Research Article

Distributed Policy Evaluation with Fractional Order Dynamics in Multiagent Reinforcement Learning

Wei Dai,¹ Wei Wang,² Zhongtian Mao,¹ Ruwen Jiang,¹ Fudong Nian,³ and Teng Li¹

¹Anhui Engineering Laboratory of Human-Robot Integration System and Intelligent Equipment, School of Electrical Engineering and Automation, Anhui University, Hefei 230601, China
²Center for Assessment and Demonstration Research, Academy of Military Science, Beijing 100091, China
³Hefei University, Hefei 230601, China

Correspondence should be addressed to Wei Wang; wwei009@mail.ustc.edu.cn

Received 15 July 2021; Revised 15 August 2021; Accepted 17 August 2021; Published 6 September 2021

Academic Editor: Zhenhua Tan

Copyright © 2021 Wei Dai et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The main objective of multiagent reinforcement learning is to achieve a global optimal policy. It is difficult to evaluate the value function with high-dimensional state space. Therefore, we transfer the problem of multiagent reinforcement learning into a distributed optimization problem with constraint terms. In this problem, all agents share the space of states and actions, but each agent only obtains its own local reward. Then, we propose a distributed optimization with fractional order dynamics to solve this problem. Moreover, we prove the convergence of the proposed algorithm and illustrate its effectiveness with a numerical example.

1. Introduction

In recent years, reinforcement learning [1] has received much attention from the society and succeeded remarkably in many areas such as machine learning and artificial intelligence [2]. As we all know, in reinforcement learning, an agent determines the optimal strategy under the feedback of rewards via constantly interacting with the environment. The function of the policy maps possible states to possible actions. Although reinforcement learning has made great achievements in single agent, it remains challenging in the application of multiagent [3]. The goal of the multiagent system is to enable several agents with simple intelligence but it is easy to manage and control to realize complex intelligence through mutual cooperation. While reducing the complexity of system modeling, the robustness, reliability, and flexibility of the system should be improved [4, 5].

In this paper, the objective of this paper is to investigate multiagent reinforcement learning (MARL), where each agent exchanges information with their neighbors in network systems [6]. All agents share the state space and action except local rewards. The purpose of the MARL is to determine the global optimal policy, and a feasible way is to construct a central controller, where each agent must exchange information with the central controller [7], which makes decisions for all of them. However, with the increase of state dimensions, the computation of the central controller becomes extremely heavy. The whole system would collapse if the central controller was attacked.

Then, we try to replace the centralized algorithm mentioned above with distributed control [8, 9]. Consistency protocol based on design enables all agents to achieve the same state [10–13]. In [14], Zhang et al. proposed a continuous-time distributed version of the gradient algorithm. As far as we know, most of the gradient methods use integer order iteration. In fact, fractional order has been developed for 300 years and used to solve many kinds of problems such as control applications and systems’ theory [15–17]. In comparison with the traditional integer order algorithm, the fractional order algorithm has more design freedom and potential to obtain better convergence performance [18, 19].

Hereinafter, the contributions of the paper are listed:
(1) We transform the multiagent strategy evaluation problem into a distributed optimization problem with a consensus constraint
(2) We construct the fractional order dynamics and prove the convergence of the algorithm
(3) We take a numerical example to verify the superiority of the proposed fractional order algorithm

The rest organization of this paper is listed as follows. Section 2 introduces some problems of formulation on MARL and fractional order calculus. Section 3 transforms the multiagent strategy evaluation problem into the optimization problem with a consensus constraint, proposes an algorithm with fractional order dynamics, and proves that the algorithm asymptotically converge to an exact solution. Section 4 presents a simulation example, and we summarize the work in Section 5.

2. Problem Formulation

2.1. Notations. Let \( \mathbb{R}, \mathbb{R}^n, \) and \( \mathbb{R}^{n \times m} \) represent the real number set, \( n \)-dimensional real column vector set, and \( n \times m \) real matrix set, respectively. \( AT \) represents the transpose of \( A \). \( \|A\| = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij}}^{1/2} \), \( \|X\|_{2} = \sqrt{X^{T}GX} \), and \( \langle A, B \rangle = A^{T}B \). \( \delta \) represents a multiagent Markov decision process (MDP), where \( \delta \) is the state space and \( \omega \) is the joint action space. \( P^{\omega} \) is the probability of transition from \( s_t \) to \( s_{t+1} \) when the agent takes the joint action \( a \) and \( P^{\omega} = E_{\pi \sim \pi}(P^{\omega}_{1s}) \), \( R(s, a) \) is the local reward when agent \( i \) takes joint action \( a \) at state \( s \) and \( \gamma \in (0, 1) \) is a discount parameter. \( \pi(a|s) \) represents the condition of probability when the agent takes joint action \( a \) at state \( s \). The reward function of agent \( i \) is defined when follows a joint policy \( \pi \) at state \( s \) as follows:

\[
R^{\pi}_{i}(s) = E_{\pi \sim \pi}[R_{i}(s, a)], \tag{1}
\]

where the right-hand side of the equation means that there is a probability for all possible choices of action \( a \), and we calculate the expected value for all rewards of agent \( i \):

\[
R^{\pi}_{i}(s) = \frac{1}{n} \sum_{i=1}^{n} R^{\pi}_{i}(s), \tag{2}
\]

where \( R^{\pi}_{i}(s) \) represents the average of the local rewards.

2.2. Graph theory. The graph is expressed as \( \mathcal{G} (\mathcal{V}, \mathcal{E}) \), where \( \mathcal{V} \) represents a graph, \( \mathcal{V} \) is the set of vertices, and \( \mathcal{E} \) is the set of edges in \( \mathcal{G} \). If any edge in the graph is undirected, the graph is named as undirected graph [20]. In graph, \( A = [a_{ij}] \in \mathbb{R}^{n \times m} \) is the adjacency matrix with \( a_{ij} \neq 0 \) if \( [i, j] \in \mathcal{E} \), \( a_{ij} = 0 \) otherwise. \( D = \text{diag}[d_{1}, d_{2}, \ldots, d_{n}] \) is the degree matrix with \( d_{i} = \sum_{j=1}^{n} a_{ij} \) and Laplacian matrix is \( L = D - A \). Moreover, if the graph is connected, \( L \) has the following two properties:

(1) Laplacian matrix is a semipositive definite matrix
(2) The minimum eigenvalue is 0 because the sum of every row of the Laplace matrix is 0

The minimum nonzero eigenvalue is defined as the algebraic connectivity of the graph.

Assumption 1. The undirected graph mentioned in the following text is connected.

\[
\begin{align*}
\frac{\partial x(\omega, t)}{\partial t} &= -\omega x(\omega, t) + u(t), \\
y(t) &= \int_{0}^{\infty} \mu_{a}(\omega) z(\omega, t) d\omega,
\end{align*}
\]

where \( \mu_{a} = \sin(\alpha t) / \omega^{a} \pi \).

Definition 1 (see [22]). The \( \alpha \)-th order Caputo derivative is

\[
\mathcal{D}^{\alpha} f(t) = \frac{1}{\Gamma(n - \alpha)} \int_{0}^{t} (t - \tau)^{n - \alpha - 1} f^{(n)}(\tau) d\tau, \tag{4}
\]

where \( \alpha \in (n - 1, n), n \in \mathbb{N}, \Gamma(t) = \int_{0}^{\infty} t^{n-1} e^{-t} dt \) is Gamma function, and \( f^{(n)}(t) \) is the \( n \)-th order derivative of \( f(t) \).

2.3. Policy Evaluation. To measure the benefits of agents in its current state, we establish the following value function, which represents the value of the cumulative return obtained by agents starting from the state \( s_t \), adopting a certain strategy \( \pi \):

\[
V^{\pi}(s) = E_{\pi} \left[ \sum_{t=1}^{\infty} \gamma^{t} R^{\pi}(s_{t}, a_{t}) \right], \tag{5}
\]

We construct Bellman equation based on \( V^{\pi} \in \mathbb{R}^{[S]} \) and \( R^{\pi} \in \mathbb{R}^{[S]}:

\[
V^{\pi} = R^{\pi} + \gamma P^{\pi} V^{\pi}. \tag{6}
\]

It is difficult to evaluate \( V^{\pi} \) directly if the dimension of the state space is very large. Therefore, we use \( V_{\theta}(s) = \phi^{T}(s) \theta \) to approximate \( V^{\pi} \), where \( \theta \in \mathbb{R}^{d} \) is the vector and \( \phi(s) : \mathcal{S} \rightarrow \mathbb{R}^{d} \), which is a particular function for state \( s \). Indeed, solving equation (6) is equivalent to obtain the vector \( \theta \) via \( V_{\theta} \approx V^{\pi} \). In other words, it means to minimize the mean square error about \( 1/2 \| V_{\theta} - V^{\pi} \|_{D}^{2} \), where \( D = \text{diag}[d_{s}, s \in S] \), \( e \in \mathbb{R}^{[S][S]} \) is a diagonal matrix determined by the stationary distribution. We construct the equation as follows:

\[
f(\theta) = \frac{1}{2} \| \Pi_{\Phi}(V_{\theta} - \gamma P^{\pi} V_{\theta} - R^{\pi}) \|_{D}^{2} + \frac{\rho}{2} \| \theta \|^{2}, \tag{7}
\]

where \( \rho \) is a regularization parameter and \( \Pi_{\Phi} \) is a projection operator in the column subspace of \( \Phi \). It is not difficult to rewrite \( \Pi_{\Phi} \) as \( \Phi(\Phi^{T} \Phi)^{-1} \Phi^{T} D \) substituting \( \Pi_{\Phi} \) into (7):
\[ f(\theta) = \frac{\rho}{2}||\theta||^2 + \frac{1}{2}||\Pi_\Phi (V_0 - \gamma P^T V_0 - R^c_v)||^2 \]

\[ = \frac{\rho}{2}||\theta||^2 + \frac{1}{2}((V_0 - \gamma P^T V_0 - R^c_v)^T \Pi_\Phi D \Pi_\Phi D^T (V_0 - \gamma P^T V_0 - R^c_v)) \]

\[ = \frac{\rho}{2}||\theta||^2 + \frac{1}{2}((V_0 - \gamma P^T V_0 - R^c_v)^T D \Phi (\Phi^T D \Phi)^{-1} \Phi^T D (V_0 - \gamma P^T V_0 - R^c_v)) \]

\[ = \frac{\rho}{2}||\theta||^2 + \frac{1}{2}||D \Phi (\Phi^T D \Phi)^{-1} \Phi^T D|| \]

\[ = \frac{\rho}{2}||\theta||^2 + \frac{1}{2}||D \Phi (\Phi^T D \Phi)^{-1} \Phi^T D|| \]

\[ = \frac{\rho}{2}||\theta||^2 + \frac{1}{2}||A\theta - b||_{C^{-1}} \]

where \( A = \Phi^T D (\Phi - \gamma P^T \Phi) \epsilon E_{p+p}, [\phi(s) (\phi(s) - \gamma \phi(s'))^T], \)

\( C = \Phi^T D \Phi \epsilon E_{p+p}, [\phi(s) \phi^T (s)], \) and \( b = \Phi^T D R^c_v = E_{p+p} [R^c_v (s)(s)]. \)

The minimum value of \( \theta \) in equation (8) is unique if \( A \) is a full rank matrix and \( C \) is a positive definite matrix. In practice, it is difficult to get the expectations in the compact form when the distribution is unknown. We replace expectation with the average as follows:

\[ \bar{A} = \frac{1}{P} \sum_{i=1}^{P} A_i, \]

\[ \bar{b} = \frac{1}{P} \sum_{i=1}^{P} b_i, \]

\[ \bar{C} = \frac{1}{P} \sum_{i=1}^{P} C_i, \]

where \( A_i = \phi(s_i) \phi^T (s_i), \phi(s_i) \phi(s_i - 1), \) \( C_i = \phi(s_i) \phi^T (s_i), \) and \( b_i = R^c_v (s)(s). \)

We assume that the sample size \( P \) approaches infinity to make sure its confidence level. In these sequences, each state is attached at least once. Then, we reconstruct equation (8) as follows:

\[ f(\theta) = \frac{1}{2}||\bar{A} \theta - \bar{b}||_{C^{-1}}^2 + \frac{\rho}{2}||\theta||^2. \]

Noteworthy, in a shared space, the agent observes the states and actions of the neighbors, but only observes the local rewards of its own. In other words, we get \( \bar{A} \) and \( \bar{C} \) except \( \bar{b} \). So, we define \( \bar{b} = (1/p) \sum_{i=1}^{P} b_i \), with \( b_{i,j} = R^c_v (s_i, a_i) \phi(s_i). \) Then, we rewrite equation (10) as follows:

\[ \min_{\theta \epsilon \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^{n} ||\bar{A} \theta - \bar{b}||_{C^{-1}}^2 + \frac{\rho}{2}||\theta||^2. \]

Hereinafter, the aim of policy evaluation becomes to minimize the object function. Now, we rewrite (11) as follows:

\[ \begin{align*}
\min_{\theta} & \quad \frac{1}{n} \sum_{i=1}^{n} ||\bar{A} \theta - \bar{b}||_{C^{-1}}^2 + \frac{\rho}{2}||\theta||^2, \\
\text{s.t.} & \quad \theta = \theta_i.
\end{align*} \]

We define \( \bar{\theta} \epsilon \mathbb{R}^n \) as a factor concatenating all \( \theta_i \); \( \bar{\theta} = [ \theta_1^T, \theta_2^T, \ldots, \theta_n^T]^T \epsilon \mathbb{R}^n \) and the aggregative function \( f \) as \( f(\bar{\theta}) = \sum_{i=1}^{n} f(\theta_i) \). As we all know, the consensus constraint (12) is expressed as

\[ \begin{align*}
\min_{\bar{\theta}} & \quad \frac{1}{2}||\bar{A} \bar{\theta} - \bar{b}||_{C^{-1}}^2 + \frac{\rho}{2}||\bar{\theta}||^2, \\
\text{s.t.} & \quad \bar{1}_{m} = 0,
\end{align*} \]

where \( \bar{b} = [ \bar{b}_1^T, \bar{b}_2^T, \ldots, \bar{b}_n^T]^T \epsilon \mathbb{R}^n, \) \( \bar{L} = L \otimes I_d \epsilon \mathbb{R}^{n \times d \times n}, \) \( \bar{A} = \bar{A} \otimes I_d \epsilon \mathbb{R}^{d \times n \times d}, \) and \( \bar{C} = \bar{C} \otimes I_d \epsilon \mathbb{R}^{d \times n \times d}. \) Based on (13), we formulate the following the augmented Lagrangian:

\[ \mathcal{L}(\bar{\theta}, \lambda) = f(\bar{\theta}) + \langle \lambda, \bar{1}_{m} \rangle + \frac{1}{2}||\bar{\theta}||_{T^\lambda}^2, \]

where \( \lambda \epsilon \mathbb{R}^n \) is the Lagrange multiplier.

It is feasible to design a fractional order continuous-time optimization algorithm from primal-dual viewpoint, gradient descend for primal variable \( \bar{\theta} \), and gradient ascent for dual variable \( \lambda \) via (14). Both of them are updated according to the fractional order law:

\[ \begin{align*}
\mathcal{D}_t^\alpha \bar{\theta}(t) &= -\nabla_{\bar{\theta}} \mathcal{L}(\bar{\theta}(t), \lambda(t)), \\
\mathcal{D}_t^\alpha \lambda(t) &= \nabla_{\lambda} \mathcal{L}(\bar{\theta}(t), \lambda(t)),
\end{align*} \]
where \(0<\alpha_1<2, \ 0<\alpha_2<1, \ \nabla^\alpha_1 \phi(t), \lambda(t), \) and \(\nabla^\lambda_1 \phi(t), \lambda(t)\) are gradient of \((\bar{\phi}(t), \lambda(t))\) on variables \(\bar{\phi}(t)\) and \(\lambda(t),\) respectively. We express the detail of (15) in Algorithm 1.

The aim of the distributed algorithm is to obtain the solution of the value function. The proposed algorithm has more potential to get better convergence performance and design freedom than the conventional integer order. Hereinafter, we provide the following convergence conclusion.

**Theorem 1.** Under Assumption 1, let \(\bar{\phi}(t)\) and \(\lambda(t)\) be generated according to Algorithm 1. If \(0<\alpha_1, \alpha_2<1,\) then \(\bar{\phi}(t)\) asymptotically converges to the optimal solution.

**Proof.** We obtain the detailed dynamics of \(\bar{\phi}(t)\) and \(\lambda(t):\)

\[
\begin{align*}
\mathcal{D}^\alpha_1 \bar{\phi}(t) &= -\left(\nabla^T \bar{\phi} - \lambda + 1\right) \bar{\phi}(t) + \lambda_1 - \lambda, \\
\mathcal{D}^\lambda \lambda(t) &= \Gamma_\lambda(\lambda(t)),
\end{align*}
\]

(16)

where \(I\) is an identity matrix. We consider the equilibrium of (16):

\[
\begin{align*}
0 &= -\left(\nabla^T \bar{\phi} - \lambda + 1\right) \bar{\phi} + \lambda_1 - \lambda, \\
0 &= \Gamma_\lambda(\lambda).
\end{align*}
\]

(17)

Then, we combine (16) and (17), and according to the facts \(\mathcal{D}^\alpha_1 \bar{\phi}(t) = 0, \ \mathcal{D}^\lambda \lambda(t) = 0,\)

\[
\begin{align*}
\mathcal{D}^\alpha_1 \bar{\phi}(t) &= -\left(\nabla^T \bar{\phi} - \lambda + 1\right) \bar{\phi}(t) - \lambda, \\
\mathcal{D}^\lambda \lambda(t) &= \Gamma_\lambda(t).
\end{align*}
\]

(18)

Through Lemma 1, we reconstruct (18) as follows:

\[
\begin{align*}
\frac{\partial \bar{\phi}}{\partial t}(\omega, t) &= -\omega \bar{z}_1(\omega, t) - \left(\nabla^T \bar{\phi} - \lambda + 1\right) \bar{\phi}(t) - \lambda, \\
\bar{\lambda}(t) &= \int_0^\infty \mu_{\alpha_2} z_2(\omega, t) d\omega.
\end{align*}
\]

(19)

and

\[
\begin{align*}
\frac{\partial z_2}{\partial t}(\omega, t) &= -\omega z_2(\omega, t) + \Gamma_\lambda(t), \\
\lambda(t) &= \int_0^\infty \mu_{\alpha_2} z_2(\omega, t) d\omega.
\end{align*}
\]

(20)

We construct the Lyapunov function as follows:

\[
V_1 = \frac{1}{2} \int_0^{\infty} \sum_{i=1}^2 \mu_{\alpha_1} \|z_i(\omega, t)\|^2 d\omega.
\]

(21)

Then,

\[
V_1 = \int_0^{\infty} \sum_{i=1}^2 \mu_{\alpha_1} \left(\frac{\partial z_i}{\partial t}\right) d\omega,
\]

(22)

We obtain the result according to the Lasalle invariance principle.

Hereinafter, we improve the convergence conclusion of Theorem 1 by extending \(\alpha_1\) from (0,1) to (1,2).

**Theorem 2.** Under Assumption 1, let \(\bar{\phi}(t)\) and \(\lambda(t)\) be generated according to Algorithm 1. If \(1<\alpha_1<2, \ \alpha_1 + \alpha_2 = 2,\) then \(\bar{\phi}(t)\) asymptotically converges to the optimal solution.

**Proof.** Under the condition \(\alpha_1 = 1 + \alpha_2,\) we rewrite the dynamics with the condition of Theorem 1 as follows:

\[
\begin{align*}
\hat{\phi}(t) &= -\left(\nabla^T \bar{\phi} - \lambda + 1\right) \bar{\phi}(t) - \lambda, \\
\hat{\lambda}(t) &= \mathcal{D}^\alpha \mathcal{D}^\lambda \hat{\lambda}(t) = \mathcal{D}^\alpha \left[\Gamma_\lambda(t)\right] = \Gamma_\lambda(t).
\end{align*}
\]

(23)

Due to \(\alpha_1 = 1 + \alpha_2\) and \(\alpha_1 + \alpha_2 = 2,\)

\[
\begin{align*}
\hat{\phi}(t) &= -\left(\nabla^T \bar{\phi} - \lambda + 1\right) \bar{\phi}(t) - \lambda, \\
\hat{\lambda}(t) &= \mathcal{D}^\alpha \mathcal{D}^\lambda \hat{\lambda}(t) = \mathcal{D}^\alpha \left[\Gamma_\lambda(t)\right] = \Gamma_\lambda(t).
\end{align*}
\]

(24)

Under the condition of (23) and (24), we obtain the frequency distributed model by Lemma 1 as follows:
Initializations: \( \theta_0 = \mathbf{0} \in \mathbb{R}^d \), \( \lambda_0 = \mathbf{0} \in \mathbb{R}^d \).

Update

For \( t \leq 50 \)

\[
\begin{pmatrix}
\dot{\theta}_a (t) = - (\tilde{\mathbf{A}}^T \tilde{\mathbf{C}}^{-1} \tilde{\mathbf{b}} + \rho \mathbf{I}) \theta_a (t) - \tilde{\mathbf{A}}^T \tilde{\mathbf{C}}^{-1} \theta_i (t) - \tilde{\mathbf{A}}^T \tilde{\mathbf{C}}^{-1} \tilde{\mathbf{b}}_i + \tilde{\mathbf{L}} \lambda (t) + \tilde{\mathbf{L}} \tilde{\lambda} (t), \\
\partial \theta_i (t, \omega, t) = - \omega \theta_i (t, \omega, t) + \tilde{\theta}_a (t), \\
\tilde{\theta} (t) = \int_0^\infty \mu \theta_i (\omega, t) d\omega, \\
\tilde{\lambda} (t) = \tilde{\mathbf{L}} \tilde{\theta} (t).
\end{pmatrix}
\]

We construct the Lyapunov function:

\[
V_2 = \frac{1}{2} \| \tilde{\theta}_a (t) \| ^2 + \frac{1}{2} \| \tilde{\lambda} (t) \| ^2.
\]

Then,

\[
V_2 = \langle \tilde{\theta}_a (t), \tilde{\theta}_a (t) \rangle + \langle \tilde{\lambda} (t), \tilde{\lambda} (t) \rangle
\]

\[
= \langle \tilde{\theta}_a (t), - (\tilde{\mathbf{A}}^T \tilde{\mathbf{C}}^{-1} \tilde{\mathbf{b}} + \rho \mathbf{I}) \tilde{\theta}_a (t) - \tilde{\mathbf{A}}^T \tilde{\mathbf{C}}^{-1} \tilde{\theta}_a (t) - \tilde{\mathbf{A}}^T \tilde{\mathbf{C}}^{-1} \tilde{\mathbf{b}}_i + \tilde{\mathbf{L}} \tilde{\lambda} (t) + \tilde{\mathbf{L}} \tilde{\lambda} (t) \rangle
\]

\[
= - \tilde{\theta}_a (t) ^ T (\tilde{\mathbf{A}}^T \tilde{\mathbf{C}}^{-1} \tilde{\mathbf{b}} + \rho \mathbf{I} + \tilde{\mathbf{L}}) \tilde{\theta}_a (t) \leq 0.
\]

Algorithm 1

Before the simulation, it is necessary to obtain the solution of the multiagent reinforcement learning:

\[
\tilde{\theta}^* = \begin{bmatrix}
0.9797 & 0.5949 & 0.1174 & 0.0855 & 0.7303 \\
0.4389 & 0.2622 & 0.2967 & 0.2625 & 0.4886 \\
0.1111 & 0.6028 & 0.3188 & 0.8010 & 0.5785 \\
0.2581 & 0.7112 & 0.4242 & 0.0292 & 0.2373 \\
0.4087 & 0.2217 & 0.5079 & 0.9289 & 0.4588
\end{bmatrix}.
\]

In this section, we provide an example to illustrate the effectiveness of the proposed algorithm. There are 20 states in the multiagent reinforcement learning. We set \( d = 5 \), regularization parameter \( \rho = 0.1 \), and discount parameter \( \gamma = 0.5 \). There are 4 agents in the connected network in Figure 1. State \( s \) is a randomly generated 5-dimensional column vector, the dimension of \( \phi (s) \) is a cosine function, and \( P \) is a randomly generated 5-dimensional matrix.

Then, we randomly generate the matrices \( \tilde{\mathbf{A}}, \tilde{\mathbf{C}}, \tilde{\mathbf{b}}_i \) as follows:

4. Experimental Simulation

Before the simulation, it is necessary to obtain the solution of the multiagent reinforcement learning:

\[
\tilde{\theta}^* = \begin{bmatrix}
0.9797 & 0.5949 & 0.1174 & 0.0855 & 0.7303 \\
0.4389 & 0.2622 & 0.2967 & 0.2625 & 0.4886 \\
0.1111 & 0.6028 & 0.3188 & 0.8010 & 0.5785 \\
0.2581 & 0.7112 & 0.4242 & 0.0292 & 0.2373 \\
0.4087 & 0.2217 & 0.5079 & 0.9289 & 0.4588
\end{bmatrix}.
\]

We show the comparison about the fractional order algorithm with the conventional integer order one. In Figures 2 and 3, the curve illustrates almost the same convergence performance as the conventional integer order when \( \alpha \) is 0.995. In Figures 4 and 5, the fractional order algorithm achieves a faster convergent rate than that of the integer order algorithm. Simulation results illustrate the convergence about the integer order and the fractional order. Furthermore, the proposed distributed
algorithm with fractional order dynamics has more design freedom to achieve a better performance than that of the conventional first-order algorithm.

5. Conclusion

In this paper, the value function problem of the multiagent reinforcement learning was transformed as a distributed optimization problem with a consensus constraint. Then, we proposed a distributed algorithm with fractional order dynamics to solve this problem. Besides, we proved the asymptotic convergence of the algorithm by Lyapunov functions and illustrated the effectiveness of the proposed algorithm with an example. In the future, we will consider applying reinforcement learning to the recommendation system, so as to get better results [23].

Data Availability

The .m and .slx data used to support the findings of this study have been deposited in the Github repository (97weiD/data_DPEFOD).
Conflicts of Interest
The authors declare that they have no conflicts of interest.

Acknowledgments
This work was supported by the National Natural Science Foundation of China (61973002 and 61902104), and the Anhui Provincial Natural Science Foundation (2008085J32 and 2008085QF295).

References