Hindawi
Complexity
Volume 2018, Article ID 7172614, 14 pages
https://doi.org/10.1155/2018/7172614

Research Article

EAQR: A Multiagent Q-Learning Algorithm for Coordination of Multiple Agents

Zhen Zhang and Dongqing Wang

School of Automation and Electrical Engineering, Qingdao University, Qingdao 266071, China

Correspondence should be addressed to Zhen Zhang; tbsunshine8@163.com

Received 3 April 2018; Revised 12 June 2018; Accepted 24 July 2018; Published 28 August 2018

Academic Editor: Wenbo Wang

We propose a cooperative multiagent Q-learning algorithm called exploring actions according to Q-value ratios (EAQR). Our aim is to design a multiagent reinforcement learning algorithm for cooperative tasks where multiple agents need to coordinate their behavior to achieve the best system performance. In EAQR, Q-value represents the probability of getting the maximal reward, while each action is selected according to the ratio of its Q-value to the sum of all actions' Q-value and the exploration rate ε. Seven cooperative repeated games are used as cases to study the dynamics of EAQR. Theoretical analyses show that in some cases the optimal joint strategies correspond to the stable critical points of EAQR. Moreover, comparison experiments on stochastic games with finite steps are conducted. One is the box-pushing, and the other is the distributed sensor network problem. Experimental results show that EAQR outperforms the other algorithms in the box-pushing problem and achieves the theoretical optimal performance in the distributed sensor network problem.

1. Introduction

Reinforcement learning (RL) uses a scalar numeric feedback from the environment to improve the behavior of the learner. In the case with only one agent, RL is an effective unsupervised learning method to solve problems with the Markov property [1, 2]. Many researchers have been trying to extend RL to optimize performance indices in circumstances where multiple agents exist and a lot of multiagent reinforcement learning (MARL) algorithms, and their applications have been proposed [3–5]. In a multiagent system (MAS), on one hand, the state transition distribution and the local immediate reward received by each agent are not determined by the behavior of any single agent but the behavior of all the agents in the system. Thus, each agent has to adapt to the environment and the other agents at the same time, which leads to the invalidity of the Markov property. On the other hand, if all the agents in the system are viewed as a single one, the joint action space will grow exponentially, which deteriorates the scalability of MARL algorithms.

This paper investigates methods coordinating multiple agents through MARL techniques. In recent years, many MARL algorithms with different assumptions and goals have been presented to solve coordination issues in MAS. Some algorithms require sharing each agent's local immediate reward; some algorithms require sharing each agent's selected actions and even value functions or Q-value functions as well. The learning goal depends on the problems at hand. Nash equilibria have been used in optimal control [6, 7] and are also adopted as the learning goal by many MARL algorithms. Hu and Wellman [8] proposed Nash-Q which could converge to a Nash equilibrium in some repeated games. However, Nash-Q needed Q-value functions to be shared as well. Infinitesimal gradient ascent (IGA) [9] was proposed and guaranteed that the agents' strategies would converge to a Nash equilibrium, or average rewards would converge to the expected rewards of a Nash equilibrium in two-player two-action repeated games. Win-or-learn fast policy with IGA (WoLF-IGA) [10] was proposed to address the issue that IGA would not converge to any Nash equilibrium in some repeated games. For IGA and WoLF-IGA, each agent has to know its own payoff matrix and the other agent's strategy. Besides, Nash-Q, IGA, and WoLF-IGA would suffer the curse of dimensionality for joint action space.

To mitigate the above problems, some algorithms with less requirement of sharing were studied. WoLF-policy hill-climbing (WoLF-PHC) [10] only needed to share states and local immediate rewards of each agent, but the convergence property was not guaranteed any more. The exponential moving average (EMA) Q-learning [11] and the weighted policy learner (WPL) [12] empirically converged to a Nash equilibrium in some typical repeated games. To design scalable MARL algorithms that can gain the optimal total sum of reward in fully cooperative games is our motivation.

New MARL algorithms can be obtained by designing new action exploration method. Babes et al. [13] pointed out that more robust algorithms could be produced by inserting tools from nonlinear dynamics into Q-learning to modify the exploration or learning rate. So far, the dynamics of independent Q-learning (IQL) in two-player two-action repeated games have been extensively studied. Tuyls and Nowé, Tuyls and Parsons, and Bloembergen et al. [14–16] firstly built the model of IQL with Boltzmann exploration in three typical repeated games. They pointed out that the IQL model was similar with dynamic replication equations, and they presented graphical representation of the relation between the temperature parameter T and the critical points. Kianercy and Galstyan [17] further studied the dynamics of IQL. They analyzed the position and the stability of the critical points of IQL in some types of two-player two-action repeated games. Babes et al. [13] analyzed the dynamics of IQL with ε-greedy exploration. For ε-greedy exploration, since the action with the maximal Q-value will be selected for exploitation, the Q-value can be viewed as the switching signal. Thus results on stability analysis for switching systems [18, 19] might be beneficial to the analysis of IQL with ε-greedy exploration. Awheda and Schwartz [11] proposed EMA Q-learning and proved its ability to converge to a Nash equilibrium in two-player two-action games.

Nash equilibrium is important when analyzing the interaction between agents. Some multiagent reinforcement (MARL) algorithms do focus on convergence on Nash equilibrium, and most of these algorithms consider general sum games. In contrast, for cooperative tasks, reaching better performance indices is more important than converging to Nash equilibrium and becomes the prime concern for MARL algorithms.

To obtain the maximal expected total cumulative reward, this paper proposes a multiagent Q-learning algorithm called exploring actions according to Q-value ratios (EAQR). In standard fictitious play [20], each player's strategy is a function of the other players' empirical frequency, while in EAQR, each agent selects an action according to its Q-value function of its own actions and updates its Q-value function only according to the frequency of its own action selection. The maximal total immediate reward can still be achieved in some cooperative repeated games, which is the first contribution. The second contribution is that EAQR can be naturally extended to apply to stochastic games. Simulation results show that EAQR outperforms the other algorithms in the box-pushing problem and achieves the theoretical optimal performance in the distributed sensor network problem.

The remainder of this paper is organized as follows. Section II introduces stochastic games and repeated games. Section III proposes EAQR in repeated games. Section IV studies the dynamics of EAQR in seven different repeated games which are analyzed. Section V compares EAQR with EMA Q-learning, WoLF-PHC, and single-agent RL in two stochastic games—box-pushing and the DSN problem. Section VI summarizes the conclusions.

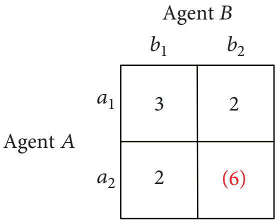
2. Preliminaries of Stochastic Games and Repeated Games

2.1. Stochastic Games. A stochastic game [5] is a tuple $<S, A_1, A_2, \dots, A_n, p, r_1, r_2, \dots, r_n>$, where n is the number of agents in the game; S is the set of environment states; A_i is the set of agent i's available actions; and A_i for all agents i constitutes the joint action set $A = A_1 \times A_2 \times \cdots \times A_n$; the state transition function $p : S \times A_1 \times A_2 \times \cdots \times A_n \times S \rightarrow [0, 1]$ is a conditional probability determining the probability of transiting to the next state s' if the joint action $a \in A$ has been executed in the current state s, and $r_i : S \times A_1 \times A_2 \times \cdots \times A_n \times S \rightarrow \mathbf{R}$ is the local immediate reward function of agent i. The global immediate reward function is the sum of local immediate reward functions of all agents and is defined as $r = \sum_{i=1}^{n} r_i$. In cooperative MAS, the learning objective is to maximize the discounted global cumulative reward at each time t,

$$R(t) = r(t+1) + \gamma r(t+2) + \gamma^2 r(t+3) + \cdots = \sum_{k=0}^{K} \gamma^k r(t+k+1), \tag{1}$$

where γ is the discount factor within (0, 1) (smaller γ values correspond to a greater importance of near future rewards); K is the ending time of an episode; and $r(t+1)$ is the global immediate reward received at time $t+1$.

2.2. Repeated Games. There still exists interacting between agents, although this paper focuses on optimization problem. Repeated game is an ideal tool to depict interaction and build the model of EAQR. In a repeated game, the set of state is null. Each agent's local immediate reward depends solely on the joint action. In a fully cooperative repeated game, we are concerned with only the global immediate reward representing the team benefit. Figure 1 shows the payoff matrix of a two-player two-action game. Each row represents an action of agent A, and each column represents an action of agent B. Each element of the payoff matrix is a numerical global immediate reward. For example, if agent A chooses action a_1 while agent B chooses action b_2, then both agents will receive a global immediate reward of 2. The optimal global immediate reward of 6 is marked with parentheses.



		Agent B	
		b_1	b_2
Agent A	a_1	3	2
	a_2	2	(6)

Figure 1: The payoff matrix of a fully cooperative repeated game.

3. EAQR: A Multiagent Q-Learning Algorithm for Coordination of Multiple Agents

EAQR is designed for optimizing performance indices of fully cooperative MAS. EAQR requires each agent to have full observation of states and local immediate reward of all agents. One merit of EAQR is that each agent does not need to observe any other agent's action. Thus the size of the Q-table maintained by agent i is $|S| \times |Ai|$. Sharing local immediate reward is to achieve the optimal global immediate reward. EAQR manages to converge to (a_2, b_2) in Figure 1 through the procedures depicted in Algorithm 1. For agent i, the probability of selecting action j is

$$p_j(t) = \begin{cases} \dfrac{Q_j(t)}{\sum_{k \in A_i} Q_k(t)} & \text{with probability } 1 - \varepsilon, \\ \dfrac{1}{|A_i|} & \text{with probability } \varepsilon, \end{cases} \qquad (2)$$

where $Q_j(t)$ is the probability of obtaining the maximum global immediate reward by taking action j at time t; the exploration rate ε is within (0,1); A_i is the action set of agent i; $|A_i|$ is the number of available actions for agent i. The nonnegativity of $Q_j(t)$ can be guaranteed by setting the learning rate α and the initial value of $Q_j(t)$ to a positive value within (0, 1). $\sum_{k \in A_i} Q_k(t)$ will be strictly greater than zero if each action is visited by infinite times. To avoid being divided by zero in practical application, we randomly select an action if $\sum_{k \in A_i} Q_k(t) = 0$. In EAQR, the exploration rate ε balances exploration and exploitation. When $\varepsilon = 0$, $p_j(t)$ is equal to the ratio of $Q_j(t)$ to $\sum_{k \in A_i} Q_k(t)$. When $\varepsilon = 1$, a random action is selected according to the uniform distribution.

In a repeated game, all agents keep their strategies unchanged and play the game for N_s times. Then they update the Q-value of each action j according to

$$Q_j(t+1) = Q_j(t) + \alpha\left(f_j(t) - Q_j(t)\right), \qquad (3)$$

where $\alpha \in (0, 1)$ is the learning rate; $f_j(t)$ is the frequency of obtaining the maximum global immediate reward by taking action j. It is evaluated according to

$$f_j(t) = \frac{n_{\max\_j}(t)}{n_j(t)}, \qquad (4)$$

where $n_j(t)$ is the number of times for agent i selecting action j during the previous N_s games, and $n_{\max\_j}(t)$ is the number of times for agent i achieving the maximum global immediate reward in history when selecting action j during the previous N_s games. Before playing the next N_s games, all agents need to update their strategies according to (3).

In stochastic games with deterministic state transition, a state can be viewed as a repeated game, and the elements of the payoff matrix are cumulative rewards if EAQR can converge to a joint action at each of its subsequent states. In this situation, the frequency of obtaining the maximum global cumulative reward by taking action j in each state is to be evaluated to update Q-value functions. In stochastic games with nondeterministic state transition, each state cannot be simply regarded as a repeated game. Yet we can try to treat each state in an optimistic way (The frequency of maximal global cumulative reward instead of the average global reward is concerned) and employ the same Q-value updating rule.

4. Dynamics of EAQR in Cooperative Repeated Games

In this section, the dynamics of EAQR in seven cooperative repeated games are analyzed. A theorem about the dynamics of EAQR is presented, and seven cases of repeated games are analyzed. If the updating of Q-value function is regarded as a continuous process, the EAQR can be modeled with differential equations. According to [14, 17], the continuous-time form of Q-value updating rule of EAQR can be obtained as follows:

$$\dot{Q}_j(t) = \alpha\left(f_j(t) - Q_j(t)\right). \qquad (5)$$

After rescaling $t \to \alpha t$, we can obtain the following:

$$\dot{Q}_j(t) = f_j(t) - Q_j(t). \qquad (6)$$

Theorem 1. *For a cooperative repeated game with n $(n \geq 3)$ players and m optimal pure joint strategies, if for any optimal pure joint strategy each of its component actions is different from the corresponding component action of the other optimal pure joint strategies, then only the m optimal pure joint strategies are the stable critical points of the model of EAQR with $\varepsilon = 0$.*

Proof 1. a_j^i is used to denote player i's component action of the optimal pure joint strategy j, and Q_j^i is used to denote the Q-value of a_j^i for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$. For any optimal pure joint strategy, each of its component actions is different from the corresponding component action of the other optimal pure joint strategies, which is saying that $\forall p \neq q$, a_p^i, and a_q^i should not be the same action for player i for $i = 1, 2, \ldots, n$. According to (6), the Q-value of actions that can never reach the optimal global reward will

```
for each agent $i$, do
    initialize $Q(a_i)$ with a number within (0,1) for $\{a_i \mid a_i \in A_i\}$,
    initialize $\varepsilon$ with a number within (0,1)
    $f(a_i) = 0$: frequency of getting the maximum global immediate reward after selecting action $a_i$
    sampleGameCnt = 0: number of sample games played
    repeat for each game
        select an action $a_i$ with the probability of
        $p(a_i) = \begin{cases} Q(a_i)/\sum_{a \in A_i} Q(a) & \text{with probability } 1-\varepsilon \\ 1/|A_i| & \text{with probability } \varepsilon \end{cases}$
        sampleGamesCnt = sampleGamesCnt + 1
    execute action $a_i$, update information about reward
        if sampleGamesCnt = $N_s$ then
            for each action $a_i \in A_i$ do
                evaluate $f(a_i)$ according to (4)
                $Q(a_i) = Q(a_i) + \alpha(f(a_i) - Q(a_i))$
            end for each action
            sampleGamesCnt = 0
        end if
     until the predefined number of games have been played
end for each agent
return Q-value function for each agent
```

Algorithm 1: EAQR for repeated games.

decrease to zero. Then the model of EAQR with $\varepsilon = 0$ can be expressed by the following equations.

$$\dot{Q}_j^i = \left(\prod_{k=1(k \neq i)}^{n} \frac{Q_j^k}{\sum_{l=1}^{m} Q_l^k} \right) - Q_j^i, \tag{7}$$

for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$. If $m = 1$, it can be proved trivially that there is only one stable critical point which is the optimal pure joint strategy. It can be obtained from (7) that the critical points have to satisfy

$$\prod_{k=1(k \neq i)}^{n} \frac{Q_j^k}{\sum_{l=1}^{m} Q_l^k} = Q_j^i, \tag{8}$$

for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$. It can be further obtained that $Q_j^1 = Q_j^2 = \cdots = Q_j^n$ for $j = 1, 2, \ldots, m$ at the critical point. Suppose $Q_j^i / \sum_{k=1}^{m} Q_k^i = p_j$ for $j = 1, 2, \ldots, m$ at the critical point, the following can be obtained according to (8):

$$p_j = \frac{\left(p_j\right)^{n-1}}{\sum_{k=1}^{m} \left(p_k\right)^{n-1}}, \tag{9}$$

for $j = 1, 2, \ldots, m$. It can be seen that the value of p_j can only be 1, 0, or $1/m$. $p_j = 1$ and $p_j = 0$ correspond to the m optimal pure joint strategies, while $p_j = 1/m$ corresponds to a mixed strategy. Thus the critical points include all the m optimal pure joint strategies and the strategy equally choosing an action that has reached the optimal global reward, namely, $Q_j^i = (1/m)^{n-1}$ for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$. The stability of the critical points can be judged by the eigenvalues λ of the Jacobin matrix J.

For the m optimal pure joint strategies, the determinant of $J - \lambda I$ can be expanded according to rows and columns step by step. The following can be obtained:

$$\det(J - \lambda I) = (\lambda + 1)^{mn}. \tag{10}$$

All eigenvalues are -1 which is negative. Thus the m optimal pure joint strategies are stable critical points.

For the mixed strategy, we just need to transform the determinant of $J - \lambda I$ and extract a common factor of it for $n = 2k + 2$ and $n = 2k + 1$, $k \in Z^+$, respectively. Although the transformation processes are different in the two cases, the following can be obtained for both the cases:

$$\det(J - \lambda I) = [\lambda - (n-2)] f(\lambda), \tag{11}$$

where $f(\lambda)$ is a polynomial of λ of degree $mn - 1$. Thus, there always exists at least one positive eigenvalue $\lambda = n - 2$ when $n \geq 3$, which means the mixed strategy is unstable. Thus, only the m optimal pure joint strategies are the stable critical points of the model of EAQR.

Cases 1–4 are two-player two-action repeated games. Case 5 and case 6 are two-player three-action repeated games. Case 7 is a three-player two-action repeated game. The corresponding payoff matrices are displayed in Figures 2–4, respectively. The numeric number represents the global immediate reward. The optimal global immediate reward is displayed in parentheses. In all cases, p_j^i represents the probability of obtaining the maximum global immediate reward when player i chooses action j. Q_j, P_j, and K_j represent the Q-value of action j for player 1, 2, and 3, respectively.

b_{11}	b_{12}
b_{21}	(a)

Case 1

(a)	b_{12}
b_{21}	(a)

Case 2

(a)	(a)
b_{21}	b_{22}

Case 3

(a)	(a)
(a)	b_{22}

Case 4

Figure 2: The payoff matrices of Cases 1–4.

(a)	b_{12}	b_{13}
b_{21}	(a)	b_{23}
b_{31}	b_{32}	(a)

Case 5

b_{11}	(a)	b_{13}
(a)	b_{22}	(a)
b_{31}	(a)	b_{33}

Case 6

Figure 3: The payoff matrices of Case 5 and Case 6.

If player 3 chooses action 1

(a)	b_{12}
b_{21}	b_{22}

If player 3 chooses action 2

c_{11}	c_{12}
c_{21}	c_{22}

Case 7

Figure 4: The payoff matrix of Case 7.

In Cases 1–4, player 1 and player 2 are literally the row player and the column player, respectively. We assume that a matrix B exists and that each element of $B - b_{ij}$ is strictly smaller than a scalar a ($b_{ij}, a \in R$). Cases 1–4 are examined first.

Case 1. There is only one optimal global immediate reward.

We can see that $p_1^1 = 0$, $p_2^1 = (\varepsilon/|A_2|) + (1-\varepsilon)(P_2/(P_1+P_2))$, $p_1^2 = 0$, and $p_2^2 = (\varepsilon/|A_1|) + (1-\varepsilon)(Q_2/(Q_1+Q_2))$. Thus, we arrive at the following equations from (6):

$$\dot{Q}_1 = -Q_1, \tag{12}$$

$$\dot{Q}_2 = \frac{\varepsilon}{2} + (1-\varepsilon)\frac{P_2}{(P_1+P_2)} - Q_2, \tag{13}$$

$$\dot{P}_1 = -P_1, \tag{14}$$

$$\dot{P}_2 = \frac{\varepsilon}{2} + (1-\varepsilon)\frac{Q_2}{(Q_1+Q_2)} - P_2. \tag{15}$$

It can be seen from (12) and (14) that Q_1 and P_1 will be stable at zero after an infinite long time. Suppose at time t_0, Q_1 and P_1 are both very close to zero. Then we can obtain the following from (13) and (15) when $t > t_0$:

$$\dot{Q}_2 = 1 - \frac{\varepsilon}{2} - Q_2, \tag{16}$$

$$\dot{P}_2 = 1 - \frac{\varepsilon}{2} - P_2. \tag{17}$$

If we let $\widehat{Q}_2 = (1-\varepsilon)/(2-Q_2)$ and $\widehat{P}_2 = (1-\varepsilon)/(2-P_2)$, then (16) and (17) can be transformed to a set of linear differential equations. And it is easy to see that $(\widehat{Q}_2^*, \widehat{P}_2^*) = (0,0)$, namely, $(Q_2^*, P_2^*) = ((1-\varepsilon)/2, (1-\varepsilon)/2)$ is a globally stable node. To sum up, there is only one globally stable critical point $(Q_1^*, Q_2^*, P_1^*, P_2^*) = (0, (1-\varepsilon)/2, 0, (1-\varepsilon)/2)$ in Case 1. This point is corresponding to the strategy $(x^*, y^*) = (Q_1^*/(Q_1^* + Q_2^*), P_1^*/(P_1^* + P_2^*)) = (0,0)$, which corresponds to the optimal global immediate reward. In Case 1, this conclusion is also valid for $\varepsilon = 0$ and $\varepsilon = 1$.

To validate our analysis, we present the plot of the learning process of EAQR in Case 1 with Figures 5–10. The learning rate α is 0.1, and the number of samples N_s is 200. Twelve different points (Q_1, Q_2, P_1, P_2) are used as initial conditions and marked with solid circles. It can be seen in Figures 5 and 8 that the learning trajectories converge to the point $(Q_1^*, Q_2^*, P_1^*, P_2^*) = (0,1,0,1)$ when $\varepsilon = 0$. It can also be seen in Figures 6 and 9 that the learning trajectories converge to the point $(Q_1^*, Q_2^*, P_1^*, P_2^*) = (0, 0.55, 0, 0.55)$ when $\varepsilon = 0.9$. Both points are literally the critical point $(Q_1^*, Q_2^*, P_1^*, P_2^*) = (0, (1-\varepsilon)/2, 0, (1-\varepsilon)/2)$ we have obtained earlier. The joint strategy (x, y) is illustrated in Figures 7 and 10. It converges to $(0, 0)$. This indicates that our analysis is reasonable.

Case 2. There are two optimal global immediate reward in diagonal positions.

We arrive at the following equations from (6):

$$\dot{Q}_1 = \frac{\varepsilon}{2} + (1-\varepsilon)\frac{P_1}{(P_1+P_2)} - Q_1, \tag{18}$$

$$\dot{Q}_2 = \frac{\varepsilon}{2} + (1-\varepsilon)\frac{P_2}{(P_1+P_2)} - Q_2, \tag{19}$$

$$\dot{P}_1 = \frac{\varepsilon}{2} + (1-\varepsilon)\frac{Q_1}{(Q_1+Q_2)} - P_1, \tag{20}$$

$$\dot{P}_2 = \frac{\varepsilon}{2} + (1-\varepsilon)\frac{Q_2}{(Q_1+Q_2)} - P_2. \tag{21}$$

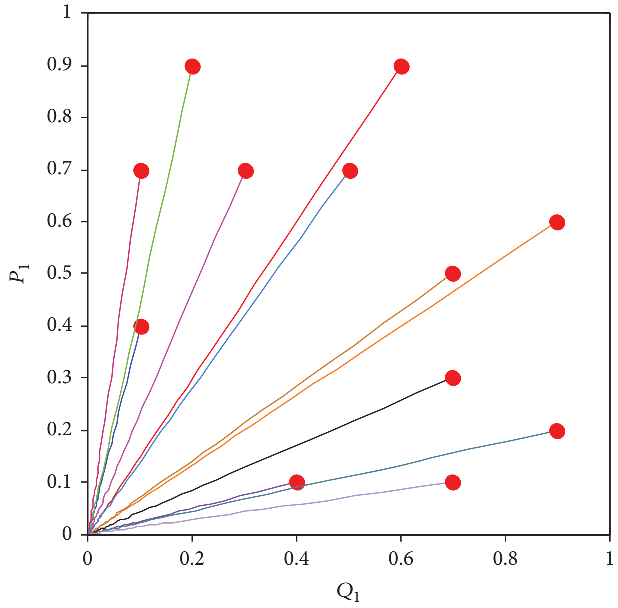


Figure 5: P_1 and Q_1 during the learning process of EAQR in Case 1 ($\varepsilon = 0$).

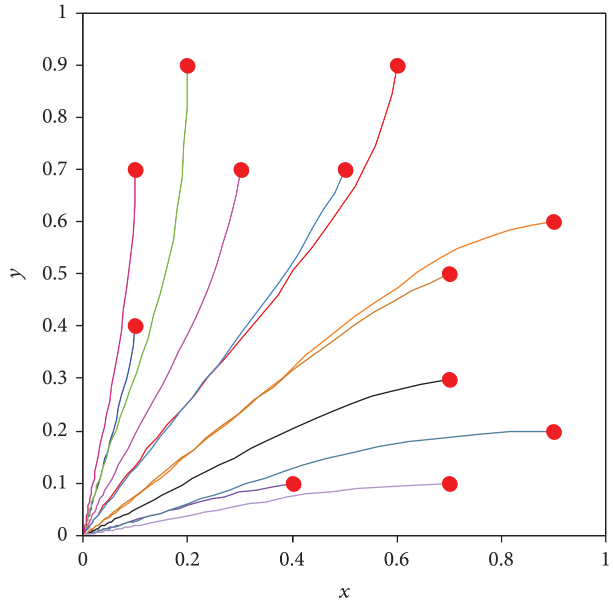


Figure 7: x and y during the learning process of EAQR in Case 1 ($\varepsilon = 0$).

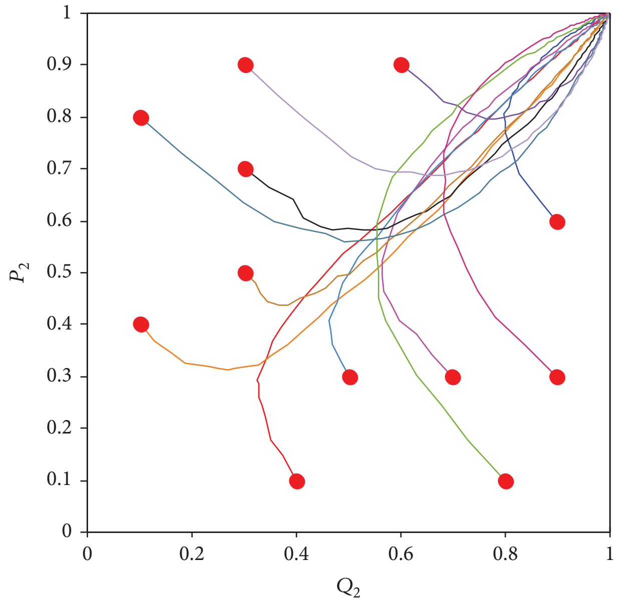


Figure 6: P_2 and Q_2 during the learning process of EAQR in Case 1 ($\varepsilon = 0$).

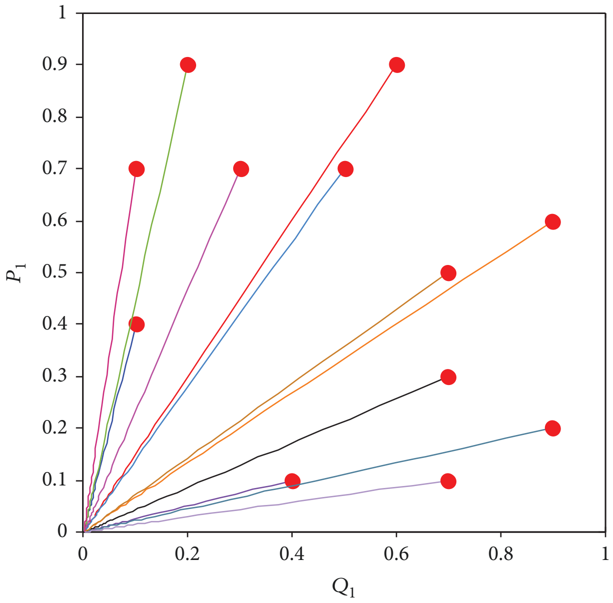


Figure 8: P_1 and Q_1 during the learning process of EAQR in Case 1 ($\varepsilon = 0.9$).

If we let $Q = Q_1 + Q_2$ and $P = P_1 + P_2$, then the following system can be derived from (18)–(21):

$$\dot{Q} = 1 - Q, \tag{22}$$

$$\dot{P} = 1 - P. \tag{23}$$

It is obvious that $(Q^*, P^*) = (1, 1)$ is a globally stable node of the above system. Suppose at time t_0, Q and P are both very close to 1. Then the system described by (18)–(21) degenerates to the following one when $t > t_0$:

$$\dot{Q}_1 = -Q_1 + (1 - \varepsilon)P_1 + \frac{\varepsilon}{2}, \tag{24}$$

$$\dot{P}_1 = (1 - \varepsilon)Q_1 - P_1 + \frac{\varepsilon}{2}. \tag{25}$$

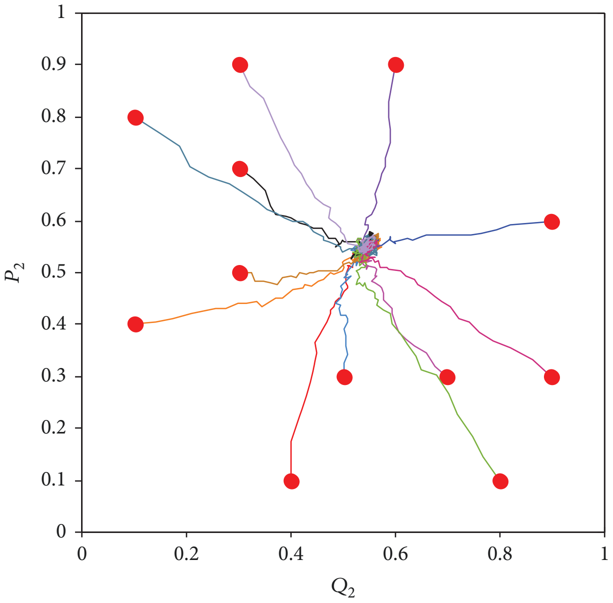


FIGURE 9: P_2 and Q_2 during the learning process of EAQR in Case 1 ($\varepsilon = 0.9$).

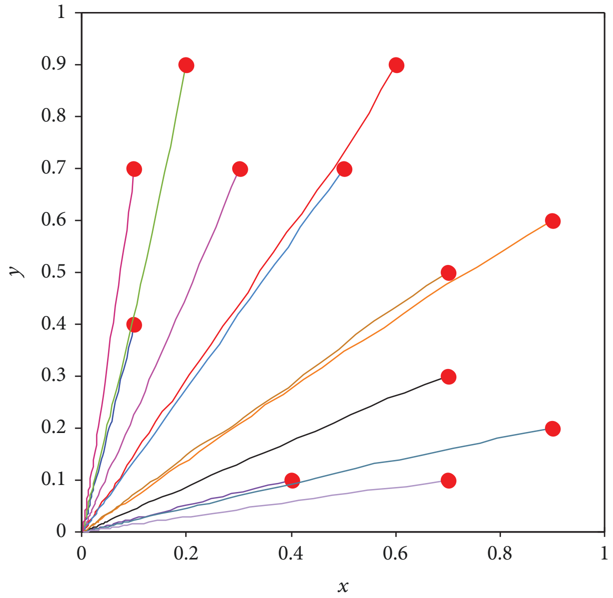


FIGURE 10: x and y during the learning process of EAQR in Case 1 ($\varepsilon = 0.9$).

The interior critical point must satisfy

$$Q_1 = (1-\varepsilon)P_1 + \frac{\varepsilon}{2},$$
$$P_1 = (1-\varepsilon)Q_1 + \frac{\varepsilon}{2}. \tag{26}$$

Thus, we have the critical point $(Q_1^*, P_1^*) = (0.5, 0.5)$ when $\varepsilon \in (0, 1]$. Then we examine the stability of this critical point. The Jacobin matrix of the system described by (24), (25) is

$$J = \begin{bmatrix} -1 & 1-\varepsilon \\ 1-\varepsilon & -1 \end{bmatrix} \tag{27}$$

of which the eigenvalues are $\lambda_{1,2} = -1 \pm (1-\varepsilon)$. When $\varepsilon \in (0, 1]$, we have $\lambda_1 < 0$, $\lambda_2 < 0$. According to the theorem of stability of almost linear systems, this critical point is stable. To sum up, there is only one stable critical point $(Q_1^*, Q_2^*, P_1^*, P_2^*) = (0.5, 0.5, 0.5, 0.5)$ in Case 2 when $\varepsilon \in (0, 1]$. This point is corresponding to the strategy $(x^*, y^*) = (Q_1^*/(Q_1^* + Q_2^*), P_1^*/(P_1^* + P_2^*)) = (0.5, 0.5)$. The greedy joint action may not correspond to either of the optimal global immediate reward.

The above conclusion does not hold when $\varepsilon = 0$. This is because we use the condition $\varepsilon \in (0, 1]$ when determining the position and the stability of the critical point of the system described by (24) and (25). When the system described by (22) and (23) is stable, the point (Q_1, Q_2) is on the line $Q_1 + Q_2 = 1$; the point (P_1, P_2) is on the line $P_1 + P_2 = 1$, and $Q_1 = P_1$. The converged strategy is determined by initial conditions.

Case 3. There are two optimal global immediate reward in the same row.

The system is described by the following differential equations:

$$\dot{Q}_1 = 1 - Q_1,$$
$$\dot{Q}_2 = -Q_2,$$
$$\dot{P}_1 = \frac{\varepsilon}{2} + (1-\varepsilon)\frac{Q_1}{(Q_1+Q_2)} - P_1, \tag{28}$$
$$\dot{P}_2 = \frac{\varepsilon}{2} + (1-\varepsilon)\frac{Q_1}{(Q_1+Q_2)} - P_2.$$

The analysis process is similar with that in Case 1. There is only one stable critical point $(Q_1^*, Q_2^*, P_1^*, P_2^*) = (1, 0, (1-\varepsilon)/2, (1-\varepsilon)/2)$ for $\varepsilon \in [0, 1]$. This point is corresponding to the strategy $(x^*, y^*) = (Q_1^*/(Q_1^* + Q_2^*), P_1^*/(P_1^* + P_2^*)) = (1, 0.5)$, which corresponds to either of the optimal global immediate reward.

Case 4. There are three optimal global immediate reward.

The system is described by the following differential equations:

$$\dot{Q}_1 = 1 - Q_1,$$
$$\dot{Q}_2 = \frac{\varepsilon}{2} + (1-\varepsilon)\frac{P_1}{(P_1+P_2)} - Q_2,$$

$$\dot{P}_1 = 1 - P_1,$$
$$\dot{P}_2 = \frac{\varepsilon}{2} + (1-\varepsilon)\frac{Q_1}{(Q_1+Q_2)} - P_2. \quad (29)$$

It is obvious that $(Q1, P1) = (1,1)$ is a globally stable node. Suppose at time t_0, Q_1 and P_1 are both very close to 1. Then the system degenerates to the following one when $t > t_0$:

$$\dot{Q}_2 = (1-\varepsilon)\frac{1}{(1+P_2)} + \frac{\varepsilon}{2} - Q_2,$$
$$\dot{P}_2 = (1-\varepsilon)\frac{1}{(1+Q_2)} + \frac{\varepsilon}{2} - P_2. \quad (30)$$

The interior critical point must satisfy

$$Q_2 = \frac{\varepsilon}{2} + (1-\varepsilon)\frac{1}{(1+P_2)},$$
$$P_2 = \frac{\varepsilon}{2} + (1-\varepsilon)\frac{1}{(1+Q_2)}. \quad (31)$$

It can be derived that the critical point is $(Q_2^*, P_2^*) = (c, c)$ where $c = ((-((1-\varepsilon)/2) + \sqrt{((1-\varepsilon)/2)^2 + 4((1-\varepsilon)/2))})/2$. We can further get $c \in [0.5, 0.618]$ when $\varepsilon \in [0,1]$. Thus at the critical point, $x = y$, and they are both within [0.618, 0.667] when $\varepsilon \in [0,1]$. The examination of stability follows the way in Case 2. It can be determined that the critical point is a stable node. This means that the converged greedy joint action corresponds to the top left optimal global immediate reward. This conclusion holds for$\varepsilon \in [0,1]$.

We want to examine cases with more than two actions. Thus, Case 5 and Case 6 repeated games with two agents and three actions are given. In Case 5 and Case 6, x_j and y_j represent the probability of selecting action j for player 1 and 2, respectively.

Case 5. There are three optimal global immediate reward in the diagonal line.

The system is described by the following differential equations:

$$\dot{Q}_j = \frac{\varepsilon}{3} + (1-\varepsilon)\frac{P_j}{(P_1+P_2+P_3)} - Q_j,$$
$$\dot{P}_j = \frac{\varepsilon}{3} + (1-\varepsilon)\frac{Q_j}{(Q_1+Q_2+Q_3)} - P_j. \quad (32)$$

If we let $Q = Q_1 + Q_2 + Q_3$ and $P = P_1 + P_2 + P_3$ and follow the way in Case 2, then it can be obtained that there is only one stable node $(Q_1^*, Q_2^*, Q_3^*, P_1^*, P_2^*, P_3^*) = (1/3, 1/3, 1/3, 1/3, 1/3, 1/3)$ when $\varepsilon \in (0,1]$. This critical point is corresponding to the strategy $(x_1^*, x_2^*, x_3^*, y_1^*, y_2^*, y_3^*) = (1/3, 1/3, 1/3, 1/3, 1/3, 1/3)$. The greedy joint action may not correspond to any optimal global immediate reward.

As in Case 2, the above conclusion does not hold when $\varepsilon = 0$. In this situation, when the system is stable, the point (Q_1, Q_2, Q_3) is on the plane $Q_1 + Q_2 + Q_3 = 1$, the point $(P_1, P_2, P_3)$is on the plane $P_1 + P_2 + P_3 = 1$, $Q_1 = P_1$, and $Q_2 = P_2$. The converged strategy is determined by initial conditions.

Case 6. There are four optimal global immediate reward.

The system is described by the following differential equations:

$$\dot{Q}_1 = \frac{\varepsilon}{3} + (1-\varepsilon)\frac{P_2}{(P_1+P_2+P_3)} - Q_1, \quad (33)$$

$$\dot{Q}_2 = \frac{2\varepsilon}{3} + (1-\varepsilon)\frac{P_1+P_3}{(P_1+P_2+P_3)} - Q_2, \quad (34)$$

$$\dot{Q}_3 = \frac{\varepsilon}{3} + (1-\varepsilon)\frac{P_2}{(P_1+P_2+P_3)} - Q_3, \quad (35)$$

$$\dot{P}_1 = \frac{\varepsilon}{3} + (1-\varepsilon)\frac{Q_2}{(Q_1+Q_2+Q_3)} - P_1, \quad (36)$$

$$\dot{P}_2 = \frac{2\varepsilon}{3} + (1-\varepsilon)\frac{Q_1+Q_3}{(Q_1+Q_2+Q_3)} - P_2, \quad (37)$$

$$\dot{P}_3 = \frac{\varepsilon}{3} + (1-\varepsilon)\frac{Q_2}{(Q_1+Q_2+Q_3)} - P_3. \quad (38)$$

If we let $M_1 = Q_1 + Q_2$, $M_2 = Q_2 + Q_3$, $N_1 = P_1 + P_2$, and $N_2 = P_2 + P_3$, the following system can be derived from (33)–(38):

$$\dot{M}_1 = 1 - M_1,$$
$$\dot{M}_2 = 1 - M_2,$$
$$\dot{N}_1 = 1 - N_1,$$
$$\dot{N}_2 = 1 - N_2. \quad (39)$$

It is obvious that $(M_1, M_2, N_1, N_2) = (1,1,1,1)$ is a globally stable node of the above system. Suppose at time t_0, the state of the above system is very close to the stable state (1, 1, 1, 1), that is

$$Q_1 = 1 - Q_2, \quad (40)$$

$$Q_3 = 1 - Q_2, \quad (41)$$

$$P_1 = 1 - P_2, \quad (42)$$

$$P_3 = 1 - P_2. \quad (43)$$

Then the system described by (33)–(38) degenerates to the following one when $t > t_0$:

$$\dot{Q}_2 = \frac{2\varepsilon}{3} + (1-\varepsilon)\frac{(2-2P_2)}{(2-P_2)} - Q_2, \quad (44)$$

$$\dot{P}_2 = \frac{2\varepsilon}{3} + (1-\varepsilon)\frac{(2-2Q_2)}{(2-Q_2)} - P_2. \quad (45)$$

The critical point has to satisfy

$$\begin{aligned} Q_2 &= \frac{2\varepsilon}{3} + (1-\varepsilon)\frac{(2-2P_2)}{(2-P_2)}, \\ P_2 &= \frac{2\varepsilon}{3} + (1-\varepsilon)\frac{(2-2Q_2)}{(2-Q_2)}. \end{aligned} \tag{46}$$

It can be obtained that there is only one critical point $(Q_2^*, P_2^*) = (c, c)$ where $c = (2-2\varepsilon)/3 - \sqrt{((2-2\varepsilon)/3)((1-2\varepsilon)/3)}$. It can be further determined that $c \in (2-\sqrt{2}, 2/3]$ for $\varepsilon \in (0,1]$. The Jacobin matrix of the system described by (44), (45) is

$$J = \begin{bmatrix} -1 & \dfrac{-2(1-\varepsilon)}{(2-P_2^*)^2} \\ \dfrac{-2(1-\varepsilon)}{(2-Q_2^*)^2} & -1 \end{bmatrix} \tag{47}$$

of which the eigenvalues are $\lambda_{1,2} = -1 \pm (2(1-\varepsilon)/(2-P_2^*)(2-Q_2^*))$. There are two repeated roots $\lambda_{1,2} = -1$ when $\varepsilon = 1$. In this situation, the system described by (33)–(38) will be stable at the point $(Q_1^*, Q_2^*, Q_3^*, P_1^*, P_2^*, P_3^*) = (1/3, 2/3, 1/3, 1/3, 2/3, 1/3)$ which corresponds to the point $(x_1^*, x_2^*, x_3^*, y_1^*, y_2^*, y_3^*) = (0.25, 0.5, 0.25, 0.25, 0.5, 0.25)$. When $\varepsilon \in (0,1)$, let $k = (2-2\varepsilon)/3$, $k \in (4/3, 2)$, then the eigenvalues can be rewrote as $\lambda_{1,2} = -1 \pm ((3k-4)/(2-k+\sqrt{k(k-1)})^2)$. We want to show that in this situation there are two different negative real eigenvalues. The following condition will suffice:

$$\left| \frac{(3k-4)}{\left(2-k+\sqrt{k(k-1)}\right)^2} \right| < 1. \tag{48}$$

It is trivial to prove (48). To sum up, the system described by (33)–(38) has a stable node for $\varepsilon \in (0,1]$, and the greedy action for both players is the second action. Unfortunately, this joint action does not correspond to any optimal global immediate reward.

When $\varepsilon = 0$, there is only one stable node that satisfies (40)–(43) and

$$Q_2 + P_2 = \frac{1}{2} Q_2 P_2 + 1. \tag{49}$$

The converged strategy is determined by initial conditions, and the greedy joint action does not necessarily correspond to any optimal global immediate reward.

Case 7. There is only one optimal global immediate reward in a three-player three-action game.

Player 1 and player 2 are literally the row player and the column player, respectively. Player 3 can be viewed as a matrix player. If player 3 chooses the first action, the left payoff matrix will be adopted. Otherwise, the right payoff matrix will be adopted. We assume that there are matrix B, matrix C, and each element of B and $C - b_{ij}$, c_{ij} is strictly smaller than a scalar a $(b_{ij}, c_{ij}, a \in R)$. Let Q_j, P_j, and K_j denote the Q-value of action j for player 1, 2, 3, respectively, and let x, y, z denote the probability of selecting the first action for player 1, 2, 3, respectively. The system is described by the following differential equations:

$$\begin{aligned} \dot{Q}_1 &= (1-\varepsilon)\frac{K_1}{(K_1+K_2)}\left[\frac{\varepsilon}{2} + (1-\varepsilon)\frac{P_1}{(P_1+P_2)}\right] \\ &\quad + \frac{\varepsilon}{2}\left[\frac{\varepsilon}{2} + (1-\varepsilon)\frac{P_1}{(P_1+P_2)}\right] - Q_1, \\ \dot{Q}_2 &= -Q_2, \\ \dot{P}_1 &= (1-\varepsilon)\frac{K_1}{(K_1+K_2)}\left[\frac{\varepsilon}{2} + (1-\varepsilon)\frac{Q_1}{(Q_1+Q_2)}\right] \\ &\quad + \frac{\varepsilon}{2}\left[\frac{\varepsilon}{2} + (1-\varepsilon)\frac{Q_1}{(Q_1+Q_2)}\right] - P_1, \\ \dot{P}_2 &= -P_2, \\ \dot{K}_1 &= \left[\frac{\varepsilon}{2} + (1-\varepsilon)\frac{P_1}{(P_1+P_2)}\right]\left[\frac{\varepsilon}{2} + (1-\varepsilon)\frac{Q_1}{(Q_1+Q_2)}\right] - K_1, \\ \dot{K}_2 &= -K_2. \end{aligned} \tag{50}$$

The analysis process is similar with that in Case 1. There is only one stable node $(Q_1^*, Q_2^*, P_1^*, P_2^*, K_1^*, K_2^*) = (((1-\varepsilon)/2)^2, 0, ((1-\varepsilon)/2)^2, 0, ((1-\varepsilon)/2)^2, 0)$ for $\varepsilon \in [0,1]$. This critical point corresponds to the strategy $(x^*, y^*, z^*) = (Q_1^*/(Q_1^*+Q_2^*), P_1^*/(P_1^*+P_2^*), K_1^*/(K_1^*+K_2^*)) = (1,1,1)$, which literally corresponds to the optimal global immediate reward.

To sum up, it can be seen that the optimal global immediate reward can be achieved in Cases 1, 3, 4, 7 and may not be achieved in Cases 2, 5, 6. In the next section, we will show the performance of EAQR in two stochastic games: box-pushing and the DSN problem.

5. Simulations on Stochastic Games

Case A. Box-pushing

The box-pushing problem is illustrated in Figure 11. Four boxes are represented with grey solid circles, and empty positions are represented with white circles. Four agents (which are not shown in the figure) need to collaborate with each other to make the boxes distribute uniformly. Each agent is responsible for moving one box and has three kinds of actions: pushing the box to the adjacent clockwise position, pushing the box to the adjacent anticlockwise position, or doing nothing. In the beginning of an episode, four boxes are located in random positions. Each agent selects an action, and the boxes are pushed to the new positions. An episode ends when the number of empty positions between any two adjacent boxes are the same, or 100 steps have occurred. If

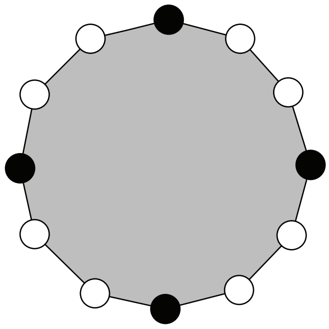


Figure 11: Four-agent/12-vertex box-pushing.

one episode ends, the positions of the four boxes, reward, and step for each agent will be reset for the next episode (but the Q-value function for each agent will be restored until the next run). Each agent receives a reward of −1 at each step and receives a reward of 10 at the end of an episode.

The rules of the box-pushing problem are as follows. First, all agents push boxes simultaneously. Second, if a conflict occurs, then the boxes in the conflict will stay still. A conflict occurs in the following cases: a box being pushed to a static box, two boxes being pushed to the same empty positions, two adjacent boxes being pushed in the opposite direction, and a string of adjacent boxes being pushed in the same direction while the head box is in a conflict. Third, a box can be pushed successfully if it is not in a conflict.

In the experiment, EMA (exponential moving average) Q-learning [11], WoLF-PHC [10], and SARSA (state-action-reward-state-action) [21] are chosen as comparison algorithms. EMA Q-learning and WoLF-PHC are MARL algorithms while SARSA is a type of single-agent RL algorithm corresponding to centralized learning in the context of multiple agents.

The parameters were fine tuned after many trials. For EAQR, the sample times $N_s = 50$. The learning rate α follows

$$\alpha = \alpha_{\text{ini}} - \frac{\alpha_{\text{ini}} n}{1.05L}, \tag{51}$$

where the initial learning rate $\alpha_{\text{ini}} = 0.7$, L is the predefined number of learning episodes, and n is number of experienced learning episodes. The exploration rate ε follows

$$\varepsilon = \begin{cases} 0.9 & 1 \le n \le 0.2L, \\ 0.8 & 0.2L < n \le 0.4L, \\ 0.7 & 0.4L < n \le 0.6L, \\ 0.6 & 0.6L < n \le 0.8L, \\ 0.5 & 0.8L < n \le L. \end{cases} \tag{52}$$

For EMA Q-learning, $\varepsilon = 0.2$, $k = 2$, and the discount factor $\gamma = 0.9$. The learning rate α follows (51) with $\alpha_{\text{ini}} = 0.7$; $\eta_l = 0.001\eta_w$ and η_w follows

$$\eta_w = \frac{1}{(10 + 0.2n)}. \tag{53}$$

Table 1: Average steps for 4-agent/12-vertex box-pushing (evaluation episodes = 50,000).

	$L = 100{,}000$	$L = 500{,}000$	$L = 1000{,}000$
Optimal	1.71	1.71	1.71
EAQR	**2.53 ± 0.11**	**1.76 ± 0.03**	**1.74 ± 0.02**
WoLF-PHC	2.83 ± 0.23	2.24 ± 0.11	1.99 ± 0.06
EMA Q-learning	4.53 ± 0.49	3.66 ± 0.40	3.47 ± 0.34
Single-agent RL	14.78 ± 0.60	3.29 ± 0.14	2.03 ± 0.06

Table 2: Average success rate for 4-agent/12-vertex box-pushing (evaluation episodes = 50,000).

	$L = 100{,}000$	$L = 500{,}000$	$L = 1000{,}000$
EAQR	**82.6%**	**98.6%**	**99.6%**
WoLF-PHC	80.7%	87.1%	91.7%
EMA Q-learning	66.7%	76.6%	78.7%
Single-agent RL	60.2%	91.2%	95.9%

Table 3: Maximal steps for 4-agent/12-vertex box-pushing (evaluation episodes = 50,000).

	$L = 100{,}000$	$L = 500{,}000$	$L = 1000{,}000$
EAQR	**2.77**	**1.85**	**1.81**
WoLF-PHC	3.45	2.55	2.18
EMA Q-learning	5.90	4.66	4.61
Single-agent RL	15.89	3.66	2.20

For WoLF-PHC, $\delta_w = 0.003$, $\delta_l = 0.01$, $\gamma = 0.9$, $\varepsilon = 0.8$, and the learning rate α follows (51), with $\alpha_{\text{ini}} = 0.7$. For SARSA, α, γ, and ε are the same as those in WoLF-PHC.

The prime performance metric is the average number of steps per episode, which needs to be minimized. The second important performance metric is the success rate, which reflects the stability of the algorithm. A success means the minimum steps are used in an episode. The theoretical minimum number of steps of an episode is determined by a specially designed program. Thus, the success rate over a number of episodes can be evaluated. The experimental results in Tables 1 and 2 are averaged over 100 runs. The standard deviation is also presented in Table 1. Table 3 shows the worst run for each algorithm. Each run has experienced L learning episodes and 50,000 evaluation episodes. During each of L learning episodes, the agents update their strategies to try to obtain more cumulative reward. During each of 50,000 evaluation episodes, the agents do not update their strategies. For the sake of fairness, in the same columns of Tables 1 and 3, the initial positions of the boxes for the evaluation episodes are the same.

In Tables 1 and 2, it is noted that all algorithms perform better as the number of learning episodes L grows. Optimal represents the theoretical minimum number of steps. EAQR presents the best performance for all different values of L, which means EAQR learns faster than any of the other

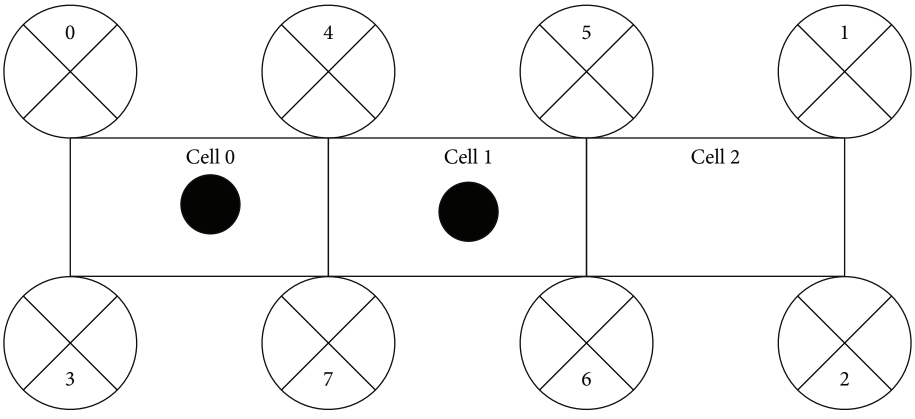


Figure 12: A distributed sensor network with eight sensors ⊗ and two targets ●.

algorithms. Besides, EAQR can obtain an average success rate of 99.6% when L is 1,000,000, which means that it can use the minimum steps to complete the box-pushing task with a probability of 99.6%. This result is sufficiently good to complete the task satisfactorily. Single-agent RL performs poorly in the beginning, but it performs fairly well when L is 1,000,000 in the aspect of average success rate. Still, single-agent RL is outperformed by EAQR. It is noted from Table 3 that EAQR also has the better worst run compared with the other algorithms.

Case B. Distributed sensor network

The DSN problem is used as the second test bed for MARL algorithms. It was part of the NIPS 2005 benchmarking workshop [22]. Figure 12 shows a DSN composed of eight sensors. Each sensor is viewed as an agent. The sensors have to cooperate to capture both targets wandering in a grid of three cells. At each time step, each target moves to its left side, moves to its right side, or keeps still with equal probability. Each cell can be occupied by only one target at any time. The targets move sequentially. Thus, if a target moves out of the grid or moves to a cell which has been occupied by another one, it just stays where it was. Each sensor also has three actions: focus on its left side, focus on its right side, or no focus at all. For example, sensor 4 can focus on cell 1 which is on its left side, focus on cell 0 which is on its right side, or make no focus. Although there is only one cell near sensor 0, 1, 2, and 3, respectively, these four sensors can still focus on the side with no cells. To capture a target, the sensors must accomplish three hits on the same target. One hit happens if at least three sensors focus on the cell occupied by a target. A target is removed from the grid if it is captured. The reward allocation rules follow [23]. If a target is captured by four sensors, the sensor with the minimum index gets null reward, and the other three sensors are rewarded by 10, respectively. The action of focus gains a local immediate reward of −1, and no focus gains a local immediate reward of 0.

The goal of the DSN problem is to capture the targets with as many cumulative rewards as possible in an episode. At the beginning of each episode, both targets are randomly located in the grid. At each time step, all sensors take actions at the same time. The judgment of focus, no focus, hit, and capture is made, and the local immediate rewards are fed back to each sensor. Then it is the turn for targets to move, and the new state is fed back to each sensor. An episode ends if both targets are captured, or 1000 time steps have elapsed. Each sensor can perceive the state and the local immediate rewards. However, they do not have any a priori knowledge like what is a hit, what is a capture, or the goal of the problem. They do not know the reward allocation rules either.

There are 37 states and $3^8 = 6, 561$ joint actions in the DSN problem. Single-agent RL algorithm needs to store and learn Q-value of $37 \times 6561 = 242, 757$ state-action pairs, and this number grows exponentially as the number of sensors increases. By learning each agent's own action instead of joint action, the number of state-action pairs will be reduced to $37 \times 3 \times 8 = 888$, and it grows linearly as the number of sensors increases.

The optimal strategy for a DSN problem is that every three sensors focus on a target while the rest of the sensors make no focus at all. It is obvious that the optimal global cumulative reward is 42, and the optimal number of steps is 3. According to the credit assignment rules in [23], there will be no punishment if all agents do not focus at all. This can lead to more steps in an episode. Thus, we select average global cumulative rewards per episode as the main performance metric and select average number of steps as the secondary performance metric. A success is made if a global cumulative reward of 42 is obtained in an episode.

For EAQR, the learning rate α is constant and is set to 0.2; N_s is set to 50, and the exploration rate ε follows (52). For EMA Q-learning, the learning rate α follows (51) with $\alpha_{\text{ini}} = 0.7$; the exploration rate ε is constant and is set to 0.8, $k = 2$, $\gamma = 0.9$, $\eta_l = 0.001\eta_w$, and η_w follows (53). For WoLF-PHC, the parameters $\delta_w = 0.003$, $\delta_l = 0.01$, the learning rate α follows (51) with $\alpha_{\text{ini}} = 0.7$; the exploration rate ε is constant and is set to 0.2, and $\gamma = 0.9$. For single-agent RL, α, ε, and γ are the same with those of WoLF-PHC.

Table 4: Success rate for the DSN problem (evaluation episodes = 5000).

	$L = 10{,}000$	$L = 50{,}000$	$L = 100{,}000$
EAQR	**47.4%**	**99.9%**	**100%**
WoLF-PHC	22.8%	34.1%	33.7%
EMA Q-learning	7.8%	6.8%	7.1%
Single-agent RL	0	0	0

Table 5: Average cumulative reward for the DSN problem (evaluation episodes = 5000).

	$L = 10{,}000$	$L = 50{,}000$	$L = 100{,}000$
EAQR	**41.23 ± 0.52**	**41.99 ± 0.003**	**42 ± 0**
WoLF-PHC	39.96 ± 0.89	40.69 ± 0.73	40.74 ± 0.68
EMA Q-learning	36.59 ± 1.76	36.14 ± 1.82	36.21 ± 1.80
Single-agent RL	29.88 ± 1.57	33.16 ± 1.33	34.96 ± 1.05

Tables 4 and 5 show that after 100, 000 learning episodes the success rate of EAQR is 100%, and it gains an average global cumulative reward of 42 which is just the theoretical optimal cumulative reward in an episode. The standard deviation is also presented in Table 5. EAQR has great advantages over the other algorithms in terms of success rate. Table 5 shows that for WoLF-PHC and single-agent RL, higher average cumulative reward might be achieved if more learning episodes are given. However, there is no such a trend for EMA Q-learning. More learning would probably not improve the performance of EMA Q-learning. Table 6 shows the worst run of cumulative reward for each algorithm. It is noted from Table 6 that EAQR also has the better worst cumulative reward compared with the other algorithms.

Table 7 shows that EAQR consumes less steps to capture both targets than the other algorithms. The standard deviation is also presented in Tables 7. Table 8 shows the worst run of steps for each algorithm. Due to the large joint action space in the DSN problem, single-agent RL shows the worst performance among all algorithms. This experiment shows that solving multiagent reinforcement problem through single-agent view is inadvisable.

EAQR shows good performance in both stochastic games, which indicates that most of the time EAQR can converge to one of the optimal global cumulative reward under any initial strategies. Otherwise, EAQR will not gain a success rate of 99.6% in Case A and 100% in Case B. EAQR also alleviates the curse of dimensionality of joint action space. Yet the same problem for joint state space remains to be addressed. For some stochastic games such as box-pushing [24, 25] and hunting game [26, 27], the circle and the grid can be viewed as images. Many states are actually the same one if the translation operation is performed on the "images". Thus, the structure of convolutional neural networks [28, 29] can be employed to realize an autoencoder [30] which automatically extracts features in the original state space and uses these features to construct a compressed state space.

Table 6: Minimal cumulative reward for the DSN problem (evaluation episodes = 5000).

	$L = 10{,}000$	$L = 50{,}000$	$L = 100{,}000$
EAQR	**39.26**	**41.97**	**42**
WoLF-PHC	37.65	38.71	38.82
EMA Q-learning	32.92	32.08	32.53
Single-agent RL	25.12	29.38	32.21

Table 7: Average steps for the DSN problem (evaluation episodes = 5000).

	$L = 10{,}000$	$L = 50{,}000$	$L = 100{,}000$
EAQR	3.65 ± 0.35	3.22 ± 0.26	**3.12 ± 0.20**
WoLF-PHC	3.64 ± 0.63	3.58 ± 0.64	3.69 ± 0.61
EMA Q-learning	3.81 ± 0.43	3.94 ± 0.44	3.93 ± 0.45
Single-agent RL	5.57 ± 0.25	5.3 ± 0.29	5.06 ± 0.27

Table 8: Maximal steps for the DSN problem (evaluation episodes = 5000).

	$L = 10{,}000$	$L = 50{,}000$	$L = 100{,}000$
EAQR	4.81	3.93	**3.77**
WoLF-PHC	5.34	5.34	5.67
EMA Q-learning	4.72	4.94	4.80
Single-agent RL	6.00	6.00	5.66

6. Conclusions

In this paper, we deal with the problem of how to achieve optimal coordination in fully cooperative multiagent systems. Firstly, we propose a cooperative multiagent Q-learning algorithm called EAQR and analyze its dynamics in seven repeated games. The results in these games show that if there is only one optimal global immediate reward, then EAQR can converge to it. However, if more than one optimal global immediate reward exist, then EAQR may not necessarily converge to any optimal global immediate reward. Secondly, we test EAQR in two stochastic games—one with four agents and the other with eight agents. EAQR shows excellent performance in both tasks. It achieves the theoretical optimal cumulative reward in the DSN problem.

We will carry on our work towards three directions in the future. Firstly, we have to find a way to depict the learning process for stochastic games to help us find out why EAQR works well in these tasks. Secondly, we will learn from solutions to consensus [31, 32] to design new action exploration methods that can be analyzed more trivially and has rigorous theoretical proof in general cases. Thirdly, we will employ convolutional neural networks and autoencoders to alleviate the curse of dimensionality of state space in some collaborative tasks.

Data Availability

All underlying data related to this paper are available by sending emails to the corresponding author Zhen Zhang with email address tbsunshine8@163.com.

Conflicts of Interest

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

Acknowledgments

The authors appreciate Professor Dongbin Zhao for his suggestions on the convergence analysis in the paper. This work was supported by Shandong Provincial Natural Science Foundation of China under Grant (ZR2017PF005, ZR2015FM017) and the National Natural Science Foundation of China (61573205).

References

[1] R. Cui, C. Yang, Y. Li, and S. Sharma, "Adaptive neural network control of AUVs with control input nonlinearities using reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 6, pp. 1019–1029, 2017.

[2] Z. Wang, L. Liu, H. Zhang, and G. Xiao, "Fault-tolerant controller design for a class of nonlinear MIMO discrete-time systems via online reinforcement learning algorithm," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 5, pp. 611–622, 2016.

[3] H. Kebriaei, A. Rahimi-Kian, and M. N. Ahmadabadi, "Model-based and learning-based decision making in incomplete information Cournot games: a state estimation approach," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 4, pp. 713–718, 2015.

[4] D. Zhao, Z. Xia, and D. Wang, "Model-free optimal control for affine nonlinear systems with convergence analysis," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 4, pp. 1461–1468, 2015.

[5] L. Busoniu, R. Babuska, and B. de Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 2, pp. 156–172, 2008.

[6] K. G. Vamvoudakis, F. L. Lewis, and G. R. Hudas, "Multi-agent differential graphical games: online adaptive learning solution for synchronization with optimality," *Automatica*, vol. 48, no. 8, pp. 1598–1611, 2012.

[7] D. Zhao, Q. Zhang, D. Wang, and Y. Zhu, "Experience replay for optimal control of nonzero-sum game systems with unknown dynamics," *IEEE Transactions on Cybernetics*, vol. 46, no. 3, pp. 854–865, 2016.

[8] J. Hu and M. P. Wellman, "Nash Q-learning for general-sum stochastic games," *Journal of Machine Learning Research*, vol. 4, pp. 1039–1069, 2003.

[9] S. Singh, M. Kearns, and Y. Mansour, "Nash convergence of gradient dynamics in general-sum games," in *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pp. 541–548, Morgan Kaufman, USA, June-July 2000.

[10] M. Bowling and M. Veloso, "Multiagent learning using a variable learning rate," *Artificial Intelligence*, vol. 136, no. 2, pp. 215–250, 2002.

[11] M. D. Awheda and H. M. Schwartz, "Exponential moving average based multiagent reinforcement learning algorithms," *Artificial Intelligence Review*, vol. 45, no. 3, pp. 299–332, 2016.

[12] S. Abdallah and V. Lesser, "A multiagent reinforcement learning algorithm with non-linear dynamics," *Journal of Artificial Intelligence Research*, vol. 33, pp. 521–549, 2008.

[13] M. Babes, M. Wunder, and M. L. Littman, *Q-Learning in Two-Player Two-Action Games*, AAMAS, 2009.

[14] K. Tuyls and A. Nowé, "Evolutionary game theory and multiagent reinforcement learning," *Knowledge Engineering Review*, vol. 20, no. 1, pp. 63–90, 2005.

[15] K. Tuyls and S. Parsons, "What evolutionary game theory tells us about multiagent learning," *Artificial Intelligence*, vol. 171, no. 7, pp. 406–416, 2007.

[16] D. Bloembergen, K. Tuyls, D. Hennes, and M. Kaisers, "Evolutionary dynamics of multi-agent learning: a survey," *Journal of Artificial Intelligence Research*, vol. 53, pp. 659–697, 2015.

[17] A. Kianercy and A. Galstyan, "Dynamics of Boltzmann *Q* learning in two-player two-action games," *Physical Review E*, vol. 85, no. 4, pp. 1145–1154, 2012.

[18] Y. Kao, T. Yang, and J. H. Park, "Exponential stability of switched Markovian jumping neutral-type systems with generally incomplete transition rates," *International Journal of Robust and Nonlinear Control*, vol. 28, no. 5, pp. 1583–1596, 2018.

[19] B. Jiang, K. Yonggui, H. R. Karimi, and C. C. Gao, "Stability and stabilization for singular switching semi-Markovian jump systems with generally uncertain transition rates," *IEEE Transactions on Automatic Control*, p. 1, 2018.

[20] J. S. Shamma and G. Arslan, "Dynamic fictitious play, dynamic gradient play, and distributed convergence to Nash equilibria," *IEEE Transactions on Automatic Control*, vol. 50, no. 3, pp. 312–327, 2005.

[21] G. A. Rummery and M. Niranjan, *On-Line Q-Learning Using Connectionist Systems*, Engineering Dept., Cambridge University, Tech. Rep. CUED/F-INFENG/TR 166, 1994.

[22] A. Syed, S. Koenig, and M. Tambe, "Preprocessing techniques for accelerating the DCOP algorithm ADOPT," in *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pp. 1041–1048, Utrecht, The Netherlands, July 2005.

[23] J. Kok and N. Vlassis, "Collaborative multiagent reinforcement learning by payoff propagation," *Journal of Machine Learning Research*, vol. 7, pp. 1789–1828, 2006.

[24] G. Chen, W. Cao, X. Chen, and M. Wu, "Multi-agent Q-learning with joint state value approximation," in *Proceedings of the 30th Chinese Control Conference*, pp. 4878–4882, Yantai, China, July 2011.

[25] A. Bab and R. Brafman, "Multi-agent reinforcement learning in common interest and fixed sum stochastic games: an experimental study," *Journal of Machine Learning Research*, vol. 9, no. 4, pp. 2635–2675, 2008.

[26] T. Taniguchi and T. Sawaragi, "Adaptive organization of generalized behavioral concepts for autonomous robots: schema-based modular reinforcement learning," in *2005 International Symposium on Computational Intelligence in Robotics and Automation*, pp. 601–606, Espoo, Finland, June 2005.

[27] L. Matignon, G. J. Laurent, and N. L. Fort-Piat, "Hysteretic Q-learning :an algorithm for decentralized reinforcement learning in cooperative multi-agent teams," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 64–69, San Diego, CA, USA, October-November 2007.

[28] Y. Bengio, A. Courville, and P. Vincent, "Unsupervised feature learning and deep learning: a review and new perspectives," p. 1, 2012, CoRR, abs/1206.5538.

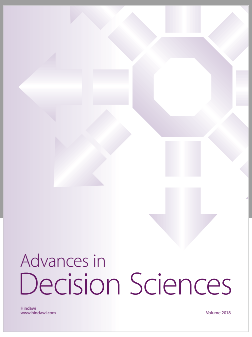
[29] D. Zhao, Y. Chen, and L. Lv, "Deep reinforcement earning with visual attention for vehicle classification," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 9, no. 4, pp. 356–367, 2017.

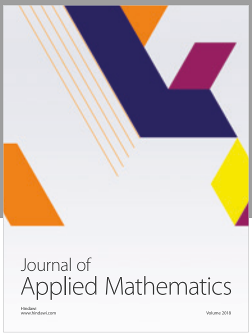
[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015, http://arxiv.org/abs/1512.03385.

[31] L. Rong and H. Shen, "Distributed containment control of second order multiagent systems with input delays under general protocols," *Complexity*, vol. 21, no. 6, pp. 112–120, 2016.

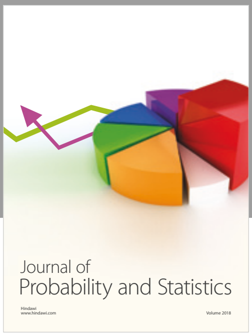
[32] H. Zhao and J. H. Park, "Dynamic output feedback consensus of continuous time networked multiagent systems," *Complexity*, vol. 20, no. 5, pp. 35–42, 2015.

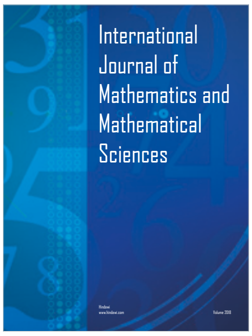
Advances in
Operations Research


Advances in
Decision Sciences

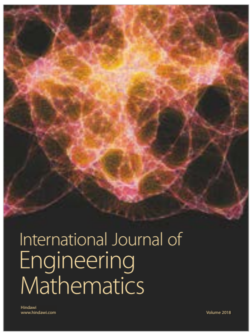

Journal of
Applied Mathematics

The Scientific
World Journal

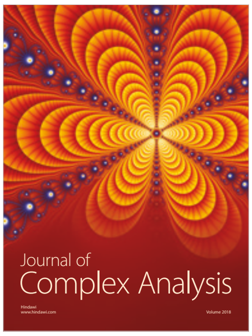

Journal of
Probability and Statistics

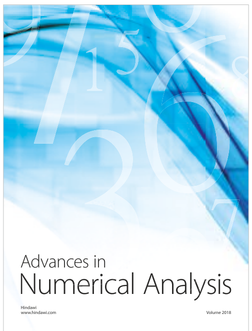

International
Journal of
Mathematics and
Mathematical
Sciences

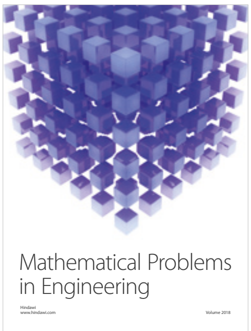
Journal of
Optimization

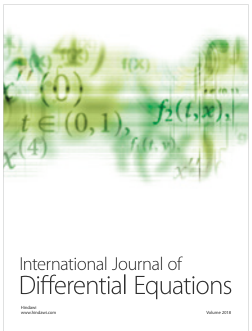

International Journal of
Engineering
Mathematics

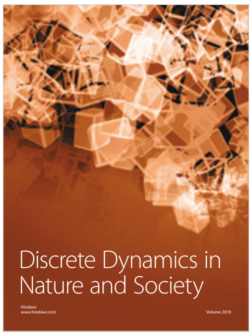
International Journal of
Analysis

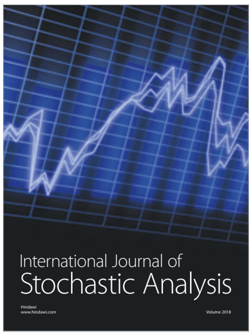

Journal of
Complex Analysis


Advances in
Numerical Analysis


Mathematical Problems
in Engineering

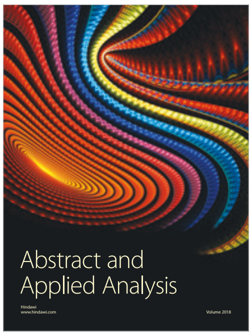

International Journal of
Differential Equations

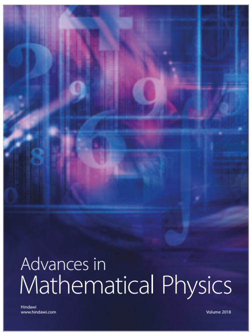

Discrete Dynamics in
Nature and Society


International Journal of
Stochastic Analysis

Journal of
Mathematics

Journal of
Function Spaces


Abstract and
Applied Analysis


Advances in
Mathematical Physics