HRL-Based Access Control for Wireless Communications With Energy Harvesting

Yingkai Wang, Qingshan Wang, *Member, IEEE*, Qi Wang, and Zhiwen Zheng

***Abstract*— This paper studies the access control problem of long-term throughput maximization in wireless communication systems with Energy Harvesting (EH). In the existing research, many access schemes based on accurate environmental information have been proposed, such as channel information and the EH process. However, access to environmental information is costly, and traditional access control frameworks are expensive to explore in high-dimensional spaces. Thus, an access control framework based on hierarchical reinforcement learning (HRL) is proposed in this paper. In HRL, the control problem in the Markov decision process (MDP) form is decomposed into a multilevel sequential control problem. It includes high-level channel number selection, mid-level channel selection, and low-level channel matching subproblems. The scheme is obtained by combining the solutions of subproblems at different level which are solved in sequence. In addition, to improve learning efficiency, the deterministic action (DA) module and the prior knowledge (PK) module are put forward. The DA module solves the channel matching problem under the additional guidance given by the previous subproblem, which selects definite good low-level actions. The PK module provides the framework with the common knowledge of the system structure learned from the hypothetical environment, so as to obtain better initial performance. Experimental results show that our framework achieves better performance and better learning efficiency compared with several recent transmission schemes.**

***Note to Practitioners*—Access control is an important issue in wireless communication systems, and users need to be scheduled to solve the constraint of limited resources, such as energy usually provided by batteries. In recent years, in order to overcome the energy limitation, energy harvesting devices have been developed and applied to wireless communication systems. However, the energy collection ability of the system is greatly influenced by the environment, which leads to the poor performance of most traditional control schemes that rely on the prior knowledge of the environment. Therefore, this paper proposes a novel hierarchical reinforcement learning (HRL)-based model-free access control framework for wireless communication system to maximize the system throughput without any prior environmental knowledge. The scheme abstracts the original control problem into three sub-control sub control problems according to tasks and solves them sequentially, thus simplifying the original control problem. This scheme can not only learn independently, but also does not depend on the prior knowledge of the environment. Moreover, this method is also suitable for the large-scale environment while the conventional end-to-end reinforcement learning is not suitable for. Compared with traditional algorithms, our method has better performance and higher learning efficiency.**

***Index Terms*— Neural network applications, decision-making, knowledge based system, access control, energy harvesting.**

I. Introduction

With the increasing popularity of the Internet of Things (IOT) in recent years [1], its application scope is also expanding [2], [3]. However, due to the limited battery capacity of IOT terminal equipment, the progress of IOT terminal equipment has been hindered by the shortage of energy [4]. Among various energy sources, energy harvesting (EH) technology is a promising solution. It collects environmental energy (such as solar energy, wind energy and heat energy) and converts it into electric energy for terminal equipment [5]. Theoretically, it provides endless power, allowing the EH series communication system to be free from the limitation of permanent wires position or limited battery capacity for constant charging [6].

Despite the benefits listed above, EH communication systems still face the challenge of operating in an uncertain and dynamic environment [7]. Because the environment has such a strong influence on the performance of EH, the energy obtained in each cycle is usually expressed as a highly volatile random value [8]. At the same time, the fast fading channel (rapidly changing) makes the state space of communication systems grow exponentially with the increase of system scale [9]. The above factors make it difficult to design an appropriate access control scheme.

Access control schemes have been the focus of intensive study in wireless communication systems, and EH devices need to schedule users to solve the constraint of limited resources. There are two significant limitations. The first limitation is that the conventional methods rely on environment information (no-causal case). These methods lack the prior knowledge of new environment that is difficult to obtain or estimate. Thus, they are almost difficult to be applied in practice. Even if the current environmental distribution is estimated from historical data, it is inevitably different

Manuscript received 18 August 2022; revised 22 November 2022; accepted 2 January 2023. This article was recommended for publication by Associate Editor A. Parisio and Editor C. Seatzu upon evaluation of the reviewers' comments. This work was supported in part by the National Natural Science Foundation of China under Grant 61571179 and in part by the Anhui Provincial Natural Science Foundation of China under Grant 2208085MF165. *(Corresponding author: Qingshan Wang.)*

Yingkai Wang, Qingshan Wang, and Qi Wang are with the School of Mathematics, Hefei University of Technology, Hefei 230001, China (e-mail: qswang@hfut.edu.cn).

Zhiwen Zheng is with the School of Mathematics, Hefei University of Technology, Hefei 230001, China.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TASE.2023.3235316.

Digital Object Identifier 10.1109/TASE.2023.3235316

from the actual current environment and cannot adapt to the dynamically changing environment. The second limitation is the scalability of some RL methods, which are hard to be used in large-scale environments because of low learning efficiency. If the scale of the system is slightly larger, there will be poor learning ability and low learning efficiency. It reduces the generalization of the method and increases the actual cost and computational cost in the face of environmental changes.

In this paper, we study a problem: given an energy harvesting module to support power and sets of fading channels and users, find the optimal transition policies such that the sum of throughput is maximized over all slots while satisfying the power constraint. In order to overcome two major limitations mentioned above, we propose a novel HRL-based model-free access control framework without any prior environmental knowledge. For the first limitation, we designed a layered architecture with three controllers. In this architecture, EH devices do not decide the matching of specific channels for each user separately. Instead, considering the fast fading channels, it selects the number of matching channels, the specific channels to participate in the matching, and the specific channel user matching orderly. By making simple decisions step by step, such as the number of matching channels, the goal of the specific channel user matching complex decisions is finally realized. In this way, the structural knowledge of the system is used to supplement the lack of environmental prior knowledge and avoid the difficulty of making and realizing complex decision goals based on sparse environmental prior knowledge. In addition, the prior knowledge (PK) module into the framework to improve the learning efficiency of samples and reduce the device's dependence on environmental information. For the second limitation, we propose a deterministic action (DA) module with low computational overhead as a low-level channel user matching controller, and a neural network-based DRL algorithm as a high-level and intermediate level controller.

The main contributions are summarized as follows:

- We formulate the access control problem as a Markov Decision Process (MDP) problem.
- We further design a novel HRL framework with three levels for access control network in task abstraction form (high-level channel number selection, mid-level channel selection and low-level channel matching). By decoupling the original optimization objectives into three levels, the complex action space in our proposed HRL framework is simplified.
- We integrate DA module and PK module into the framework to make better use of some structural features of the system shown by the decoupling method. The DA module directly solves the problem of low-level channel matching, instead of traditional inefficient random exploration, that is, reducing the potential exploration cost by finding the optimal action that meets the constraints. The PK module improves the efficiency of samples and avoids repeated learning of public knowledge by acquiring common knowledge in the hypothetical environment.

The rest of this paper is organized as follows: Section II introduces the related work. Section III present our system model and section IV formulates the maximization throughput problem. Section V propose our HRL-based framework and learning module in detail. Section VI presents performance evaluation setup details and results. Finally, Section VII concludes this paper and outlines future works.

II. Related Work

In this section, we discuss two different types of existing approaches for EH and some HRL works [10], [11].

A. Conventional Approaches Without RL

Most of the existing research is based on traditional methods, such as heuristic, mixed integer linear programming and dynamic programming. Authors often rely on unrealistic prior knowledge. Specifically, there are two types: one type is to assume that we already know the distribution of the energy arrival model or the specific channel model, then use the dynamic programming method or mixed integer linear programming to find the optimal decision [12], [13]. Another type is to assume that we already know environment prior knowledge, that is, before transmitting, we can get the energy harvesting and channel changes at the subsequent time, and use heuristic method to allocate the power of each time slot [9], [14], [15], [16], [17].

Although all the above methods are effective in the scenarios they consider, each method is based on the strong environment assumption, either directly or indirectly. As we mentioned earlier, accurate environmental assumptions are extremely difficult to obtain. Even if the current environment distribution is derived from historical data, the obtained offline algorithm is incapable of adapting to the constantly changing environment. In contrast, our model-free HRL methods can work without any prior knowledge.

B. RL Based Approaches

RL is a promising model-free algorithm for achieving a given goal by learning directly from the environment in an unknown environment [18]. The first RL work of designing EH wireless communication transmission strategy based on RL is [19], in which RL is applied to EH point-to-point communications system, and its performance is better than that of traditional offline methods. In both [20] and [21], the EH point-to-point communication system is discussed, and Q-learning algorithm is applied to learn a transmission strategy which maximizes the amount of data arriving at the destination. In [22], a well-designed Q-learning-based architecture was designed to find the best Antenna selection scheme and achieved ideal results. With the development of technology, neural network (NN) has been applied in RL. The first DRL work is [23], in which the DRL is used to design energy allocation and multiple access control strategy for a multi-access system that transmits data to multiple users based on EH module access control access point. In [23], [24] and [25], the DRL-based access scheme of EH multiple

access system has achieved better results than traditional algorithms without prior knowledge. In [26], a distributed multi-agent DRL algorithm is proposed, which is based on the same reward function for all nodes. In [27], the DRL framework for online operation of multi-hop EH-WSNs is proposed for the first time, and distributed and centralized architectures are designed. According to the channel status information, battery status and packet priority, the node adjusts its selective transmission strategy. Especially, in [28], the DRL-based optimal transmission strategy of single link EH-WSN is proposed by using a monotone neural network. However, these RL methods and frameworks can only be used in some simple environments, and do not work well in the high-dimensional search space.

C. Some HRL Works

To adapt to the situation of high-dimensional space, a reinforcement learning extension framework-hierarchical reinforcement learning (HRL) began to be studied, known for some methods: HAM [29], Options [30] and MAXQ [31]. The key to HRL is an abstraction, by removing irrelevant or redundant information or adding transcendental upper-level guidance. There are some abstractions in HRL, including but not limited to state space abstraction [32], task abstraction [33], and time abstraction [34]. In [35], a proactive VNE algorithm relying on HRL was proposed, which better than stat-of-the-art VNE algorithms and better robustness when the type of VNR changes. Moreover in many large-scale projects [36], [37], [38], HRL shows far more better learning ability than DRL. However, there is no universal abstraction method that can obtain generally satisfactory results in all problems which means that the abstract methods in each problem need to be designed separately.

III. System Model

The framework of the system is shown in Fig. 1. The system provides communication service for K users (UE) through N orthogonal channels (CH) by sending packets, running in the form of time slots. Especially, this system only uses an EH module to supply power, and carries a battery to store the power collected by the EH module.

A. Channel Model

Considering the characteristic that orthogonal frequency division multiplexing channel can only be assigned to one user at one time, the overall channel state in each time slot t is denoted as a $N \times K$ matrix $cs[t]$ where the element $cs_{i,j}[t]$ stands for i orthogonal channel assigned to user j. The channel state is fast fading which means that the channel state changes rapidly between the unit duration and the CH information in each time slot is uncertain. Therefore, $cs_{i,j}[t]$ is random, and the random distribution is strongly related to the environment. Obviously, the shape of the channel state matrix $cs[t]$ is determined by the number of users K and the number of orthogonal channels N contained in the system.

In the current wireless communication system, only a few predefined discrete communication models are actually supported which corresponding to different channel coding rates [39]. Let $r_{ap} = \{r_1, r_2, \cdots, r_m\}$ $(r_1 < r_2 < \cdots < r_m)$ to represent the m kinds of discrete transmission rate and $tr_i \in \{tr_1, tr_2, \ldots tr_m\}$ to represents the corresponding minimum received signal power to the transmission rates in r_{ap}. In order to support the operation of the minimum transmission rate $r_k (1 \leq k \leq m)$, the accepted power(attenuated fixed transmitting power)is required to be at least greater than the minimum decoding power tr_k but less than tr_{k+1} (if $k = m$ then $tr_{k+1} = \infty$), that is, the channel state $cs_{i,j}[t]$ satisfies $tr_k \leq p * cs_{i,j}[t]$ but $p * cs_{i,j}[t] < tr_{k+1}$, so as to obtain:

$$tr_k / p \leq cs_{i,j}[t] < tr_{k+1}/p \tag{1}$$

where p is the fixed system transmitting power. From Eq. 1, it can be seen intuitively that the channel state $c_{i,j}[t]$ can be measured by the minimum received signal power $tr_k (1 \leq k \leq m)$.

Furthermore, we use transmission rate to represent the current channel state $cs[t]$. Therefore, all the channel states $cs_{i,j}[t]$ mentioned hereafter are referred to as the maximum transmission rate can be sent through the j channel to the i user.

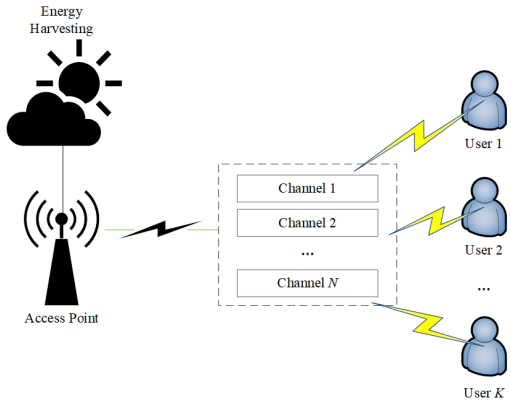


Fig. 1. Framework of the system.

B. Access Control

Similar to the representation method of channel state, the access control action in time slot t is denoted as a $N \times K$ matrix $a[t]$ and the element $a_{i,j}[t]$ in $a[t]$ satisfies $a_{i,j}[t] \in \{0, 1\}$, where $a_{i,j}[t] = 0$ indicates that channel i is not been assigned to user j, and $a_{i,j}[t] = 1$ indicates assigning.

Obviously, if channel state $cs[t]$ and access control $a[t]$ are given, the total transmission rate $r[t]$ can be obtained by multiplying the access control action $a[t]$ and the channel state $cs[t]$, i.e.:

$$r[t] = a[t] \circ cs[t] = \sum_{i=1}^{N} \sum_{j=1}^{K} a_{i,j}[t] * cs_{i,j}[t]$$

where $\circ$ means Hadamard Product (each element is the product of corresponding position elements of the original two matrices).

C. Energy Harvesting Model And Battery Model

The EH module starts working at the same time as AP and the energy collected by the EH module in time slot t is denoted as $e[t]$, which is regarded as a Poisson process with capacity constraint, similar to many previous works [40], [41].

The battery has a limit of capacity b_{max}, and the amount of energy stored in the battery is expressed as $b[t]$. The energy consumption due to work in each time slot is denoted as $p[t]$ and can be directly obtained by

$$p[t] = (\sum_{i=1}^{N}\sum_{j=1}^{K} a_{i,j}[t]) \times p$$

where p is the fixed system transmitting power. In particular, access control action cannot be executed when the energy consumption $p[t]$ of the action is greater than the current energy $b[t]$ of the battery.

Like many other works, the whole workflow involving energy harvesting and energy consumption is considered as a typical Markov decision process [42]. At the beginning of time slot t, the workflow calculates the remaining energy $b[t]$ of the previous time slot. Following, at the end of the time slot t, the workflow calculates the working energy $p[t]$ and energy $e[t]$ collected. Then the energy $b[t+1]$ at the beginning of the next time slot can be definitely calculated by Markov decision process:

$$b[t+1] = min(b_{max}, b[t] + e[t] - p[t])$$

Specially, we discretize both energy of the battery $b[t]$ and harvesting energy $e[t]$. Assume that sending a data packet consumes a unit power.

D. System Overview

For the considered system, the transmission strategy used is binary transmission strategy, where there are only two transport options available: full service and no service. The total access control policy π makes access control decisions $a[t]$ based on the current channel state $cs[t]$, EH module performance $e[t]$ and electric quantity $b[t]$, which can be expressed as:

$$\pi\,(cs[t], e[t], b[t]) \longrightarrow a[t]$$

Therefore, the workflow of the whole system can be summarized as follows: firstly, at the beginning of t time slot the system decides to allocate action $a[t]$ based on the access control policy π. Then according to the allocation action $a[t]$ and $cs[t]$, the total transmission rate $r[t]$ and the corresponding energy consumption $p[t]$ are calculated respectively. Finally, according to the battery storage $b[t]$ at the beginning of the round, the energy $e[t]$ collected in the current round and the energy consumption $p[t]$, the electric energy $b[t+1]$ in the battery at time slot $t+1$ is calculated. The whole cycle goes to the next one.

IV. Problem Formulation

In this section, the access control problem is formulated in MDP form, and then the RL framework is adopt to solve it.

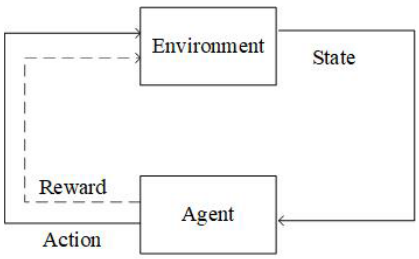


Fig. 2. MDP based reinforcement learning process.

A. Markov Decision Process

The MDP can be represented by a quad: state space S, action space A, reward function R and transition function. The problem we considered is a finite-horizon MDP with T slots. The interaction process is shown in Fig. 2 and the four parts are given below:

1) State space S: the element $s[t]$ in state space S represents the system state at time slot t, consisting of channel state $cs[t]$ and battery energy $b[t]$. Thus, the $s[t]$ can be written as:

$$s[t] \in S = \{cs[t] \oplus b[t]\}.$$

2) Action space A: the action space is a set of all possible access control action:

$$a[t] \in A$$

where $a_{i,j}[t] \in \{0,1\}$ and $a_{i,j}[t] = 0$ indicates that the i channel is not assign to the j user, and $a_{i,j}[t] = 1$ indicates assigning.

3) Reward function R: the reward function is defined as the total transmission rate in a single time slot:

$$R(s[t], a[t]) = a[t] \circ cs[t] = \sum_{i=1}^{N}\sum_{j=1}^{K} a_{i,j}[t] * cs_{i,j}[t]$$

where $\circ$ means Hadamard Product. Obviously, higher transmission rate mean higher throughput. Therefore, the original problem is converted to get more reward in the MDP.

4) Transition Function $ts[t]$: the transition function $ts[t] : S \times A \rightarrow S'$ is defined as how the system sate $s[t]$ in current time changes if an action $a[t]$ is adapted.

The communication system needs to determine the policy $\pi : S \rightarrow A$, which maps each state to the corresponding action.

We aim to develop the optimal policies $\pi^{\star}$ for the communication system so as to maximize its expected sum-throughput (the expected total reward R_{sum} under the MDP model) overall T slots, subject to the channel using constraint and the total power. The MDP problem is therefore formulated as:

$$\max_{\pi} R_{sum} = \mathrm{E}\left[\sum_{t=1}^{T} R(s[t], a[t]))\right]$$
$$\text{s.t. } p[t] < b[t], 1 \le t \le T$$

MDP is the optimal solution to the problem. Generally, such an optimization problem certainly can be solved by

some traditional methods, such as value iteration [43] and policy iteration [44]. However, these traditional methods require the explicit transition model for optimization. But in many environments, the transition model is unavailable to the relaying system. For instance, the channel state of a link at certain time or the energy of EH over a period of time is both affected by many complex factors, which are difficult, sometimes even impossible to be modeled. A simple solution [43], [44] is to estimate the transition model by sampling, but the performance of this method becomes very sensitive to sampling quality and is difficult to achieve good results in a dynamic environment. Therefore, we deploy deep reinforcement learning (DRL) which requires neither explicit transition function, i.e., prior knowledge of the environment.

B. Standard RL And DRL

(1) Standard RL: Q-learning is the representative algorithm of standard reinforcement learning (RL). In Q-learning, agent interacts with the environment directly and in every interaction step t, the agent take an action $a[t]$ to the state $s[t]$ then obtain the feedback, i.e., the reward $r[t]$ and the state $s[t]$ goes to the next state $s[t+1]$. Throughout the interaction, Q-learning maintains a lookup table of the state-action values, i.e., Q-values $Q(s,a)$ and updates Q-value by the temporal difference (TD) method:

$$Q(s,a) \longleftarrow Q(s,a) + \alpha[R + \gamma \max_{a' \in A}(Q(s',a') - Q(s,a))]$$

where s' is the successor state, $\alpha \in (0,1]$ is learning rate and $\gamma \in [0,1]$ is discount rate. The TD method is performed recursively for all the experiences until convergence. The key idea of Q-learning is to obtain the optimal policy by selecting the optimal action (the one with the highest Q-value $Q^{\star}(s,a)$) for every state in Q-table.

However, for a system with large state space and action space, the calculation of Q-table is quite time-consuming and space-consuming. What is more, updating the Q-table becomes difficult and the convergence speed becomes very slow because a single trajectory is very sparse in overall space.

(2) Deep learning form: To solve the limitations imposed by the space, the deep neural network with a set of weights θ is used as an approximation function to calculate the Q-value $Q(s,a)$ e.g., $Q(s,a) \approx Q(s,a,|\theta)$. In deep Q-network (DQN), the loss function is the error between the target and actual function value, as defined below:

$$L_i(\theta_i) = E[(r_{t+1} + \gamma \max Q(s_{t+1}, a', |\theta_i^-) - Q(s_t, a, |\theta_i))^2]$$

where $\gamma \in [0,1]$, θ_i^- is precursor weights and a' is the precursor action. By reducing the error between the objective function value and actual function value, the gradient of the objective function value is obtained and optimized. Finally, weight θ is updated by utilizing:

$$\nabla_{\theta_i} L_i(\theta_i) = E[(r_{t+1} + \gamma \max Q(s_{t+1}, a', |\theta_i^-) - Q(s_t, a, |\theta_i))^2] \nabla_{\theta_i} Q(s_t, a, |\theta_i)^2$$

where θ_i^- is precursor weights. The framework diagram of DRL algorithms is shown in Fig. 3.

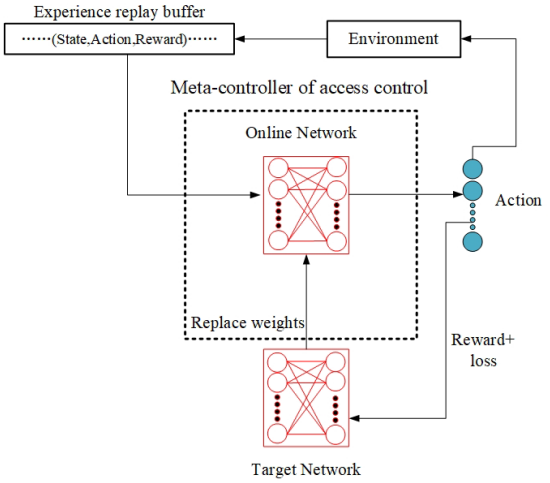


Fig. 3. DRL based framework.

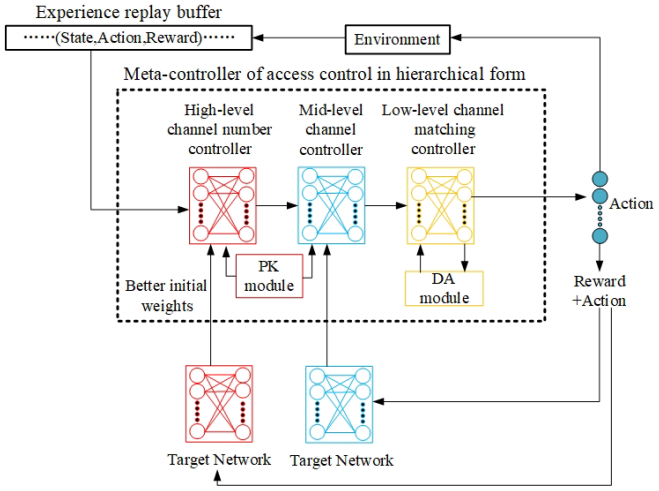


Fig. 4. HRL based framework.

In essence, DQN only achieves small-space storage Q-table through the generalization ability of neural network, but does not solve the problem of low efficiency in exploration. The neural network can only infer the quality of other actions from the existing Q value, but can not make the correct judgment of Q value autonomously. Therefore, the generalized problem is still unsolved: RL is not good at judging unexplored strategies, and RL still needs to try almost every alternative possibility. To solve these problems, this paper adopts a hierarchical structure based on DRL with two additional modules, PK and DA modules. It aims to further reduce the action space at the level of high abstract dimension, avoids some obvious bad action choices and improves the exploration efficiency.

V. Proposed HRL Framework

In this section, the details of our hierarchical policy framework are shown in Fig. 4. There are three sequential task controllers in the hierarchical policy framework. The downstream sub-task controller learns tasks under the constraints of the results of the upstream controller, that is,

they are guided by the upstream controller. In addition, based on the prior knowledge of system structure dynamics, two methods to improve learning efficiency of the HRL framework are proposed.

A. Hierarchical Agent Composition

This section illustrates the links between the meta-controller (corresponds to the meta access control task) and sequence controllers (corresponds to sequence sub-tasks). The policy in this paper is denoted as $\pi(s \mid g) : S \rightarrow A$, where goal g is restriction (guidance) from upstream controller. It is mentioned that goal g is not necessary. If there is no goal g, policy $\pi(s) : S \rightarrow A$ means there is no restriction (guidance) from upstream controller,which is exactly normal notation.

We manually decompose the meta access control policy $\pi_{Meta}(s) : S \rightarrow A$ into three crucial sequence steps and design three corresponding controllers to learn the sub-policy, namely high-level controller $\pi_h(s) : S \rightarrow G_1$, middle-level controller $\pi_m(s \mid g_1) : S \rightarrow G_2$ $(g_1 \in G_1)$ and low-level controller $\pi_l(s \mid g_2) : S \rightarrow A$ $(g_2 \in G_2)$. The goal G_1 is a set of all number of channels selected for transmission in the single turn and G_2 is a set of all combinations of users be served in a single turn. The details of the three controllers are shown below:

- The high-level controllers $\pi_h(s) : S \rightarrow G_1$ is responsible for determining the number of channels is used for the turn based on the channel status, takes the original state $s \in S$ as the inputs, and chooses the subgoal g_1 from G_1 as its output action.
- The middle-level controller $\pi_m(s \mid g_1) : S \rightarrow G_2$ is responsible for determining which users is serviced based on the power consumption selected by the high-level controller and the current channel state, takes the original state $s \in S$ masked by the subgoal of the high-level controller g_1 as the inputs, then chooses a subgoal g_2 from G_2 as its output action.
- The low-level controller $\pi_l(s \mid g_2) : S \rightarrow A$ is responsible for determining the best channel and user allocation pair for transmission, takes a state $s \in S$ and the subgoal of the middle-level controller G_2 as the inputs, and chooses a goal action $a \rightarrow A$ as its output action.

In essence, goals g_1, g_2 of the sequential sub-task is the intermediate state of the mapping from the original input to the original output, i.e., the meta-policy:

$$\pi_{Meta}(s) : s \rightarrow g_1 \rightarrow g_2 \rightarrow a$$

each of goals corresponds to a step. Obviously, by increasing the intermediate states, the original space is divided into smaller subsets and conversely, for each action a, goals g_2 and g_1 are existing.

Moreover, we use an example to illustrate the hierarchical selection process. Suppose that $r_{ap} = \{1, 2, 3\}$ and the system has $K = 3$ orthogonal channels and $N = 3$ UEs. For a certain time slot t, the battery state of the AP is $b[t] = 3P = 3E_0$ and the system channel state is given as:

$$cs[t] = \begin{pmatrix} 3 & 2 & 0 \\ 1 & 2 & 1 \\ 1 & 0 & 2 \end{pmatrix}$$

We assume that, based on $cs[t]$, the high-level controller selects a goal $G_1 = 2$, which means the high-level controller decided to use 2 orthogonal CH to transmit in this time slot. Then, based on the $cs[t]$ and $G_1 = 2$, we assume that the high-level controller selects a target $G_2[t] = [1, 1, 0]$, which means the mid-level controller decided to provide data transmission services for UE_1 and UE_2 in this time slot. Finally, based on the $cs[t]$ and $G_2[t] = [1, 1, 0]$, we obtain the masked

$$\overline{cs[t]} = \begin{pmatrix} 3 & 2 & 0 \\ 1 & 2 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

and then assume that the low-level controller selects an action:

$$a[t] = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

which means the low-level controller decided to provides data transmission services for UE_1 through orthogonal channels C_1 and UE_2 through orthogonal channels C_2 in this time slot. From the currently selected $a[t]$ and $s[t]$, we can directly compute R with the Hadamar product of $cs[t]$ and $a[t]$:

$$\begin{aligned} R(s[t], a[t]) &= a[t] \circ cs[t] \\ &= \sum_{i=1}^{N} \sum_{j=1}^{K} a_{i,j}[t] * cs_{i,j}[t] \\ &= 3 * 1 + 2 * 1 = 5 \end{aligned}$$

We can intuitively verify that of all possible actions, the action that produces the greatest reward is the currently selected action $a[t]$.

From the above example, the sub-controller has fewer space and better explanations than the meta-controller, and it is easier to learn stable policy. However, the action space of the low-level controller (i.e. action space of the original problem) is not reduced. Although learning efficiency is improved, the large scale of neural network output layer limits the overall architecture design of neural network.

Since the output of the high level controller is $a \in [0, \cdots, N]$, the output of the mid level controller can be calculated as $\binom{N}{a}$. Furthermore, the output of the low level controller is on the order of $\binom{N}{a} * \frac{N!}{(N-a)!}$. Even if N is 10^1, the lower level controller has an action space output dimension of over 10^5. Therefore, we propose deterministic action learning module to avoid the representation difficulty of lower level controller.

B. Deterministic Action (DA) Learning Module

The exploration strategy used in reinforcement learning is random exploration strategy, which leads to low learning efficiency. However, not all potential actions need to be explored. Under the same conditions, some actions are definitely worse than others, for example, sending fewer packets with the same energy is definitely not the potential best.

Considering that the layered reinforcement learning strategy brings better knowledge of structural dynamics, the best action is sometimes deterministic under the guidance of the

upper level, and random search is not needed. Based on the mentioned above, firstly, the low-level channel matching problem is formulated into an optimization problem with constraints g_1, g_2 determined by the mid-level controller and the low-level controller as:

$$\max_{a\in A} \sum_{i}^{N}\sum_{j}^{K} a \circ s \tag{2}$$

$$\text{s.t. } a_{i,j} \in \{0,1\}, \quad \forall i \in \{1,\cdots,N\}, \forall j \in \{1,\cdots,K\}$$

$$\sum_{i}^{N}\sum_{j}^{K} a_{i,j} = g_1 \tag{3}$$

$$\sum_{j}^{K} a_{i,j} = 1, \text{ if } i \in g_2 \tag{4}$$

$$\sum_{j}^{K} a_{i,j} = 0, \text{ if } i \notin g_2 \tag{5}$$

$$\sum_{i}^{N} a_{i,j} \leq 1, \forall j \in \{1,\cdots,K\} \tag{6}$$

$$\sum_{j}^{K} a_{i,j} \leq 1, \forall i \in \{1,\cdots,N\} \tag{7}$$

where a is the target action and s is the current state. Eq. (4) means that the channel is selected to work, that is, the channel number is present in target G. On the other hand, Eq. (5) means that the channel is not selected. Eq. (6) and Eq. (7) represents the restriction of the orthogonal channel.

Because the target action a is represented by a matrix and satisfies $a_{i,j} \in \{0,1\}$, our problem becomes a deterministic 0-1 integer programming problem (IP), which Eq. (2) is an NP-hard problem. However, because of Eq. (4) and Eq. (5), our optimization goal becomes a special kind of IP problem – the maximum weight matching problem.

Theoretically, it's possible to loop over all the possible combinations and calculate it's reward but the time computation is too high. Since the Kuhn-Munkres (KM) method is an effective approach for solving the maximum weighted matching problem, we propose a KM method based algorithm Alg. 1 to find the optimal action a under constraint g_2. In Alg. 1, we first initial the optimal action in step 1 and rewrite the original optimization problem by masking the original state in step 2. Then in step 3 to step 4 we use the KM algorithm to obtain the matching result with the largest weight. In step 5 to step 9, we generate the corresponding selection action based on the largest weight in step 4. Finally, in step 10, return the optimal action a^*. The complexity of Algorithm 1 is $O((N \times K)^3)$.

Algorithm 1 Optimal Action Search Algorithm

Require: cs: channel state; N: number of users; K: number of channel; g_1: high-level target; g_2: mid-level target;
Ensure: Optimal action a^* under g_1 and g_2
1: Initialize $a^* \leftarrow [0]_{N\times K}$
2: Mask the rows of cs according to g_2 and invert each element in cs
3: Use Kuhn-Munkres method to solve the matching problem of cs
4: Find the $\overline{a}$ with largest maximum reward
5: **for** i, j in $\overline{a}$ **do**
6: **if** $\overline{a}_{i,j} \neq 0$ **then**
7: $a^*_{i,j} \leftarrow 1$
8: **end if**
9: **end for**
10: **return** a^*

C. Prior Knowledge (PK) Learning Module

In this section, PK module is proposed to improve the efficiency of sample selection, which makes agent adapt to different learning tasks faster. The key to improving sample efficiency is to find common knowledge between different tasks. In fact, for the considered system, there are some learnable structural features that are independent of environmental dynamics (channel gain process and energy harvesting process). Especially when using the HRL framework to decompose the decision into continuous sub decisions, the functions of these structural features are also decoupled. For example, from the energy point of view, if both EH energy and battery energy are at a high level (almost full), the action of the high-level controller always tends to select more channels for transmission, thus preventing the waste caused by excessive consumption of low energy. On the other hand, once the action of the high (middle) level controller is selected, the selection of the low-level controller has nothing to do with the EH process and only depends on the channel quality and the number of transmission channels (selected users).

The above shows that there are some common knowledge between tasks in the system under consideration. We believe that even if the hypothetical environment is very different from the real working environment, at least some structural features may still help agents to better capture the environmental features at the beginning of learning. Therefore, we introduce a good prior for RL by assigning an appropriate initial value to the neural network. The initial value is obtained by training the agent in a proper hypothetical environment.

Therefore, the overall algorithm is shown in Alg. 2:

In Alg. 2, we use two groups of NN as high-level controller and middle-level controller while the low-level controller is replaced by DA module. In step 1 to step 4, we initialize the neural network with the pretrained weights from PK module. Then in the loop (step 6 to step 16), we initialize the environment and conduct corresponding experiments and update the neural networks used to maintain the high-level controller and the middle-level controller respectively. Specifically, from step 8 to step 9, the related actions of three levels are carried out separately. In step 10, the environment generates reward and transfers it to the next environment. In step 11 to 12, the experience are stored separately in experience replay buffer. Finally, in step 13 to 15, a minibatch of experiences are extracted from the experience replay buffer

Algorithm 2 HRL-Based Framework

Initialize DNN parameters θ_h for high-level Q-network and θ_m for mid-level Q-network with random weights
Initialize experience replay buffer: β_h for higher level, and β_m for mid-level
Train HRL in arbitrary transition model to get the pretrained weights θ_h^{pre} and θ_m^{pre}
Set $\theta_h = \theta_h^{pre}$ and $\theta_m = \theta_m^{pre}$.
repeat
  Initialize the environment, obtain initial state s
  for time $t = 1, 2, ..., t_{max}$ **do**
    Choose sequence high-level action a_h and mid-level action a_m using ϵ-greedy method
    Choose low-level action by DA module a_l
    Execute action a_l, receive reward r_t from the environment and observe next state s'
    Collect and save the tuple (s, a_h, r, s') in β_h
    Collect and save the tuple (s, a_l, r, s') in β_l
    Sample a minibatch in β_h to finely tune θ_m with θ_m^-
    Sample a minibatch in β_l to finely tune θ_l with θ_l^-
    Update current state $s = s'$
  end for
until Convergence

and used to update the neural network. This loop is repeated until the neural network converges.

VI. Performance Evaluation

In this section, we first introduce the setup of experimental environment. Then we carry out experiments to evaluate the proposed algorithms.

A. Setup

To implement our proposed framework, we use two separate dueling DQN (target network and online network) which share the same structure for both high-level controller and mid-level controller. The number of neurons in input layer is equal to the sum of numbers of states and sub-goals, and the number of neurons in output layer corresponds to the dimensions of action. Both dueling DQN include two hidden layers, and the number of neurons in each hidden layer is 1.5 times that of the output layer corresponding to the dimensions of action space. The activation function for hidden layers is Relu function. The target network is updated every step, and the online network is updated every fixed step to break the data correlation by copying the weights from the target network. The low-level controller is made of DA module. The detailed parameters used in RL experiments are shown in the Tab. I.

In order to present smoother and more general performance comparisons, the rewards given in all numbers are further averaged by locally weighted linear regression smoothing method. All the simulations are performed on the deep learning framework in Keras. We set same random seeds to ensure that each round of experiments is fair under different environment settings.

TABLE I
RL Settings

RL parameter	Setting
learning rate	0.002
ϵ-decays	0.1 to 0.001 goes up with training
reward discount factor γ	0.9
replay buffer size	2000
replace target step	100
maximum training rounds	4000

TABLE II
System Settings

System parameter	Setting
user number	8
channel number	6
data rate	[0,3,5,7,10]
transmission energy consumption	E_0
maximum battery capacity	$20E_0$
initial battery energy	$20E_0$

The maximum supportable data rates of the channels w.r.t the UEs in each time slot are subjected to a certain environmental distribution $f(r)$ (channel model) given in Tab. III. Thus, the probability to support a maximum data rate $r_m (1 \leq k \leq m)$ is given by $\int_{r_k}^{r_{k+1}} f(r)dr$. The energy harvesting rate in each time slot is subjected to a Poisson arrival process with different arriving rates (given in Tab. II). We set the transmitting power at the AP as $P = E_0$.

The main performance metric is the long-term average throughput of the system. The state, reward, primitive action spaces, and hierarchical action spaces was discussed in the previous article. In the considered system, the state and action spaces increase exponentially with the number of channels and UEs. For both efficiency and equity of proposed framework, we conduct experiments for system with a moderate number of UE and channels (high-dimensional action space leads to the large dimension of DQN output layer, which limits the size of the overall neural network), i.e., the following multi-channel wireless communication system is considered as in Tab. II.

Although the scale of the system seems not large, but in fact, the corresponding action space and state increase exponentially with it to a great extent. Take the considered system with 6 channels and 8 users as an example. The action space can be calculated by $\sum_{i=0}^{6} C_6^i \times A_8^i \approx 10^5$. The state space can be calculated by $5^{6\times 8} \approx 10^{33}$. Therefore, the corresponding combination (s, a) can be calculated as 10^{38}. Too many pairs of potential actions and states may cause great difficulties in policy convergence. Because the dimension of Q-value of neural network output layer is equal to that of action space, too many neural network parameters will cause memory overflow, limit the scale of neural network.

B. Methods Compared

Our framework is compared against the following methods in the experiments:

- **Random:** For each time slot, the agent randomly selects an action to execute communication with random

TABLE III
ENVIRONMENTAL SETTINGS

	Channel prob	EH prob
Env 1	Normal N($\mu = 2$,$\sigma = 1$)	Normal N($\mu = 2.0E_0$,$\sigma = 1$)
Env 2	Normal N($\mu = 2$,$\sigma = 1$)	Normal N($\mu = 3.5E_0$,$\sigma = 1$)
Env 3	Normal N($\mu = 2$,$\sigma = 1$)	Normal N($\mu = 5.0E_0$,$\sigma = 1$)
Env 4	Uniform	Normal N($\mu = 2.0E_0$,$\sigma = 1$)
Env 5	Uniform	Normal N($\mu = 3.5E_0$,$\sigma = 1$)
Env 6	Uniform	Normal N($\mu = 5.0E_0$,$\sigma = 1$)
Env 7	Normal N($\mu = 2$,$\sigma = 1$)	Normal N($\mu = 7.0E_0$,$\sigma = 1$)
Env 8	Normal N($\mu = 2$,$\sigma = 1$)	Normal N($\mu = 11.0E_0$,$\sigma = 1$)
Env 9	Uniform	Normal N($\mu = 7.0E_0$,$\sigma = 1$)
Env 10	Uniform	Normal N($\mu = 11.0E_0$,$\sigma = 1$)

transmission power. As a degenerate strategy without learning knowledge, random scheme plays a baseline here.

- **DRL:** We adapt the traditional dueling DQN framework, which is the most common DRL scheme in recent works. Due to the limitation of output action space, the total number of layers of Q-network used is only two layers.
- **HRL:** We adopt the traditional HRL framework, which decomposed the original problem as shown described in Section V-A. The HRL framework is not contain PK and DA module. In this framework, each controller is served by a separate neural network.
- **HRL+PK:** On the basis of HRL, PK module is added.
- **HRL+DA:** On the basis of HRL, the low-level controller is replaced with DA module.
- **COMB+DA:** On the basis of HRL+DA, the high-level controller and the middle-level controller of HRL are replaced with the combined controller (COMB) to directly determine which channels participate in the matching.
- **HRL+DA+PK:** On the basis of HRL, the low-level controller is replaced with DA module, and a better set of weights is obtained from PK module for HRL. The other settings are exactly the same as HRL.

Then, we evaluate the above methods in ten different environments as shown in Tab. III. Among them, Env1-Env6 are prepared for normal scale experiments, and Env7-Env10 are prepared for large-scale experiments.

C. Results And Discussion

In this section, we evaluate the performance (throughput), different selection schemes, scalability, and learning efficiency of the algorithm numerically. Considering that the simulation experiment environment in which our system is located is highly random, we make locally weighted scatter plots smoothing on the performance result graph of the experiment, so as to make a more intuitive comparison.

1) Performance: In Fig. 5, the performance of the proposed HRL framework (marked as HRL+DA+PK) is compared with other methods mentioned before. All the models used in the comparison of Fig. 5 have been trained. The experimental results as shown in Fig. 5. It can be seen from Fig. 5 that the performance of HRL methods are better than DRL and random methods. Specifically, the average throughput in

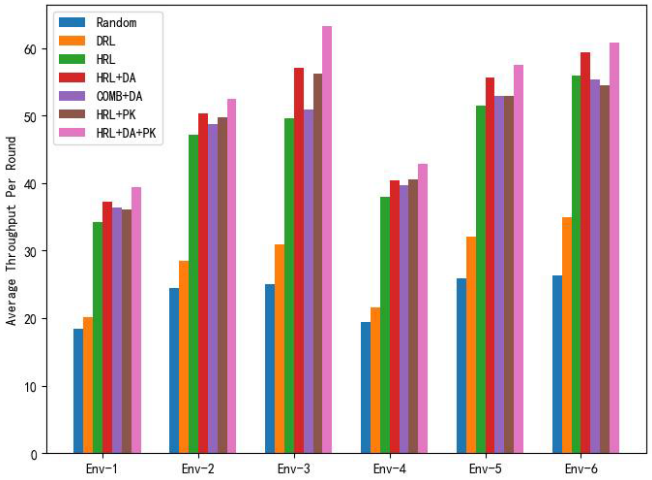


Fig. 5. Performance comparison in different environments.

TABLE IV
SCALABILITY ENVIRONMENTAL RESULTS-1

ENV \ U-C scale	8U-6C	16U-12C	24U-18C
Env 2	41.832	-	-
Env 7	-	86.683	-
Env 8	-	-	122.962

six environments shows that the effect of HRL, HRL+PK, HRL+DA, and HRL+DA+PK methods are about 39.85%, 54.10%, 56.42%, and 65.35% higher than that of DRL, respectively. The reason is that too large space and not good EH energy condition cause the DRL method not learning much knowledge. The experimental results support that HRL framework has better learning ability than DRL framework. In addition, the average throughput of our framework (HRL+DA+PK) is the highest among all approaches in every environment. The experimental results in Fig. 5 support that PK and DA learning modules improve the performance of access control in long-term throughput of the system.

2) Different Selection Schemes: The combination of the three different selection schemes proposed is the highest level abstraction of the original matching scheme. Correspondingly, the DRL scheme can be considered as the lowest level abstraction (no abstraction) of the original matching problem.

Therefore, to investigate the effect of the scheme with lower abstract level than the scheme proposed, a combined controller (COMB) is used to replace the high-level controller and the mid-level controller in HRL to directly make decisions on the users participating in the matching. The combined controller is a neural network of the same size as that used in DRL. The COMB + DA module is compared with the proposed HRL + DA scheme.

It can be seen from Fig. 5 that the average throughput of COMB + DA is worse than that of HRL + DA in six different environments. This indicates that the multilevel division can help selection schemes achieve better performance.

3) Scalability: The scalability is to evaluate the throughput of proposed method when the U-C (user-channel) scale is

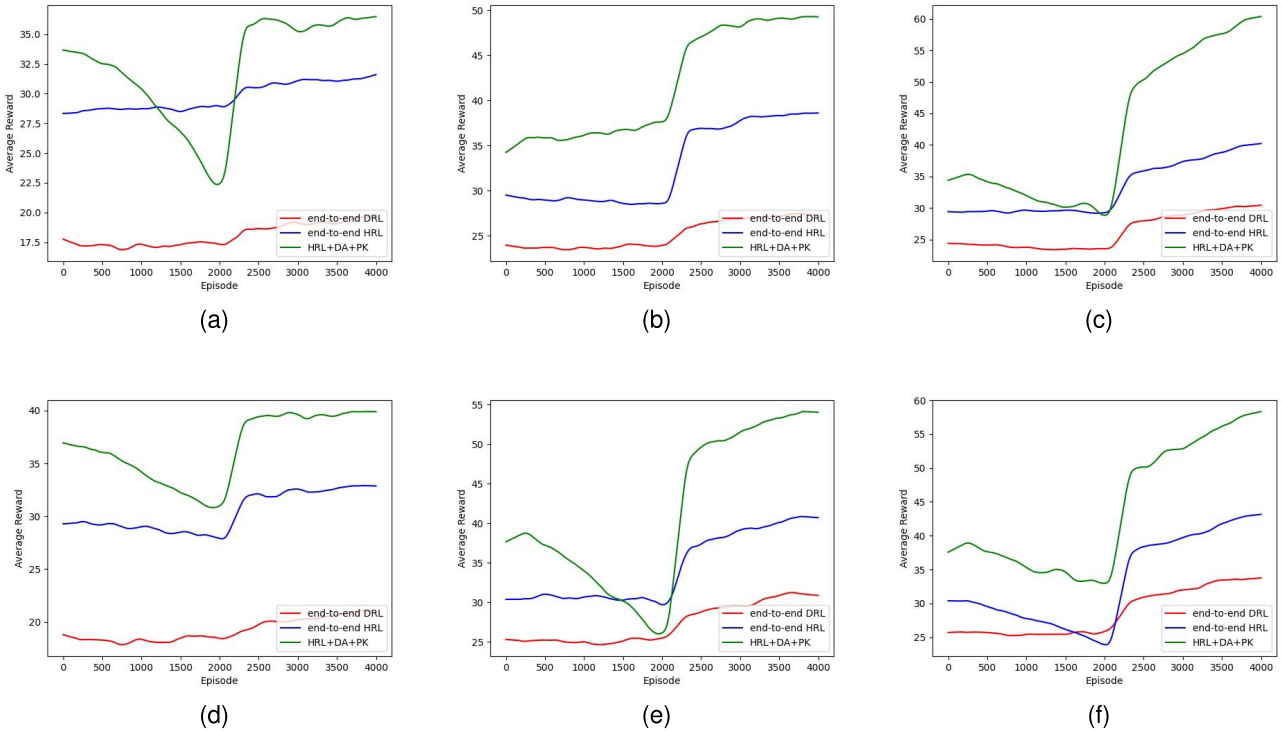


Fig. 6. Learning efficiency in different environments.

TABLE V
SCALABILITY ENVIRONMENTAL RESULTS-2

ENV \ U-C scale	8U-6C	16U-12C	24U-18C
Env 5	48.628	-	-
Env 9	-	88.971	-
Env 10	-	-	130.866

expanded. Moreover, to eliminate the influence of energy on the throughput, the EH value is correspondingly expanded. In the experiment, based on the original U-C scale (8 users and 6 channels) and the original EH value (Normal N ($\mu = 3.5E_0$, $\sigma = 1$)), the U-C scale and EH value are expanded to twice and triple the original scale (value). Under two different channel gain conditions, the experiment is carried out based on four simulation environments. Tab. IV and Tab. V present experimental results of groups (env-2 and env-5) with same channel gain.

It can be seen from the diagonal grid of the Tab. IV and Tab. V, the average throughput of the experimental group is twice or triple original U-C scale experimental group. This shows that the proposed method has a good scalability.

4) Learning Efficiency: In Fig. 6, the performances of these frameworks during the training process are presented. Among the results in Fig. 6, our framework achieved better initial performance at the very beginning and showed a higher learning efficiency. By comparing with the DRL framework, our framework is better all the time for both initial performance and learning efficiency. In Fig. 6(a) and Fig. 6(e), even if it falls into the local minimum point in several environments, our framework soon starts to rise and quickly achieves better results than HRL. The initial decline can be explained as the difference between the environment in which the PK model is located and the real environment (because we use the same PK model and fixed random seeds, the trend is similar in different environmental models). Due to the existence of DA module, the invalid low-level action space is greatly reduced. Therefore, from 2000 steps of interaction, all the effects have been greatly improved. If the final results of the HRL framework are taken as the benchmark, it only takes about 60% of the time of the HRL framework for our model to reach the benchmark.

VII. CONCLUSION

In this paper, we propose a HRL based access control framework to dynamically select access control, in order to maximize the long-term system average throughput. Unlike traditional studies, our method does not require any assumptions about channel distribution, but relies on the interaction between the agent and the communication environment. The proposed HRL framework decomposes the original control problem into three sequence control sub-problems through task abstraction, namely, high-level channel number selection problem, mid-level channel selection problem and low-level channel matching problem. Each sub-problem is solved in sequence and the final optimal scheme is obtained by combining the sub-problem schemes in the form of hierarchical. In order to improve the learning efficiency, two learning modules, DA module and PK module, are proposed.

The DA module solves the channel matching problem under the additional constraints given by the previous sub-problem, which selects definite good lower-level matching actions instead of traditional random exploration. The PK module provides the framework with the common knowledge of the system learned from the hypothetical environment for better initial performance. Simulation experiment compares our scheme with recently HRL, DRL and several conventional transmission schemes. The experiment results show that the transmission strategy obtained by our proposed framework achieves better throughput performance and better learning efficiency.

Our HRL method provides a novel way for the research of access control in the field of wireless communication. However, in the proposed framework, the transmission rate in the considered system is discretized into enumerable power levels, which can be further increased to a continuous level. In the future, we will explore new methods applicable to continuous action space, and adaptively control the transmission power to exploit the fading channel and multi-user diversity resulting in higher throughput.

REFERENCES

[1] J. Zhang, G. Xie, G. Han, Z. L. Yu, Z. Gu, and Y. Li, "Compressive sensing-based power allocation optimization for energy harvesting IoT nodes," *IEEE Trans. Wireless Commun.*, vol. 21, no. 6, pp. 4535–4548, Jun. 2021.

[2] Z. Yang, W. Xu, Y. Pan, C. Pan, and M. Chen, "Energy efficient resource allocation in machine-to-machine communications with multiple access and energy harvesting for IoT," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 229–245, Feb. 2018.

[3] S. Ulukus et al., "Energy harvesting wireless communications: A review of recent advances," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 3, pp. 360–381, Jan. 2015.

[4] J. Rinne, J. Keskinen, P. R. Berger, D. Lupo, and M. Valkama, "Feasibility and fundamental limits of energy-harvesting based M2M communications," *Int. J. Wireless Inf. Netw.*, vol. 24, no. 3, pp. 291–299, Sep. 2017.

[5] M.-L. Ku, W. Li, Y. Chen, and K. J. R. Liu, "On energy harvesting gain and diversity analysis in cooperative communications," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 12, pp. 2641–2657, Dec. 2015.

[6] G. Rohini, "Energy harvesting from machineries for industries: Vibration as a source of energy," in *Proc. Int. Conf. Syst., Comput., Autom. Netw. (ICSCAN)*, Jul. 2020, pp. 1–5.

[7] T. K. Thuc, E. Hossain, and H. Tabassum, "Downlink power control in two-tier cellular networks with energy-harvesting small cells as stochastic games," *IEEE Trans. Commun.*, vol. 63, no. 12, pp. 5267–5282, Dec. 2015.

[8] M.-L. Ku, W. Li, Y. Chen, and K. J. R. Liu, "Advances in energy harvesting communications: Past, present, and future challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1384–1412, 2nd Quart. 2016.

[9] K. Tutuncuoglu and A. Yener, "Optimum transmission policies for battery limited energy harvesting nodes," *IEEE Trans. Wireless Commun.*, vol. 11, no. 3, pp. 1180–1189, Mar. 2012.

[10] I. Ahmed, K. T. Phan, and T. Le-Ngoc, "Optimal stochastic power control for energy harvesting systems with delay constraints," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3512–3527, Dec. 2016.

[11] I. Ahmed, S. Yan, D. B. Rawat, and C. Pu, "Dynamic resource allocation for IRS assisted energy harvesting systems with statistical delay constraint," *IEEE Trans. Veh. Technol.*, vol. 71, no. 2, pp. 2158–2163, Feb. 2022.

[12] O. Ozel, K. Tutuncuoglu, J. Yang, S. Ulukus, and A. Yener, "Transmission with energy harvesting nodes in fading wireless channels: Optimal policies," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 8, pp. 1732–1743, Sep. 2011.

[13] F. Amirnavaei and M. Dong, "Online power control optimization for wireless transmission with energy harvesting and storage," *IEEE Trans. Wireless Commun.*, vol. 15, no. 7, pp. 4888–4901, Jul. 2016.

[14] J. Yang and S. Ulukus, "Optimal packet scheduling in an energy harvesting communication system," *IEEE Trans. Commun.*, vol. 60, no. 1, pp. 220–230, Jan. 2012.

[15] F. Yuan, Q. T. Zhang, S. Jin, and H. Zhu, "Optimal harvest-use-store strategy for energy harvesting wireless systems," *IEEE Trans. Wireless Commun.*, vol. 14, no. 2, pp. 698–710, Feb. 2015.

[16] M. B. Khuzani and P. Mitran, "On online energy harvesting in multiple access communication systems," *IEEE Trans. Inf. Theory*, vol. 60, no. 3, pp. 1883–1898, Mar. 2014.

[17] F. Liu, C. Jiang, and W. Xiao, "Multistep prediction-based adaptive dynamic programming sensor scheduling approach for collaborative target tracking in energy harvesting wireless sensor networks," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 2, pp. 693–704, Apr. 2021.

[18] R. S. Sutton and A. G. Barto, "Reinforcement learning: An introduction," *IEEE Trans. Neural Netw.*, vol. N-9, no. 5, p. 1054, Sep. 1998.

[19] P. Blasco, D. Gündüz, and M. Dohler, "A learning theoretic approach to energy harvesting communication system optimization," *IEEE Trans. Wireless Commun.*, vol. 12, no. 4, pp. 1872–1882, Apr. 2013.

[20] A. Masadeh, Z. Wang, and A. E. Kamal, "Look-ahead and learning approaches for energy harvesting communications systems," *IEEE Trans. Green Commun. Netw.*, vol. 4, no. 1, pp. 289–300, Mar. 2020.

[21] R. Wang, A. Yadav, E. A. Makled, O. A. Dobre, R. Zhao, and P. K. Varshney, "Optimal power allocation for full-duplex underwater relay networks with energy harvesting: A reinforcement learning approach," *IEEE Wireless Commun. Lett.*, vol. 9, no. 2, pp. 223–227, Feb. 2020.

[22] D. Ouyang, R. Zhao, Y. Li, R. Guo, and Y. Wang, "Antenna selection in energy harvesting relaying networks using Q-learning algorithm," *China Commun.*, vol. 18, no. 4, pp. 64–75, Apr. 2021.

[23] M. Chu, X. Liao, H. Li, and S. Cui, "Power control in energy harvesting multiple access system with reinforcement learning," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 9175–9186, Oct. 2019.

[24] F. Ait Aoudia, M. Gautier, and O. Berder, "RLMan: An energy manager based on reinforcement learning for energy harvesting wireless sensor networks," *IEEE Trans. Green Commun. Netw.*, vol. 2, no. 2, pp. 408–417, Jun. 2018.

[25] M. Chu, H. Li, X. Liao, and S. Cui, "Reinforcement learning-based multiaccess control and battery prediction with energy harvesting in IoT systems," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2009–2020, Apr. 2019.

[26] M. K. Sharma, A. Zappone, M. Assaad, M. Debbah, and S. Vassilaras, "Distributed power control for large energy harvesting networks: A multi-agent deep reinforcement learning approach," *IEEE Trans. Cognit. Commun. Netw.*, vol. 5, no. 4, pp. 1140–1154, Dec. 2019.

[27] H. Al-Tous and I. Barhumi, "Reinforcement learning framework for delay sensitive energy harvesting wireless sensor networks," *IEEE Sensors J.*, vol. 21, no. 5, pp. 7103–7113, Mar. 2021.

[28] K. Wu, F. Li, C. Tellambura, and H. Jiang, "Optimal selective transmission policy for energy-harvesting wireless sensors via monotone neural networks," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 9963–9978, Dec. 2019.

[29] R. Parr and J. Stuart Russell, "Reinforcement learning with hierarchies of machines," in *Advances in Neural Information Processing Systems*, M. I. Jordan, M. J. Kearns, and S. A. Solla, Eds. Denver, CO, USA: MIT Press, 1997, pp. 1043–1049.

[30] R. Ramesh, M. Tomar, and B. Ravindran, "Successor options: An option discovery framework for reinforcement learning," 2019, *arXiv:1905.05731*.

[31] T. G. Dietterich, "Hierarchical reinforcement learning with the MAXQ value function decomposition," *J. Artif. Intell. Res.*, vol. 13, no. 1, pp. 227–303, Aug. 2000.

[32] D. Abel, D. Arumugam, L. Lehnert, and L. Michael Littman, "State abstractions for lifelong reinforcement learning," in *Proc. 35th Int. Conf. Mach. Learn. (ICML)*, J. G. Dy and A. Krause, Eds. Stockholm, Sweden, Jul. 2018, pp. 10–19.

[33] Y. Fu, Z. Xu, F. Zhu, Q. Liu, and X. Zhou, "Learn to human-level control in dynamic environment using incremental batch interrupting temporal abstraction," *Comput. Sci. Inf. Syst.*, vol. 13, no. 2, pp. 561–577, 2016.

[34] A. Neitz, G. Parascandolo, S. Bauer, and B. Schölkopf, "Adaptive skip intervals: Temporal abstraction for recurrent dynamical models," 2018, *arXiv:1808.04768*.

[35] J. Cheng, Y. Wu, Y. Lin, Y. E, F. Tang, and J. Ge, "VNE-HRL: A proactive virtual network embedding algorithm based on hierarchical reinforcement learning," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 4, pp. 4075–4087, Dec. 2021.

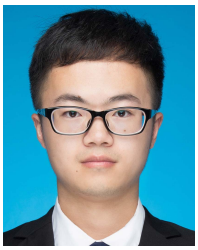
[36] C. Berner et al., "DOTA 2 with large scale deep reinforcement learning," 2019, *arXiv:1912.06680*.

[37] K. Arulkumaran, A. Cully, and J. Togelius, "Alphastar: An evolutionary computation perspective," 2019, *arXiv:1902.01724*.
[38] P. Dayan and E. G. Hinton, "Feudal reinforcement learning," in *Advances in Neural Information Processing Systems*, S. J. Hanson, J. D. Cowan, and C. L. Giles, Eds. Denver, CO, USA: Morgan Kaufmann, 1992, pp. 271–278.
[39] W. Wicke, N. Zlatanov, V. Jamali, and R. Schober, "Buffer-aided relaying with discrete transmission rates for the two-hop half-duplex relay network," *IEEE Trans. Wireless Commun.*, vol. 16, no. 2, pp. 967–981, Feb. 2017.
[40] F. Zeng et al., "Throughput maximization for two-way buffer-aided and energy-harvesting enabled multi-relay networks," *IEEE Access*, vol. 7, pp. 157972–157986, 2019.
[41] P. Sakulkar and B. Krishnamachari, "Online learning schemes for power allocation in energy harvesting communications," *IEEE Trans. Inf. Theory*, vol. 64, no. 6, pp. 4610–4628, Jun. 2018.
[42] P. Kamalinejad, C. Mahapatra, Z. Sheng, S. Mirabbasi, V. C. M. Leung, and Y. L. Guan, "Wireless energy harvesting for the Internet of Things," *IEEE Commun. Mag.*, vol. 53, no. 6, pp. 102–108, Jun. 2015.
[43] F. Ozcelik, G. Uctu, and E. Uysal-Biyikoglu, "Minimization of transmission duration of data packets over an energy harvesting fading channel," *IEEE Commun. Lett.*, vol. 16, no. 12, pp. 1968–1971, Dec. 2012.
[44] A. Aprem, C. R. Murthy, and N. B. Mehta, "Transmit power control policies for energy harvesting sensors with retransmissions," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 5, pp. 895–906, Oct. 2013.

Yingkai Wang is currently pursuing the master's degree with the School of Mathematics, Hefei University of Technology, China. His current research interests are in the areas of deep reinforcement learning, machine learning, and game AI.

Qingshan Wang (Member, IEEE) received the Ph.D. degree in computer science from the University of Science and Technology of China (USTC), Hefei, China, in 2007. He was a Visiting Scholar at Cornell University from 2009 to 2010. He is currently a Professor with the Department of Mathematics, Hefei University of Technology. His research interests include human action recognition, delay tolerant networks and ad hoc networks protocol design, and network coding.

Qi Wang received the Ph.D. degree in computer science from the Hefei University of Technology, Hefei, China, in 2010. She was a Visiting Scholar at Temple University from 2014 to 2015. She is currently an Associate Professor with the Department of Mathematics, Hefei University of Technology. Her research interests include delay tolerant networks, scheduling algorithm, and network coding.



Zhiwen Zheng received the B.S. degree in information and computing science from the Hefei University of Technology, Hefei, China, in 2018, where he is currently pursuing the Ph.D. degree with the Laboratory of Big-Data, School of Mathematics. His research interests include human action recognition and sign language translation.