Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 2672537, 12 pages
https://doi.org/10.1155/2020/2672537

Research Article

Exploration Entropy for Reinforcement Learning

Bo Xin,[1] **Haixu Yu**,[1] **You Qin**,[1] **Qing Tang**,[2] **and Zhangqing Zhu**[1]

[1] *Department of Control and Systems Engineering, School of Management and Engineering, Nanjing University, Nanjing 210093, China*
[2] *Nanjing Research Institute for Agricultural Mechanization, Ministry of Agriculture and Rural Area, Nanjing 210014, China*

Correspondence should be addressed to Zhangqing Zhu; zzqing@nju.edu.cn

Received 25 September 2019; Revised 2 November 2019; Accepted 20 November 2019; Published 9 January 2020

Academic Editor: Francesco Lolli

The training process analysis and termination condition of the training process of a Reinforcement Learning (RL) system have always been the key issues to train an RL agent. In this paper, a new approach based on *State Entropy* and *Exploration Entropy* is proposed to analyse the training process. The concept of *State Entropy* is used to denote the uncertainty for an RL agent to select the action at every state that the agent will traverse, while the *Exploration Entropy* denotes the action selection uncertainty of the whole system. Actually, the action selection uncertainty of a certain state or the whole system reflects the degree of exploration and the stage of the learning process for an agent. The *Exploration Entropy* is a new criterion to analyse and manage the training process of RL. The theoretical analysis and experiment results illustrate that the curve of *Exploration Entropy* contains more information than the existing analytical methods.

1. Introduction

Reinforcement learning (RL) has become an efficient solution to deal with the Markov Decision Processes (MDPs), such as traffic self-adaptive control [1], smart grid prediction [2], and activity-travel patterns in a city [3]. Along with the development of artificial neural networks [4], the application of RL has made great successes in the field of nonlinear and large-scale system decision problem. At the same time [5], the learning performance of RL and the judgment of the convergence for a certain RL algorithm has been the core problem for training an agent all the time. An agent learns the action selection strategy by interacting with the environment through trial-and-error and the acquisition of rewards [6]. Different values of reward corresponding to all state-action pairs are used to update the value function from which the action selection probability is generated in each training round. During the training process, the state transition probability indicates the character of the environment, and the action selection probability is metabolic which indicates the learning process of the agent [7]. In order to get more information of the training process and understand the learning problem more deeply, it is necessary to analyse the learning process with more efficient methods. Therefore, a new criterion is necessary to judge when to terminate the training process or whether an algorithm is better and faster than another one.

Meanwhile, some literature has analysed the training process for RL based on the parameters α, γ, and some other parameters which are also important for the convergence rate [8]. The influence to the convergence in RL from the major parameters, algorithmic complexity, the reward designing, and the training data is analysed in [9]. The storage complexity and exploration complexity are defined to analyse the complexity of RL for some complex problems, especially for quantum control systems [10]. However, these studies have only paid attentions to the change of the convergence results introduced by the parameters' variation, instead of the learning process of the agent.

On the other hand, the concept of *Entropy* that describes the randomness or uncertainty of a physical system has been introduced into information theory [11], called as *Information Entropy* which solved the problem of information quantification and is used to indicate the amount of information in the information source. It has been successfully applied in many fields such as information processing, artificial intelligence, and statistics [12]. In the field of MDPs [8], entropy has been used to optimize the decision results.

In the field of RL, some entropy-based methods have also been studied in depth. In [13], the maximum causal entropy framework was extended to discount reward setting in inverse RL. A kind of maximum entropy-based RL [14] was also applied to the problem of speech recognition for telephone speech. Ramicic and Bonarini used the entropy-based prioritized sampling method to optimize experience replay for deep Q-Learning [15]. Nevertheless, as far as we know, there is no result reported on the learning process for MDP based on entropy.

In addition, the causal sparse tsallis entropy regularization was applied into the sparse Markov decision with RL [16]. The computation and estimation for general entropy functions including classical Shannon and Rényi entropies in Markov chains are introduced [17]. A kind of estimators for the entropy of the ergodic homogeneous Markov chains with countable state spaces was constructed [18]. Girardin and Limnios defined an entropy rate and extended the Shannon-McMillan-Btriman theorem for a kind of countable discrete-time semi-Markov process [19]. Inspired by these studies, we focus on the change of the transition probability and action selection probability in the whole training process of RL. The concept of entropy will be used to define a new norm to illustrate the training process of an RL agent.

In consideration of the value function or the probability of action selection in RL, the agent has no knowledge about the environment before learning. The value function for each state cannot reflect the real and accurate information of the state. The probability of action selection for the agent at each state is random. The agent can get more than one possible paths along which it can get to the target, although most of them are not optimal for its mission. During the learning process, as the count of the trial-and-error increases, the value function of the agent can reflect the environment and the mission more and more accurately. For each state, the action selection becomes more and more certain. At last, the agent will get one or more optimal policies for the mission through the training process. This means that the value function for the RL system indicates some potential information through the change of the uncertainty of action selection, which has not been realized clearly until now. In [20–22], a kind of strategy entropy-based algorithm to accelerate the learning speed through self-adaptive learning rates was introduced, but the relationship between the strategy entropy and the learning process has not been explained clearly. In this paper, we define the concept of *Exploration Entropy* (EE) corresponding to each state and the whole system to explain the learning process of RL.

The rest of the paper is organized as follows. Section 2 introduces the concept of RL and Exploration Strategy. In Section 3, the definition of *Exploration Entropy* for RL is introduced, and related issues are analysed. In Section 4, the EE is applied to two kinds of typical RL systems to demonstrate the application of EE for the learning process of RL. Conclusions are given in Section 5.

2. Preliminaries

2.1. Reinforcement Learning. An RL agent learns a map between the environment state space and the action space through its interaction with the environment including observing the system's state, selecting and executing actions, and getting numerical action reward [23]. The mathematical theoretical basis of RL is discrete-time finite-state MDPs [24]. In a general way, a five-element tuple $\left\{S, A_{s_i}, p_{s_i s_j}(a), r_{(s_i,a)}, V : s_i, s_j \in S, a \in A_{s_i}\right\}$ is often used to define the MDP model for RL [25, 26]:

(i) $S = \{s_1, s_2, \dots, s_n\}$ is the state space

(ii) A_{s_i} is the action space for state s_i

(iii) $p_{s_i s_j}(a)$ is the probability of the states transition $s_i \longrightarrow s_j$ with executing action a at state s_i

(iv) $r_{(s_i,a)}$ is a reward function getting from environment after executing action a at state s_i

(v) V is a criterion function or objective function of all the $r_{(s_i,a)}$ in the whole process

The whole process of an MDP is made up of a series of state-action pairs: $\{s_0, a_0, s_1, a_1, \dots, s_{n-1}, a_{n-1}, s_n\}$, which is determined by the probability of states transition. Also, the state transition probability matrix P_s that consists of $\left\{p_{s_i s_j}(a) : s_i, s_j \in S\right\}$ reflects the inherent properties of the system under the executed action. The policy π for the agent is a sequence $\{\pi_0, \pi_1, \dots,\}$ which is a strategy for action selection based on the action selection probability $\pi(s, a)$. In RL, $\pi(s, a)$ is computed from the criterion function, such as a kind of value function matrix. The criterion function implying knowledge and experience is established on historical learning process and would be used to select future actions.

The goal of RL is to learn an optimal policy that the agent would get the maximal accumulated rewards starting from the initial state to the target state in an episode. In the RL algorithm, the accumulated rewards $v_\pi(s)$ called "Value function" [27] can be defined formally by

$$\begin{aligned} v_\pi(s_t) &\doteq E\left\{r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots \mid s_t = s, \pi\right\}, \\ &= E\left\{r_{t+1} + \gamma V_\pi(s_{t+1}) \mid s_t = s, \pi\right\}, \\ &= \sum_{a \in A_s} \pi(s,a)\left[r_{(s,a)} + \gamma \sum p_{s_i s_j}(a) V_\pi(s_{t+1})\right], \end{aligned} \tag{1}$$

where $\gamma \in [0, 1]$ is the discount factor which reflects the influence degree of the future states to the current action. According to the Bellman equation, the value function will be described as

$$v_\pi^*(s) = \max_{a \in A_s}\left[r_{(s,a)} + \gamma \sum p_{s_i s_j}(a) V_\pi(s_{t+1})\right], \tag{2}$$

where $v_\pi^*(s)$ is the state value function for the optimal policy π^*:

$$\pi^* = \arg\max_\pi v_\pi(s). \tag{3}$$

During the training process, taking the method of temporal difference (TD) as an example, the value function will be updated as

$$v(s_t) \longleftarrow v(s_t) + \alpha(r_t + \gamma v(s_{t+1}) - v(s_t)), \tag{4}$$

where $\alpha \in (0, 1)$ is the learning rate.

In a general RL training process, when the system, including the environment and agent, is at a certain state s_t, the agent gets the state information through observation of environment, and then, chooses an action a_{s_t} based on $\pi(s, a)$. The system transits to a new state s_{t+1} based on P_{s_t} caused by the executed action a_{s_t}. The agent gets a numerical reward which shows the usefulness degree of a_{s_t} for the agent's target. The agent updates V using the reward and other parameters.

2.2. Exploration Strategy. The performance of an agent's policy can only be observed until the end of an episode, and the value function describes the long-term accumulated rewards of all the actions that have been executed. The agent should try every action to get the reward for every state in order to find the best action by comparing the accumulated rewards. More generally, the rewards for an action at a certain state fit into some probability distribution. To get the average reward, the agent should execute the action many times. It is a key problem how to select an action for a certain state based on the value function in the training process [28].

There is an exploration-exploitation dilemma in action selection which is based on the value function. In a training process, if an agent gives preference to the maximal value for the current state which is called exploitation, then the training process may converge quickly. However, the agent may get to the local optimal policy but not the global optimal policy. On the contrary, the episodes of training process will be increased on the preference of exploration. To balance exploration and exploitation [26], some action selection strategies have been applied to RL, such as ϵ-greedy, *softmax*, and so on. In the method of ϵ-greedy where $\epsilon \in (0, 1)$, the value function at a certain state will be classified to the maximum and others. The agent will select the maximum value function with the probability of $1 - \epsilon$ and select other actions with the probability of ϵ. In the method of *softmax*, the probabilities will be arranged from large to small based on the value function. However, all of these methods set the probability for action selection directly and simply based on value function without consideration of the system change.

Therefore, whether it can converge and the convergence rate of the strategy rely heavily on the action selection strategy and the parameters of RL such as discount factor γ, learning rates α, reward value, and so on. However, these parameters are set according to experts experience or some tricks which rarely base on any mathematical principle. In this paper, we try to reveal the characters of convergence process of RL by taking advantage of *Exploration Entropy* defined below which may provide a new angle to improve RL algorithm.

3. Exploration Entropy for RL

RL algorithm has been proved to be an effective method in MDPs [7]. However, little research has studied the training process [9]. In this section, we first give the definition of *Exploration Entropy* (EE) and formulate the reinforcement learning procedure with EE regarding Q-learning and probabilistic Q-learning. Then, we present the general performance analysis methods for RL , including convergence analysis and termination conditions. Finally, EE is applied to the measurement of multiple optimal solutions for a certain RL problem.

3.1. Exploration Entropy. The optimal policy for RL is represented by the probability distribution of actions for each state [29]. As a trial-and-error process, the rewards got by interacting with the environment are used to update the value function. The probability distribution of actions which represents the uncertainty of the action selection is calculated from the value function. It means that the learning process is essentially the process of reducing the uncertainty of action selection strategy on which actions should be chosen at each state. Hence, the performance of RL and the balance between exploration and exploitation can also be described with the degree of uncertainty for action selection [22]. Here, we introduce a new notion of EE to measure the degree of uncertainty. *Shannon entropy* (i.e., Shannon measure of uncertainty) has been well used in information theory [11], where the amount of uncertainty is measured by a probability distribution function p on a finite set [30]:

$$S(p) = -\sum_{x \in X} p(x) \log_2 p(x), \tag{5}$$

where X is the universal set, x is a element of the finite set X, and $p(x)$ is the probability distribution function on X.

Similar to *Shannon entropy*, the concept of *State Entropy* (SE) is defined based on the probability distribution of action selections to measure the uncertainty of action selection for a certain state s. The resulting function is [22]

$$\mathrm{SE}(s) = -\sum \pi(a_i \mid s) \log_2 \pi(a_i \mid s). \tag{6}$$

For an RL system, the global uncertainty of action selection can be described with *Exploration Entropy EE*(s):

$$\mathrm{EE}(\mathrm{SE}) = \frac{\sum_{j=1,2,\ldots,n} \mathrm{SE}(s_j)}{n \log_2 m}, \tag{7}$$

where $S = \{s_1, s_2, \ldots, s_n\}$ is the universal state set.

For an RL agent with m actions in each state, it can be proved [11] that the SE (uncertainty of action selection in a certain state) will be maximum and equal to $\log_2 m$ when all the probabilities are equal to $1/m$. Also, when every SE is maximal, the EE will also be maximal because the denominator in (7) is constant with the training process in an RL system. The maximum EE means that all the action selection probabilities of all states are equal, which is always the situation at the initialization without any prior knowledge about the environment. Along with the learning process, EE (SE) will tend to decrease and reach its minimum when the learning process converges and gives us the optimal policy.

For example, when there are only four alternative actions at state s, the probabilities for these actions will be $\{c_1, c_2, c_3, c_4\}$ $(c_1 + c_2 + c_3 + c_4 = 1)$. The maximum *State Entropy* $\mathrm{SE}(s)$ for state s is 2, which is obtained when all the action selection probabilities of the state are equal to 1/4. This result indicates that when all the probabilities are equal, the uncertainty of action selection will be at a maximum. This conclusion can be generalized to the states with m action choices. For the state s that has m action choices, the maximum $\mathrm{SE}(s)$ and EE (SE) will be obtained when all the probabilities of actions are equal to $1/m$. Formally,

$$\begin{aligned} \mathrm{SE}_{(s)}(c_1, c_2, \ldots, c_m) &\le \mathrm{SE}_{(s)}\left(\frac{1}{m}, \frac{1}{m}, \ldots, \frac{1}{m}\right), \\ &= \mathrm{SE}^*_{(s)}, \\ &= \log_2 m, \\ \mathrm{EE}\Big(\mathrm{SE}_{(s_1)}, \mathrm{SE}_{(s_2)}, \ldots, \mathrm{SE}_{(s_m)}\Big) &\le \mathrm{EE}\Big(E^*_{(s_1)}, E^*_{(s_2)}, \ldots, E^*_{(s_m)}\Big) = 1. \end{aligned} \tag{8}$$

In this paper, the *Exploration Entropy* will be computed in two fundamental RL algorithms: QL with softmax and PQL, as shown in Alg.1, 2 to analyse the RL training process. Softmax strategy is used to select actions in Algorithm 1. At a certain state, each action has a variable probability which is related to the q value and will be used to calculate $\mathrm{SE}(s_i)$ and EE. As the training process proceeds, the agent will nearly traverse every state. As a result, a stable Q-table and convergent EE will be generated. The other algorithm is shown in Algorithm 2, probabilistic Q-learning. The difference between the two algorithms is that their probability of action selection has different computing ways and updating methods. It is worth to point out that the exploration entropy-based method will not work effectively in the standard ϵ-greedy because the action selection probability is completely dependent on the epsilon parameter which is set depending on experts' experience or other tricks. The same phenomenon appears in QL with *Greedy* algorithm.

```
Initialize $Q(s,a)$ arbitrarily
Initialize the policy $\pi : P^{\pi} = (p^{\pi}(s,a))_{n\times m}$
**repeat**
  Initialize $s$, $t = 1$, $\tau$
  **repeat**
    a $\longleftarrow$ action $a_j$ with probability $p_{s_i a_j}$ for $s_i$
    Take action $a$, observe reward $r$, and next state $s'$
    $Q(s,a) \longleftarrow Q(s,a) + \alpha_t (r + \gamma \max_{a'} Q(s',a') - Q(s,a))$
    Update $P^{\pi}$ with Softmax strategy
    $\mathrm{SE}(s_i) = -\sum_j^m (p_{s_i a_j} \log_2 p_{s_i a_j})$
    $s \longleftarrow s', t \longleftarrow t + 1$
  **until** $s_i$ is destination
  $\mathrm{EE} = -\sum_i^n \sum_j^m (p_{s_i a_j} \log_2 p_{s_i a_j}) / n \log_2 m$
**until** the learning process ends
```

Algorithm 1: Q-learning.

```
Initialize $Q(s,a)$ arbitrarily
Initialize the policy $\pi : P^{\pi} = (p^{\pi}(s,a))_{n\times m}$
**repeat**
  Initialize $s$, $t = 1$
  **repeat**
    a $\longleftarrow$ action $a_j$ with probability $p_{s_i a_j}$ for $s_i$
    Take action $a$, observe reward $r$, and next state $s'$
    $Q(s,a) \longleftarrow Q(s,a) + \alpha_t (r + \gamma \max_{a'} Q(s',a') - Q(s,a))$
    $p(s,a_j) \longleftarrow p(s,a_j) + k(r + \max_{a'} Q(s',a'))$
    Normalize $p(s,a_j) \mid j = 1, 2, \ldots, m$
    $\mathrm{SE}(s_i) = -\sum_j^m (p_{s_i a_j} \log_2 p_{s_i a_j})$
    $s \longleftarrow s', t \longleftarrow t + 1$
  **until** $s_i$ is destination
  $\mathrm{EE} = -\sum_i^n \sum_j^m (p_{s_i a_j} \log_2 p_{s_i a_j}) / n \log_2 m$
**until** the learning process ends
```

Algorithm 2: Probabilistic Q-learning.

3.2. Performance Analysis Using Exploration Entropy. In this subsection, the *Exploration Entropy* is used to analyse the training process of RL which reflects some important characters of the system and the exploration strategy.

3.2.1. Convergence Analysis in Training Process. It has been known that the policy of the agent is made up of a series of (s, a) pairs. Also, a is selected according to s and the knowledge of the environment. At the beginning of the training process, as to any s the value function is 0 and the reward for any $a \in A$ is unknown. The uncertainty of the policy is max because the agent has to select an action haphazard. Along with the training process, more states are traversed, more actions are executed, and much more rewards are got. The value function $v(s)$ is updated by equation (4). So, the agent can make more efficient action which is definitely better than others. The uncertainty has been decreased with the decrease of the uncertainty of action selection, SE reflects the intelligibility of the state, and EE which includes all the SE reflects the convergence degree of the policy.

3.2.2. Termination Conditions in Training Process. In the field of RL algorithms [9], the convergence rate is the major problem. The key indicators used to reflect the advantages and disadvantages of the algorithm are the accumulation of rewards and steps number. However, in most algorithms with such exploration strategy as ϵ-greedy, *softmax* [10], the two key indicators are often influenced by the algorithm parameters of the action selection strategy, such as ϵ, t, and so on. At the same time, for some complex MDP, it is almost impossible to get the optimal policy in limited time. The aim of RL algorithm is to get the second best solution. In the above case, the curves based on the only two key indicators cannot always reflect physical truth of the convergence process for training.

On the other hand, as the update of $v(s)$ continues [7], the EE of the system is also being constantly updated. As the key indicator of system uncertainty, it is easy to use EE to estimate or distinguish the convergence time for a certain RL algorithm.

3.3. Measurement of Multiple Optimal Solutions Using Exploration Entropy. Normally, there may be multiple optimal solutions for a certain MDP. However, to the extent of the author's knowledge, there is no RL algorithm that has concerned this issue. The *Exploration Entropy* provides an angle to analyse this problem.

For an RL system, the value function updating process will converge to the Bellman equation, if the RL system is convergent. It means that the value function for a certain state will not change anymore. Also, the agent can get the optimal policy with *Greedy* strategy to select action based on the static value function. As to the agent, if there is only one path to get to the target state from the start state, it means that the agent will select only one action which has the maximal accumulated rewards value. According to the previous definition, the SE of every state and EE will be both 0 finally in the one-optimal-solution situation in ideal conditions. On the other hand, if there is more than one path to get to the target state, it means that the agent could get several (take 2 as an example) equivalent actions which have the same accumulated rewards value at some states (take state s_n as an example). Based on the *Greedy* principle, each one of the two actions with the same maximal reward value is chosen with the probability of 0.5. All other actions are selected with the probability of 0. The SE of s_n is 1, and obviously the EE is bigger than 0. This can be concluded as the EE value bigger than 0 indicates multiple optimal solutions.

Considering a simple grid MDP described as in Figure 1, each square in the grid represents a state in the problem. An agent at each state has four actions to select: east, west, south, and north. The agent will move a step by executing an action and get a reward of -1 at all states except state A. All actions at state A will take the agent to state B and get a reward of $+8$, as shown in Figure 1(a). After training the agent with *random strategy* in which the agent selects actions with the same probability at every state, the optimal policy and convergent value function which satisfies the *Bellman equation* will be obtained, as shown in Figures 1(b) and 1(c).

In this simple path planning problem, it is obvious that there are more than one path for the agent to get to state A from an arbitrary state, as shown in Figure 1(b). As to the value function, taking *state B* as example, the agent has two optimal selections, west and north, while each selection probability of the two actions is 0.5. The SE of *state B* is $\mathrm{SE}(s_B) = 1$. On the contrary, the SE of *state 1* is $\mathrm{SE}(s1) = 0$ because the agent has the only action *east* to execute. For all the states, EE = 0.17. Based on the previous analysis, when the value function gets to convergence, the *nonzero Exploration Entropy* reflects that there are multiple optimal paths for the MDP.

4. Experiment Using Exploration Entropy

To test the proposed EE and related analysis methods, we carry out several groups of experiments using two typical RL control examples, i.e., a classical control problem of indoor robot navigation and a quantum control problem of two-level quantum systems.

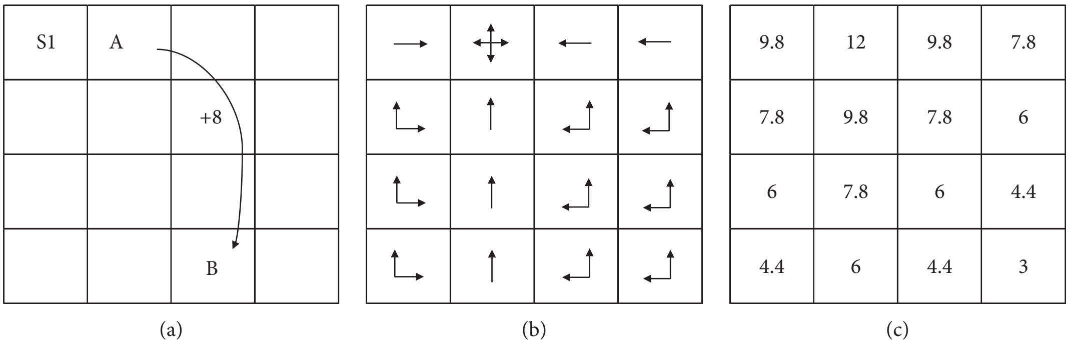
4.1. Indoor Robot Navigation. Problem description: As is shown in Figure 2, there is a 5×5 maze. Each square represents a state which belongs to the state set S. Also, the finite action set A contains 4 directions: *up, down, left, and right*. The agent should arrive to the *Goal point* from the *Start point* without going through the *Obstacle point*, as to G, S, and O in Figure 2. If the agent falls into the *Obstacle*, it will get a PUNISH which is -10. Also, if the agent reaches the destination, it will get a REWARD, 11 for PQL and 10 for *Softmax*. The experiment settings for all these algorithms are listed as follows: all Q-values are initialized as 0, the discount factor $\gamma = 0.9$, and the learning rate $\alpha = 0.1$. For Q-learning with *Softmax*, τ is initialized to 5.2 and then gradually reduced. For PQL, the update step of probability $k = 1$. In this section, the *Entropy*-based analytical method will be explained in the problem of path planning with two strategies shown in Algorithms 1 and 2.

After training with 1000 episodes, the experimental results are shown below. Figures 3 and 4 show the change of *Exploration Entropy* and *steps*. To demonstrate the effectiveness of our method, we have done the experiment 10 times and then computed the average steps and exploration entropy. As is shown in Figure 3, there are two stages in Softmax strategy. The first stage is from the first episode to the 200th episode, where the Q-table keeps updating and gradually converges, and the same to steps. In the second stage, after the 200th episode, the *Exploration Entropy* and steps are both basically stable. This indicates that the Q-table is basically stable and the convergence process is basically completed. Similar to PQL strategy, while the first stage is from the start to the 100th episode, and the second stage is after the 100th episode. The similar result for Maze B is shown in Figure 4.

On the other hand, the convergent process of every state is shown in Figures 5–8 intuitively for Maze A and Maze B, respectively. Taking Figures 5 and 6 as an example, the depth of grey color in any square is proportional to the *State Entropy* value. The black squares represent a big SE value which means that action selection of the corresponding states is with high uncertainty. The relatively white squares represent a relatively small entropy value, which means the action selection of these states is relatively clear. Obviously, as the iteration goes on, more and more squares become closer to white, which means their action selection is more and more clear. Finally, the figures become stable. It is worth saying why the *Exploration Entropy* is not 0 and the two algorithms have different EE at the end of the two experiments (this cannot influence the conclusion that EE can inflect the degree of convergence and act as a terminational condition of the training process).

(a) Obviously there are multiple optimal paths; therefore, the EE only can be close to 0 theoretically.

(b) Training episodes may not be enough. So, the entropy value may be still relatively big. Besides, the



9.8	12	9.8	7.8
7.8	9.8	7.8	6
6	7.8	6	4.4
4.4	6	4.4	3

Figure 1: A simple grid world example.

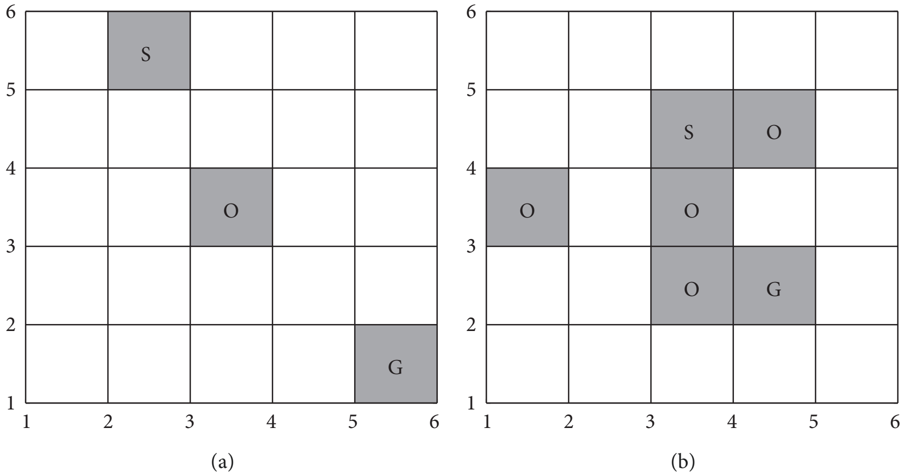


Figure 2: A navigation problem in a 5 × 5 maze. (a) Maze A. (b) Maze B.

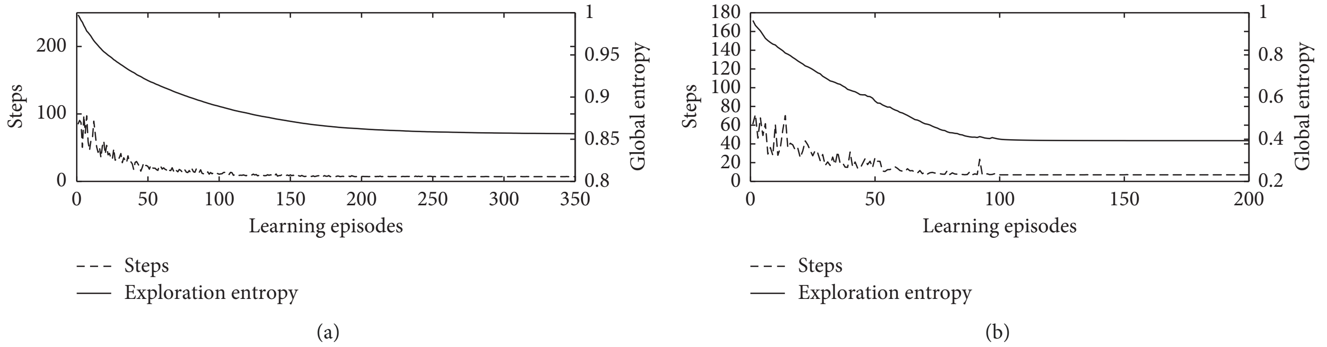


Figure 3: Steps and EE Learning performance for (1) Q-learning with Softmax strategy and (2) PQL, respectively, in Maze A. (a) Softmax (Maze A). (b) PQL (Maze A).

agent cannot traverse each state enough times in the actual situation, and the number of exploring times for a certain state is not the same between the two algorithms.

(c) The experimental final EE can be influenced to some extent by some parameter settings such as REWARD value, PUNISH value, and so on. Besides, it can also be changed by the computing way of the probability of action selection.

It is noticed that the *Exploration Entropy* and the steps have almost the same variation tendency. Therefore, EE can reflect the stage of training process. Besides, the entropy and the steps become convergent nearly at the same time. So, we can use EE as a termination condition.

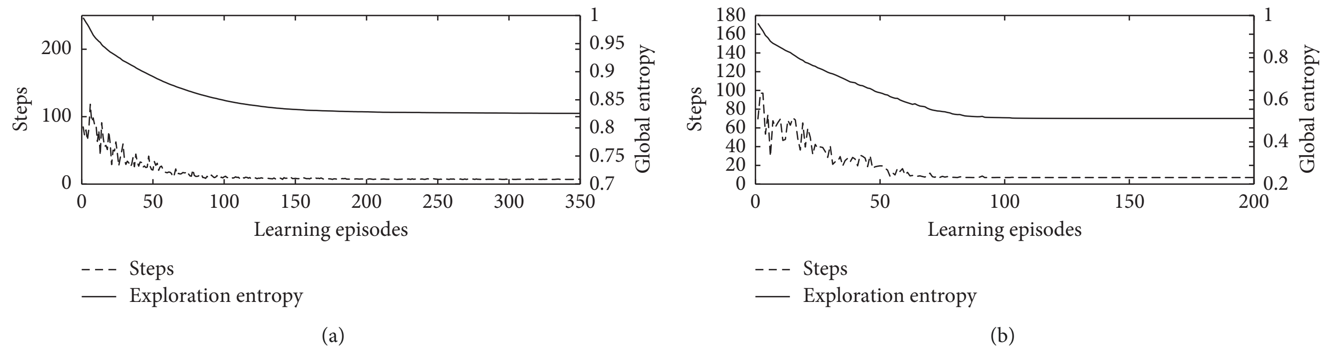


Figure 4: Steps and EE Learning performance for (1) Q-learning with Softmax strategy and (2) PQL, respectively, in Maze B (a) Softmax (Maze B). (b) PQL (Maze B).

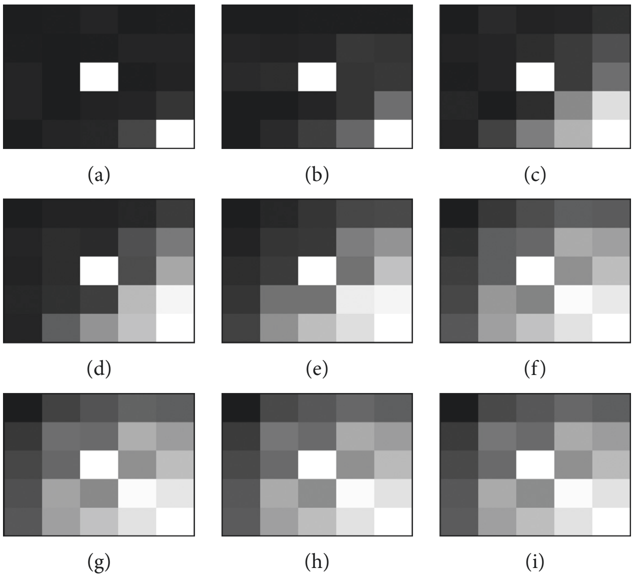


Figure 5: *State Entropy* of all states at the (a) 10th, (b) 20th, (c) 40th, (d) 60th, (e) 100th, (f) 200th, (g) 300th, (h) 800th, and (i) 1000th iteration with *Softmax strategy* in Maze A.

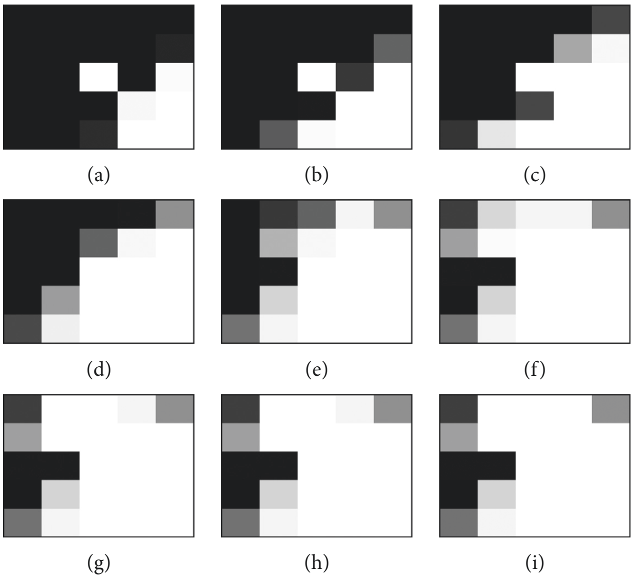


Figure 6: *State Entropy* of all states at the (a) 10th, (b) 20th, (c) 40th, (d) 60th, (e) 100th, (f) 200th, (g) 300th, (h) 800th, and (i) 1000th iteration with *Probabilistic strategy* in Maze A.

4.2. Quantum Control

4.2.1. Problem Description. Here, we consider the control problem of finite-dimensional (N-level) quantum systems [31]. We can denote the eigen states of the free Hamiltonian H_0 of an N-level quantum system as $D = \{|\phi_i\rangle\}_{i=1}^{N}$. The evolving state $|\psi_{(t)}\rangle$ of a controlled quantum system can be expressed as the eigen states in the set D:

$$|\psi_{(t)}\rangle = \sum_{i=1}^{N} c_i(t)|\phi_i\rangle, \tag{9}$$

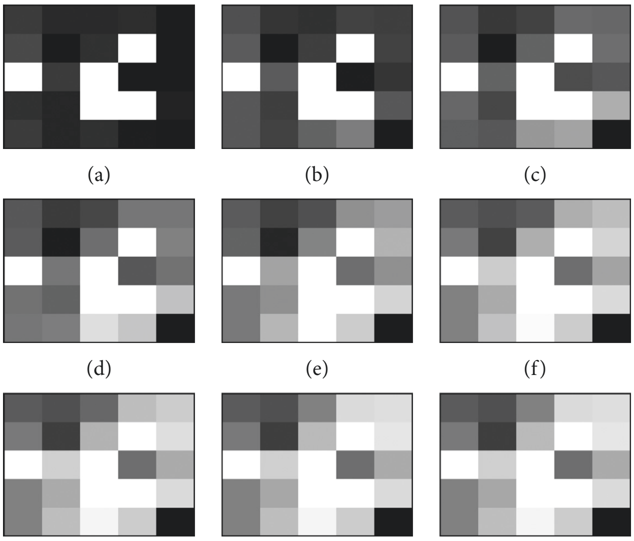
where complex numbers $c_i(t)$ satisfy $\sum_{i=1}^{N}|c_i(t)|^2 = 1$. Introducing a control $\varepsilon(t) \in L^2(\mathbf{R})$ acting on the system via a time-independent interaction Hamiltonian H_I and denoting $|\psi_{(t=0)}\rangle$ as $|\psi_0\rangle$, $C(t) = (c_i(t))_{i=1}^{N}$ evolves according to the Schrödinger equation [32]:

$$\begin{gathered} i\hbar\dot{C}(t) = [A + (t)B]C(t), \\ C(t=0) = C_0, \end{gathered} \tag{10}$$

where $i = \sqrt{-1}$, $C_0 = (c_{0i})_{i=1}^{N}$, $c_{0i} = \langle\varphi_i \mid \psi_0\rangle$, $\sum_{i=1}^{N}|c_{0i}|^2 = 1$, $\hbar$ is the reduced Planck constant, and the matrix A and B correspond to H_0 and H_1, respectively.

The propagator $U_{(t_1\longrightarrow t_2)}$ is a unitary operator such that for any state $|\psi_{(t_1)}\rangle$, the state $|\psi_{(t_2)}\rangle = U_{(t_1\longrightarrow t_2)}|\psi_{(t_1)}\rangle$ is the solution at time $t = t_2$ of (1) and (2) with the initial condition $|\psi_{(t_1)}\rangle$ at time $t = t_1$. For the sake of convenient representation, $U_{(t_1\longrightarrow t_2)}$ is simplified as $U_{(t)}$, $t \in [t_1, t_2]$. Furthermore, the control set $\{\varepsilon_j, j = 1, \ldots, m\}$ is given, where each control ε_j corresponds to a unitary operator U_j. Then, we define the performance function [33]:

$$J(\varepsilon) = \mathrm{tr}\Big(U_{(\varepsilon,T)}|\psi_0\rangle\langle\psi_0|U^{+}_{(\varepsilon,T)}|\psi_f\rangle\langle\psi_f|\Big), \tag{11}$$

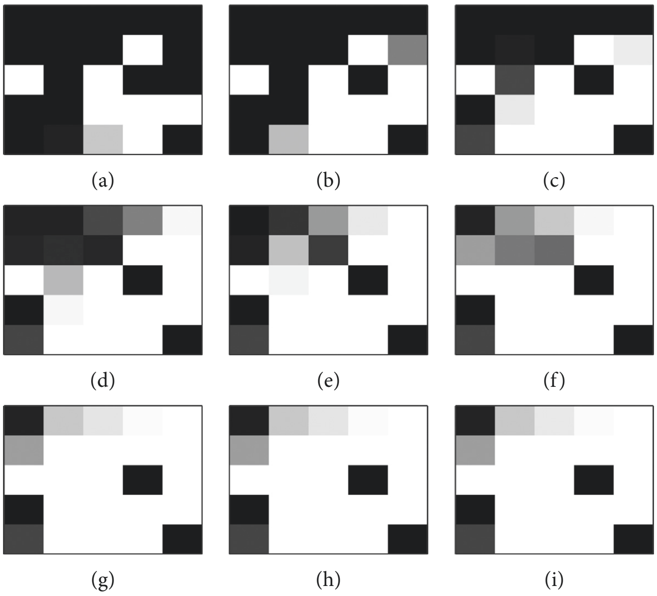


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 7: *State Entropy* of all states at the (a) 10th, (b) 20th, (c) 40th, (d) 60th, (e) 100th, (f) 200th, (g) 300th, (h) 800th, and (i) 1000th iteration with *Softmax strategy* in Maze B.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 8: *State Entropy* of all states at the (a) 10th, (b) 20th, (c) 40th, (d) 60th, (e) 100th, (f) 200th, (g) 300th, (h) 800th, and (i) 1000th iteration with *Probabilistic strategy* in Maze B.

where $\mathrm{tr}(\cdot)$ is the trace operator, $|\psi_0\rangle$ is the initial state, $|\psi_f\rangle$ is the target state, and U^+ is the adjoint of U. Thus, the task of the learning control system can be transformed into finding a global optimal control policy.

$$\varepsilon^* = \arg\max_{\varepsilon} J(\varepsilon). \tag{12}$$

The learning control problem of the two-level quantum system is so simple and typical that it is important to solve it [34–37]. Here, we focus on the spin-1/2 system, which is a typical two-level quantum system with important theoretical research and practical application. The state $|\psi\rangle$ of the spin-1/2 quantum system can be written as

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle, \tag{13}$$

where $\theta \in [0,\pi]$ and $\varphi \in [0,2\pi]$. For the quantum control problem, its permitted controls are U_1, U_2, and U_3, called no control input, a positive pulse control, and a negative pulse control, respectively [28]. Figure 9 shows the effect of one-step control on the evolution of the quantum system. More specifically, the propagators $\{U_i, i = 1,2,3\}$ are demonstrated as follows:

$$\begin{aligned} U_1 &= e^{-iI_z(\pi/15)}, \\ U_2 &= e^{-i\left(I_z+0.5I_x\right)(\pi/15)}, \\ U_3 &= e^{-i\left(I_z-0.5I_x\right)(\pi/15)}, \end{aligned} \tag{14}$$

where

$$\begin{aligned} I_z &= \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \\ I_x &= \frac{1}{2}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \end{aligned} \tag{15}$$

4.2.2. RL Control of Spin-1/2 Systems. The learning control objective is to drive the spin-1/2 system from the initial state $|\psi_{\text{initial}}\rangle(\theta = (\pi/60), \varphi = (\pi/30))$ to the target state $|\psi_{\text{target}}\rangle(\theta = (41\pi/60), \varphi = (29\pi/30))$ and minimize the control steps by three-switch control with all the propagators $\{U_i, i = 1,2,3\}$ and Bang-Bang control with the propagators $\{U_i, i = 2,3\}$, as shown in Figure 9. To solve the quantum control problem using RL, here, we make the following hypothesis: we can discretize the state of the spin-1/2 system into a finite-state set $S = \{s_i = |\psi_i\rangle\}$ $i = 1,2,\ldots,n$ and have the finite action (propagator) set $A = \{a_j = u_j\}$, $j = 1,\ldots,m$. More specifically, the system state set S is discrete by 30 warps and 30 wefts, the initial state $s_{\text{initial}} = |\psi_{\text{initial}}\rangle$, and the target state $s_{\text{target}} = |\psi_{\text{target}}\rangle$. For the three-switch control method, $A = U_j$, $j = 1,2,3$, and for Bang-Bang control, $A = U_j$, $j = 2,3$.

We use Q-learning [7, 38] and PQL [28] by two control methods (three-switch control and Bang-Bang control). The goal of all RL algorithms is to find an optimal policy π^*, which corresponds to the optimal control ε^* in (12). The experiment settings for all these algorithms are listed as follows: all Q-values are initialized as 0, the discount factor $\gamma = 0.99$, and the learning rate $\alpha = 0.01$. If the agent gets target state S_{target}, it will receive a reward $r_{\text{target}} = 1000$, and otherwise, it will get a reward $r_{\text{step}} = 0$. For Q-learning with ϵ-greedy strategy, ϵ is initialized to 0.5 and then gradually reduced to 0. For PQL, the update step of probability $k = 0.005$.

4.2.3. Experimental Results of Three-Switch Control. Figure 10 illustrates the control effect using RL algorithms (Q-learning and PQL), and the two-level quantum system

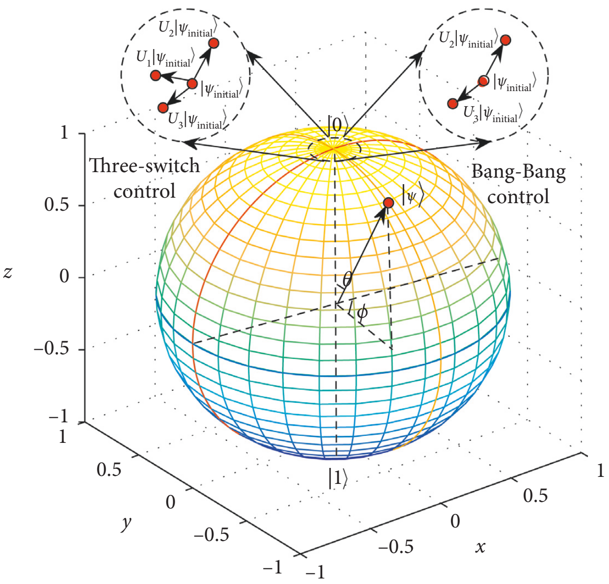
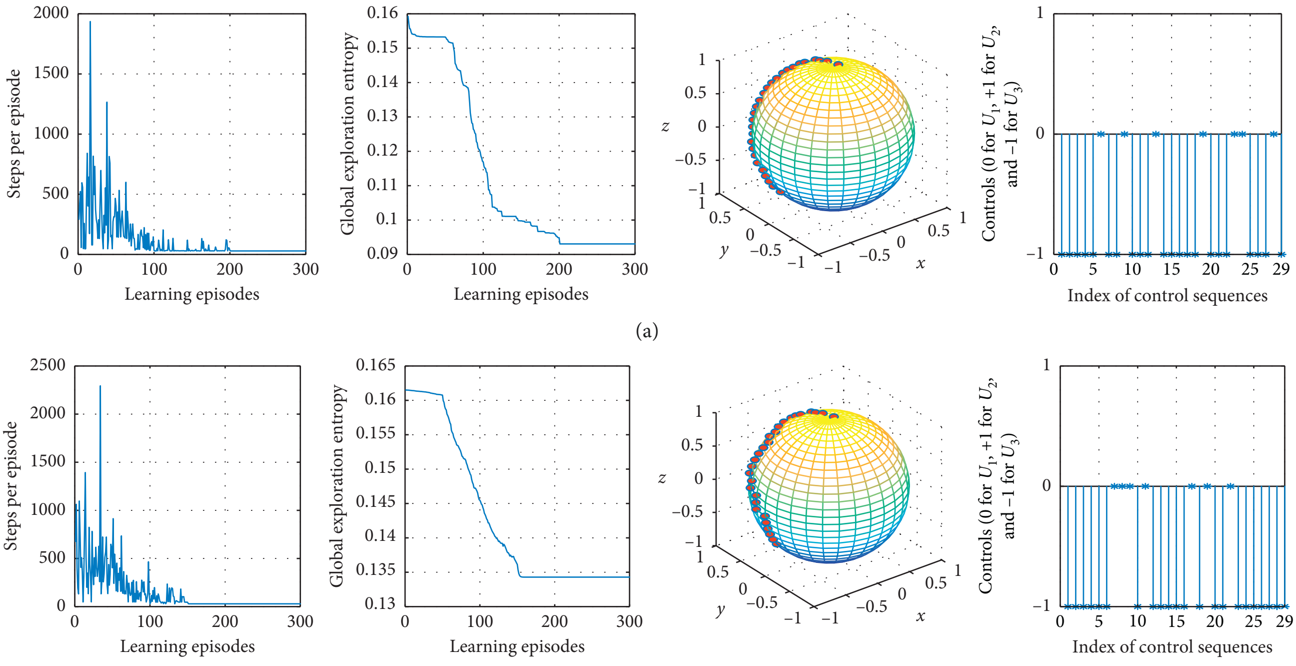

$U_2|\psi_{\text{initial}}\rangle$
$U_1|\psi_{\text{initial}}\rangle$
$|\psi_{\text{initial}}\rangle$
$U_3|\psi_{\text{initial}}\rangle$
$U_2|\psi_{\text{initial}}\rangle$
$|\psi_{\text{initial}}\rangle$
$U_3|\psi_{\text{initial}}\rangle$
$|0\rangle$
Three-switch control
Bang-Bang control
$|\psi\rangle$
θ
ϕ
$|1\rangle$
z
1
0.5
0
−0.5
−1
y
1
0.5
0
−0.5
−1
x
−1
−0.5
0
0.5
1

Figure 9: Demonstration of the spin-1/2 system with a Bloch sphere in a 3D Cartesian coordinates and one-step control effect of an initial quantum state $|\psi_{\text{initial}}\rangle$ by two control methods (three-switch control and Bang-Bang control).



(a)

(b)

Figure 10: The learning control performance of RL algorithms is used by the three-switch control method. For each algorithm, the step convergence effect, the *Exploration Entropy* effect, the quantum system state transition path, and the control sequence learned (0 for no pulse, −1 for negative pulse, and +1 for positive pulse) are shown separately. (a) Q-learning. (b) PQL.

can be controlled from the initial state to the target state with a certain number of control sequences (a learned control strategy), which is proof of the effectiveness of RL algorithms in solving such quantum control problem.

For Q-learning, the learning process converges after about 200 episodes, while PQL requires about 150 episodes. In addition, through the picture of *Exploration Entropy*, we can see that with the learning episodes increasing, the EE shows a downward trend. Moreover, the EE curves of Q-learning and PQL converge to approximately 200 learning episodes and 150 learning episodes, respectively, which is almost the same as the number of learning episodes when the step convergence curves of the two learning algorithms converge. As for the reason that EE cannot eventually converge to 0 (theoretical value), it may be because the agent has

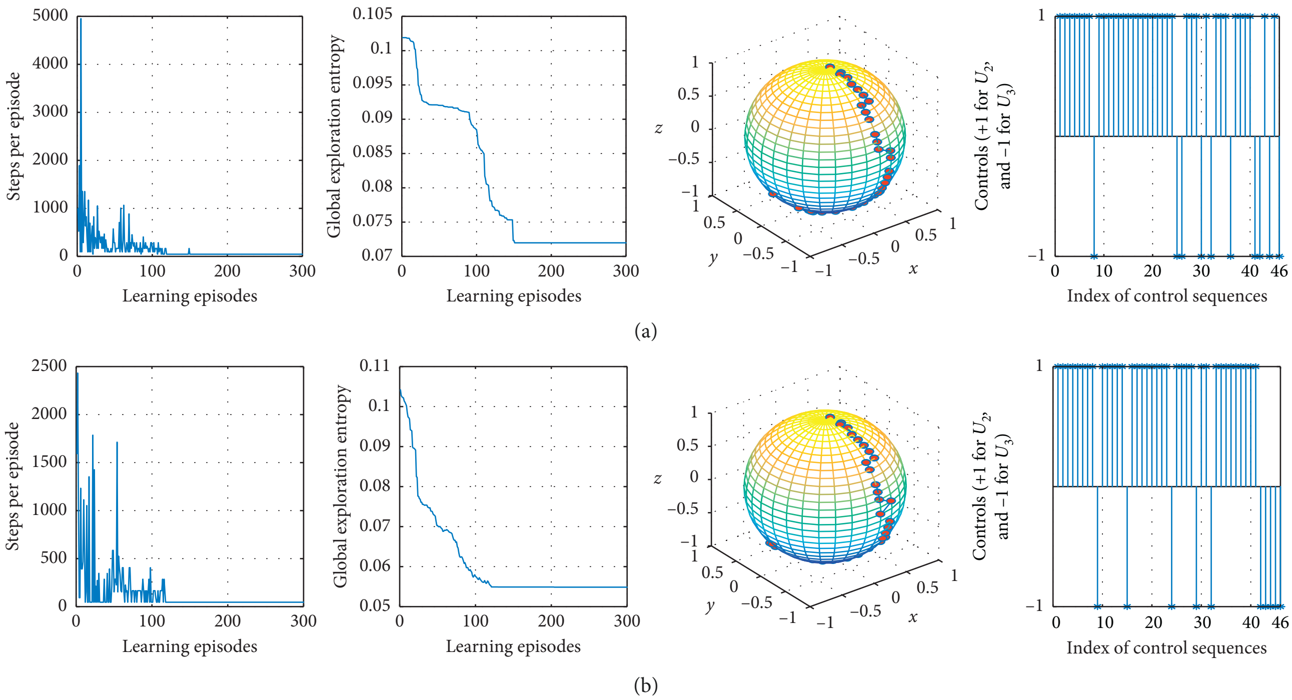


Figure 11: The learning control performance of RL algorithms is used by the Bang-Bang control method. For each algorithm, the step convergence effect, the *Exploration Entropy* effect, the quantum system state transition path, and the control sequence learned (−1 for negative pulse and +1 for positive pulse) are shown separately. (a) Q-learning. (b) PQL.

learned the control strategy without exploring all the states. We can not ignore the phenomenon that the EE graph can not only indicate the end of the agent's training (RL algorithm's convergence) from another angle, but also visually show the convergence degree of the agent, which is mainly reflected in the change process of the agent's exploration strategy.

4.2.4. Experimental Results of Bang-Bang Control. The learning control performance for RL algorithms is shown in Figure 11. We can see that RL algorithms are effective for solving this kind of quantum control problem because the two-level quantum system can be transferred from the initial state to the target state with a learned control strategy.

The learning process converges after about 150 episodes using Q-learning, and PQL needs about 120 episodes to find an optimal control sequence. Similarly, both the step convergence curve and the *Exploration Entropy* curve can represent the learning process information of the agent by the Bang-Bang control method, in which the EE curve is in a downward trend as a whole. Also, the reason why the EE curve does not converge to 0 at the end is also the same as that of the three-switch method (the agent learns the optimal control strategy without exploring all the states). It is not difficult to find that the EE graph can not only indicate the degree of convergence in the agent's training process, but also reflect the termination condition of the training.

5. Conclusion

In this paper, a new approach to analyse the training process of reinforcement learning is proposed, which includes *State Entropy* and *Exploration Entropy*. The SE reflects the training level a single state of the agent in a whole training process. The EE reflects more important information of the agent's training which includes the convergence degree and termination condition of training. The experimental results of the two typical problems, i.e., indoor navigation and quantum control in simulation environment illustrate the advantages of the proposed approach, when used to analyse the improved RL algorithms. In addition, it could be used to accelerate the RL training process because the method based on entropy can judge the end of the training process more exactly. Our future work will focus on the extension of EE to continuous cases of RL with deep neural networks and multiagent RL systems [39].

Data Availability

The data used to support the findings of this study are available form the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (no. 2016YFD0702100), the National Natural Science Foundation of China (no. 71732003), and the Fundamental Research Funds for the Central Universities (no. 011814380035).

References

[1] L. H. Xu, X. H. Xia, and Q. Luo, "The study of reinforcement learning for traffic self-adaptive control under multiagent markov game environment," *Mathematical Problems in Engineering*, vol. 2013, Article ID 962869, 10 pages, 2013.

[2] L. Tian, Y. Li, and B. Li, "Reinforcement learning based novel adaptive learning framework for smart grid prediction," *Mathematical Problems in Engineering*, vol. 2017, Article ID 8192368, 8 pages, 2017.

[3] M. Yang, Y. Yang, W. Wang, H. Ding, and J. Chen, "Multiagent-based simulation of temporal-spatial characteristics of activity-travel patterns using interactive reinforcement learning," *Mathematical Problems in Engineering*, vol. 2014, Article ID 951367, 11 pages, 2013.

[4] H. V. Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," 2015, https://arxiv.org/abs/1509.06461.

[5] Z. Wang, C. Chen, H.-X. Li, D. Dong, and T.-J. Tarn, "Incremental reinforcement learning with prioritized sweeping for dynamic environments," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 2, pp. 621–632, 2019.

[6] K. M. Jagodnik, P. S. Thomas, A. J. van den Bogert, M. S. Branicky, and R. F. Kirsch, "Training an actor-critic reinforcement learning controller for arm movement using human-generated rewards," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 10, pp. 1892–1905, 2017.

[7] R. S. Andrew and G. Barto, *Reinforcement Learning: An Introduction*, MIT Press Cambridge, Cambridge, UK, 2nd edition, 2018.

[8] M. V. C. Guelpeli, B. S. de Oliveira, M. A. Pinto, and R. C. dos Santos, "The apprentice modeling through reinforcement with a temporal analysis using the q-learning algorithm," in *Proceedings of the 2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE)*, vol. 1, pp. 296–300, Zhangjiajie, China, May 2012.

[9] B. Li, L. Xia, and Q. Zhao, "Complexity analysis of reinforcement learning and its application to robotics," in *Proceedings of the 2017 13th IEEE Conference on Automation Science and Engineering (CASE)*, pp. 1425-1426, Xi'an, China, August 2017.

[10] C. Chen and D. Dong, "Complexity analysis of quantum reinforcement learning," in *Proceedings of the 29th Chinese Control Conference*, pp. 5897–5901, Beijing, China, July 2010.

[11] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, July 1948.

[12] W. Fan, N. Bouguila, S. Bourouis, and Y. Laalaoui, "Entropy-based variational bayes learning framework for data clustering," *IET Image Processing*, vol. 12, no. 10, pp. 1762–1772, 2018.

[13] M. Bloem and N. Bambos, "Infinite time horizon maximum causal entropy inverse reinforcement learning," in *Proceedings of the 53rd IEEE Conference on Decision and Control*, pp. 4911–4916, Los Angeles, CA, USA, December 2014.

[14] C. Molina, N. B. Yoma, F. Huenupán, C. Garretón, and J. Wuth, "Maximum entropy-based reinforcement learning using a confidence measure in speech recognition for telephone speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 5, pp. 1041–1052, 2010.

[15] M. Ramicic and A. Bonarini, "Entropy-based prioritized sampling in deep Q-learning," in *Proceedings of the 2017 2nd International Conference on Image, Vision and Computing (ICIVC)*, pp. 1068–1072, Chengdu, China, June 2017.

[16] K. Lee, S. Choi, and S. Oh, "Sparse markov decision processes with causal sparse tsallis entropy regularization for reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1466–1473, 2018.

[17] G. Ciuperca, V. Girardin, and L. Lhote, "Computation and estimation of generalized entropy rates for denumerable Markov chains," *IEEE Transactions on Information Theory*, vol. 57, no. 7, pp. 4026–4034, 2011.

[18] G. Ciuperca and V. Girardin, "Estimation of the entropy rate of a countable Markov chain," *Communications in Statistics—Theory and Methods*, vol. 36, no. 14, pp. 2543–2557, 2007.

[19] V. Girardin and N. Limnios, "Entropy rate and maximum entropy methods for countable semi-Markov chains," *Communications in Statistics—Theory and Methods*, vol. 33, no. 3, pp. 609–622, 2004.

[20] X. Zhuang and Z. Chen, "Strategy entropy as a measure of strategy convergence in reinforcement learning," in *Proceedings of the 2008 First International Conference on Intelligent Networks and Intelligent Systems*, pp. 81–84, Wuhan, China, November 2008.

[21] X. Zhuang, "The strategy entropy of reinforcement learning for mobile robot navigation in complex environments," in *Proceedings of the 2005 IEEE International Conference on Robotics & Automation*, Barcelona, Spain, April 2005.

[22] B. Masoumi, M. Asghari, and M. R. Meybodi, "Utilizing learning automata and entropy to improve the exploration power of rescue agents," in *Proceedings of the 2010 Second WRI Global Congress on Intelligent Systems*, Wuhan, China, December 2010.

[23] B. Xin, K. Tang, L. Wang, and C. Chen, "Knowledge transfer between multi-granularity models for reinforcement learning," in *Proceedings of the 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 2881–2886, Miyazaki, Japan, October 2018.

[24] Z. Ren, D. Dong, H. Li, and C. Chen, "Self-paced prioritized curriculum learning with coverage penalty in deep reinforcement learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 6, pp. 2216–2226, 2018.

[25] D. Dong, C. Chen, H. Li, and T.-J. Tarn, "Quantum reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 38, no. 5, pp. 1207–1220, 2008.

[26] X.-L. Chen, L. Cao, C.-X. Li, Z.-X. Xu, and J. Lai, "Ensemble network architecture for deep reinforcement learning," *Mathematical Problems in Engineering*, vol. 2018, Article ID 2129393, 6 pages, 2018.

[27] D. Daoyi, C. Chunlin, and H. Li, "Reinforcement strategy using quantum amplitude amplification for robot learning," in *Proceedings of the 2007 Chinese Control Conference*, pp. 571–575, Hunan, China, July 2007.

[28] C. Chen, D. Dong, H. Li, J. Chu, and T. Tarn, "Fidelity-based probabilistic q-learning for control of quantum systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 5, pp. 920–933, 2014.

[29] D. Xu, Z. Zhu, and C. Chen, "An event-based probabilistic q-learning method for navigation control of mobile robots," in *Proceeding of the 11th World Congress on Intelligent Control and Automation*, pp. 587–592, Shenyang, China, June 2014.

[30] D. Zhang, "Entropy—a measure of uncertainty of random variable," *Systems Engineering & Electronics*, vol. 11, pp. 3–7, 1997.

[31] C. Chen, D. Dong, J. Lam, J. Chu, and T.-J. Tarn, "Control design of uncertain quantum systems with fuzzy estimators," *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 5, pp. 820–831, 2012.

[32] D. Dong and I. R. Petersen, "Quantum control theory and applications: a survey," *IET Control Theory & Applications*, vol. 4, no. 12, pp. 2651–2671, 2010.

[33] R. Chakrabarti and H. Rabitz, "Quantum control landscapes," *International Reviews in Physical Chemistry*, vol. 26, no. 4, pp. 671–735, 2007.

[34] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, Cambridge, UK, 2010.

[35] R. Jackiw, *Intermediate Quantum Mechanics*, CRC Press, Boca Raton, FL, USA, 2018.

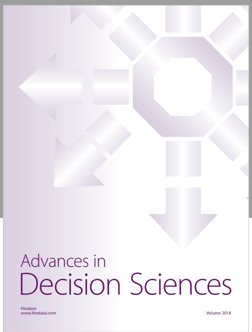
[36] D. D'alessandro and M. Dahleh, "Optimal control of two-level quantum systems," *IEEE Transactions on Automatic Control*, vol. 46, no. 6, pp. 866–876, 2001.

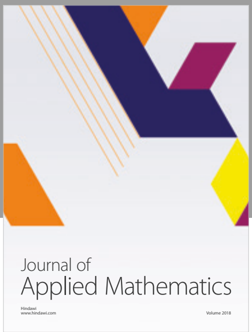
[37] A. M. Brańczyk, P. E. Mendonça, A. Gilchrist, A. C. Doherty, and S. D. Bartlett, "Quantum control of a single qubit," *Physical Review A*, vol. 75, no. 1, Article ID 012329, 2007.

[38] C. J. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

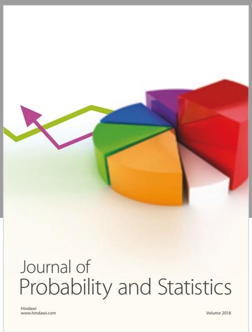
[39] L. Zhou, P. Yang, C. Chen, and Y. Gao, "Multiagent reinforcement learning with sparse interactions by negotiation and knowledge transfer," *IEEE Transactions on Cybernetics*, vol. 47, no. 5, pp. 1238–1250, 2017.

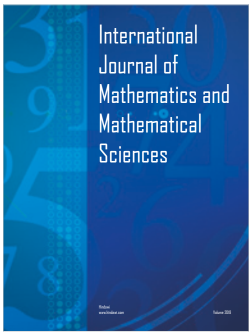
Advances in
Operations Research


Advances in
Decision Sciences

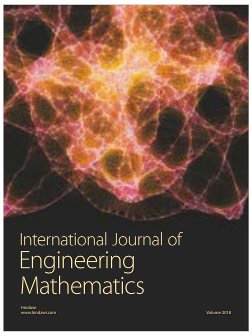

Journal of
Applied Mathematics

The Scientific
World Journal

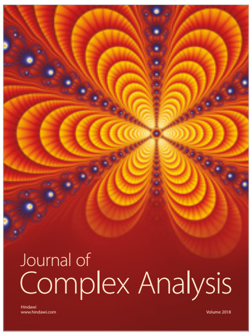

Journal of
Probability and Statistics

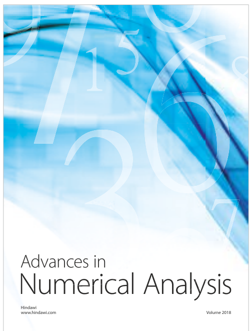

International
Journal of
Mathematics and
Mathematical
Sciences

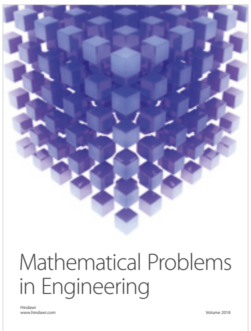
Journal of
Optimization

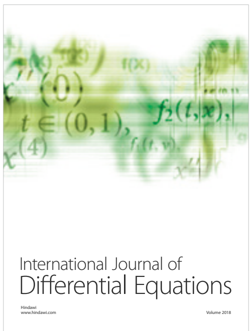

International Journal of
Engineering
Mathematics

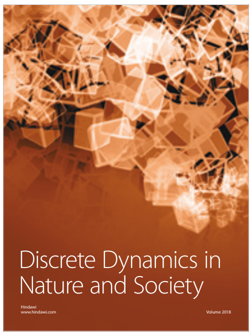
International Journal of
Analysis

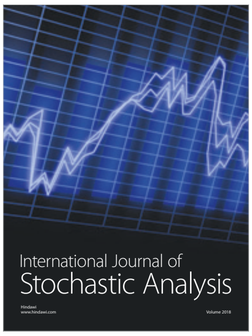

Journal of
Complex Analysis


Advances in
Numerical Analysis


Mathematical Problems
in Engineering

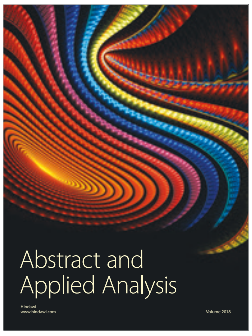

International Journal of
Differential Equations

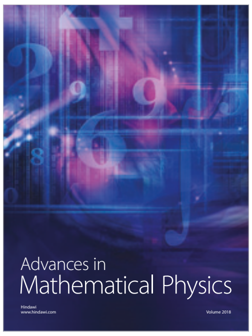

Discrete Dynamics in
Nature and Society


International Journal of
Stochastic Analysis

Journal of
Mathematics

Journal of
Function Spaces


Abstract and
Applied Analysis


Advances in
Mathematical Physics