Journal of Systems Engineering and Electronics
Vol. 33, No. 3, June 2022, pp.693 – 705

Day-ahead scheduling based on reinforcement learning with hybrid action space

CAO Jingyu[1], DONG Lu[2], and SUN Changyin[1,*]

1. School of Automation, Southeast University, Nanjing 210096, China;
2. School of Cyber Science and Engineering, Southeast University, Nanjing 211189, China

Abstract: Driven by the improvement of the smart grid, the active distribution network (ADN) has attracted much attention due to its characteristic of active management. By making full use of electricity price signals for optimal scheduling, the total cost of the ADN can be reduced. However, the optimal day-ahead scheduling problem is challenging since the future electricity price is unknown. Moreover, in ADN, some schedulable variables are continuous while some schedulable variables are discrete, which increases the difficulty of determining the optimal scheduling scheme. In this paper, the day-ahead scheduling problem of the ADN is formulated as a Markov decision process (MDP) with continuous-discrete hybrid action space. Then, an algorithm based on multi-agent hybrid reinforcement learning (HRL) is proposed to obtain the optimal scheduling scheme. The proposed algorithm adopts the structure of centralized training and decentralized execution, and different methods are applied to determine the selection policy of continuous scheduling variables and discrete scheduling variables. The simulation experiment results demonstrate the effectiveness of the algorithm.

Keywords: day-ahead scheduling, active distribution network (ADN), reinforcement learning, hybrid action space.

DOI: 10.23919/JSEE.2022.000064

1. Introduction

With the rapid development of science and technology, the load demand of users continues to increase and the requirements for environmental protection are getting higher. Therefore, it is necessary to improve the traditional distribution method of uniformly generating electricity from large power plants and then flowing to the load nodes through the superior grid, because this method has the problems of high supply pressure during peak hours and large power loss during transmission. These problems can be solved by introducing the distributed generation (DG) units and battery energy storage systems (BESS). Meanwhile, the energy consumption can be reduced by interrupting some unnecessary load on the user-side. The traditional distribution network can no longer achieve the purpose of active management. Therefore, active distribution network (ADN) is proposed, which can actively manage the DG units, the BESS and the user-side.

In practical applications, optimal scheduling is the key point of active management of the ADN. Many researches have focused on the BESS and user-side due to their controllability and flexibility of scheduling. For example, the mixed-integer conic programming (MICP) is applied to the scheduling of energy storage [1]. In [2–4], electric vehicles were regarded as the BESS, and then different optimization algorithms were used to obtain the optimal charging or discharging scheduling. The similar algorithms are also applied to demand response of the user-side [5–7]. The basic idea of these methods is to formulate the scheduling problem as a mixed integer nonlinear programing (MINLP), and then the optimal policies are explored through different optimization algorithms. For example, in [8], the MINLP was linearized to mixed integer linear programing (MILP), and the branch and bounded method was used to observe the optimal solution. The authors of [9] directly applied the teaching & learning based optimization (TLBO) algorithm to obtain the optimal value of the MINLP. These scheduling methods have all been verified to be effective, but only considered from a single aspect of the BESS or the user-side, which may lead to poor performance in other aspects. Therefore, many scheduling approaches for the overall architecture of the ADN have been proposed. For instance, the studies in [10] proposed a multi-stage optimization approach for the scheduling of the ADN. In addition, the ADN was regarded as a whole for modeling, and then different opti-

Manuscript received July 19, 2021.
*Corresponding author.
This work was supported by the National Key R&D Program of China (2018AAA0101400), the National Natural Science Foundation of China (62173251; 61921004; U1713209), and the Natural Science Foundation of Jiangsu Province of China (BK20202006).

mization methods were used to obtain the optimal scheduling scheme. The authors of [11] directly used the general algebraic modeling system (GAMS) to solve the formulated MINLP problem. The rolling optimization method and the robust optimization method were applied in [12] and [13], respectively. The intelligent algorithms were also used by many scholars to solve the overall programming problem of the ADN, such as the particle swarm optimization (PSO) algorithm in [14], the grey wolf algorithm in [15] and the hybrid algorithm based on dynamic programming (DP) and the genetic algorithm (GA) in [16].

The above-mentioned optimization methods are carried out on the basis of the established model, so there exists the problem of excessive dependence on the model. However, in the actual day-ahead scheduling problem, the electricity price and residential load cannot be known in day-ahead and fluctuate dynamically within a certain range, so it is difficult to establish an accurate model. Therefore, reinforcement learning (RL) is introduced. It does not require a model and obtains the optimal solution based on the interaction between the agent and the environment. A lot of related work has been done in literature. For the charging scheduling of the BESS, Q learning was used in [17] and deep Q network (DQN) was used in [18]. These two methods regarded the selection of charging behavior of the BESS as a discrete variable, and then the RL methods for the Markov decision process (MDP) with discrete action space were applied. In practice, the charge or discharge capacity of the BESS can be any value within the maximum range, that is, treating it as a continuous variable can obtain a better scheduling scheme. Similarly, the authors of [19,20] applied DQN in user-side demand response. For the scheduling of DG units, the double DQN (DDQN) was proposed in [21]. Although RL has not been widely applied to the day-ahead scheduling of the ADN, the autonomous household energy management of smart homes with independent generators can be extended to the whole ADN. In [22], a deep neural network (DNN) was built and its parameters were trained to obtain the optimal solution for scheduling. The studies in [23] proposed an algorithm that combined DNN and Q learning to improve the optimization performance. DQN was applied in [24,25] and deep deterministic policy gradient (DDPG) was applied in [26–28]. In addition, the RL algorithm has been combined with other methods to achieve better results. For example, fuzzy reasoning was introduced into RL in [29].

It is worth noting that these papers have formulated the optimal scheduling problem as an MDP with fully continuous action space or fully discrete action space, and then the appropriate RL methods have been applied to obtain the optimal solution. Obviously, these formulations are idealized. In practice, some schedulable variables are continuous, such as charging or discharging capacity of the BESS and the interrupted load of the user-side. While some schedulable variables are discrete, such as the number of the operating DG units. Therefore, a new RL algorithm is required to obtain the optimal solution of the MDP with continuous-discrete hybrid action space. In the literature, the methods for the MDP with hybrid action space are mainly divided into two categories. One is to discretize the continuous action space, so that this problem is transformed into the MDP with fully discrete action space. For example, fuzzy rules were used in [30] to discretize the continuous variables. However, this method of approximation through discretization made the control accuracy decrease a lot. The other is to make the discrete action space continuous. The algorithm based on the multi-agent DDPG proposed in [31] was applied to obtain the optimal solution, and then performed inverse discretization to obtain the discrete controllable variables. This method greatly increased the complexity of the action space. Therefore, a more reasonable method is to apply two different algorithms to update the selection policies of discrete actions and continuous actions [32]. The authors of [33] proposed an algorithm called p-DQN that combined DQN and DDPG, where DQN was used to select discrete actions and DDPG was used to select continuous actions. Afterwards, some papers proposed improved algorithms on the basis of the p-DQN according to the practical problem, such as multi-pass DQN (MP-DQN) in [34] and deep multi-agent parameterized Q-networks (Deep MAPQN) in [35]. However, these algorithms are applied to the problems with parameterized action space, that is, continuous actions are the parameters of discrete actions. For the parallel structure of discrete action space and continuous action space proposed in this paper, when the dimensionality of the discrete actions increases, the complexity of the algorithm will increase exponentially. To the best of our knowledge, the application of RL in the optimal day-ahead scheduling problem of the ADN with hybrid action space has not been reported in the literature.

In this paper, the optimal day-ahead scheduling of the ADN is formulated as an MDP with continuous-discrete hybrid action space. The objective of this problem is to obtain the optimal scheduling scheme to minimize the total cost of the ADN. A novel RL structure is proposed to determine the optimal scheduling scheme. The main contributions of this papers are as follows:

(i) A multi-agent hybrid RL (HRL) based algorithm is proposed for the MDP with continuous-discrete hybrid action space. In this algorithm, the advantage actor-critic and DDPG are applied for the selection of discrete schedulable variables and continuous schedulable variables, respectively. Moreover, the HRL adopts the structure of centralized training and decentralized execution. Due to the parallel relationship between the actor networks, when the dimensionality of discrete actions increases, the complexity of the algorithm will not increase significantly.

(ii) The objective function is designed as the sum of the costs of different aspects of the ADN. The optimal scheduling scheme which is obtained based on this objective function can reduce the total cost of the ADN in one day and alleviate the supply pressure on the superior grid during peak hours.

(iii) In the proposed method, Gaussian distribution is applied to the establishment of the forecasting models, which can effectively increase the robustness of the forecasting models.

The rest of this paper is organized as follows. The problem formulation is presented in Section 2. After that, the forecasting model and the multi-agent HRL-based algorithm are introduced in Section 3. In Section 4, simulation results based on actual application scenarios are presented to demonstrate the effectiveness of the proposed algorithm. Finally, conclusions are drawn in Section 5.

2. Problem formulation

The optimal day-ahead scheduling problem proposed in this paper is aimed to minimize the total cost of the ADN. The framework of the ADN is shown in Fig. 1. The red arrows in the figure indicate the electricity exchange between each single aspect and the ADN. It can be seen that the cost of the ADN mainly includes the cost of electricity exchanged with the superior grid, the BESS, the user-side, and the DG units.

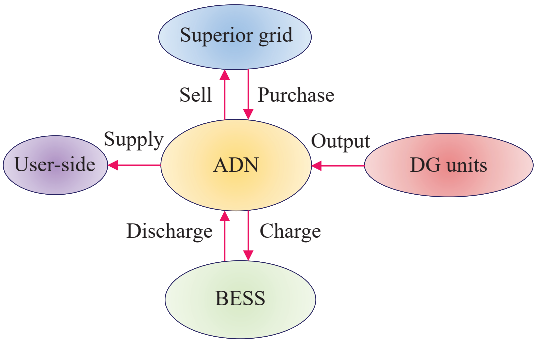


Fig. 1 Framework of ADN

This section is mainly divided into four parts. First of all, to simplify the optimization process, this paper makes some appropriate assumption according to the actual scenarios in Subsection 2.1. Afterwards, in Subsection 2.2, the objective functions which cover the four aspects of the ADN are introduced in detail. In Subsection 2.3, the constraints of some parameters are explained. In Subsection 2.4, this problem is formulated as an MDP with discrete time steps of one hour. The specific descriptions are as follows.

2.1 Assumptions

(i) The electricity consumption during the transmission is zero.

(ii) The maximum interruptible load cannot exceed 30% of the total residential load at the current hour.

(iii) The difference in dissatisfaction of individual users is ignored, and the interruption of residential load is carried out uniformly by the ADN.

2.2 Objective function

The objective function of this optimal scheduling problem is to minimize the total cost of the ADN, which is defined as

$$\min C_{\text{total}} = C_{\text{GE}} + C_{\text{BESS}} + C_{\text{UD}} + C_{\text{DG}} \tag{1}$$

where C_{GE} denotes the cost of electricity exchanged with the superior grid, C_{BESS} indicates the sum cost of the BESS internal loss and the transmission of charging or discharging, C_{UD} represents the cost of user dissatisfaction caused by interrupting the residential load, and C_{DG} is the operating cost of the DG units.

In particular,

$$C_{\text{GE}} = \sum_{t=0}^{23}((\alpha_{\text{pur}}(t)\cdot c_{\text{pur}}(t) - \alpha_{\text{sel}}(t)\cdot c_{\text{sel}}(t))\cdot P_{\text{GE}}(t)) \tag{2}$$

where t denotes every hour of the day, α_{pur} and α_{sel} are 0−1 variables. For time t, $\alpha_{\text{pur}}(t)=1$ represents the ADN purchases electricity from the superior grid and $\alpha_{\text{sel}}(t)=1$ represents the ADN sells the surplus electricity to the superior grid. It is worth noting that $\alpha_{\text{pur}}(t)+\alpha_{\text{sel}}(t)<1$. $c_{\text{pur}}(t)$ represents the total amount of electricity purchased and $c_{\text{sel}}(t)$ represents the total amount of electricity sold. $P_{\text{GE}}(t)$ represents the electricity price per megawatt (MW).

$$C_{\text{BESS}} = \sum_{t=0}^{23}(\alpha_{\text{loss}}((|B(t)-B_{\max}\times 20\%|)+(|B(t)-B_{\max}\times 80\%|))+P_{\text{tr}}(\alpha_{\text{ch}}(t)\cdot c_{\text{ch}}(t)+\alpha_d(t)\cdot c_d(t)))+\mu(B_i-B_{24}) \tag{3}$$

where α_{loss} represents the loss factor of the BESS aging. $B(t)$ denotes the electricity of the BESS at time t. When $B(t)$ is in the range of 20%−80% of the maximum capacity of the BESS $B_{\max}$, the aging cost of the BESS is low. However, when it exceeds this range, the aging cost of the BESS will increase. P_{tr} indicates the transmission price of the charging or discharging process. α_{ch} and α_d are 0−1 variables. For time t, $\alpha_{\text{ch}}(t)=1$ means the BESS is charging and $\alpha_d(t)=1$ means the BESS is discharging. Similarly, $\alpha_{ch}(t)+\alpha_d(t)<1$. $c_{\text{ch}}(t)$ and $c_d(t)$ represent the amount of charge and discharge, respectively. B_i denotes the initial electricity of the BESS at the beginning of the day and B_{24} denotes the remaining electricity of the BESS at the end of the day. μ indicates the coefficient relationship between the reduced electricity of the BESS in the day and the extra cost of the BESS.

$$C_{\mathrm{UD}}=\sum_{t=0}^{23}(\beta_{\mathrm{dis}}\cdot c_{\mathrm{IL}}^{2}(t)) \tag{4}$$

where $c_{\mathrm{IL}}(t)$ denotes the amount of interrupted residential load according to the current electricity price. The cost of user dissatisfaction is proportional to the square of the interrupted load. β_{dis} is the coefficient that represents the relationship between the cost of user dissatisfaction and the square of the interrupted load. This parameter can be set according to the preference of the user-side.

$$C_{\mathrm{DG}}=\sum_{t=0}^{23}c_{\mathrm{dg}}(t), \tag{5}$$

$$c_{\mathrm{dg}}(t)=\begin{cases}c_{\mathrm{dg}}^{0}, & o(t)=0\\ c_{\mathrm{dg}}^{1}, & o(t)=1\\ c_{\mathrm{dg}}^{2}, & o(t)=2\end{cases}, \tag{6}$$

where $o(t)$ represents the number of operating DG units at time t. c_{dg}^{0}, c_{dg}^{1} and c_{dg}^{2} denote the sum of the generation cost and the maintenance cost of the DG units when the number of operating DG units is 0, 1 and 2, respectively.

2.3 Constraints

(i) BESS constraint:

$$0\leqslant B(t)\leqslant B_{\max} \tag{7}$$

(ii) Transmission electricity constraints:

$$0\leqslant c_{\mathrm{ch}}(t)\leqslant c_{\max}, \tag{8}$$

$$0\leqslant c_{d}(t)\leqslant c_{\max}, \tag{9}$$

where $c_{\max}$ represents the maximum charging or discharging electricity within one hour.

(iii) Load interruption constraint:

$$0\leqslant c_{\mathrm{IL}}(t)\leqslant I_{\max} \tag{10}$$

where $I_{\max}$ represents the maximum interruptible load within one hour.

2.4 RL

The RL is a theoretical framework for simulating the randomness policy and the received reward of the agents in an environment where the state has Markov properties. The framework is constructed based on a set of interactive objects, namely agents and environment. This paper takes the decision makers of the scheduling scheme as the agents, and the influencing factors of the scheduling scheme as the environment. Therefore, the optimal scheduling problem is formulated as an MDP, and then the objective is achieved through interactive learning between the agents and the environment. The specific descriptions are as follows.

(i) State:

$$s_t=\{P_{t-1},P_t,L_{t-1},L_t,h_t,B_t\} \tag{11}$$

where P_{t-1} and P_t denote the electricity price in the previous hour and the current hour. L_{t-1} and L_t denote the total residential load in the previous hour and the current hour. h_t represents the current hour. B_t indicates the electricity of the BESS in the current hour. The above state variables provide a reference for the decision-making of policies.

In addition, this paper proposes an additional state variable Ex_t, which represents the amount of electricity exchanged between the ADN and the superior grid, denoted as s_{at}. This additional state variable will not affect the decision of the next actions, but it can be helpful to evaluate the current selected actions. Therefore, s_{at} is entered into the critic network but not into the actor networks.

The additional state variable Ex_t can be calculated as

$$Ex_t=L_t+C_t-I_t-D_t \tag{12}$$

where C_t denotes the charging or discharging electricity of the BESS. The value of C_t is positive for charging and negative for discharging. The amount of electricity change is expressed by the absolute value of C_t, while C_t is zero means that the BESS is neither charging nor discharging in the current hour. I_t indicates the interrupted load of the user-side. D_t denotes the amount of electricity generated by the DG units. When the calculated value of the variable Ex_t is positive, it means that the ADN purchases electricity from the superior grid. Otherwise, it means that the ADN sells the surplus electricity to the superior grid.

(ii) Action:

$$a_t=\{C_t,I_t,O_t\} \tag{13}$$

where O_t indicates the number of operating DG units. Among the action variables, C_t and I_t are continuous variables, and they can be any value within the restricted range. While O_t is a discrete variable which can only be selected from the discrete action space $A_d=\{0,1,2\}$. Therefore, the action space of this optimal scheduling problem is a continuous-discrete hybrid action space.

(iii) Policy:

$$\pi(a_t\mid s_t)=P(a_t\mid s_t) \tag{14}$$

where $\pi(a_t\mid s_t)$ reflects the conditional probability distribution of each action a_t according to the state s_t.

(iv) Reward:

$$r_t = -C_{\text{total}}. \tag{15}$$

Reward is the feedback from the environment to the agents after the agents execute the actions. The agents use it to evaluate the performance of selected actions. Finally, the maximum value of cumulative reward can be obtained through interactive learning between the agents and the environment. Therefore, the minimum value of the total cost can be obtained. Each parameter in the calculation formula of C_{total} has a certain corresponding relationship with the state or action parameters. Therefore, the optimal value of the parameters in C_{total} can be determined by the decision-making of policies, and then the optimal day-ahead scheduling scheme is determined.

(v) Return:

$$G = R_1 + \gamma R_2 + \gamma^2 R_3 + \cdots = \sum_{k=0}^{23} \gamma^k R_{k+1} \tag{16}$$

where G is return, which represents the cumulative reward of the day after being weighted by the discount factor. γ is the discount factor between 0 and 1. When γ is close to 0, the agent is shortsighted. When γ is close to 1, the agent is foresighted.

(vi) State-action value function:

$$Q_\pi(s,a) = E_\pi\left[\sum_{k=0}^{K} \gamma^k \cdot r_{t+k} \mid s_t = s, a_t = a\right] \tag{17}$$

where $Q_\pi(s,a)$ denotes the state-action value function which evaluates the performance of the obtained scheduling scheme. The objective of the day-ahead scheduling problem is to obtain the optimal policy π^*, i.e., a sequence of actions for the user-side, the BESS and the DG units, to maximize the state-action value function.

(vii) State transition:

$$s_{t+1} = f(s_t, a_t) \tag{18}$$

where $s_{t+1} = \{P_t, P_{t+1}, L_t, L_{t+1}, h_{t+1}, B_{t+1}\}$ represents the next state, which can be expressed as function of s_t and a_t.

With the aforementioned definitions in RL framework and constraints in Subsection 2.3, the remark is created.

Remark 1 The constraints proposed in Subsection 2.3 are accomplished by restricting the value of the action variables.

For BESS, the electricity of the BESS B_{t+1} is obtained by

$$B_{t+1} = B_t + C_t. \tag{19}$$

Thus as long as C_t is restricted to satisfy $-B_t \leqslant C_t \leqslant B_{\max} - B_t$, the constraint of the BESS can be accomplished. At the same time, the value of C_t needs to be guaranteed that $-c_{\max} \leqslant C_t \leqslant c_{\max}$ in order to satisfy the constraint of transmission electricity. Furthermore, in order to satisfy the constraint of load interruption, the action variable I_t needs to take a value between 0 and $I_{\max}$.

3. Multi-agent HRL for optimal scheduling of the ADN

It is difficult to determine the optimal day-ahead scheduling scheme in the case that the future electricity price and residential load are unknown. Moreover, through the problem formulation, the scheduling problem is transformed into the problem of an RL with continuous-discrete hybrid action space. Therefore, how to obtain the optimal solution of the MDP with hybrid action space is a more important but difficult point in this paper. In view of the above difficulties, first of all, the values of electricity price and residential load fluctuate within a certain range with a 24-hour cycle, so the neural networks are used to fit the forecasting models of electricity price and residential load. Then, to obtain the optimal solution of the MDP with hybrid action space, a multi-agent HRL algorithm is proposed in this paper. The details are as follows.

3.1 Forecasting model

Electricity price and residential load are both time-related variables. However, in actual application scenarios, to avoid the security risks of grid caused by sudden changes in electricity price or residential load, the previous values are usually used as reference for limiting the current value. Therefore, the current value is determined by both the previous values and time. The relationship between them is expressed as follows:

$$P_{t+1} = f(P_{t-1}, P_t, t), \tag{20}$$

$$L_{t+1} = f(L_{t-1}, L_t, t). \tag{21}$$

Then the neural networks are used to fit unknown variables. In order to increase the robustness of the forecast models, the input variables are sampled from $\mathrm{N}(V_a, 0.1^2)$, V_a denotes the actual values of electricity price or residential load.

3.2 Multi-agent HRL algorithm

The multi-agent HRL algorithm adopts an actor-critic architecture since this basic architecture can be applied to both continuous and discrete action spaces. In addition, for the policy selections of different aspects, the multi-agent HRL algorithm adopts an architecture of centralized training and decentralized execution in order that each single aspect can obtain the optimal scheduling scheme independently. Consequently, the architecture of the multi-agent HRL algorithm contains several parallel actor networks for execution and a single critic network for training, which is shown in Fig. 2.

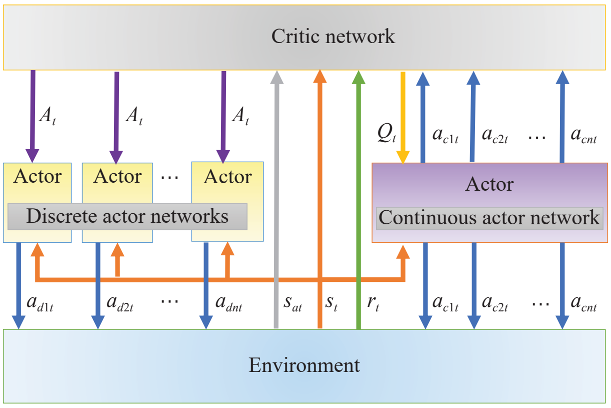


Fig. 2 Architecture of multi-agent HRL algorithm

In order to learn stochastic policies more effectively, different RL algorithms are adopted for policy selection of continuous and discrete actions. Decision making of discrete action policies is mainly based on the advantage actor-critic, and there is a one-to-one correspondence between the discrete action variables and the discrete actor networks. However, the different continuous action variables are only determined by one continuous actor network based on the DDPG algorithm. The lines with arrows indicate the information flow between the decentralized actor networks, the centralized critic network, and the environment. To begin with, each actor network perceives the state s_t, and then executes the discrete actions a_{d1t}, a_{d2t}, $\cdots$, a_{dnt} and the continuous actions a_{c1t}, a_{c2t}, $\cdots$, a_{cnt}. As the result of these actions, the state is transformed from s_t to s_{t+1} and additional state s_{at+1} and reward r_t are generated, which are transmitted together to the single critic network to evaluate the decision-making of policies.

For a certain discrete actor network, its output is the action probability density function $\pi_{\theta_{di}}(a_{dit}|s_t)$, which represents the probability of each discrete action being selected under the state s_t. Then the specific discrete action is obtained through sampling. Therefore, the state value function $V(s_t)$ is generally used to evaluate the policy, and then optimize the discrete actor network parameters θ_{di} by increasing the probability of good action being selected. However, as for continuous actor network, according to the DDPG algorithm, its outputs are deterministic action values a_{ct} and the currently selected actions are evaluated by the state-action value function $Q(s_t, a_{ct})$. Based on the above analysis, a state-action (continuous) value function is proposed to approximate the expected return G_t if the policy π_θ is executed. The function is defined by Bellman equation [36]:

$$\begin{aligned}Q(s_t, s_{at}, a_{ct}) =& E_{\pi_\theta}[r(s_t, s_{at}, a_{ct}, a_{dt})+\\ &\gamma Q(s_{t+1}, s_{at+1}, \pi_{\theta_c}(s_{t+1}))]\end{aligned} \tag{22}$$

where s_{at} represents the additional state defined in the RL framework. a_{ct} and a_{dt} represent all continuous actions and all discrete actions respectively. θ refers to parameters of all actor networks. $\pi_{\theta_c}(s_{t+1})$ denotes the continuous actions a_{ct+1} generated by the continuous actor network based on the state s_{t+1}.

The method of temporal-difference (TD) learning is used to update the value of $Q(s_t, s_{at}, a_{ct})$:

$$\begin{aligned}&Q(s_t, s_{at}, a_{ct}) \leftarrow Q(s_t, s_{at}, a_{ct})+\\ &\alpha[r_{t+1} + \gamma Q(s_{t+1}, s_{at+1}, a_{ct+1}) - Q(s_t, s_{at}, a_{ct})]\end{aligned} \tag{23}$$

where α denotes the update rate. The update objective of the TD method is $r_{t+1} + \gamma Q(s_{t+1}, s_{at+1}, a_{ct+1})$.

Then TD error δ_t is defined as

$$\delta_t = r_{t+1} + \gamma Q(s_{t+1}, s_{at+1}, a_{ct+1}) - Q(s_t, s_{at}, a_{ct}). \tag{24}$$

In order to evaluate the policy of continuous actions π_{θ_c} and the policy of discrete actions π_{θ_d}, the performance objectives are defined.

For continuous actor network, the performance objective $J(\pi_{\theta_c})$ is defined as

$$J(\pi_{\theta_c}) = \mathrm{E}[Q(s_t, s_{at}, \pi_{\theta_c}(s_t))] \tag{25}$$

where θ_c represents the parameters of continuous actor networks.

The optimal policy of continuous actions $\pi^*_{\theta_c}$ is the policy that maximizes $J(\pi_{\theta_c})$ [37]:

$$\pi^*_{\theta_c} = \underset{\pi_{\theta_c}}{\operatorname{argmax}}\, J(\pi_{\theta_c}). \tag{26}$$

For a certain discrete actor network, first of all, define an advantage function:

$$A(s_t, s_{at}, a_{ct}, a_{dit}) = Q(s_t, s_{at}, a_{ct}, a_{dit}) - Q(s_t, s_{at}, a_{ct}) \tag{27}$$

where $Q(s_t, s_{at}, a_{ct})$ is a baseline function that has nothing to do with discrete action a_{dit}. Subtracting this baseline function can reduce the variance but does not change the gradient itself. As defined in [38], $d^\pi(s)$ is a discounted weighting of encountered states. Then, proof is as follows:

$$\begin{aligned}&\mathrm{E}[\nabla_{\theta_{di}} \ln \pi_{\theta_{di}}(a_{dit}|s_t, s_{at}) Q(s_t, s_{at}, a_{ct})] =\\ &\sum_{s_t, s_{at}} d^{\pi_{\theta_{di}}}(s_t, s_{at}) \sum_{a_{dit}} \nabla_{\theta_{di}} \pi_{\theta_{di}}(a_{dit}|s_t, s_{at}) Q(s_t, s_{at}, a_{ct}) =\\ &\sum_{s_t, s_{at}} d^{\pi_{\theta_{di}}}(s_t, s_{at}) Q(s_t, s_{at}, a_{ct}) \nabla_{\theta_{di}} \sum_{a_{dit}} \pi_{\theta_{di}}(a_{dit}|s_t, s_{at}) =\\ &\sum_{s_t, s_{at}} d^{\pi_{\theta_{di}}}(s_t, s_{at}) Q(s_t, s_{at}, a_{ct}) \nabla_{\theta_{di}} 1 = 0.\end{aligned} \tag{28}$$

Thus the performance objective $J(\pi_{\theta_{di}})$ [39] is defined as

$$J(\pi_{\theta_{di}}) = \mathrm{E}[\ln \pi_{\theta_{di}}(a_{dit}|s_t, \theta_{di}) A(s_t, s_{at}, a_{ct}, a_{dit})] \tag{29}$$

where θ_{di} represents the parameters of discrete actor networks.

Similarly, the optimal discrete policy $\pi^*_{\theta_{di}}$ is the policy that maximizes $J(\pi_{\theta_{di}})$:

$$\pi^*_{\theta_{di}} = \underset{\pi \quad \theta_{di}}{\arg\max}\ J(\pi_{\theta_{di}}). \tag{30}$$

As for the calculation of the advantage function $A(s_t, s_{at}, a_{ct}, a_{dit})$, it is troublesome to use two sets of parameters to approximate $Q(s_t, s_{at}, a_{ct}, a_{dit})$ and $Q(s_t, s_{at}, a_{ct})$ respectively, so the TD error is usually used directly to approximate the advantage function. It can be proved that δ_t defined in (24) is an unbiased estimate of $A(s_t, s_{at}, a_{ct}, a_{dit})$:

$$\begin{gathered}\mathrm{E}[\delta_t | s_t, s_{at}, a_{ct}, a_{dit}] = \\ \mathrm{E}[r_{t+1} + \gamma Q(s_{t+1}, s_{at+1}, a_{ct+1}) - \\ Q(s_t, s_{at}, a_{ct}) | s_t, s_{at}, a_{ct}, a_{dit}] = \\ \mathrm{E}[r_{t+1} + \gamma Q(s_{t+1}, s_{at+1}, a_{ct+1}) | s_t, s_{at}, a_{ct}, a_{dit}] - \\ Q(s_t, s_{at}, a_{ct}) = Q(s_t, s_{at}, a_{ct}, a_{dit}) - \\ Q(s_t, s_{at}, a_{ct}) = A(s_t, s_{at}, a_{ct}, a_{dit}).\end{gathered} \tag{31}$$

However, it can be seen from the definition that the value function $Q(s_t, s_{at}, a_{ct})$ is a recursive equation. Therefore, it is impossible to calculate the value of Q through recursion every time in practical applications, so the single critic network is used to approximate the value of Q:

$$Q_{\theta_w}(s_t, s_{at}, a_{ct}) \approx Q(s_t, s_{at}, a_{ct}) \tag{32}$$

where θ_w denotes the parameter of critic network. The result is the approximate value of Q obtained through θ_w.

Algorithm 1 shows how to train the network parameters θ of the overall architecture of the mutli-agent HRL algorithm. The inputs are the real electricity price and residential load of the previous day. After the HRL algorithm training is completed, the trained network parameters θ are output, including the single critic network parameters θ_w, the continuous actor network parameters θ_c, and the discrete actor networks parameters θ_d.

Algorithm 1 Multi-agent HRL algorithm

```
Input: Electricity price, residential load
Output: Network parameters $\theta$
Randomly initialize the estimated network parameters $\theta$.
Initialize the target network parameters $\bar{\theta} = \theta$.
for Epoch=1:51 000 do
    Obtain the initial state $s_{t_i}$ and the initial additional state $s_{at_i}$.
    for Time $t$=$t_i$:23 do
        Select continuous actions $a_{ct}$ and discrete actions $a_{dt}$.
        Execute actions $a_{ct}$ and $a_{dt}$, observe reward $r_t$ and transition to the next state $s_{t+1}$ and the next additional state $s_{at+1}$.
        Store transition ($s_t$, $s_{at}$, $a_{ct}$, $a_{dt}$, $r_t$, $s_{t+1}$, $s_{at+1}$) in experience pool $D$.
        while Epoch > 1 000 do
            Draw minibatch of transitions $\mathcal{F} = \{(s_j, s_{aj}, a_{cj}, a_{dj}, r_j, s_{j+1}, s_{aj+1})\}_{j=1}^{\#\mathcal{F}}$ from $D$.
            Calculte the target state-action (continuous) value $y_j \leftarrow r_j + \gamma Q(s_{j+1}, s_{aj+1}, \pi_{\bar{\theta}_c}(s_{j+1}); \bar{\theta}_w)$.
            Calculte the loss function $L(\theta_w) = \frac{1}{N}\sum_{j=1}^{\#\mathcal{F}}[y_j - Q(s_j, s_{aj}, a_{cj}; \theta_w)]^2$.
            Update the critic network parameters
            $\theta_w \leftarrow \theta_w - l_c \nabla_{\theta_w} L(\theta_w)$
            Calculte the performance objectives
            $J(\pi_{\theta_c}) = \frac{1}{N}\sum_{j=1}^{\#\mathcal{F}} Q(s_j, s_{aj}, \pi_{\theta_c}(s_j))$,
            $J(\pi_{\theta_d}) = \frac{1}{N}\sum_{j=1}^{\#\mathcal{F}}[\ln \pi_{\theta_d}(a_{dj} | s_j, s_{aj})\delta_j]$.
            Update the actor network parameters
            $\theta_c \leftarrow \theta_c - l_a \nabla_{\theta_c}(-J(\pi_{\theta_c})), \theta_d \leftarrow \theta_d - l_a \nabla_{\theta_d}(-J(\pi_{\theta_d}))$.
            Update the target network parameters
            $\bar{\theta} \leftarrow \tau\theta + (1-\tau)\bar{\theta}$.
        end while
    end for
end for
```

First of all, the estimated network parameters θ are initialized randomly. Then the target network parameters $\bar{\theta}$ are initialized to the same value as θ. After that, the storage of state transition pairs and the update of network parameters are performed in the loop of 51 000 epochs. Each epoch starts at a random hour t of the day. This randomness can improve the robustness of the neural network to avoid overfitting, so that the optimal scheduling policy can still be obtained when the forecasted electricity price is slightly deviated. Then the initial state s_{t_i} and the initial additional state s_{at_i} are obtained. At each time step, the exploration and exploitation of discrete actions a_{dt} are based on the randomness of selecting actions according to the probability distribution of discrete actions. As for the exploration and exploitation of continuous actions a_{ct}, Gaussian noise is added to the decision-making process of a_{ct} to change it from a deterministic process to a random process, and then a_{ct} is obtained by sampling from this random process. Then the actions a_{ct} and a_{dt} are executed to complete the state transition and the reward r_t of environmental feedback is observed, and thus a sequence of state transition pairs are formed and stored in experience pool D. While epoch > 1 000, the experience pool is full, and then these state transition pairs in the experience pool are used to update the parameters θ. Specifically, N state transition pairs are drawn from experience pool as samples. As a reference objective for optimization, the target state-action (continuous) value y_j is calculated as

$$y_j \leftarrow r_j + \gamma Q(s_{j+1}, s_{aj+1}, \pi_{\bar{\theta}_c}(s_{j+1}); \bar{\theta}_w) \tag{33}$$

where $\bar{\theta}_w$ represents the target critic network parameters, $\pi_{\bar{\theta}_c}(s_{j+1})$ denotes the target continuous actions $\bar{a}_{cj+1}$ generated by the target continuous actor network with the parameters $\bar{\theta}_c$.

Therefore, with the minibatch samples, the loss function is calculeted as

$$L(\theta_w) = \frac{1}{N}\sum_{j=1}^{\#\mathcal{F}}[y_j - Q(s_j, s_{aj}, a_{cj}; \theta_w)]^2 \tag{34}$$

which denotes the mean square error between the target state-action (continuous) value y_j and the state-action (continuous) value $Q(s_j, s_{aj}, a_{cj}; \theta_w)$ approximated by the estimated critic network parameters θ_w. Then, along the gradient direction that minimize the loss function, the parameters of the critic network are updated as

$$\theta_w \leftarrow \theta_w - l_c \nabla_{\theta_w} L(\theta_w) \tag{35}$$

where l_c indicates the learning rate of the critic network parameters and $\nabla_{\theta_w} L(\theta_w)$ denotes the gradient of $L(\theta_w)$ drop.

Then, based on the above definition of the performance objectives of the continuous actor network and discrete actor networks, the performance objectives of these minibatch samples are calculated as

$$J(\pi_{\theta_c}) = \frac{1}{N}\sum_{j=1}^{\#\mathcal{F}} Q(s_j, s_{aj}, \pi_{\theta_c}(s_j)), \tag{36}$$

$$J(\pi_{\theta_d}) = \frac{1}{N}\sum_{j=1}^{\#\mathcal{F}} [\ln \pi_{\theta_d}(a_{dj}|s_j, s_{aj}, a_{cj}, \theta_d)\delta_j]. \tag{37}$$

Similar to the update of the critic network parameters, the parameters of the actor networks are updated as

$$\theta_c \leftarrow \theta_c - l_a \nabla_{\theta_c}(-J(\pi_{\theta_c})), \tag{38}$$

$$\theta_d \leftarrow \theta_d - l_a \nabla_{\theta_d}(-J(\pi_{\theta_d})), \tag{39}$$

where l_a indicates the learning rate of actor network parameters. Different from the update of the critic network parameters, the update of the actor network parameters is along the gradient direction that maximizes the performance objectives.

The update of the target network parameters adopts soft update, that is, the parameters of the target networks will be updated every step but the update rate is very small. The update is according to the following formula:

$$\bar{\theta} \leftarrow \tau\theta + (1-\tau)\bar{\theta} \tag{40}$$

where τ denotes the update rate.

Remark 2 Due to the structure of parallel distributed actor networks, the computational complexity of the proposed multi-agent HRL algorithm is $O(n_c + n_d)$, where n_c and n_d represent the number of agents with continuous action space and discrete action space. When the number of agents increases, the computational complexity of the proposed algorithm increases linearly, while the computational complexity of the previous algorithms [33] increases exponentially.

4. Experimental results

In this section, the effectiveness of the proposed algorithm is verified through simulation results. This section is divided into two parts. Subsection 4.1 introduces the experimental setup in detail. Then, the simulation results and discussion are presented in Subsection 4.2.

4.1 Experimental setup

The proposed algorithm is a price-based scheduling method, so the evaluation of its performance is based on real-world hourly electricity price. However, the settings of the BESS and DG units are hypothetical based on the actual situations. If the proposed algorithm is applied to the real-world scenarios, it only needs to adjust some coefficients according to the local actual conditions. In this experiment, some parameters are set as follows. The $B_{\max}$ and $c_{\max}$ do not exceed 3000 MW and 210 MW, respectively. The α_{loss} is set to 0.08. The P_{tr} is 0.14 k/MW. The coefficient μ is set as 0.5.

In the algorithm structure, each actor network consists of an input layer, an output layer, and a hidden layer with 32 hidden neurons. While the critic network contains an input layer, an output layer, and two hidden layers in which the number of hidden neurons are 64 and 32, respectively. In addition, the settings of some hyperparameters in this algorithm are presented in Table 1.

Table 1 Hyperparameters in the algorithm

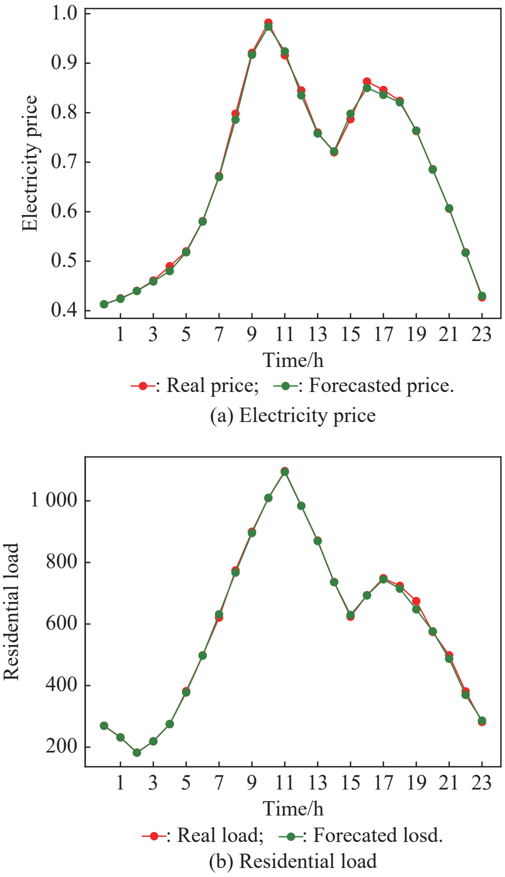
Parameter	Value
Learning rate of the actor networks l_a	0.001
Learning rate of the critic network l_c	0.01
Update rate of the target networks τ	0.01
Discount factor γ	0.99
Capacity of the experience pool	10000
Batch size	128

Remark 3 The hyperparameters in this algorithm are set by referring to [26]. Among them, the learning rate of the actor networks l_a is an order of magnitude lower than

the learning rate of the critic network l_c. Then these hyperparameters are adjusted according to the neural network structure and the actual training results of this experiment.

4.2 Simulation results and discussion

(i) Performance of the forecasting models: The past real data of electricity price and residential load are used to train the neural network for 5 000 epochs, and thus the forecasting models are obtained. Then the real electricity price and residential load at 23 o'clock on the previous day and 0 o'clock on the day are input into this model. After continuous state transitions, the forecasted electricity price and residential load for the next day are output. The comparison between the forecasted values and the real values are presented in Fig. 3.



(a) Electricity price

(b) Residential load

Fig. 3 Comparison of the forecasted values and true values

As shown in Fig. 3, though there exist slight deviations between the forecasted values and the real values, the deviations are within a reasonable range. Therefore, it is effective to use the value of the current hour, the value of the previous hour and the current time t to fit the forecasting models. Accordingly, as long as the value of the last hour of the previous day and the first hour of the day are known, the day-ahead forecast can be carried out.

(ii) Performance of the day-ahead scheduling algorithm: The multi-agent HRL algorithm proposed in this paper is used to perform day-ahead scheduling experiments in a simulation scenario, where the scheduling of the ADN is limited to the charge or discharge of the BESS, the interrupted load of the user-side and the number of operating DG units. In order to reduce the influence of different magnitudes of the schedulable variables on the determination of the scheduling scheme, the schedulable variables are normalized. The optimal day-ahead scheduling scheme obtained after 50 000 epochs training are shown as follows.

Fig. 4 shows the hourly charging or discharging of the BESS according to the day-ahead forecasted electricity price. It can be seen that the optimal scheduling scheme guides the BESS to charge when the electricity price is low and to discharge when the electricity price is high. Fig. 5 shows the remaining electricity in the BESS per hour after charging or discharging. After a whole day of charging and discharging, the electricity at the end of the day is approximately equal to the value at the beginning of the day, which is beneficial to the long-term scheduling of the BESS. In addition, the electricity of the whole day is maintained within the range of 20% to 80% of $B_{\max}$, which is conducive to prolonging the service life of the BESS.

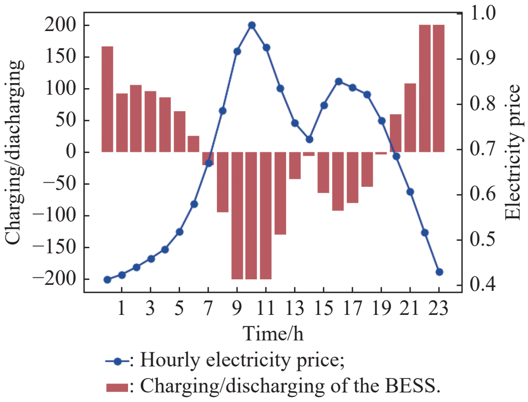


Fig. 4 Hourly electricity price and charging/discharging of the BESS

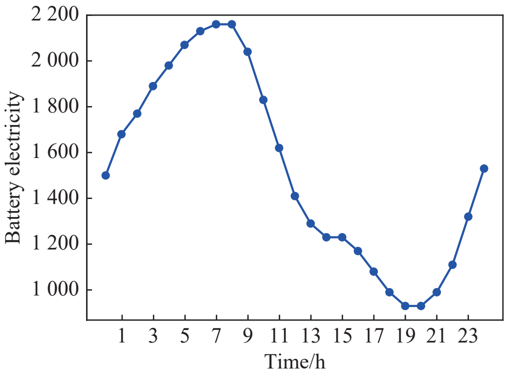


Fig. 5 Electricity of the BESS per hour

For the interrupted load of the user-side, when it increases, the electricity cost of the user-side under the current electricity price decreases, but the dissatisfaction cost increases. The purpose of the algorithm proposed in this paper is to learn the policy that can balance the electricity cost and the user dissatisfaction cost according to the preference of the user-side. Fig. 6 shows the relationship between the interrupted load of the user-side and the electricity price when the user dissatisfaction factor β is 0.001 7. It can be seen that when the electricity price increases, the interrupted load of the user-side increases. Otherwise, it decreases. Therefore, the effectiveness of the proposed algorithm is proved. Fig. 7 shows the comparison of the interrupted load when the user dissatisfaction factor β takes different values. It can be observed that a larger β corresponds to a larger interrupted load. This is because a larger β means that the user-side prefers to concern about the dissatisfaction cost rather than the electricity cost.

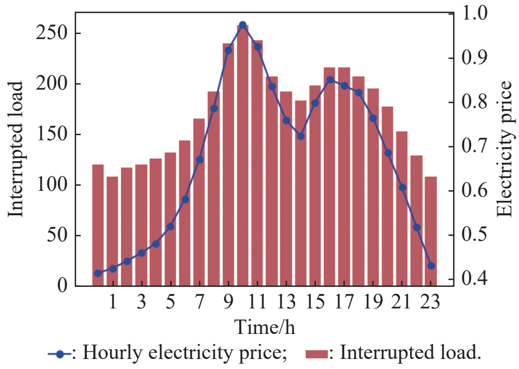


Fig. 6 Hourly electricity price and the interrupted load of the user-side

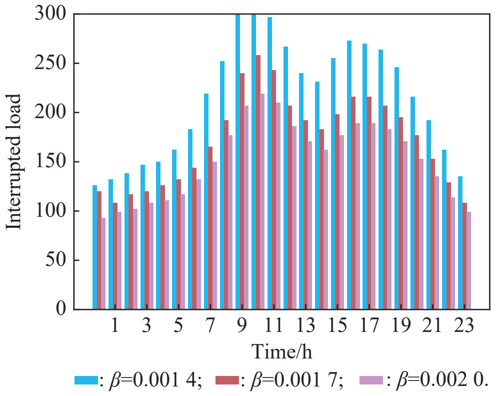


Fig. 7 Comparison of the interrupted load under different β

Fig. 8 shows the number of local DG units in operation per hour according to the forecasted electricity price. As shown in the figure, when the electricity price is in the low-range (0.4 − 0.6 k/MW), it is more economical to purchase electricity from the superior grid, so the DG units are shut down. When the electricity price is in the mid-range (0.6 − 0.8 k/MW), the maintenance cost of operating the two DG units at the same time is relatively high, so only one of the DG units is turned on, which can reduce part of the electricity cost. When the electricity price is in the peak-range (0.8 − 1.0 k/MW), the cost of electricity purchase and sale is high. Therefore, the two DG units operate at the same time to make full use of the electricity supply of the DG units and the excess electricity will be sold to the superior grid, which can improve the economy of the ADN.

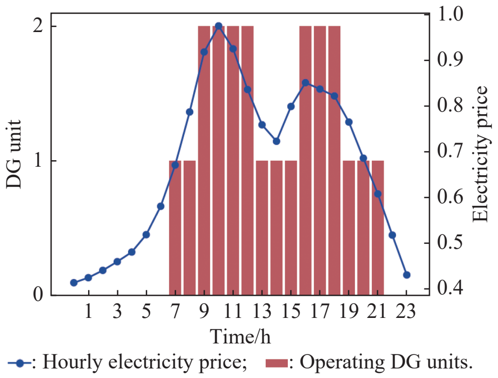


Fig. 8 Hourly electricity price and the number of operating DG units

The above-mentioned different aspects scheduling together constitute the optimal day-ahead scheduling scheme of the ADN. The total cost after scheduling is 8 259.35, which is a reduction of 22.68% compared with the value of 10 682.29 before scheduling. Moreover, the comparison of the electricity exchanged between the ADN and the superior grid before and after the optimal day-ahead scheduling is shown in Fig. 9. It is proved that the proposed scheduling scheme can effectively reduce the total cost of the ADN within a day and alleviate the supply pressure of the superior grid during peak hours.

In order to evaluate the performance of the proposed multi-agent HRL algorithm in solving the problems of MDP with hybrid action space, the previous methods for solving the problems of MDP with hybrid action space are compared [33]. Among them, the p-DQN algorithm adopts the same network architecture as the HRL algorithm. For the DQN algorithm, the charging behavior of the BESS is discretized into two types: charging or discharging, and the load interruption behavior of the user-side is discretized into two types: interruption or non-interruption. Other parameters of the DQN are the same as the HRL. When the networks start to be updated, the scheduling scheme of the whole episode from 0 o'clock to 23 o'clock is completed every 5 epochs, then output the episode return to observe the entire optimization process of these algorithms. The experimental result of DDPG is not included here because it fails to obtain the optimal scheduling scheme in the application scenario of this paper. The training results of different algorithms are compared from the following three dimensions. The comparison of episode return during the training process is shown in Fig. 10. Moreover, the comparison of the total electricity cost reduction rate of 24 hours ΔR_c and the variance reduction rate of the electricity exchanged between the ADN and the superior grid of 24 hours ΔR_v after the optimal scheduling are presented in Table 2. The greater the variance reduction rate of the exchanged electricity, the stronger the ability of the algorithm to alleviate the supply pressure on the superior grid during peak hours. It can be seen that the performance of the proposed algorithm is better than the previous algorithm. Therefore, the superiority of the proposed algorithm is proved.

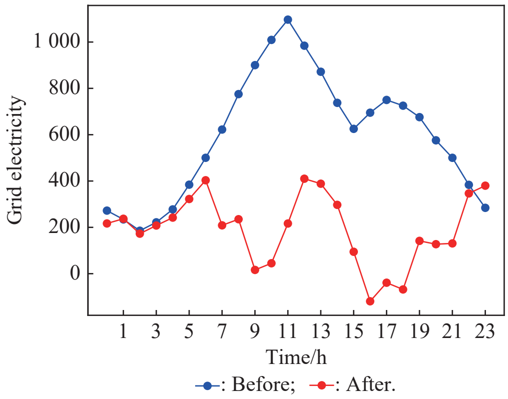


Fig. 9 Comparison of the electricity exchanged between the ADN and the superior grid before and after the optimal day-ahead scheduling

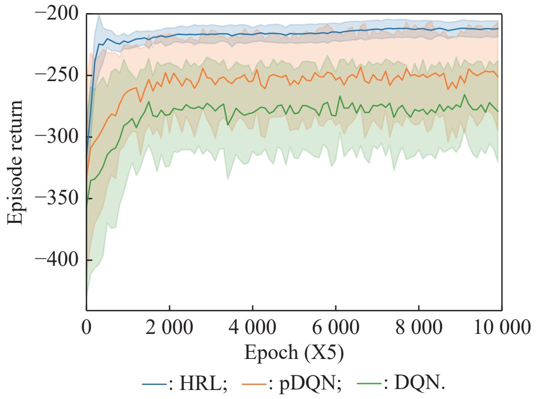


Fig. 10 Episode return of the proposed multi-agent HRL algorithm compared with the DQN and p-DQN algorithm

Table 2 Performances of different algorithms %

Algorithm	ΔR_c	ΔR_v
HRL	20.73	69.48
pDQN	17.24	53.95
DQN	9.09	48.39

5. Conclusions

In this paper, a multi-agent HRL algorithm is proposed to solve the optimal day-ahead scheduling problem of the ADN, where continuous schedulable variables and discrete schedulable variables coexist. With the aiming of minimizing the total cost of the ADN within a day, the optimal scheduling problem is formulated as an MDP with continuous-discrete hybrid action space. In the proposed approach, forecasting models are established to overcome the uncertainty of the future electricity price and residential load. Then, the multi-agent HRL algorithm is proposed to learn the optimal scheduling scheme, which adopts actor-critic and DDPG to the selection of discrete schedulable variables and continuous schedulable variables, respectively. Simulation results show that the multi-agent HRL algorithm can minimize the total cost and alleviate the supply pressure during the peak hours. Furthermore, the previous algorithms are compared, which indicates the superiority of the proposed algorithm.

References

[1] FANG X, HODGE B M, BAI L Q, et al. Mean-variance optimization-based energy storage scheduling considering day-ahead and real-time LMP uncertainties. IEEE Trans. on Power Systems, 2018, 33(6): 7292–7295.

[2] ZHONG Q W, BUCKLEY S, VASSALLO A, et al. Energy cost minimization through optimization of EV, home and workplace battery storage. Science China Technological Sciences, 2018, 61(5): 761–773.

[3] WANG X Y, SUN C, WANG R T, et al. Two-stage optimal scheduling strategy for large-scale electric vehicles. IEEE Access, 2020, 8: 13821–13832.

[4] WANG Y Y, JIAO X H. Multi-objective energy management for PHEV using pontryagin’s minimum principle and particle swarm optimization online. Science China Information Sciences, 2021, 64(1): 1–3.

[5] YU D M, BRESSER C. Peak load management based on hybrid power generation and demand response. Energy, 2018, 163: 969–985.

[6] AGHAJANI G R, SHAYANFAR H A, SHAYEGHI H. Demand side management in a smart micro-grid in the presence of renewable generation and demand response. Energy, 2017, 126: 622–637.

[7] MURALITHARAN K, SAKTHIVEL R, SHI R. Multi objective optimization technique for demand side management with load balancing approach in smart grid. Neurocomputing, 2016, 177: 110–119.

[8] DAI B J, WANG R, ZHU K, et al. A demand response scheme in smart grid with clustering of residential customers. Proc. of the IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids, 2019: 1–6.

[9] DERAKHSHAN C, SHAYANFAR H A, KAZEMI A. The optimization of demand response programs in smart grids. Energy Policy, 2016, 94: 295–306.

[10] ZHU X J, HAN H T, GAO S, et al. A multi-stage optimization

approach for active distribution network scheduling considering coordinated electrical vehicle charging strategy. IEEE Access, 2018, 6: 50117–50130.

[11] MAZIDI M, MONSEF H, SIANO P. Incorporating price-responsive customers in day-ahead scheduling of smart distribution networks. Energy Conversion and Management, 2016, 115: 103–116.

[12] YAN Y, ZHANG C H, LI K, et al. Synergistic optimal operation for a combined cooling, heating and power system with hybrid energy storage. Science China Information Sciences, 2018, 61(11): 110202.

[13] SOROUDI A, SIANO P, KEANE A. Optimal DR and ESS scheduling for distribution losses payments minimization under electricity price uncertainty. IEEE Trans. on Smart Grid, 2015, 7(1): 261–272.

[14] LI S Y, ZHONG S, PEI Z, et al. Multi-objective reconfigurable production line scheduling for smart home appliances. Journal of Systems Engineering and Electronics, 2021, 32(2): 297–317.

[15] SALEHI J, ABDOLAHI A. Optimal scheduling of active distribution networks with penetration of PHEV considering congestion and air pollution using DR program. Sustainable Cities and Society, 2019, 51: 101709.

[16] LI F, SUN B, ZHANG C H. Operation optimization for integrated energy system with energy storage. Science China Information Sciences, 2018, 61(12): 129207.

[17] LEI W H, CUI H, NEMETH T, et al. Deep reinforcement learning-based energy management of hybrid battery systems in electric vehicles. Journal of Energy Storage, 2021, 36: 102355.

[18] WAN Z Q, LI H P, HE H B, et al. Model-free real-time EV charging scheduling based on deep reinforcement learning. IEEE Trans. on Smart Grid, 2018, 10(5): 5246–5257.

[19] LU R Z, HONG S H, YU M M. Demand response for home energy management using reinforcement learning and artificial neural network. IEEE Trans. on Smart Grid, 2019, 10(6): 6629–6639.

[20] CAO J Y, DONG L, XUE L. Load scheduling for an electric water heater with forecasted price using deep reinforcement learning. Proc. of the Chinese Automation Congress, 2020: 2500–2505.

[21] XI L, YU L, XU Y C, et al. A novel multi-agent DDQN-AD method-based distributed strategy for automatic generation control of integrated energy systems. IEEE Trans. on Sustainable Energy, 2019, 11(4): 2417–2426.

[22] LI H P, WAN Z Q, HE H B. A deep reinforcement learning based approach for home energy management system. Proc. of the IEEE Power & Energy Society Innovative Smart Grid Technologies Conference, 2020: 1–5.

[23] XU X, JIA Y W, XU Y, et al. A multi-agent reinforcement learning-based data-driven method for home energy management. IEEE Trans. on Smart Grid, 2020, 11(4): 3201–3211.

[24] TSANG N, CAO C, WU S, et al. Autonomous household energy management using deep reinforcement learning. Proc. of the IEEE International Conference on Engineering, Technology and Innovation, 2019. DOI: 10.1109/ICE.2019.8792636.

[25] WANG Y D, LIU H, ZHENF W B, et al. Multi-objective workflow scheduling with deep-q-network-based multi-agent reinforcement learning. IEEE Access, 2019, 7: 39974–39982.

[26] YU L, XIE W W, XIE D, et al. Deep reinforcement learning for smart home energy management. IEEE Internet of Things Journal, 2019, 7(4): 2751–2762.

[27] CHUNG H M, MAHARJAN S, ZHANG Y, et al. Distributed deep reinforcement learning for intelligent load scheduling in residential smart grid. IEEE Trans. on Industrial Informatics, 2020, 17(4): 2752–2763.

[28] LEE S, CHOI D H. Energy management of smart home with home appliances, energy storage system and electric vehicle: a hierarchical deep reinforcement learning approach. Sensors, 2020, 20(7): 2157.

[29] ALFAVERH F, DENAI M, SUN Y C. Demand response strategy based on reinforcement learning and fuzzy reasoning for home energy management. IEEE Access, 2020, 8: 39310–39321.

[30] ZHOU S Y, HU Z J, GU W, et al. Artificial intelligence based smart energy community management: a reinforcement learning approach. CSEE Journal of Power and Energy Systems, 2019, 5(1): 1–10.

[31] LOWE R, WU Y, TAMAR A, et al. Multi-agent actor-critic for mixed cooperative-competitive environments. Proc. of the 31th International Conference on Neural Information Processing Systems, 2017: 6382–6393.

[32] MASSON W, RANCHOD P, KONIDARIS G. Reinforcement learning with parameterized actions. Proc. of the AAAI Conference on Artificial Intelligence, 2016: 1934–1940.

[33] XIONG J C, WANG Q, YANG Z R. Parametrized deep q-networks learning: reinforcement learning with discrete-continuous hybrid action space. https://arxiv.org/abs/1810.06394.

[34] BESTER C J, JAMES S D, KONIDARIS G D. Multi-pass *q*-networks for deep reinforcement learning with parameterized action spaces. https://arxiv.org/abs/1905.04388.

[35] FU H T, TANG H Y, HAO J Y, et al. Deep multi-agent reinforcement learning with discrete-continuous hybrid action spaces. https://arxiv.org/abs/1903.04959.

[36] BELLMAN R. Dynamic programming. Science, 1966, 153(3731): 34–37.

[37] LILLICRAP T P, HUNT J J, PRITZEL A, et al. Continuous control with deep reinforcement learning. https://arxiv.org/abs/1509.02971.

[38] SUTTON R S, MCALLESTER D A, SINGH S P, et al. Policy gradient methods for reinforcement learning with function approximation. Proc. of the 12th International Conference on Neural Information Processing Systems, 1999: 1057–1063.

[39] HAARNOJA T, ZHOU A, HARTIKAINEN K, et al. Soft actor-critic algorithms and applications. https://arxiv.org/abs/1812.05905.

Biographies

CAO Jingyu was born in 1999. She received her B.S. degree in the School of Control and Computer Engineering from North China Electric Power University, Beijing, China, in 2019. She is currently pursuing her Ph.D. degree with the School of Automation, Southeast University, Nanjing, China. Her research interests include machine learning, deep reinforcement learning, optimal control, and multi-agent cooperative control.

E-mail: cjy0564@seu.edu.cn

DONG Lu was born in 1990. She received her B.S. degree in the School of Physics and her Ph.D. degree in the School of Automation from Southeast University, Nanjing, China, in 2012 and 2017, respectively. She is currently an associate professor with the School of Cyber Science and Engineering, Southeast University, Nanjing, China. Her research interests include adaptive dynamic programming, event-triggered control, and multi-agent reinforcement learning.
E-mail: ldong90@seu.edu.cn

SUN Changyin was born in 1975. He received his B.S. degree in applied mathematics from the College of Mathematics, Sichuan University, Chengdu, China, in 1996, and his M.S. and Ph.D. degrees in electrical engineering from Southeast University, Nanjing, China, in 2001 and 2004, respectively. He is currently a professor with the School of Automation, Southeast University, Nanjing, China. His current research interests include intelligent control, flight control, and optimal theory.
E-mail: cysun@seu.edu.cn