Hidden Markov Random Field for Multi-Agent Optimal Decision in Top-Coal Caving

YI YANG1, (Member, IEEE), ZHIWEI LIN2, BINGFENG LI1, XINWEI LI1, LIZHI CUI1, AND KEPING WANG1
1School of Electrical Engineering and Automation, Henan Polytechnic University, Jiaozuo 454003, China
2School of Computing, Ulster University, Newtownabbey BT37 0QB, U.K.
Corresponding author: Bingfeng Li (libingfeng@hpu.edu.cn)
This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFC0604502, in part by the Henan Province Scientific and Technological Project of China under Grant 192102210100 and Grant 172102210270, and in part by the Key Scientific and Research Projects of Universities in Henan province of China under Grant 19A413008 and Grant 17A480007.

ABSTRACT Applying model-based learning for the optimal decision of the multi-agent system is not trivial due to the expensive price or even the impossibility of obtaining the ground truth for training the model of the complex environment. Such as learning the optimal action of hydraulic supports in the top-coal caving, the optimal action could not accessible as the ground truth of the corresponding state in the intricate processes. Regarding the latent ground truth as the hidden variable is an effective method in the hidden Markov model. This paper extends the hidden variable of ground truth to the multi-agent system and proposes the hidden Markov random field (HMRF) model with reinforcement learning for optimizing the action decision of the multi-agent. In the HMRF model, the input states and the output actions of the multi-agent are considered as an observable random field and a latent Markov random field, respectively. Based on the HMRF model, the optimal decision is inferred by the maximum posterior probability with the prior probability obtained by Q-learning. Meanwhile, the parameters of the HMRF model are estimated by the expectation maximum algorithm. In the experiment, the top-coal caving demonstrates the effectiveness of the proposed method that the recall of top-coal is improved prominently with a very small price of increasing the rock-rate. Furthermore, the proposed method is employed to deal with the predator-preys problem in the gym. The experiment result shows that the communication between agents by the HMRF increases the reward of the preys.

INDEX TERMS Hidden Markov random field, optimal decision, multi-agent, top-coal caving.

I. INTRODUCTION

Top-coal caving is the most efficient mining method in the underground thick coal seam at present [1]. As shown in Fig. 1, the hydraulic supports (HSs) is the key equipment of the fully mechanized caving workspace (FMCW) for security and top-coal exploiting. Each hydraulic support is equipped with a window (e.g, the component C in Fig. 1b) for top-coal exploiting. In the FMCW, there are hundreds of HSs to jack up the coal-roof, and the coal-cutter cuts coal in front of HSs. As the HSs move forward, the top-coal falls under gravity and be captured by the drag conveyor if the windows at the back of the HSs are open.

The associate editor coordinating the review of this manuscript and approving it for publication was Chemping Hou.

Controlling the top-caving windows of HSs are the key to maximize the output of top-coal exploit as in general practice. Ideally, the windows should keep open as long as possible so that more coal can fall into the drag conveyor. However, the rocks from the immediate-roof will inevitably fall during the top-coal exploiting. If the windows could not close in time, the rocks in the immediate-roof or even in the main-proof will fall into the drag conveyor too, which increases the cost of the mining due to the need to filter out the rocks subsequently. Therefore, the optimal decision of windows’ action is one of the most important issues in top-coal caving to maximize the top-coal capturing meanwhile minimize the number of rocks falling into drag conveyor.

At present, there are two methods to control the windows open or close, by hand or by the preset timer based on the experience. However, the hundreds of HSs need many
exponentially with the increasing number of the agents [5].

That makes it difficult to optimize the action of windows since the hundreds of HSs in the FMCW will bring a large state space. In order to address these issues, this paper proposes a probabilistic graphical model (PGM) [6] for the multi-agent system to obtain the optimal decision with the following motivation:

Because in the PGM of a multi-agent, the agents and their relationships can be depicted uniformly by the nodes and edges, hence the model architecture is explicit to describe the relationship between the current agent and its neighbors, and the optimal decision could be inferred by the PGM directly. However, training the PGM needs the samples with ground-truth, and unfortunately, it is impossible for the top-coal caving. Inspired by the hidden Markov model (HMM) training the action of each agent as a hidden variable, this paper extends hidden variables of the multi-agent to the random field and proposes the optimal decision methodology incorporated with the hidden Markov random field (HMRF) [7] for the multi-agent.

The proposed methodology mainly contains two parts: the HMRF architecture and the Q-learning algorithm. In the first part, the optimal decision model is established by HMRF, in which the input and output are the state random field and action Markov random field of the multi-agent, respectively. Meanwhile, the correlation of the agents is depicted by the nodes and edges of the HMRF. As a supervised learning method, it is impossible for the HMRF model to carry out training without a large number of samples. However, the complex dynamic environment of FMCW makes the training sample acquisition hard. Hence the action of the agent is regarded as the hidden variable, which means the agent cannot know the optimal action during the inference process, then the expectation maximization (EM) algorithm [8] is employed to train the parameters of HMRF model. In the Q-learning part, the trained Q-value function produces the prior probability for HMRF and the HMRF model learns the parameters without labels, essentially, the proposed methodology is an unsupervised machine learning. The main contribution of this work includes:

(I) The actions of HSs window for top-coal caving are transformed into the multi-agent decision methodology to approach optimal performance.

(II) The HMRF for the multi-agent decision is proposed to establish the unified model of both current-agent and its neighbors, in which the action of each agent is considered as a hidden random variable and the neighbors’ actions are embedded into the HMRF to achieve the optimal decisions collaboratively. Hence, the optimal decision is transformed into a probabilistic inference based on the HMRF model.

(III) Based on the HMRF, the optimal decisions of multi-agent are formulated as a maximum a posterior (MAP) of the actions with the joint probabilistic distribution, and the Q-learning algorithm is employed to obtain the prior probability for the MAP. In consequence, training the HMRF model without labels can acclimatize the complex environment of top-coal caving.

experienced workers which increases the cost significantly and may result in health and safety issues. Moreover, in the FMCW, the structures of top-coal, immediate-roof and main-roof layers are very complicated, and it is hard to decide the boundaries between them [1], [3], [4]. Hence, presetting a timer to control each window individually is impossible to get the maximum top-coal mining, especially during the coal cutter moving forward, the dynamic processes of top-coal, immediate-roof and main-roof would deteriorate the intricate boundary and results in the rock funnel areas.

In order to maximize the output of top coal with lower rocks by automating the windows’ action, it is important to develop models that can control the windows collaboratively. This is due to the fact that the boundary straightness between coal and rocks could be improved to avoid the rock funnel areas and increase the top-coal exploiting by controlling the windows collaboratively. Essentially, it is an optimizing problem of the multi-agent system, in which each window can be regarded as an agent. Also, the window action decision could be considered as a random process since the action of each window is independent of the past environment states and has the characters of the Markov process. Hence, each decision maker of the window is an agent which can learn to predict what action to take from the experience.

In this multi-agent system of the FMCW, the dimension of state space and the computation complexity will expand exponentially with the increasing number of the agents [5].
The rest of the paper is organized below: Section II presents the related works on the optimal decision by PGM and reinforcement learning (RL). Section III introduces the hidden Markov random field for the multi-agent optimal decision. Section IV, present the Q-learning to obtain the MAP for HMRF. Section V demonstrates the automating top-caving and the predator-preys to prove the effectiveness of the algorithm. Section VI gives the conclusion.

II. RELATED WORKS

For the multi-agent system, the signal-agent could be abstracted to the node of the PGM [9], meanwhile, the communication between the agents could be tred as the edges of the PGM. Hence, the temporal-spatial characteristic of the agents is described by the node and edges.

RL [10] is an effective method to get the optimal decision [11]–[13]. The optimal policy is obtained by iterating the learning mechanism based on the reward from the environment or an established neural network to approximate the environment model by the critic-actor architecture [14], [15]. Hence in the multi-agent systems, the RL is widely applied [16], such as in traffic control [17], [18], mobile robots [19]–[21], resource management [22], [23].

RL achieves the optimal policy by revising the possible state transition from the end [24]. Hence, the multi-agent system encounters the dimension curse dramatically. Furthermore, the reward of each agent is dependent, and the performance index function of the system should be optimized overall the rewards [25], [26], sometime it should equilateral between the entirety and the individuals [27]. Moreover, the agent executes its policy locally while effecting the other agents, and the different policy combination produces different environment change, hence the environment is non-stationary during the agents executing its policy [25], [26]. That means it is difficult to get train data who can cover all the states’ combinations, especially there is a large number of agents.

Hence, combining the advantages of PGM and RL is naturally chosen for multi-agent optimal decision, because the communication topology of multi-agent is explicitly formulated by PGM, and the temporal-spatial characteristic of the multi-agent actions is depicted clearly by the output of the PGM. Hence the combination could alleviate the difficulty confronted in the RL system [9].

Bayesian network and Markov random field (MRF) are the two major kinds of PGM. A series of methods based on the Bayesian method has been proposed for RL with the signal agent system [28]–[30], and the multi-agent system [31]–[33]. Conditional random field (CRF) is one of the classical MRF, in which the input and output random variables form their own MRF. Meanwhile, the output CRF could be obtained by probabilistic inference with joint conditional probability. The CRF used for the multi-agent can be found in [34], [35], in which the incidence relation of each agent is formulated as the potential functions whose parameters are estimated by maximum likelihood estimation based on the data pair of input and output.

However, one of the most important challenges for optimal decisions via PGM is the parameter estimating of the approximation model since it is hard to obtain the ground truth for training data, especially the process and environment is complex and non-stationary. Regarding the ground truth of the training data as a hidden variable is an effective method to estimate the parameters of PGM [36]. Hence, the Hidden Markov model (HMM) is proposed to deal with the hidden parameters system, now it has been well applied in many control and decision making system, such as robot control [37]–[40], autonomous manufacturing [41]–[43], fault diagnosis [44], [45].

In the RL system, the HMM is often used to deal with the partially observable Markov decision problems [46], [47]. In [48], [49] the HMM has been employed to describe the hidden variable in the RL framework, and the parameters are estimated effectively. In [50], the Restricted Boltzmann machine with the hidden variables has been adopted to approximate the action and state of the RL, and the hidden variables can be estimated by state-action-reward-state-action (SARSA) algorithm. In [51], the HMM is used with RL to predict the zone in the 5G system. Similar work can be found in [52]. However, the methods mentioned above are just applied to the signal-agent in the RL framework.

Extending the HMM to the multi-agent optimal decision is the most direct method. Naturally, the single random variable for denoting the input or output of a single agent should be promoted to the random variables for the multi-agent by the available way, named Hidden Markov random field (HMRF) [7].

The HMRF has been proposed to deal with the image segmentation [7], [53], [54], in which the input image and the segmentation output are respectively considered as an observable random field and a hidden Markov random field. Currently, HMRF is extending to the cone penetration tests [55], light-field stereo matching problems [56] and electronic medical records analysis [57], etc. For image segmentation, the best result could be considered as the optimal decision of choosing the label for the pixels by unsupervised learning. If the segmentation is understood from another perspective that the pixels make the decision regulated by HMRF, the pixels could be regarded as the multi-agent and the label set could be transformed into the action space of the multi-agent. Naturally, the HMRF could be used for inferring the optimal decision of the multi-agent.

Since the PGM could depict the relationship between the elements of the systems, recently, the graph property is combined with other machine learning methods to address the issue of combinatorial generalization which is considered as the top priority for artificial intelligence [58]. The graph neural network [59], [60] is the most famous branch and rising rapidly which has been expanded into RL [61] for robot controlling [62], solving boolean satisfiability [63], and crowd navigation [64], etc. Although the graph neural
network is in the different categories of the method proposed in this paper, the direction of how to describe the relationship between the elements in the system is the same.

Motivated by the serial references mentioned above and aiming the optimal action of the windows of HSS for top-coal caving, this paper proposes the HMRF methodology based on probability inference and RL to optimize the multi-agent decision.

III. HIDDEN MARKOV RANDOM FIELD FOR MULTI-AGENT OPTIMAL DECISION

In a multi-agent system, the states of all agents can be denoted by a random field, meanwhile, the action decision process of all the agents meets Markov random process and all the actions can be considered as a Markov random field [65]. The states and actions are the input and output of the multi-agent, and the two random fields are the foundation of employing the HMRF [7] to get the optimal decision.

A. PRELIMINARIES

For a multi-agent system, let \( A = \{a_1, a_2, \ldots, a_L\} \) be the action space of agent, where \( L \in \mathbb{R} \) is the number of the actions for a single agent, and \( a_i \) is the \( i \)th action value. The state space is \( S = \{x_1, x_2, \ldots, x_D\} \) where \( D \in \mathbb{R} \) is the number of states for a signal agent, and \( x_i \) is the \( i \)th state value.

As a random field, the states of a multi-agent system is denoted by \( X = \{X_1, X_2, \ldots, X_N\}, N \in \mathbb{R} \) is the number of the agents, \( X_i \in S \) is a discrete random variable. The configuration of \( X \) is \( x = \{x_1, x_2, \ldots, x_N\} \), and the set of all configurations is defined as \( \chi = \{x = \{x_1, x_2, \ldots, x_N\}|x_i \in S, i = 1, \ldots, N\} \).

The Markov random field of the actions is denoted by \( Y = \{Y_1, Y_2, \ldots, Y_N\}, N \in \mathbb{R} \), \( Y_i \in A \). The configuration is \( y = \{y_1, y_2, \ldots, y_N\} \) and the set \( \gamma = \{y = \{y_1, y_2, \ldots, y_N\}|y_i \in A, i = 1, \ldots, N\} \) denotes all the configurations of \( Y \).

For the multi-agent action decision, given an action Markov random field \( Y \) that every random variable of states \( X_i \) follows a known conditional probability distribution \( p(X_i = x_i|Y_i = y_i) = p(x_i|y_i) \) which can be formulated as the same function \( f(x_i; \theta_{y_i}) \), i.e.,

\[
p(x_i|y_i) = f(x_i; \theta_{y_i})
\]

where \( \theta_{y_i} \) is the parameters associated with the action \( y_i \). Note that, \( X_i \) and \( Y_i \) are discrete random variable, hence \( \theta_{y_i} \) should meet the following condition

\[
f(x_i; \theta_{y_i}) \geq 0
\]

\[
\sum_{x_i} f(x_i; \theta_{y_i}) = 1 \tag{2}
\]

Suppose each state and action in \( x \) and \( y \) is conditionally independent, that means for any random configuration \( y \in \gamma \), the conditional probability distribution of \( x \) corresponded to \( y \) is independent for every \( x_i \). Therefore, the following formulation is yielded

\[
p(x|y) = \prod_i p(x_i|y_i)
\]

(3)

B. HIDDEN MARKOV RANDOM FIELD FOR MULTI-AGENT SYSTEM

In a Markov random field, the neighborhood set of agent \( i \) is denoted by \( N_i \), that \( N_i \subset A \) and \( i \not\in N_i \). According to the Markov local characteristic [7], [65], the joint conditional probability distribution of state and action of each agent could be yielded as below

\[
p(x_i, y_i|N_i) = p(x_i|y_i, N_i) p(y_i|N_i)
\]

\[
= p(x_i|y_i) p(y_i|N_i)
\]

(4)

The marginal probability distribution of \( x_i \) under the condition of the neighborhood of \( N_i \) could be described as follows

\[
p(x_i|N_i) = \sum_{y_i \in A} p(x_i, y_i|N_i)
\]

(5)

By Eq. (4),

\[
p(x_i|N_i) = \sum_{y_i \in A} p(x_i|y_i) p(y_i|N_i)
\]

(6)

Since every probabilistic distribution \( p(x_i|y_i) \) has the same function described as shown Eq. (1), and by Eq. (6) that Eq. (5) can be rewritten as the hidden Markov random field model [7]

\[
p(x_i|N_i) = \sum_{y_i \in A} f(x_i; \theta_{y_i}) p(y_i|N_i)
\]

\[
= p(x_i|N_i; \theta_{y_i})
\]

(7)

where \( \theta_{y_i} \) is the parameter of the distribution function. If the random variable \( X_i \) follows a Gaussian distribution, i.e., \( f(x_i; \theta_{y_i}) \), is formulated as

\[
f(x_i; \theta_{y_i}) = \frac{1}{\sqrt{2\pi \sigma_{y_i}^2}} \exp \left( -\frac{(x_i - \mu_{y_i})^2}{2\sigma_{y_i}^2} \right)
\]

(8)

where \( \theta_{y_i} = (\mu_{y_i}, \sigma_{y_i}). \) And HMRF can be rewritten as

\[
p(x_i|N_i; \theta_{y_i}) = \sum_{y_i \in A} \frac{1}{\sqrt{2\pi \sigma_{y_i}^2}} \exp \left( -\frac{(x_i - \mu_{y_i})^2}{2\sigma_{y_i}^2} \right) p(y_i|N_i)
\]

(9)

Remark 1: In this paper, the state variable \( X \) and the action variable \( Y \) of the multi-agent are discrete, hence the probability distribution function \( f(x_i; \theta_{y_i}) \) should meet the condition of Eq. (2), that means \( 0 \leq f(x_i; \theta_{y_i}) \leq 1 \).

Remark 2: The form of \( f(x_i; \theta_{y_i}) \) showed Eq. (8) is different from the probability distribution function for the continuous random variable, in which \( f(\cdot) > 1 \) will be held if the standard deviation \( \sigma \) is set as a very small constant. Hence if the probability distribution function for the discrete random variable as Eq. (8), \( \sigma_{y_i} > \rho \) must be held to meet \( 0 \leq f(x_i; \theta_{y_i}) \leq 1 \). Actually, the simulation experiment
verifies this condition should be held, if there is no condition \( \sigma_i > \rho \), the result is in confusion; while if the condition is obeyed, the result is reasonable.

C. MAXIMUM A POSTERIORI WITH HMRF FOR OPTIMAL DECISION

Based on the HMRF, the multi-agent system optimal decision could be transformed into a MAP problem. Given the observable random field of state \( X = \{X_i\}, X_i \in S \), the aim is to find the maximum probability of hidden actions \( Y = \{Y_i\}, Y_i \in A \). If the configuration of \( x \) and \( y \) is respectively \( x = \{x_1, \ldots, x_N\} \) and \( y = \{y_1, \ldots, y_N\} \), multi-agent system optimal decision could be obtained by MAP

\[
y^* = \arg \max_y P(x|y) P(y) \quad (10)
\]

Since the conditional independent characteristic Eq. (3), and \( P(x_i|y_i) \) has the same probability distribution as shown in Eq. (1), meanwhile the element of random variable \( Y = \{Y_i\}, Y_i \in A \) is independent, the optimal action decision \( y^* \) is determined by the following formulation

\[
y^* = \arg \max_{y_i \in A, i = 1, \ldots, N} \prod_i p(x_i|y_i) p(y_i) = \arg \max_{y_i \in A, i = 1, \ldots, N} \prod_i f(x_i; \theta_i) p(y_i) \quad (11)
\]

where \( p(y_i) \) is the prior probability, \( p(x_i|y_i) \) is an adjustment term to improve the output precision based on \( p(y_i) \). In HMRF, the optimal decision shown as Eq. (11) could be formulated as below

\[
y^* = \arg \max_{y_i \in A, i = 1, \ldots, N} \prod_i p(x_i|y_i, N_i) p(y_i|N_i) \quad (12)
\]

By Eq. (4) and Eq. (1),

\[
y^* = \arg \max_{y_i \in A, i = 1, \ldots, N} \prod_i p(x_i|y_i) p(y_i|N_i) = \arg \max_{y_i \in A, i = 1, \ldots, N} \prod_i f(x_i; \theta_i) p(y_i|N_i) \quad (13)
\]

D. PARAMETER ESTIMATION FOR OPTIMAL DECISION VIA EXPECTATION-MAXIMIZATION ALGORITHM

Parameter estimation is one of the keys to infer the probability by PGM. However, the non-stationary dynamic process of multi-agent makes it hard to obtain the ground truth for the training data, hence in HMRF, the actions for training data are considered as the latent variables.

The EM algorithm [8] is a famous method to estimate the parameters of the model with latent variables. In the multi-agent system, the state of each agent \( X = \{X_1, \ldots, X_N\} \) is known as the discrete random variable, while the action \( Y = \{Y_1, \ldots, Y_N\} \) is the ground truth as the latent random variable. The aim is to estimate \( \theta = \{\theta_i, Y_i \in A\} \) in Eq. (11) then to obtain the optimal actions.

The joint probability distribution of \( x \) with the parameter \( \theta \) is \( p(x; \theta) \) and the likelihood can be calculated as below

\[
L(\theta) = \sum_i \log p(x_i; \theta) = \sum_i \log \left( \sum_{y_i \in A} p(x_i, y_i; \theta_i) \right) \quad (14)
\]

Suppose that \( \Theta_i \) is a probability distribution over \( y_i \), and

\[
\sum_{y_i} \Theta_i(y_i) = 1, \quad \Theta_i(y_i) \geq 0 \quad (15)
\]

By Jensen’s inequality [8] and Eq. (14), the infimum of the likelihood \( L_1(\theta) \) is calculated as below

\[
L_1(\theta) = \sum_i \log \left( \sum_{y_i \in A} \Theta_i(y_i) \frac{p(x_i, y_i; \theta_i)}{\Theta_i(y_i)} \right) \geq \sum_i \Theta_i(y_i) \log \left( \frac{p(x_i, y_i; \theta_i)}{\Theta_i(y_i)} \right) \quad (16)
\]

Note that \( \Theta_i(y_i) \) is a probability distribution, hence the right hand of the inequality Eq. (16) can be represented as \( E_{y_i \sim \Theta_i} \left\{ \log \frac{p(x_i, y_i; \theta_i)}{\Theta_i(y_i)} \right\} \) that means the expectation of

\[
\log \frac{p(x_i, y_i; \theta_i)}{\Theta_i(y_i)} \quad (17)
\]

where \( C \) is a constant, the inequality Eq. (16) gets the greatest low boundary. By Eq. (17)

\[
\sum_{y_i} p(x_i, y_i; \theta) = \sum_{y_i} \Theta_i(y_i) = C \quad (18)
\]

Substitute the result of Eq. (18) into Eq. (17)

\[
\frac{p(x_i, y_i; \theta_i)}{\Theta_i(y_i)} = \sum_{y_i} p(x_i, y_i; \theta_i) \quad (19)
\]

The probability distribution \( \Theta_i \) can be obtained

\[
\Theta_i(y_i) = \sum_{x_i} p(x_i, y_i; \theta_i) = \sum_{x_i} \frac{p(x_i, y_i; \theta_i)}{p(x_i; \theta_i)} = p(y_i|x_i; \theta_i) \quad (20)
\]

That is the E-step of the EM algorithm. And the next is to find the maximum of \( L_1(\theta) \) by

\[
\theta \leftarrow \arg \max_{\theta} \sum_{i} \sum_{y_i \in A} \Theta_i(y_i) \log \left( \frac{p(x_i, y_i; \theta_i)}{\Theta_i(y_i)} \right) \quad (21)
\]
The $\theta$ will substitute into Eq. (20) to calculate the $\Theta_i(y_i)$ for an iteration till the result convergence.

The key to Eq. (11) is to estimate the parameter $\theta = \{\theta_y, y_i \in A\}$ in HMRF by EM algorithm. Since the latent variable in Eq. (11) is $Y = \{Y_1, \ldots, Y_N\}$, so $\Theta_i(y_i)$ is calculated based on Eq. (20)

$$\Theta_i(y_i) = \frac{p(y_i, x_i|N; \theta_y)}{p(x_i|N)} = \frac{f(x_i; \theta_y) p(y_i|N)}{\sum_{y_i} f(x_i; \theta_y) p(y_i|N)}$$

and $\theta$ can be derived by the following equation

$$\frac{\partial}{\partial \theta_y} \sum_i \sum_{y_i \in A} \Theta_i(y_i) \log \frac{p(x_i, y_i|N; \theta_y)}{\Theta_i(y_i)} = 0$$

By Eq. (11)

$$\sum_i \sum_{y_i \in A} \Theta_i(y_i) \log \frac{p(x_i, y_i|N; \theta_y)}{\Theta_i(y_i)} = \sum_i \Theta_i(y_i) \log \frac{f(x_i; \theta_y) p(y_i|N)}{\Theta_i(y_i)} = \sum_i \sum_{y_i \in A} \Theta_i(y_i) \log \frac{f(x_i; \theta_y) p(y_i|N)}{\Theta_i(y_i)}$$

Hence,

$$\frac{\partial}{\partial \theta_y} \sum_i \sum_{y_i \in A} \Theta_i(y_i) \log \frac{f(x_i; \theta_y) p(y_i|N)}{\Theta_i(y_i)} = \sum_i \Theta_i(y_i) \frac{\partial}{\partial \theta_y} \log f(x_i; \theta_y)$$

where $f(x_i; \theta_y)$ is formulated as Eq. (8), $\theta_y = \{\mu_y, \sigma_y\}$

$$\sum_i \Theta_i(y_i) \frac{\partial}{\partial \mu_y} \log f(x_i; \theta_y) = \sum_i \Theta_i(y_i) \frac{x_i - \mu_y}{\sigma_y^2} = 0$$

Hence

$$\mu_y = \frac{\sum_i \Theta_i(y_i) x_i}{\sum_i \Theta_i(y_i)}$$

And in the same way,

$$\sigma_y^2 = \frac{\sum_i \Theta_i(y_i) (x_i - \mu_y)^2}{\sum_i \Theta_i(y_i)}$$

### IV. HMRF WITH Q-LEARNING ALGORITHM FOR MULTI-AGENT SYSTEM OPTIMAL DECISION

Q-learning algorithm [10] is one of the effective off policy RL methods. The core of Q-learning is learning the value of the Q matrix which represents the relationship between action and the state. If the action decision is represented by probability, the output of the RL system could be considered as the prior probability for the MAP.

#### A. PRIOR PROBABILITY ESTIMATION BY Q-LEARNING ALGORITHM

Consider the multi-agent in the MRF, the state space and action space are the same for every agent. If the configuration action and state of each agent are respective $y = \{y_1, y_2, \ldots, y_N\}$, $y_i \in A$, and $x = \{x_1, x_2, \ldots, x_N\}$, $x_i \in S$, and the reward of each agent is $\{R_1, R_2, \ldots, R_N\}$ in a step of the Markov process, we can define the action-value function $Q(\bar{x}, \bar{y})$, $\bar{x} \in x$, $\bar{y} \in y$. Each agent acts with the same Q-value function under the state of its own. The learning mechanism of the Q matrix is shown below

$$Q(\bar{x}, \bar{y}) \leftarrow Q(\bar{x}, \bar{y}) + \alpha \Psi$$

where $\alpha$ denotes the learning step; $\beta > 0$ is the discount rate; $\alpha > 0$ is the learning step-size.

Since the Q-value function of every agent is the same, the dimension of the Q matrix is $L \times D, L$ and $D$ is respective the dimension of action space and state space.

The agents learn the same Q-matrix independently during interacting with the environment by the reward and action. Then, the softmax result of the Q-matrix could be considered as the prior knowledge of policy decision for the multi-agent in the HMRF, hence the prior probability of $P (y_i|N_i)$ in Eq. (13) could be transformed as the follows

$$P (y_i|N_i) = Z^{-1} \exp(\alpha U (y_i|N_i))$$

where $U (y_i|N_i)$ is the energy function, and $Z$ is the normalization constant. Hence $U (y_i|N_i)$ and $Z$ are set as

$$U (y_i|N_i) = Q (\bar{y}_i = y_i, \bar{x}_i = x_i)$$

$$Z = \sum_{y_i} Q (\bar{y}_i = y_i, \bar{x}_i = x_i)$$

According to Eq. (8), (13), and (31), the optimal actions of multi-agent could be inferred by MAP based on the HMRF as below

$$y^* = \arg \max_{y_i \in A \cup \{1, 2, \ldots, N\}} \prod_i \Phi_1(y_i)$$

$$\Phi_1(y_i) = \frac{1}{Z \sqrt{2\pi \sigma^2_{y_i}}} \exp \left( -\frac{(x_i - y_i)^2}{2\sigma^2_{y_i}} \right) \exp(\alpha U (y_i|N_i))$$

The likelihood of $\prod_i \Phi_1(y_i)$ is

$$L_2 (\theta) = -\log \prod_i \Phi_1(y_i)$$

$$= \sum_i \frac{(x_i - \mu_y)^2}{2\sigma_y^2} + \log \sqrt{2\pi} \sigma_y + U (y_i|N_i) + \log (Z)$$
where $\theta = \{\mu_y, \sigma_y\}$. And consider the likelihood of $f(x_i; \theta_y)$, the optimal decision of $y$ could be obtained as below

$$y^* = \arg\min_{y_i \in \mathcal{A}} \sum_i \Phi_2(y_i) \tag{34}$$

$$\Phi_2(y_i) = \frac{(x_i - \mu_{y_i})^2}{2\sigma^2_{y_i}} + \log(\sigma_{y_i}) + U(y_i|N_i) \tag{35}$$

It should be noted that the state of the multi-agent is the parameter that describes the agent in the environment. For the algorithm of this paper, the state parameter could be continuous or discrete, but the action of each agent should be discrete and could be depicted by probabilistic form. The reward model is one of the keys to the RL system, and it is hard to get a perfect model. In our methodology, the Q value should be trained as the prior probability, hence the reward depends on the application. In the top-coal caving of this paper, the reward depends on the capture of each agent, and the model is given in Eq. (37).

### B. ALGORITHMS INTEGRATION

The methodology proposed in this paper includes 4 blocks, HMRF, EM, Q-learning, and MAP. The HMRF provides the model for the optimal decision. The EM estimates the parameters of HMRF. The Q-learning produces the prior probability for the MAP, and the MAP calculates the optimal action for each agent. The architecture is shown in Fig. 2.

In Fig. 2(a), the input of the multi-agent system is the observable states $x = \{x_1, \ldots, x_N\}$, and the output optimal decision is $y^*$ by Eq. (35). The HMRF model in Eq. (8) provides the optimal model for MAP, and the prior probability produces by the probability form of Q-value in Eq. (32). In Fig. 2(b), the iteration of E-step Eq. (22) and M-step Eq. (27), Eq. (28) are used to estimate the parameters of the HMRF. The methodology should be implemented by two steps which are respectively shown in Algorithm 1 and Algorithm 2.

In algorithm 1, $Q(\bar{x}, \bar{y})$ is shared by all the agents, and it is trained by signal-agent with the Q-learning algorithm [10].

![Algorithm 1 Q-Value Training by Signal-Agent](algorithm.png)

![Figure 2. The methodology of optimal decision for the multi-agent. HMRF and RL are the two branches in this methodology. Since MAP and EM are established based on HMRF, hence they are embedded into the HMRF; The RL is implemented by Q-learning with probability form which is sent to MAP as the prior probability of HMRF.](image.png)
**Algorithm 2 Multi-Agent Optimal Decision by HMRF**

**Input:** states $x$, prior probability $p(y_i|N_i)$

1. Initialize parameters of Eq.9, $\mu_{y_i}, \sigma_{y_i}, y_i \in \mathcal{A}$
2. Set the maximum number of loop $N_m$, the terminal state $S_E, S_\Phi \in \mathcal{S}$, the small constant $\zeta$ of convergence condition of Eq.36.
3. while all the state $x_i \neq S_E, i = 1, \ldots, N$ do
   
   for $m \leftarrow 1$ to $N_m$ do
   
   Calculate $f(x_i; \theta_{y_i})$ by Eq. (8)
   
   Get $\theta_i(y_i)$ by Eq. (22)
   
   Update $\mu_{y_i}, \sigma_{y_i}$ by Eq. (27) and (28)
   
   Calculate $\Delta$ by Eq. (36)
   
   if $\Delta < \zeta$ then
   
   Break
   
   end
   
   Output $f(x_i; \theta_{y_i})$ by Eq. (8)
   
   Calculate the optimal decision $y^*$ by Eq. (13)
   
   Output the action $y^*$
   
   Update the states $x$

end

The output of algorithm 1 with probability form is used as the prior probability of $p(y_i|N_i)$ for HMRF.

Algorithm 2 is to get the optimal decision of actions for the multi-agent based on the prior probability $p(y_i|N_i)$. The environment is changing during the multi-agent actions are executing, that states of the multi-agent are variational, and the optimal decisions should be updated following the environment changes. Hence, the optimal decision in Algorithm 2 is a transient result.

Remark 3: Generally, in Algorithm 2, the optimal decision is calculated by Eq. (13) with the iteration method, such as the iterated conditional modes [7]. In this paper, the prior probability with HMRF is obtained by the RL, hence it could be calculated by Eq. (13) directly, or the iteration number is 1.

Remark 4: In Algorithm 2, the condition of breaking the EM iteration is $\mu_{y_i}, \sigma_{y_i}$ approaching convergence, i.e., $\sum_{i} \Phi_2(y_i)$ in Eq.35 is a constant. In the implementation, the condition often set as below.

$$\Delta = T_{m+1} - T_m < \zeta$$  \hspace{1cm} (36)

where $T_{\Phi} = \sum_{i} \Phi_2(y_i)$, $m$ is the counter of EM iteration, and $\zeta$ is a small constant.

**V. EXPERIMENT AND RESULT ANALYSIS**

**A. TOP-COAL CAVING EXPERIMENT**

1) TOP-COAL CAVING SIMULATION PLATFORM

In order to validate the proposed methodology of this paper, we develop a simulation platform by Matlab based on the open source of a two-dimension discrete element model (DEM), named dice [66]. We set 3 kinds of particles in the DEM to denote the coal, rock of the immediate-roof, and rock of the main-roof, respectively. And we set 5 windows to demonstrate the dynamic process of top-coal caving with the decision by the proposed methodology. The main output is shown in Fig. 3. In the coal particle area, there are 60 rock particles with a random location. Meanwhile, in the rock area, there are 10 coal particles with a random location. Furthermore, both sides of the boundary between coal and rock, there are 20 opposite particles with a random location.

The source code can be accessed at “https://github.com/YangYi-HPU/Reinforcement-learning-simulation-environmnet-for-top-coal-caving” for non-commercial uses only.

2) MODEL OF HMRF AND LEARNING PROCESS

The action space of HS is $\mathcal{A} = \{a_1, a_2\}$, $a_1$ and $a_2$ respectively denote the window being open and close. In this paper, the state of the HS is the ratio of coal in the total particles near the window, we set 12 levels $\mathcal{S} = \{s_1, \ldots, s_{12}\}$, and the corresponding ratio (or interval of the ratio ) is $0, (0, 0.1], (0.1, 0.2], \ldots, (0.9, 1.0], 1.0$. Hence the dimension of the Q matrix is set as $2 \times 12$. Each of the windows is considered as an agent, hence the state random variable of the multi-agent is $X = \{X_1, \ldots, X_5\}$, and the hidden random variable is $Y = \{Y_1, \ldots, Y_5\}$. The conditional probability distribution is Eq. (8) with the constraint $\sigma_{y_i} > \rho$. Since $y_i \in \mathcal{A} = \{a_1, a_2\}$ the parameter of HMRF can be denoted as $\theta = \{\mu_{a_1}, \sigma_{a_1}; \mu_{a_2}, \sigma_{a_2}\}$. Based on the HMRF model, the learning process of the experiment is regulated as below.

3) LEARNING THE Q MATRIX BY Q-LEARNING ALGORITHM

In this process, the reward is calculated by the following equation.

$$R = n_r r_r + n_c r_c$$  \hspace{1cm} (37)

where $n_r, n_c$ is respective the number of rock and coal obtained by caving; $r_r, r_c$ is respectively the reward of getting one rock particle and one coal particle. In the experiment, $r_r = -3, r_c = 1$, and the parameter of the Q-learning algorithm is set as $\alpha = 0.2, \beta = 0.1$. After trained, the output of Q matrix with probability form is shown in Tab. 1.
4) LEARNING THE HMRF MODEL
In this process, the HMRF model is trained following the Algorithm 2, and the parameters are initialized as $\mu_{a1} = \mu_{a2} = 0$, $\sigma_{a1} = \sigma_{a2} = 1$; the constraint conditional is set as $\rho = 1$; the iteration number of EM is 10; the condition of breaking the EM iteration is $\zeta = 0.0001$.

5) EXPERIMENT RESULT ANALYSIS
The simulations are carried out to compare the effectiveness of the proposed methodology with the RL method [10] and reinforcement learning with equivalent action [16] in which the communication between the current agent and its neighbors is depicted by the equivalent action based on mean field theory. In the following analysis, the three compared methods are denoted by RL, MF, and HMRF, respectively.

The main performance indexes are coal-recall (CR) and the ratio of rock (RR) in all mining, and the formulations are shown below.

$$CR = \frac{n_c}{n_{c, total}}$$

$$RR = \frac{n_r}{n_c + n_r}$$

where $n_c$ and $n_r$ is the number of rock and coal obtained by caving, $n_{c, total}$ is the number of all the coal particles.

Since the location of some particles is randomly assigned, in this paper 10 tests with different random locations are carried out to validate the effectiveness. The parameter dynamic process of $T_\phi$ which indicates the convergence during the training is shown in Fig. 4.

In Fig. 4, the convergence index $T_\phi$ is the term of $\sum_i \Phi_2 (y_i)$ in Eq. (35). It should be noted that the environment of action is non-stationary during the windows being regulated. Hence, if the environment changed, the convergence index would have a step change, then converges to the minimum value. In the middle of the process, the state of the windows changes frequently since the states approach the boundary, hence the dynamic of the convergence index is complex. While at the end of the process, the coal has been exploited and the environment is more stationary, hence the convergence index could approach to the steady value.

The performance indexes CR and RR of experiment results are shown in Tab. 2 and Fig. 5. And the Fig. 6 shows 2 typical final scenes of the top-coal caving result, in which Figure 6(a) and (b) are the typical results of adopting the RL method, Figure 6(c) and (d) are the results of the MF method, and Figure 6(e) and (f) are produced by the HMRF method.

According to Tab. 2 and Fig. 5, it is obvious to find that the RR is increased prominently by HMRF method with a small price of RR, in which the average CR is increased from 0.5693 to 0.9188 with 0.3495 growth, and the RR is just pushed up from 0.0605 to 0.1530 with 0.0925 growth. Meanwhile, the HMRF obtains a higher reward than MF. The intuitive advantage can be found in Fig. 6, the top-coal can be exploited more fully by the HMRF method.

It is clearly shown in Tab. 1 that if the state is $s_9$ and below, the probability of open by RL regulating is less than close. In Fig. 7, the action record of a window is shown, in
FIGURE 6. Final scenes of top-coal caving. The Figure (a) and (b) are the final scenes of top-coal caving by RL; Figure (c) and (d) are the final scenes of top-coal caving by MF; and Figure (e) and (f) are the final scenes of top-coal caving by HMRF.

FIGURE 7. Action record of a window for top-coal caving. The value of an action is 1 and 2 which means the window is open and close. In the RL regulating process, if the state is $s_9$ and below, the window is closed, otherwise it is opened. While in MF and HMRF regulating processes, the action dynamic is more complex, and the window action is not just depending on the static state but also the action of the neighbors.

which Fig. 7 (a), (b) and (c) displays the entire action process regulated by RL, MF, and HMRF method, respectively. That the actions of the three methods are different since RL does not consider the neighbors of each agent, it could not obtain a high RR, and the top-coal cannot be exploited completely, while actions of MF and HMRF are based on the neighbors which bring the high CR.

B. PREDATOR-PREYS EXPERIMENT

1) EXPERIMENT ENVIRONMENT

This experiment is carried out on the multi-agent competitive environment [67] based on OpenAI’s reinforcement learning toolkit Gym [68]. There are two preys and a predator in the restricted area. The predator “eat” the preys and the preys avoid that case. To validate the algorithm of this paper, at first, the three competitive agents are trained, hence the predator can learn its action policy and the two preys can act collaboratively with a different algorithm. Then the preys act as the multi-agent to test the performance of different algorithms.

The reward is defined as: (1) Collision reward: the predator owns a positive reward 10 if it collides with a prey, meanwhile the prey will be punished with a negative reward $-10$. (2) Outside punishment: if the agent runs out of the area, it will be punished the distance between the agent location and area boundary.

There are 12 state parameters of each agent, the position (2 states) and velocity (2 states) of themselves, the relative positions (4 states) and velocities (4 states) of other agents. In this paper, the two preys are the cooperative multi-agent and avoid being eaten. For the method of this paper, the state of each prey is the relative position of the predator. And the HMRF model is formalized as below

$$f(x_t, y_t; \theta_y) = \frac{1}{2\pi \sigma_{x_t} \sigma_{y_t}} \exp \left( -\frac{(x_t - \mu_{x_t})^2}{2\sigma_{x_t}^2} - \frac{(y_t - \mu_{y_t})^2}{2\sigma_{y_t}^2} \right)$$ (39)
where $\theta_i = \{\sigma_{x_i}, \sigma_{y_i}, \mu_{x_i}, \mu_{y_i}\}$, $x_i$ and $y_i$ are the relative position of predator along $x$-axis and $y$-axis. The action space is $\mathcal{A} = \{0, 1, 2, 3\}$, and $y_i \in \mathcal{A}$, these values denote the action direction \{right, left, up, down\}. According to Eq. (9), the HMRF model of prey formulated as

$$p(x_i|N_i; \theta_{y_i}) = \sum_{y_i \in \mathcal{A}} f(x_i, x_j; \theta_{y_i}) p(y_i|N_i)$$ (40)

The optimal action can be obtained for the below equation which is deduced by the same method in section IV

$$y^* = \arg \min_{y_i \in \mathcal{A}, i=1,...,N} \sum_i \Phi_2(y_i)$$

$$\Phi_2(y_i) = \frac{(x_i - \mu_{x_i})^2}{2\sigma_{x_i}^2} + \frac{(y_i - \mu_{y_i})^2}{2\sigma_{y_i}^2}$$

$$+ \log (\sigma_{x_i}) + \log (\sigma_{y_i}) + U(y_i|N_i)$$ (41)

By the analysis method of section III, the update of parameters are the same with Eq. (27) and Eq. (28).

2) EXPERIMENT AND ANALYSIS

In the experiment, 4 typical reinforcement learning algorithms, Q-learning, deep Q-network, deep policy gradient, actor-critic are carried out to compare the optimal decision performance. In the following analysis, the above algorithms and our method are denoted by RL, DQNFT, DPG, AC, and HMRF, respectively.

The code of RL and DQN is based on the reference [67]. In the RL method, a neural network used to approximate the Q-value, and the DQN applies the network architecture of the Q-value while adopts the fixed target network to update the parameters. The core of DPG is the reinforce algorithm [69] and the actor-critic algorithm is based on the reference [10].

At first, the RL, DQN, and HMRF are trained for 20000 episodes, DPG and AC takes 40000 episodes training. In the HMRF, the prior probability comes from the training result of RL. After training, the agent of predator is fixed and the HMRF algorithm is applied to the agent of preys. Then, 100 tests are carried out to compare the reward of each agent, in which the initial location of each agent is random. The experiment result is shown in Fig. 8.

As shown in Fig. 8, the two preys with HMRF method improves the reward of most of the 100 tests, and the predator decreases the reward in the corresponding episode. Because
To the best of our knowledge, it is the first time that the HMRF is employed to deal with the optimal decision of multi-agents in such complex environments. To extend and improve this work, future work includes:

1. The training process of HMRF is divided into two processes: the prior probability obtaining by Q-learning and parameter learning of the HMRF model by EM algorithm. Hence a further study is needed to train the prior probability and parameters simultaneously.

2. The Q-value with the probability form Eq. (31) is similar to the Gibbs distribution. Actually, the output actions from the Markov random field of the multi-agent is equivalent to the Gibbs distribution [70]. Hence, how to incorporate the Hammersley Clifford theorem into the HMRF will be very beneficial for improving the current approach.

3. In the predator-preys experiment, there are 12 states for the Q-learning algorithm, while just 2 states are selected for HMRF since it is complex to formulate the Markov random field model for the system with multi-state variables. Hence, how to find a simple way to establish the Markov random field for multi-state variables needs further study.

REFERENCES

[1] T. D. Le, C. Zhang, J. Oh, R. Mitra, and B. Hebblewhite, “A new cavability assessment for longwall top coal caving from discontinuum numerical analysis,” Int. J. Rock Mech. Mining Sci., vol. 115, pp. 1–10, Nov. 2019.

[2] I Company. Fully Mechanized Coal Mining Work Space of Tang Gong Ta Coal Mine. Accessed: Apr. 21, 2019. [Online]. Available: https://www.mmgmsy.com/products_info.aspx?ProductsID=141

[3] C. Liu, H. Li, and H. Miriti, “Effect of strata conditions on shield pressure and surface subsidence at a longwall top coal caving working face,” Rock Mech. Rock Eng., vol. 52, no. 5, pp. 1523–1537, May 2019.

[4] X. Yang, E. Wang, X. Ma, G. Zhang, R. Huang, and H. Lou, “A case study on optimization and control techniques for entry stability in non-pillar longwall mining,” Energies, vol. 12, no. 3, p. 391, 2019.

[5] R. Lowe, Y. Wu, A. Tamar, J. Harb, O. P. Abbeel, and I. Mordatch, “Multi-agent actor-critic for mixed cooperative-competitive environments,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 6379–6390.

[6] D. Kollet, N. Friedman, and F. Bach, Probabilistic Graphical Models: Principles and Techniques, Cambridge, MA, USA: MIT Press, 2009.

[7] Y. Zhang, M. Brady, and S. Smith, “Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm,” IEEE Trans. Med. Imag., vol. 20, no. 1, pp. 45–57, 2001.

[8] T. K. Moon, “The expectation-maximization algorithm,” IEEE Signal Process. Mag., vol. 13, no. 6, pp. 47–60, Nov. 1996.

[9] S. Levine, “Reinforcement learning and control as probabilistic inference: Tutorial and review,” 2018, arXiv:1805.00909. [Online]. Available: http://arxiv.org/abs/1805.00909

[10] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction. Cambridge, MA, USA: MIT Press, 2018.

[11] Z. Zhang, D. Zhao, J. Gao, D. Wang, and Y. Dai, “FMRQ—A multiagent reinforcement learning algorithm for fully cooperative tasks,” IEEE Trans. Cybern., vol. 47, no. 6, pp. 1367–1379, Jun. 2017.

[12] M. L. Littman, “Markov games as a framework for multi-agent reinforcement learning,” in Machine Learning Proceedings 1994. Amsterdam, The Netherlands: Elsevier, 1994, pp. 157–163.

[13] W. Jiang, G. Feng, S. Qin, T. S. P. Yum, and G. Cao, “Multi-agent reinforcement learning for efficient content caching in mobile D2D networks,” IEEE Trans. Wireless Commun., vol. 18, no. 3, pp. 1610–1622, Mar. 2019.

[14] B. Luo, D. Liu, H.-N. Wu, D. Wang, and F. L. Lewis, “Policy gradient adaptive dynamic programming for data-based optimal control,” IEEE Trans. Cybern., vol. 47, no. 10, pp. 3341–3354, Oct. 2017.
[15] Q. Wei and D. Liu, “Adaptive dynamic programming for optimal tracking control of nonlinear systems with application to coal gasification,” IEEE Trans. Autom. Sci. Eng., vol. 11, no. 4, pp. 1020–1036, Oct. 2014.

[16] Y. Yang, R. Luo, M. Li, M. Zhou, W. Zhang, and J. Wang, “Mean field multi-agent reinforcement learning,” 2018, arXiv:1802.05438. [Online]. Available: http://arxiv.org/abs/1802.05438

[17] H. Van Dong, B. Q. Khanh, N. T. Lich, and N. T. N. Anh, “Integrating multi-agent system, geographic information system, and reinforcement learning to simulate and optimize traffic signal control,” in Proc. Int. Conf. Comput. Inf. Technol. Cham, Switzerland: Springer, 2018, pp. 145–154.

[18] D. A. Vidyate and P. Kulkarni, “Cooperative multi-agent reinforcement learning models (CMRLM) for intelligent traffic control,” in Proc. 1st Int. Conf. Intell. Syst. Inf. Manage. (ICSIM), Oct. 2017, pp. 325–331.

[19] D. Luviano-Cruz, F. Garcia-Luna, L. Pérez-Dominguez, and S. Gadi, “Multi-agent reinforcement learning using linear fuzzy model applied to cooperative mobile robots,” Symmetry, vol. 10, no. 10, p. 461, 2018.

[20] M. L. Patkin and G. N. Rogachev, “Construction of multi-agent mobile robots control system in the problem of perception with using a modified reinforcement learning method based on neural networks,” IOP Conf. Ser., Mater. Sci. Eng., vol. 312, Feb. 2018, Art. no. 012018.

[21] P. Long, T. Fanl, X. Liao, W. Liu, H. Zhang, and J. Pan, “Towards optimally decentralized multi-agent collision avoidance via deep reinforcement learning,” in Proc. IEEE Int. Conf. Robot. Automat. (ICRA), May 2018, pp. 6252–6259.

[22] S. Lall, A. K. Saha, A. Konar, K. K. Mallik, and S. Ghosh, “Multi-agent reinforcement learning for stochastic power management in cognitive radio network,” in Proc. Int. Conf. Micronetworks, Comput. Commun. (MicroCom), Jan. 2016, pp. 1–6.

[23] E. B. Dounia, Y. Ali, and B. Jauoud, “Multi-agent system based on the fuzzy control and extreme learning machine for intelligent management in hybrid energy system,” in Proc. Intell. Syst. Comput. Vis. (ISCV), Apr. 2017, pp. 1–6.

[24] R. Bellman, “Dynamic programming,” Science, vol. 153, no. 3731, pp. 34–37, 1966.

[25] L. Matignon, G. J. Laurent, and N. Le Fort-Piat, “Independent reinforcement learners in cooperative Markov games: A survey regarding coordination problems,” Knowl. Eng. Rev., vol. 27, no. 1, pp. 1–31, Feb. 2012.

[26] L. Bajoniu, R. Babuška, and B. De Schutter, “Multi-agent reinforcement learning: An overview,” in Innovations in Multi-Agent Systems and Applications I, Berlin, Germany: Springer, 2010, pp. 183–221.

[27] J. Hu and P. Wellman, “Multiagent reinforcement learning: Theoretical framework and an algorithm,” in Proc. Int. Conf. Mach. Learn., vol. 98, 1998, pp. 242–250.

[28] S. Ross and J. Pineau, “Model-based Bayesian reinforcement learning in structured domains,” in Uncertainty in Artificial Intelligence, Bethesda, MD, USA: NIH Public Access, 2008, p. 476.

[29] F. Doshi-Velez, D. Pfau, F. Wood, and N. Roy, “Bayesian nonparametric methods for partially-observable reinforcement learning,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 37, no. 2, pp. 394–407, Feb. 2015.

[30] F. Doshi-Velez, D. Wingate, N. Roy, and J. B. Tenenbaum, “Nonparametric Bayesian models for predicting and diagnosing based on acoustic emission and hidden semi-Markov model,” Int. J. Adv. Mach. Learn., vol. 90, nos. 5–8, pp. 2027–2036, May 2017.

[31] M. G. Don and F. Khan, “Process fault prognosis using hidden Markov model-Bayesian networks (HMM-BN) hybrid model,” Ind. Eng. Chem. Res., vol. 58, no. 27, pp. 12041–12053, Jul. 2019.

[32] T. Jaakkola, S. P. Singh, and M. I. Jordan, “Reinforcement learning algorithm for partially observable Markov decision problems,” in Proc. Adv. Neural Inf. Process. Syst., 1995, pp. 345–352.

[33] F. Doshi-Velez, “The infinite partially observable Markov decision process,” in Advances in Neural Information Processing Systems, Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, Eds. Red Hook, NY, USA: Curran Associates, 2009, pp. 477–485.

[34] N. Dethlefs and H. Cuayañú, “Hierarchical reinforcement learning and hidden Markov models for task-oriented natural language generation,” in Proc. 49th Annu. Meeting Assoc. Comput. Linguistics, Hum. Lang. Technol., vol. 2, 2011, pp. 654–659.

[35] C. Daniel, H. van Hoof, J. Peters, and G. Neumann, “Probabilistic inference for determining options in reinforcement learning,” Mach. Learn., vol. 104, nos. 2–3, pp. 337–357, Sep. 2016.

[36] J. Papamarkos and G. E. P. Box, “Reinforcement learning with factored states and actions,” J. Mach. Learn. Res., vol. 5, pp. 1063–1088, Aug. 2004.

[37] X. Tian, C.-B. Chang, Y. Su, K.-B. Lim, and C.-K. Chui, “Robot-assisted training in laparoscopy using deep reinforcement learning,” IEEE Robot. Autom. Lett., vol. 4, no. 2, pp. 485–492, Apr. 2019.

[38] Y. Li, X. Hu, Y. Zhang, Z. Gao, P. Zhang, and N. El-Shemy, “Deep reinforcement learning (DRL): Another perspective for unsupervised wireless localization,” IEEE Internet Things J., early access, Dec. 5, 2019, doi:10.1109/JOT.2019.2957778.

[39] S. P. Chatzis and G. Tsetchenakis, “The infinite hidden Markov random field model,” IEEE Trans. Neural Netw., vol. 21, no. 6, pp. 1004–1014, Jun. 2010.

[40] H. Wang and J. Shi, “SAR image segmentation algorithm based on contourlet domain AFMRF model,” IET Image Process., vol. 12, no. 7, pp. 1124–1130, Jul. 2018.

[41] H. Handa, “EMR-based medical knowledge representation and inference via Markov random field model,” in Proc. 41st Int. Joint Conf. Artif. Intell., 2019, pp. 242–251.

[42] C. Zhao, J. Jiang, Y. Guan, X. Guo, and B. He, “EMR-based medical knowledge representation and inference via Markov random fields and distributed representation learning,” Artif. Intell., vol. 87, pp. 49–59, May 2018.

[43] P. W. Battaglia et al., “Relational inductive biases, deep learning, and graph networks,” 2018, arXiv:1806.01261. [Online]. Available: http://arxiv.org/abs/1806.01261

[44] S. S. Seyhan, F. N. Alpaslan, and M. Yayav, “Simple and complex behavior for cloud computing,” IEEE Trans. Services Comput., vol. 10, pp. 287–294, Jul. 2017.

[45] S. Ross, D. Stirling, Z. Pan, and F. Naghdy, “Learning trajectories for robot programming by demonstration using a coordinated mixture of factor analyzers,” IEEE Trans. Cybern., vol. 46, no. 3, pp. 706–717, Mar. 2016.

[46] N. Vuković, M. Mitić, and Z. Miljković, “Activity trajectory learning and reproduction for differential drive mobile robots based on GMM/HMM and dynamic time warping learning using demonstration framework,” Eng. Appl. Artif. Intell., vol. 45, pp. 388–404, Oct. 2015.

[47] S. S. Seyhan, D. Stirling, Z. Pan, and F. Naghdy, “Hidden Markov model-based autonomous manufacturing task orchestration in smart shop floors,” Robot. Comput.-Integr. Manuf., vol. 61, Feb. 2020, Art. no. 101845.

[48] J. Yu, S. Liang, D. Tang, and H. Liu, “A weighted hidden Markov model approach for continuous-state tool wear monitoring and tool life prediction,” Int. J. Adv. Manuf. Technol., vol. 91, nos. 1–4, pp. 201–211, Jul. 2017.

[49] W. Wei, X. Fan, H. Song, X. Fan, and J. Yang, “Imperfect information dynamic Stackelberg game based resource allocation using hidden Markov model for cloud computing,” IEEE Trans. Services Comput., vol. 11, no. 1, pp. 78–89, Jan. 2018.

[50] H. Wu, Z. Yu, and Y. Wang, “Real-time FDM machine condition monitoring and diagnosis based on acoustic emission and hidden Markov model,” Int. J. Adv. Manuf. Technol., vol. 90, nos. 5–8, pp. 2027–2036, May 2017.

[51] X. Han, C. Park, X. Han, K. R. Pattipati, D. L. Kleinman, and W. G. Kemple, “Hidden Markov model and auction-based formulations of sensor coordination mechanisms in dynamic task environments,” IEEE Trans. Syst., Man, Cybern. A, Syst. Humans, vol. 41, no. 6, pp. 1102–1106, Nov. 2011.
His research interests include transfer learning and object detection.

XINWEI LI received the Ph.D. degree in instrumentation science and technology from the School of Mechano-Electronic Engineering, Xidian University, Xi’an, China, in 2011. He is currently a Lecturer with the School of Electrical Engineering and Automation, Henan Polytechnic University, Jiaozuo, China. His research interests include multimedia signal processing and video.

LYUN BING received the Ph.D. degree in pattern recognition and intelligent system from the Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China, in 2017. He is currently a Lecturer with the School of Electrical Engineering and Automation, Henan Polytechnic University, Jiaozuo, China. His research interests include transfer learning and object detection.

YI YANG received the Ph.D. degree in control science and engineering from the School of Information Science and Engineering, East China University of Science and Technology, Shanghai, China, in 2015. He is currently a Lecturer with the School of Electrical Engineering and Automation, Henan Polytechnic University, Jiaozuo, China. His research interests include generative adversarial network and signal processing.

KEPING WANG received the Ph.D. degree in signal and information processing from the Beijing University of Posts and Telecommunications, Beijing, China, in 2011. She is currently an Associate Professor with the School of Electrical Engineering and Automation, Henan Polytechnic University, Jiaozuo, China. Her research interests include deep learning, reinforcement learning, and image process.

* * *

[59] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, “The graph neural network model,” IEEE Trans. Neural Netw., vol. 20, no. 1, pp. 61–80, Jan. 2009.

[60] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, “Graph neural networks: A review of methods and applications,” 2018, arXiv:1812.08434. [Online]. Available: http://arxiv.org/abs/1812.08434

[61] V. Zambaldi, D. Raposo, A. Santoro, V. Bapst, Y. Li, I. Babuschkin, K. Tuyls, D. Reichert, T. Lillicrap, E. Lockhart, M. Shanahan, V. Langston, R. Pascanu, M. Botvinick, O. Vinyals, and P. Battaglia, “Relational deep reinforcement learning,” 2018, arXiv:1806.01830. [Online]. Available: http://arxiv.org/abs/1806.01830

[62] A. Khan, E. Tolstaya, A. Ribeiro, and V. Kumar, “Graph policy gradients for large scale robot control,” 2019, arXiv:1907.03822. [Online]. Available: http://arxiv.org/abs/1907.03822

[63] V. Kurin, S. Godil, S. Whiteson, and B. Catanzaro, “Improving SAT solver heuristics with graph networks and reinforcement learning,” 2019, arXiv:1909.11830. [Online]. Available: http://arxiv.org/abs/1909.11830

[64] C. Chen, S. Hu, P. Nikdel, G. Mori, and M. Savva, “Relational graph learning for crowd navigation,” 2019, arXiv:1909.13165. [Online]. Available: http://arxiv.org/abs/1909.13165

[65] S. Geman and D. Geman, “Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images,” in Readings in Computer Vision. Amsterdam, The Netherlands: Elsevier, 1987, pp. 564–584.

[66] G. Zhao. DICE2D An Open Source DEM. Accessed: Jun. 2018. [Online]. Available: http://www.dembox.org/

[67] K. Sanlav and A. Nabi, “Learning complex group behaviours in a multi agent competitive environment,” Zenodo, Tech. Rep., Dec. 2018: doi: 10.5281/zenodo.2528551.

[68] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, “OpenAI gym,” 2016, http://www.openai.com.

[69] R. J. Williams, “Simple statistical gradient-following algorithms for connectionist reinforcement learning,” Mach. Learn., vol. 8, nos. 3–4, pp. 229–256, 1992.

[70] J. Besag, “Spatial interaction and the statistical analysis of lattice systems,” J. Roy. Stat. Soc., B, Methodol., vol. 36, no. 2, pp. 192–225, Jan. 1974.

[71] YI YANG (Member, IEEE) was born in Lichuan, Hubei, China. He received the B.S. degree in electrical engineering and automatization and the M.S. degree in control theory and control engineering from Henan Polytechnic University, Henan, China, in 2004 and 2007, respectively, and the Ph.D. degree in pattern recognition and intelligent system from the School of Automation Science and Electrical Engineering, Beihang University, Beijing, China, in 2017. He is currently a Lecturer with the School of Electrical Engineering and Automation, Henan Polytechnic University, Jiaozuo, China. His research interests include artificial intelligence, reinforcement learning, and adaptive dynamic programming and their application in coal mining.