_1} and Q_{θ_2}. When calculating the target action value, TD3 algorithm uses the minimum value of the two Critic networks, that is,

$$y_i = r_i + \gamma \min_{i=1,2} Q_{\theta_i'}\left(s_i', \pi_{\phi'}\left(s_i'\right)\right) \tag{14}$$

  where r_i is the reward, γ is the discount factor, $Q_{\theta_i'}$ and $\pi_{\phi'}$ are the target Critic network and the target Actor network, respectively. The purpose of this is to avoid overestimation bias, and improve the quality of the policy.
- Delayed policy update: TD3 algorithm uses a delayed policy update mechanism, that is, only after the Critic network is updated d times, the Actor network and the target network are updated once. The purpose of this is to reduce the frequency of policy update, and reduce the variance of the policy, and improve the stability of the policy.
- Target policy smoothing: TD3 algorithm uses a target policy smoothing technique, that is, when calculating the target action value, a random noise is added to the action generated by the target Actor network, that is,

$$y_i = r_i + \gamma \min_{i=1,2} Q_{\theta_i'}\left(s_i', \pi_{\phi'}\left(s_i'\right) + \epsilon\right) \tag{15}$$

  where $\epsilon \sim N(0, \sigma)$ is a random noise that follows a normal distribution, σ is the standard deviation of the noise, γ is the discount factor, $Q_{\theta_i'}$ and $\pi_{\phi'}$ are the target Critic network and the target Actor network, respectively. The purpose of this is to prevent the policy from being too greedy, and increase the exploration of the policy, and improve the robustness of the policy.

The process of TD3 algorithm is basically consistent with that of DDPG algorithm, except that the three techniques mentioned above are used when calculating the target action value, updating the Critic network, and updating the Actor network. The pseudocode of TD3 algorithm is shown in Algorithm 2, and the architecture diagram of TD3 algorithm is shown in Figure 3.

Algorithm 2 TD3

1: Initialize the Actor network π_ϕ of and the Critic network Q_{θ_1} and Q_{θ_2}, as well as the target Actor network $\pi_{\phi'}$ and the target Critic network $Q_{\theta_1'}$ and $Q_{\theta_2'}$, where the target network parameters are the same as the network parameters, i.e., $\phi^{'} \leftarrow \phi, \theta_i' \leftarrow \theta_i$, $i = 1, 2$.
2: Initialize a replay buffer D, which is used to store the sampled tuples (s_t, a_t, r_t, s_t'), where s_t is the state, a_t is the action, r_t is the reward, s_t' is the next state in the MDP..
3: **for** each episode $t = 1$ to T **do**
4: Initialize a random process N, which is used to add exploration noise to the Actor network.
5: Observe the initial state s_1 from the environment.
6: **for** each time step t **do**
7: According to the current Actor network, select an action $a_t = \pi_\phi(s) + N(0, \sigma)$, where σ is the standard deviation of the noise and then execute the action.
8: Observe the reward r_t and the next state $s^{'}$.
9: Store the transition tuple $(s_t, a_t, r_t, s^{'})$ in the replay buffer D.
10: Randomly sample a mini-batch of tuples from the replay buffer D, denoted as (s_i, a_i, r_i, s_i'), where $i = 1, 2, ..., N$ and N is the batch size.
11: For each sampled transition tuple, calculate the target action value $y_i = r_i + \gamma \min_{i=1,2} Q_{\theta_i'}\left(s_i', \pi_{\phi'}(s_i') + \epsilon\right)$, where γ is the discount factor and ϵ is a random noise sampled from $N(0, \sigma)$.
12: Use the mean squared error as the loss function, update the critic network parameters, i.e., $\theta_i \leftarrow \theta_i - \alpha \bigtriangledown_{\theta_i} \frac{1}{N}\sum_{i=1}^{N}\left(Q_{\theta_i}(s_i, a_i) - y_i\right)^2, i = 1, 2$, where α is the learning rate.
13: If the current training step is an integer multiple of the delayed update frequency, i.e., $stepmodd = 0$, then update the actor network parameters, i.e., $\phi \leftarrow \phi - \alpha \bigtriangledown_\phi \frac{1}{N}\sum_{i=1}^{N} Q_{\theta_1}\left(s_i, \pi_\phi(s_i)\right)$, where α is the learning rate and d is the delayed update frequency.
14: Update the target networks by using soft updates, i.e., $\theta_i^{'} \leftarrow \tau\theta_i + (1-\tau)\theta_i^{'}, \phi^{'} \leftarrow \tau\phi + (1-\tau)\phi^{'} i = 1, 2$, where τ is the update rate. Repeat step 3 until the preset number of training epochs is reached or the termination condition is satisfied.
15: **end for**

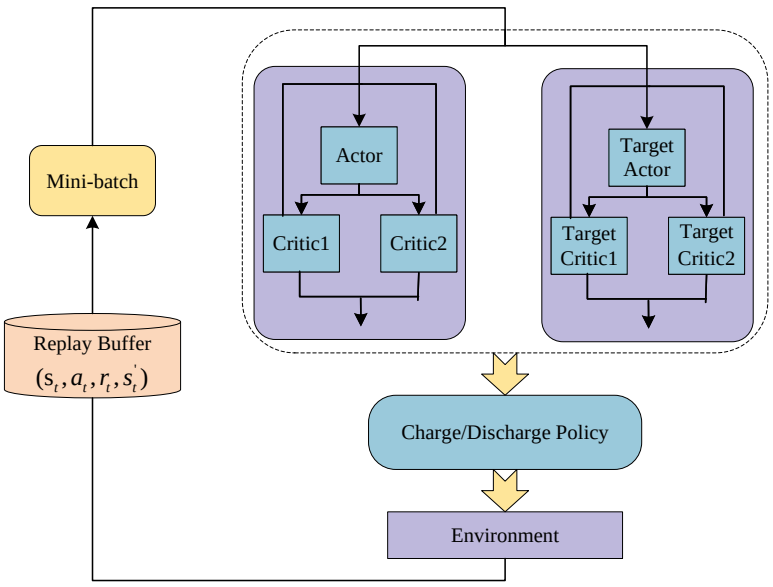


Figure 3. The structure of TD3 algorithm.

4.2.3. The O2 Complexity of TD3 Algorithm

The computational complexity of TD3 algorithm refers to the amount of computational resources and time required to perform one iteration of the algorithm. Generally speaking, the lower the computational complexity, the more efficient the algorithm.

The computational complexity of TD3 algorithm depends on several factors:

- The dimensionality of the state space S
- The dimensionality of the action space A
- The number of parameters P in the neural networks
- The size N of the experience replay buffer
- The batch size B for sampling from the buffer
- The frequency F for updating the policy and target networks

According to the pseudocode of TD3 algorithm, we can roughly estimate the computational complexity of TD3 algorithm as

$$O(TD3) = O(sample) + O(uplate) \tag{16}$$

where $O(sample) = O(B)$,$O(update) = O(B(S + A + P) + FP)$. Therefore, the O2 complexity of TD3 algorithm can be expressed as

$$O(TD3) = O(B) + O(B(S + A + P) + FP) \tag{17}$$

5. Simulation Results and Analysis

5.1. Simulation Parameters

In this section, in order to verify the effectiveness and superiority of the power scheduling scheme based on the TD3 algorithm proposed in this paper in a multi-parks distributed PV grid scenario, comparative experiments are conducted in this paper between this scheme and two other benchmark schemes, the DDPG algorithm and the SAC algorithm. These algorithms are run on a Python-based simulator. The parameters are shown in Table 1.

Table 1. Parmeter table.

Notation	Definition	Value
$p_{t,s}^{grid}$	Price sold to the grid	0.1 Yuan/kW·h
$p_{t,b}^{grid}$	Price bought from grid	0.3 Yuan/kW·h
$E_{i,t}^{load}$	Park electrical load	$[200, 1000]$ kW·h
$E_{i,t}^{pv}$	Photovoltaic power generation	$[0, 1200]$ kW·h
$E_{i,t,max}^{ess}$	The maximum of action	1500 kW·h
SOC_{max}	The maximum of storage	3000 kW·h
α, β	penalty factors	20

In the simulation, we generated the PV cases by referring to some practical scenarios in [25,26]; however, we did not directly use the original data models. Instead, we changed the PV data to a uniform distribution within a certain range, which can make the model have generalization and robustness. The park electrical load $E_{i,t}^{load}$ and the photovoltaic power $E_{i,t}^{pv}$ change over time slot t. The interval of $E_{i,t}^{load}$ is $[200, 1000]$ kW·h and the interval of $E_{i,t}^{pv}$ is $[0, 1200]$ kW·h. They are all uniformly distributed and have no memory. When the simulation system starts running, each park would have a randomized value in the interval at the time slot t.

5.2. Convergence Performance of Different Parameters

In order to analyze the convergence performance of the TD3 algorithm, this section conducts experiments with different learning rates (lr) and discount factors (gamma) in a scenario where the number of parks is six and the random seed is fixed to ensure that

the time sequence of electric load and power generation is the same. We refer to the TD3 algorithm parameterization in [27]. Figure 4 shows the convergence performance of TD3 algorithm w.r.t. different learning rate lr and discount factor γ. The learning rate lr is considered as the ratio of new sample information to the previously acquired information, which is usually fixed as a small value to ensure the stability and convergence of the learning process. The discount factor γ determines the importance of future reward in learning. It is observed that all curves converge around 200 episodes. The TD3 reaches a higher reward when lr is 0.0001 and γ is 0.9. Therefore, in the following simulation, the learning rate is fixed as $lr = 0.0001$, and the discount factor is fixed as $\gamma = 0.9$.

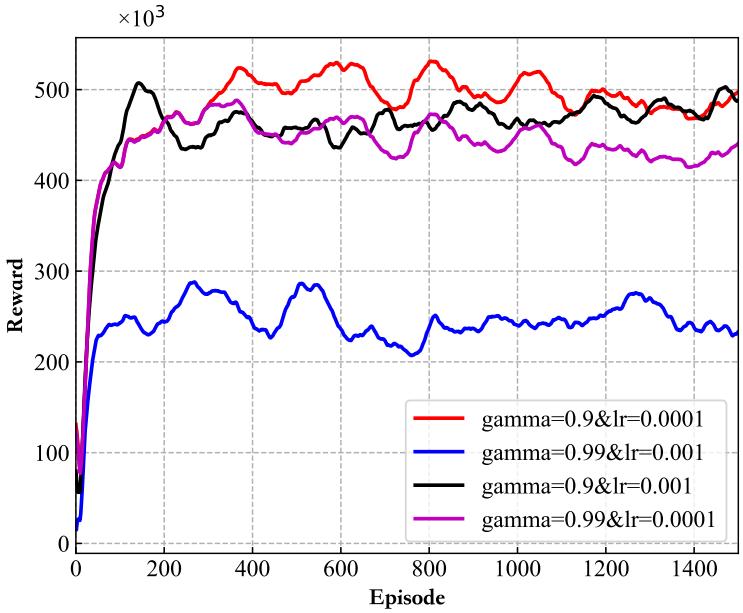


Figure 4. Convergence performance of different parameters. The episode is the t in Algorithm 2.

5.3. Convergence Performance Comparison with Benchmark Algorithms

In order to verify the superior convergence performance of the power scheduling strategy based on the TD3 algorithm, experiments are conducted with different algorithms set to the same lr and gamma γ with a number of six parks and the random seed is fixed to ensure that the time sequence of electric load and power generation is the same. The parameter comparison table of TD3 algorithm, DDPG algorithm and Soft Actor-Critic (SAC) algorithm is shown in Table 2. We set the learning rate to a low value to ensure stability and convergence, the discount factor to a high value to enhance long-term planning ability, the target update rate to a low value to maintain the smoothness and stability of the target network parameters, the replay buffer size to a large value to increase sample diversity and utilization efficiency, and the batch size to a moderate value to balance sampling efficiency and training efficiency.

Table 2. The parameter comparison table of TD3 algorithm and benchmark algorithm.

Parameter	TD3	DDPG	SAC
Number of critic networks	Two	One	Two
Target value calculation	Minimum of two target critics	Single target critic	Minimum of two target critics
Action selection	Target actor output plus noise	Actor output	Reparametrized stochastic actor output
Exploration noise	Gaussian noise	Ornstein-Uhlenbeck	None
Target update rate	0.005	0.001	0.005
Entropy regularization	No	No	Yes
Learning rate α	10^{-4}	10^{-4}	10^{-4}, 3×10^{-4}
Discount factor γ	0.9	0.9	0.9
Replay buffer size	10^6	10^6	10^6
Batch size	256	256	256

Figure 5 shows the convergence performance of TD3, DDPG and SAC algorithms. From Figure 5, it is observed that the reward of TD3 is much higher than DDPG and SAC when it reaches stability and TD3 has lower fluctuation, which means the performance of TD3 is relatively stable.

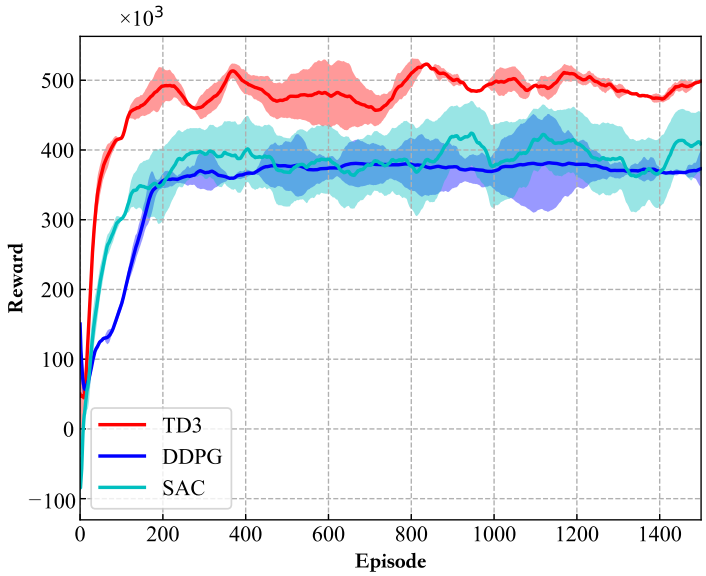


Figure 5. Reward of different algorithms. The shadowed area represents the range of reward fluctuation of three curves in 1500 episodes. The reward in each episode will not exceed the shadowed area.

5.4. System Average Cost Comparison with Benchmark Algorithms for Different Number of Parks

In order to evaluate the performance of power scheduling strategies based on different algorithms under different numbers of parks, this section compares the total system average cost of the TD3 algorithm-based strategy proposed in this paper with other benchmark strategies in a scenario with fixed PV generation and park electrical load.

Figure 6 shows the cost performance of different algorithms w.r.t. the number of parks in the system. From the figure, it is observed that the TD3 algorithm achieves a better result for its lower-cost performance.

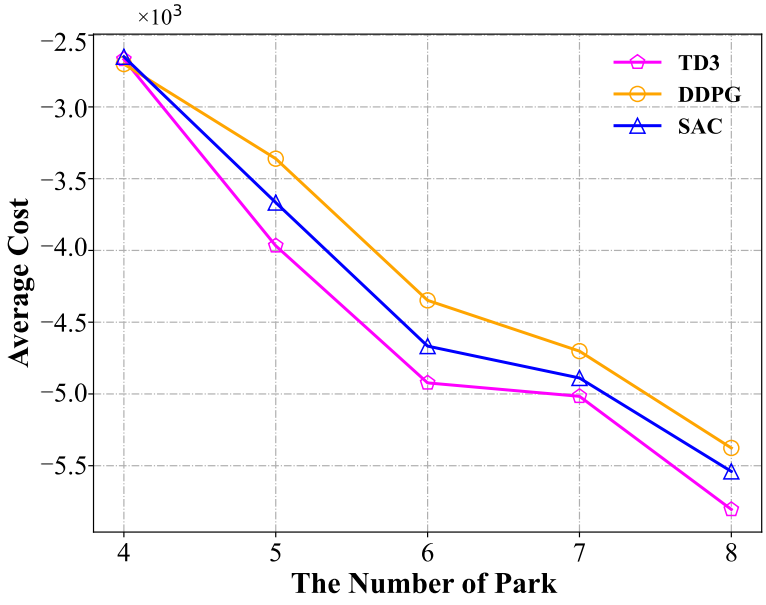


Figure 6. Average cost of algorithms for different number of parks.

6. Conclusions

In this paper, a novel method of electricity scheduling in cloud-assisted multi-park distributed PV power architecture was considered. The study aimed to minimize the cost of all parks while satisfying the constraints of the balance of power supply and demand. The problem was modeled as an MDP problem, and a TD3 algorithm was employed to solve the MDP problem. Simulation results were presented to show the effectiveness and stability of total cost of electricity used by the parks compared with other benchmarks. The future work will deploy a large-scale number of the parks in the system, and jointly consider how to efficiently implement online training.

Author Contributions: Conceptualization, H.Z., F.L. and B.M.; methodology, H.Z., F.L. and S.Z.; software, Y.P. and F.L.; validation, F.L. and Y.P.; formal analysis, F.L.; investigation, H.Z., F.L., B.M. and J.M.; resources, H.Z., B.M. and S.Z.; data curation, F.L. and Y.P.; writing—original draft preparation, H.Z., F.L. and S.Z.; writing—review and editing, H.Z., F.L., J.M., S.Z. and R.G.; visualization, F.L.; supervision, J.M. and R.G.; project administration, H.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by State Grid Corporation of China Project (5700-202316254A-1-1-ZN).

Data Availability Statement: All datasets are publicly available.

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

References

1. Wang, X.; Huang, G.; Qiao, Z.; Li, X. Contribution of distributed energy resources management to peak carbon dioxide emissions and carbon neutralization. In Proceedings of the 2021 IEEE Sustainable Power and Energy Conference (iSPEC), Nanjing, China, 23–25 December 2021; pp. 2101–2107. [CrossRef]
2. Palu, S.M.P.H.S.W. Resiliency oriented control of a smart microgrid with photovoltaic modules. *Glob. Energy Interconnect.* **2021**, *4*, 441–452.
3. Di, G.; Zhifei, C.; Jitao, N.; Liang, Y. Reactive Power Response Behaviour Modelling and Prediction Algorithm of Distributed Photovoltaic Generation. In Proceedings of the 2021 11th International Conference on Power and Energy Systems (ICPES), Shanghai, China, 18–20 December 2021; Volume 1, pp. 761–766. [CrossRef]
4. Sutikno, T.; Arsadiando, W.; Wangsupphaphol, A.; Yudhana, A.; Facta, M. A Review of Recent Advances on Hybrid Energy Storage System for Solar Photovoltaics Power Generation. *IEEE Access* **2022**, *10*, 42346–42364. [CrossRef]
5. Gaddala, R.K.; Singh, V.; Pasupuleti, P. Energy Management and Control of Combined Hybrid Energy Storage and Photovoltaic Systems in DC Microgrid Applications. In Proceedings of the 2023 IEEE Kansas Power and Energy Conference (KPEC), Manhattan, KS, USA, 25–26 April 2023; pp. 1–5. [CrossRef]
6. Zhou, X.; Wang, J.; Wang, X.; Chen, S. Review of microgrid optimization operation based on deep reinforcement learning. *J. Glob. Energy Interconnect.* **2023**, *6*, 240–257.
7. Heider, A.; Kundert, L.; Schachler, B.; Hug, G. Grid Reinforcement Costs with Increasing Penetrations of Distributed Energy Resources. In Proceedings of the 2023 IEEE Belgrade PowerTech, Belgrade, Serbia, 25–29 June 2023; pp. 1–6. [CrossRef]
8. Song, H.; Lee, Y.; Seo, G.S.; Won, D. Electric Vehicle Charging Management in Smart Energy Communities to Increase Renewable Energy Hosting Capacity. In Proceedings of the 2023 11th International Conference on Power Electronics and ECCE Asia (ICPE 2023—ECCE Asia), Jeju, Republic of Korea, 22–25 May 2023; pp. 453–458. [CrossRef]
9. Morstyn, T.; Hredzak, B.; Agelidis, V.G. Control Strategies for Microgrids with Distributed Energy Storage Systems: An Overview. *IEEE Trans. Smart Grid* **2018**, *9*, 3652–3666. [CrossRef]
10. Sun, H.; Zhu, L.; H.Y. Capacity configuration method of hybrid energy storage system in microgrids based on a non-cooperative game model. *J. Glob. Energy Interconnect.* **2021**, *4*, 454–463.
11. Anderson, A.A.; Suryanarayanan, S. Review of energy management and planning of islanded microgrids. *CSEE J. Power Energy Syst.* **2020**, *6*, 329–343. [CrossRef]
12. Nguyen, T.A.; Crow, M.L. Stochastic Optimization of Renewable-Based Microgrid Operation Incorporating Battery Operating Cost. *IEEE Trans. Power Syst.* **2016**, *31*, 2289–2296. [CrossRef]
13. Rahbar, K.; Chai, C.C.; Zhang, R. Energy Cooperation Optimization in Microgrids with Renewable Energy Integration. *IEEE Trans. Smart Grid* **2018**, *9*, 1482–1493. [CrossRef]
14. Wu, X.; Zhao, W.; Wang, X.; Li, H. An MILP-Based Planning Model of a Photovoltaic/Diesel/Battery Stand-Alone Microgrid Considering the Reliability. *IEEE Trans. Smart Grid* **2021**, *12*, 3809–3818. [CrossRef]
15. Qiu, H.; Gu, W.; Xu, Y.; Yu, W.; Pan, G.; Liu, P. Tri-Level Mixed-Integer Optimization for Two-Stage Microgrid Dispatch with Multi-Uncertainties. *IEEE Trans. Power Syst.* **2020**, *35*, 3636–3647. [CrossRef]

16. Su, L.; Li, Z.; Zhang, Z. A coordinated operation strategy for integrated energy microgrid clusters based on chance-constrained programming. *Power Syst. Prot. Control* **2021**, *49*, 9.
17. Xu, S.; Wu, W.; Zhu, T.; Wang, Z. Convex relaxation based iterative solution method for stochastic dynamic economic dispatch with chance constrains. *Autom. Electr. Power Syst.* **2020**, *40*, 43–51.
18. Nguyen, T.T.; Nguyen, N.D.; Nahavandi, S. Deep Reinforcement Learning for Multiagent Systems: A Review of Challenges, Solutions, and Applications. *IEEE Trans. Cybern.* **2020**, *50*, 3826–3839. [CrossRef] [PubMed]
19. Luong, N.C.; Hoang, D.T.; Gong, S.; Niyato, D.; Wang, P.; Liang, Y.C.; Kim, D.I. Applications of Deep Reinforcement Learning in Communications and Networking: A Survey. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 3133–3174. [CrossRef]
20. Yang, Z.; Hu, J.; Ai, X.; Wu, J.; Yang, G. Transactive Energy Supported Economic Operation for Multi-Energy Complementary Microgrids. *IEEE Trans. Smart Grid* **2020**, *12*, 4–7. [CrossRef]
21. Hua, H.; Qin, Y.; Hao, C.; Cao, J. Optimal energy management strategies for energy Internet via deep reinforcement learning approach. *Appl. Energy* **2019**, *239*, 598–609. [CrossRef]
22. Lu, X.; Xiao, X.; Xiao, L.; Dai, C.; Peng, M.; Poor, H.V. Reinforcement Learning-Based Microgrid Energy Trading with a Reduced Power Plant Schedule. *IEEE Internet Things J.* **2019**, *6*, 10728–10737. [CrossRef]
23. Zhang, G.; Li, J.; Xing, Y.; Bamisile, O.; Huang, Q. A Deep Deterministic Policy Gradient Based Method for Distribution System Load Frequency Coordinated Control with PV and ESS. In Proceedings of the 2022 4th Asia Energy and Electrical Engineering Symposium (AEEES), Chengdu, China, 25–28 March 2022; pp. 780–784. [CrossRef]
24. Sun, J.; Zheng, Y.; Hao, J.; Meng, Z.; Liu, Y. Continuous multiagent control using collective behavior entropy for large-scale home energy management. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 922–929.
25. Wang, J.; Zhou, Y.; Li, Z. Hour-ahead photovoltaic generation forecasting method based on machine learning and multi objective optimization algorithm. *Appl. Energy* **2022**, *312*, 118725. [CrossRef]
26. Liu, B.; Shi, L.; Yao, Z. Multi-objective optimal reactive power dispatch for distribution network considering pv generation uncertainty. In Proceedings of the 10th Renewable Power Generation Conference (RPG 2021), Online, 1–2 March 2021; Volume 2021, pp. 503–509. [CrossRef]
27. Dankwa, S.; Zheng, W. Modeling a Continuous Locomotion Behavior of an Intelligent Agent Using Deep Reinforcement Technique. In Proceedings of the 2019 IEEE 2nd International Conference on Computer and Communication Engineering Technology (CCET), Beijing, China, 16–18 August 2019; pp. 172–175. [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.