Hindawi
Scientific Programming
Volume 2021, Article ID 9948156, 9 pages
https://doi.org/10.1155/2021/9948156

Research Article

Experience Weighted Learning in Multiagent Systems

Yi Zou,[1] Jijuan Zhong,[2] Zhihao Jiang,[3] Hong Zhang,[4] and Xuyu Pu[1]

[1] *School of Management and E-Business, Zhejiang Gongshang University, Hangzhou 310018, China*
[2] *School of Economics, Nankai University, Tianjin 300110, China*
[3] *Department of Information Management, School of Management, Shanghai University, Shanghai 200444, China*
[4] *School of Management, Wuhan University of Science and Technology, Wuhan 430081, China*

Correspondence should be addressed to Hong Zhang; savagegardenzh@163.com

Received 26 March 2021; Revised 8 August 2021; Accepted 25 October 2021; Published 27 November 2021

Academic Editor: Pengwei Wang

Agents face challenges to achieve adaptability and stability when interacting with dynamic counterparts in a complex multiagent system (MAS). To strike a balance between these two goals, this paper proposes a learning algorithm for heterogeneous agents with bounded rationality. It integrates reinforcement learning as well as fictitious play to evaluate the historical information and adopt mechanisms in evolutionary game to adapt to uncertainty, which is referred to as experience weighted learning (EWL) in this paper. We have conducted multiagent simulations to test the performance of EWL in various games. The results demonstrate that the average payoff of EWL exceeds that of the baseline in all 4 games. In addition, we find that most of the EWL agents converge to pure strategy and become stable finally. Furthermore, we test the impact of 2 import parameters, respectively. The results show that the performance of EWL is quite stable and there is a potential to improve its performance by parameter optimization.

1. Introduction

As a trial and error mechanism, reinforcement learning (RL) has been used to optimize actions in the presence of uncertainty. Early literature has examined the performance of a single agent in static environments, but recent progress has been witnessed in multiagent reinforcement learning (MARL). Since the dynamic nature and complex correlation in distributed systems closely match the feature of MARL, it has been introduced to applications in disaster response [1] and wireless network [2]. Different types of goals have been proposed in the literature of reinforcement learning, and we briefly classify them into three categories: stability of the learning dynamics [3], adaptation ability to uncertainty, and both of them [4–6]. Since the action and reward of an agent depend much on the behavior of its counterparts that is also learning, it is more challenging to pursue the stability of the learning algorithm and ensure adaptability at the same time.

A great deal of work of RL assumes learning as a Markov decision process. But the Markov property no longer holds in a lot of MAS models, where dynamics are created by random and distributed interactions. Evolutionary game theory, particularly replicator dynamics, is evaluated as a theoretical model for the study of agent dynamics in a multiagent system. A line of recent studies begins to discuss how to optimize policies by integrating MARL with evolutionary game theory [7, 8]. For instance, Hou et al. [7] studied the evolutionary knowledge transfer process with MARL. Zhou et al. [9] used a similar framework to explore the problems in negotiation. But both of them built models based on the Markov decision process. Tuyls et al. [10] applied the replicator dynamics (RD) on population level which can only be used for homogeneous players on this level. But the heterogeneous agents in a repeated game face higher uncertainty compared with homogeneous ones because their counterparts are random through periods and have different learning rates. Unfortunately, RD can only be applied on individual level when agents are heterogeneous. Thus, we aim to investigate the individual learning of heterogeneous agents in MAS, where the Markov property no longer holds and challenges arise for improving the performance of an algorithm.

Several factors have been discussed in previous literature about the efficiency of MARL such as the population structure, information, and type of tasks [11, 12]. Among them, evaluating information is fundamental to agent policies. There are two classic approaches to treat historical information, one is reinforcement learning (RL) and the other is fictitious play (FP). RL focuses on its own experience while FP records the strategy history of its opponent and forms some belief about his future action [13, 14]. They are complementary in information observation perspective to some extent. Camerer and Hua [15] attempted to integrate reinforcement learning and fictitious play to predict agent behavior. The learning dynamics have demonstrated to be successful in repeated games but have not been applied in multiagent systems with heterogeneous agents. Since Tuyls et al. [10] have identified the relation between a general RL model and RD in MAS, we intend to explore further to integrate RL and FP in such context to achieve better performance. To this end, we propose experience weighted learning (EWL) in this paper. It evaluates experience with integration of FP and RL for adaptability and adopts evolutionary dynamics for stability.

The research questions are as follows. First, can EW learning help players to improve their profit compared with the baseline in a population of heterogeneous agents? Second, does the proposed learning algorithm perform stably under nonstationary circumstances? Third, do the agents converge to equilibria finally? This paper is organized as follows to address these research questions. In Section 2, we review the related work of our study. In Section 3, we propose a novel learning algorithm and build different game models to apply it. We conduct experiments to simulate the learning process in games and analyze the results in Section 4 and present discussion and conclusion in Section 5.

2. Related Work

Multiagent reinforcement learning (MARL) has originated from single-agent reinforcement learning and has become popular in recent years [8, 16]. It finds a variety of applications in fields including sensor networks [17], traffic signal control [18], and robotics [19]. MARL has been proved to be advantageous in at least three aspects: parallel computation, distributed layout, and communication between agents. Although agents in MAS have higher capability to adapt to a complex environment, several challenges arise in the real world [20]. For example, QMIX [21] and VDN [22] that incorporate deep reinforcement learning are designed for team tasks with networks. Similar research such as Qatten [23] was proposed recently to respond to challenges when facing limited communication ability, and QTRAN [24] covers a much wider class of MARL tasks. There are two classic kinds of targets that the MARL agents aim to achieve: adaptation and stability. One stream of research aims to achieve desirable adaptation, while the other stream tries to propose algorithms for stability.

Rationality is defined as the one standard for adaptability when one agent converges to a best response and other agents remain stationary [4]. Early literature started to investigate single-agent RL applied in static or repeated games [19, 25]. Claus and Boutilier [26] investigated the performance of two kinds of agents in repeated games in early research, the independent learner and the joint action learner based on Q-learning [27]. Recently, Zhang et al. [28] used learning automata to optimize performance in cooperative tasks. WRFMR [29] adopts a weight parameter and the action probability to balance exploration and exploitation and accelerate convergence to the optimal joint action. Nonstationarity arises in MARL because all the agents in the system are learning simultaneously. The reward of an agent depends on changing counterparts through the periods. Each agent is therefore faced with a moving-target learning problem: the best policy changes as other agents' are adapting [30]. Hence, traditional algorithms based on fixed repeated games find difficulty in capturing dynamic features under such contexts. There is a growing body of literature that focuses on achieving higher adaptability in nonstationary environments, which can arise from matching mechanism, population structure, and so on. Hao et al. [31] proposed algorithms with random matching mechanism and evaluated them in both deterministic and stochastic games. Tang et al. [32] investigated the dynamic network for interaction and studied the reinforcement social learning under a rewiring mechanism. Camerer and Hua [15] attempted to integrate reinforcement learning and fictitious play to predict agent behavior. This line of research paves the way to achieve adaptability in dynamic environments, but most of them adopt a Markov decision process. We intend to discuss the adaptation in nonstationary environment where the Markov property is violated.

Stability is another object that many algorithms pursue to obtain. Convergence to equilibria is regarded as a basic stability standard in a line of literature [3, 33]. Though convergence to a Nash equilibrium is not explicitly required, it arises naturally if all the agents in the system are rational and convergent. An opponent-independent agent that ignores the behavior of others and finally converges to a strategy is considered as an equilibrium solution [34]. It is much easier to maintain stability in static or repeated games, while the complexity arises when parallel interaction exists in MAS. Parallel computation in MAS can reduce the load for solving complicated problems but increase instability because all the agents learn simultaneously and update their strategies constantly. Each agent is facing moving partners whose policies are mutually affected. Hou et al. [7] found that the intrinsic nature of parallelism of evolution in a population is ideal for MAS and thus presented an algorithm by integrating the evolution theory with reinforcement learning to analyze the knowledge transfer which happens simultaneously in a MAS system. Tuyls and Nowe [35] also investigated the relationship between MARL and evolutionary game theory, focusing on static tasks [36]. Prediction is another standard for convergence, Camerer and Hua [15] attempted to use experimental data to estimate the parameters and validate the model by predicting the behavior with the integration of reinforcement learning and fictitious play. They demonstrated that the integration improves the fitness of predicting the behavior of players. This line of research has enlightened us to pursue stability through incorporating evolutionary game theory, but most of them focus on the Markov decision process or homogeneous repeated games. We aim to achieve convergence to equilibrium when agents interact randomly in each period. Additionally, our rules for agent interaction induce more complex policy dynamics by assuming heterogeneity and parallel learning in MAS. It is valuable to explore the stability as well as adaptability of MARL further in the context of dynamic environment.

3. The Model

3.1. Problem Description. We adopt 4 games described in Table 1 as the testbeds for our multiagent learning algorithm, including the hawk dove, stag hunt, battle of sexes, and prison dilemma. Each game contains two strategies: *a* and *b*. When an agent chooses one of them, we call it action. These game models represent different types of competition and common interests. The players interact randomly to play a single stage game that repeats several rounds in a multiagent system. After each round, the players receive their payoffs and corresponding rewards. They adjust their policies for actions, which are distributions of probabilities.

3.2. The Learning Framework. There are two populations of players who interact pairwise to play a game and learn through adjusting their policies. A policy is a combination of probabilities mapping a set of actions. A game is repeated several periods where players meet different counterparts who are randomly matched by the system. In a traditional repeated game, a player has the same counterpart. However, in practice, his partner may change frequently and randomly, resulting in a more complex learning pattern than with fixed pairs. In our model, the system randomly chooses a pair in each interaction. The agents are assumed to know their own payoffs and actions in the game and adjust the policies after each period. The agents evaluate their experience and update the policies based on EW learning. We use the following notation in Table 2 to denote the variables used in the model.

3.3. The Baseline. To test the performance of the EWL, we use independent Q-learning as a baseline for comparison [27]. Q-learning has been demonstrated by numerous works to be an efficient classic model of reinforcement learning. In an algorithm, we call it a greedy action if you always select the action with the highest value. When you select one of these actions, we say that you are exploiting your current knowledge of the values of the actions. Instead, if you select one of the nongreedy actions, then we say you are exploring. We choose ε-greedy, which is one of the most common algorithms setting a small probability ε for nonoptimal actions, to balance exploitation and exploration. The Q-learning algorithm is illustrated in Algorithm 1.

Table 1: The matrixes of game payoff.

(1) Hawk dove			(2) Battle of the sexes		
	a	*b*		*a*	*b*
a	3, 3	1, 4	*a*	2, 1	0, 0
b	4, 1	0, 0	*b*	0, 0	1, 2
(3) Prison dilemma			(4) Stag hunt		
	a	*b*		*a*	*b*
a	3, 3	0, 5	*a*	1, 1	0, 0.5
b	5, 0	1, 1	*b*	0.5, 0	0.5, 0.5

Table 2: Notations.

Variables	Explanation
$N(t)$	The observed count of interactions in time t
a_t	The action of an agent in time t
$\pi(a)$	The policy mapping action a to probability
ρ	The discount rate for experience
Φ	The decay of utility with respect to time
$r(a)$	The reward of action a
$q(a)$	The utility of taking action a

3.4. Experience Weighted Learning. We integrate reinforcement learning with fictitious play to evaluate the interaction experience and adopt replicator dynamics to update the policy during period iterations. We assume that the agents have bounded rationality and heterogeneous initial policies. Each agent chooses an action and randomly matches its opponent to play a game. The game repeats until the time reaches a predefined maximum value. The agents calculate their rewards at the end of each period and update their policies for the next period. An agent can use a policy that is a probability combination of different actions. The experience weighted learning algorithm is illustrated in Algorithm 2. We describe the basic procedure and calculation formulas of experience, utility, and learning dynamics as follows. We assume that the agents have a natural decay experience, and their beliefs are updated by depreciating the previous counts by ρ (≤ 1) and adding one for the action chosen by the players:

$$N_i(t) = \rho \times N_i(t-1) + 1, \tag{1}$$

where $N_i(t) = N_i(t-1)$ is its observed count of interactions in time t $(t-1)$ of action i. Similar to the weighted fictitious play model, an agent form its belief to take action i by probability $p(a_i(n))$ expressed in (2) where the numerator is the product of $N_i(n)$ and total reward of action i and the denominator is the sum of product values of all the actions.

$$p(a_i(n)) = \frac{N_i(n)\sum_{t=1}^{n} r(a_i(t))}{\sum_{i=1}^{m}\sum_{t=1}^{n} N_i(t) r(a_i(t))}. \tag{2}$$

We evaluate the experience with reinforcement learning in (3), but we replace the policy with replicator dynamics because of the bounded rationality assumption. We skip the deduction of the utility of integrating RL and FP in this paper, which is illustrated in the work of Camerer and Hua [15]. We calculate the expected utility of an agent for each period according to

$$\begin{aligned} q(a_i(t)) &= [1 - \Phi \times p(a_i(t))] \times q(a_i(t-1)) \\ &\quad + \Phi \times p(a_i(t)) \times r(a_i(t)), \end{aligned} \tag{3}$$

where $q(a_i)$ denotes return of taking action a_i at time t. The expected utility of an agent is determined by the reward in

```
repeat
    $i = 0$
    Initialize $Q\ (s,\ a)$
    repeat
        Choose an action A using policy derived from $Q$ (e.g., $\varepsilon$-greedy)
        Choose an opponent randomly
        Take action A and observe $R$, $S'$
            $Q(S, A) \longleftarrow Q(S, A) + \alpha[R + \gamma \max_a Q(S', a) - Q(S, A)]$
            $S \longleftarrow S'$
    until $S$ is terminal
    $i = i + 1$.
until $i =$ the total number of all the agents
```

Algorithm 1: Q-learning.

```
repeat
    repeat
        $i = 1,\ t = 1$
            repeat
                Choose an opponent randomly
                    $N_i(t) = \rho \times N_i(t-1) + 1$
                Choose an action according to strategy $q(a_t)$
                    $q(a_i(t)) = [1 - \phi \times p(a_i(t))] \times q(a_i(t-1)) + \phi \times p(a_i(t)) \times \mathrm{r}(a_i(t))$
        $t = t + 1$
        until $t$ equals the maximum period number
    Update the probability of actions according to
        $\mathrm{d}\pi(a_i(t))/\mathrm{dt} = q(a_i(t))/q(t)$
          $i = i + 1$
    until $i =$ the total number of all the agents
until predefined iteration number
```

Algorithm 2: Experience weighted learning.

time t and the previous return in the last period, which is discounted by Φ, similar to the weight in reinforcement learning. $r(a_i)$ is the reward of action a_i at time t. Before the next period, the agent learns and modifies the probabilities of the policy. The dynamic equation of policy update is

$$\frac{\mathrm{d}\pi(a_i(t))}{\mathrm{dt}} = \frac{q(a_i(t))}{q(t)}, \tag{4}$$

where $\pi(a_i(t))$ is the policy mapping action a to probability and $q(t)$ is the expected return of all the actions of this agent.

4. Simulation

To test the performance of the proposed algorithm, we conduct numerical simulations by matching a pair of agents randomly to play a game. The initial parameter setting is $\rho = 0.8$ and $\Phi = 0.9$. We use 4 classic game models described in Table 1 to test the algorithm performance in different situations. The first one in Table 1 is the hawk dove game which involves tough (strategy b) and mild (strategy a) strategies between the players. The second one is the battle of the sexes that represents different preferences of players, where the row player prefers strategy a while the column player prefers strategy b. When both sides choose the same strategy, they can realize the Pareto efficiency; otherwise, they will obtain a lower payoff. This model is desirable for analyzing common interests with different preferences and focuses more on cooperation than the first model does. We use the stag hunt game to model the situation when the cooperation between players is highly valuable. The prison dilemma represents the condition when the Nash equilibrium is not Pareto efficient.

Each experiment is repeated 10 times to evaluate the performance EWL in different games since the game process is not deterministic. We set Q-learning as the baseline group. We evaluate the average payoff, strategy convergence, and payoff evolution in the process in the following sections.

4.1. The Average Payoff. We compare the average payoffs of 2 groups of agents, respectively, with column charts in Figure 1, including EWL and Q-learning. 4 games in Figure 1 are testbeds for the agents, and the results in Figure 1(a) are from hawk dove game, those in Figure 1(b) are from battle of the sexes, those in Figure 1(c) are from prison dilemma, and those in Figure 1(d) are from stag hunt.

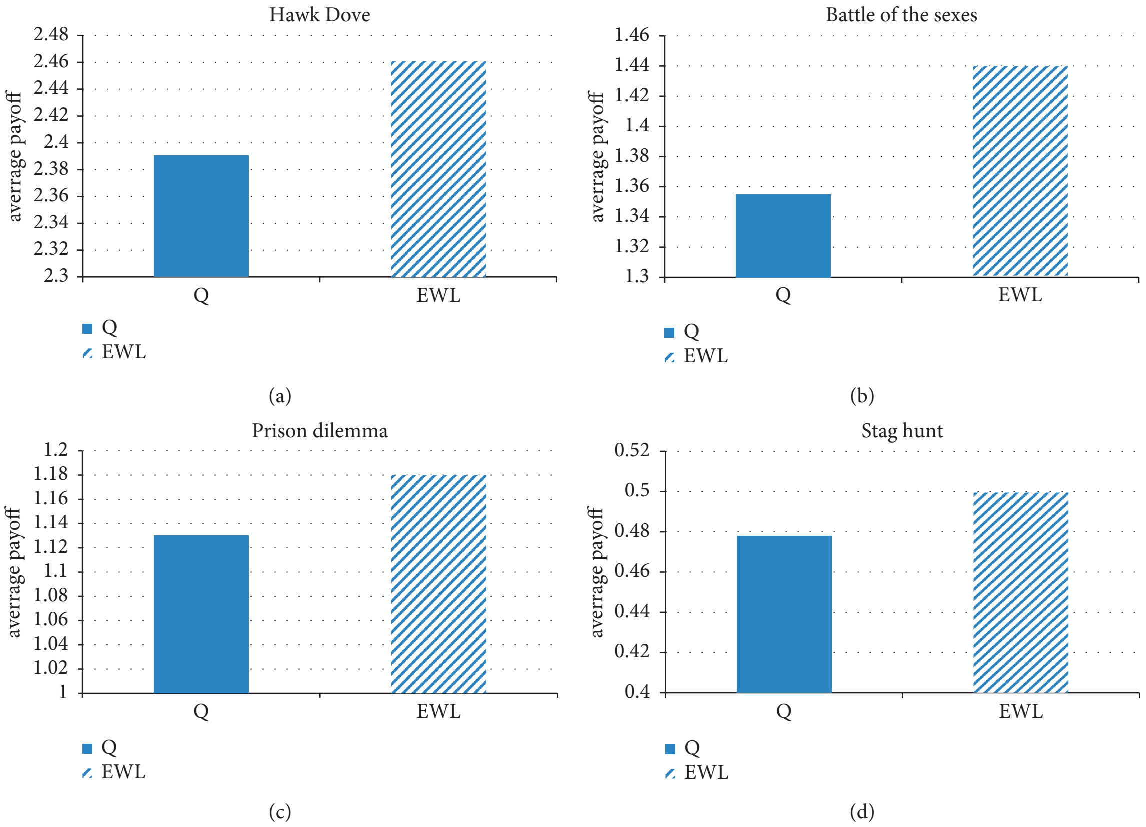


Figure 1: The average payoff: (a) Hawk dove; (b) battle of the sexes; (c) prison dilemma; (d) stag hunt.

We calculate the average payoff of all agents of the final period in 10 repeated experiments, and it is illustrated in Figure 1 that the EWL outperformed Q-learning in all 4 games. Since the major difference between EWL and RL lies in the evaluation of experience by integrating the FP and RL, the results demonstrate that mechanism of experience evaluation is more satisfactory.

4.2. The Equilibrium Distribution. Since we adopt replicator dynamics for policy update, which is equilibrium dynamics, it is necessary to discuss if the learning process converges to a stable status or fluctuates constantly. Therefore, we set a random initial state of strategy (action combination) for each agent and illustrate its final strategy distributions in Figure 2 of the 4 games mentioned above.

Each learning process stops till the period reaches a maximum number, and we repeat the process of each game 10 times. We calculate the final strategy distribution of EWL and illustrate it in Figure 2, where the x axis represents the number of agents converging to strategy a and y axis represents the number of agents converging to strategy b.

The red squares represent the strategy of column players in one population, and the blue triangles represent the row players in the other population. The points scatter along a straight line in Figure 2(a). This result indicates that most agents finally converge to a pure strategy since the sum of strategies a and b is equal or close to 20, which is the total number of agents in each population. The points in the other 3 games are concentrated in the corner, indicating that most agents converge to strategy a, and few of them converge to b. In addition, we find that there are less than 40 points on these charts because some points overlap. Note that the equilibria of hawk dove game are (a, b) and (b, a) while those for the other games are (a, a) or (b, b). Therefore, the convergence results match the equilibrium distribution of each game and most agents converge to equilibria from the perspective of population. We can find that very few agents fail to converge to pure strategy for the following possible reason. The terminal condition for convergence has a precision level which can stop some convergence occasionally. Second, there exist uncommon 0 denominators in the calculation that disturb the convergence. Moreover, not all the agents in a population converge to the same pure strategy because the initial state is fully random and the learning speeds are different for agents.

4.3. The Learning Processes. We find that EWL produces different strategy convergence distributions in Figure 2. Furthermore, we discuss the learning process when these results are formed in this section. We illustrate the average payoffs of 2 algorithms in 4 games evolved through periods, respectively, in Figure 3, where Q represents Q-learning. The average payoffs of EWL agents are higher than those of Q-learning in hawk dove, battle of the sexes, and stag hunt, where they increase smoothly at first and become flat gradually. It is slightly better and very close to prison

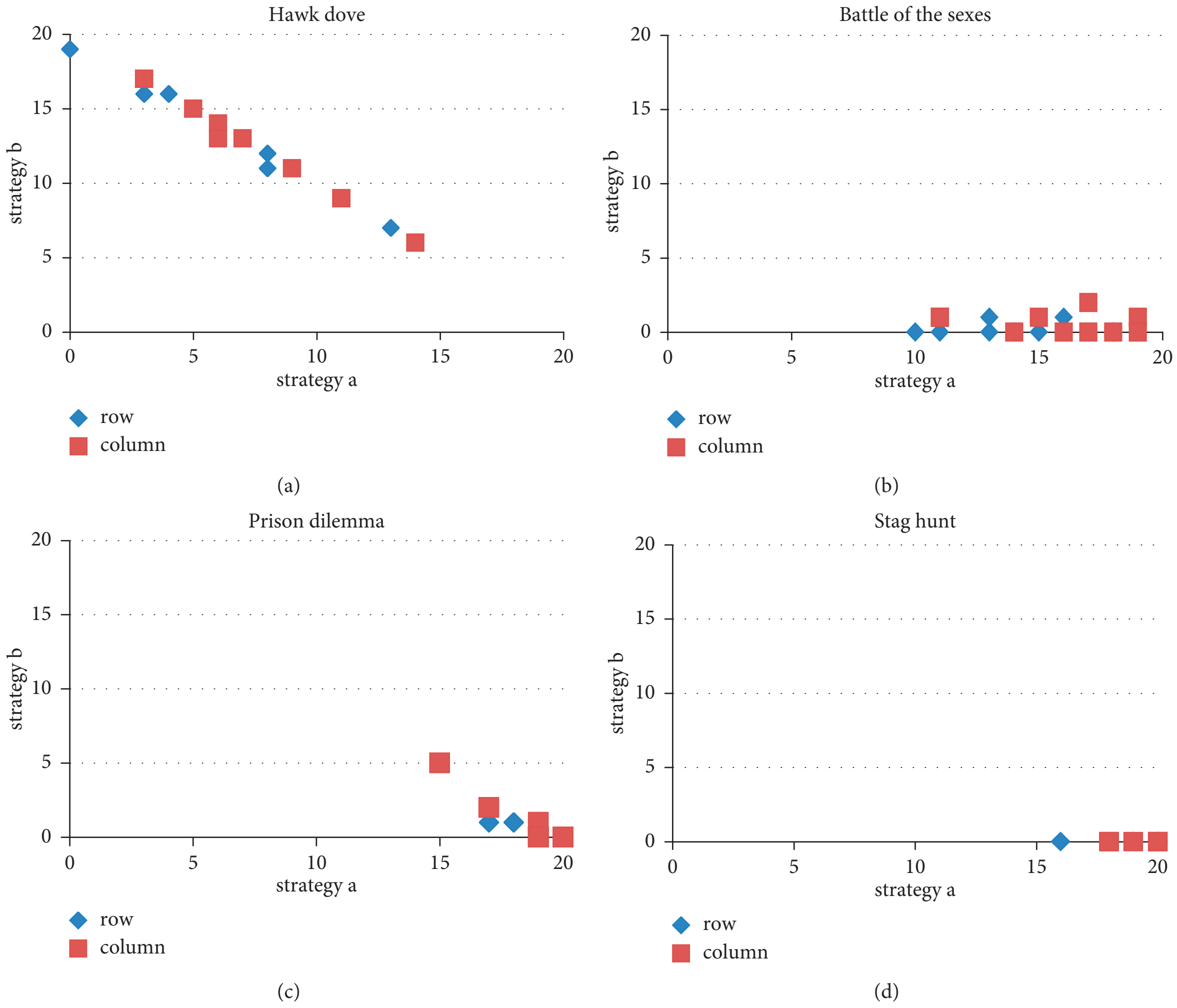


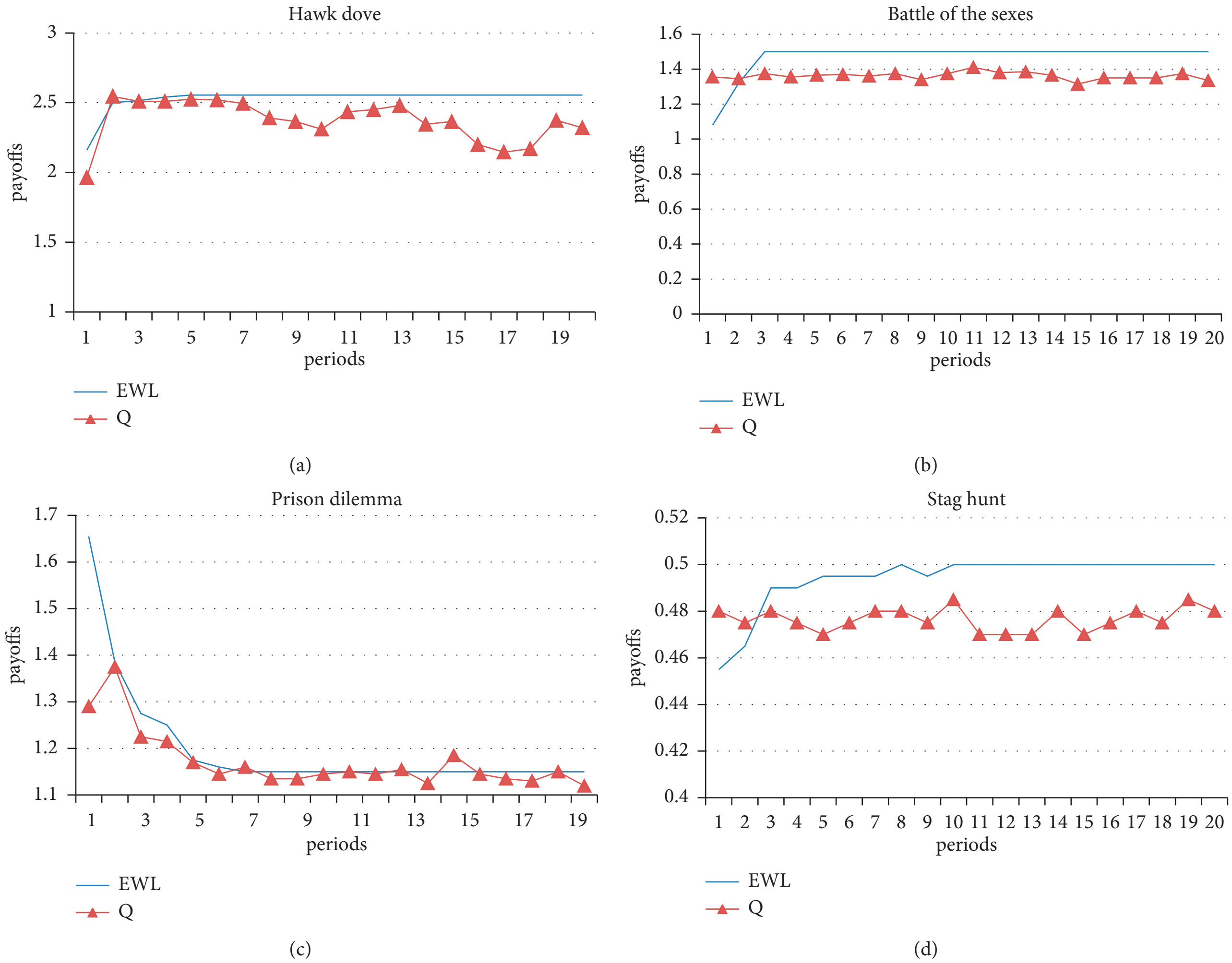
Figure 2: The equilibrium distributions: (a) hawk dove; (b) battle of the sexes; (c) prison dilemma; (d) stag hunt.

dilemma, where the payoffs decrease smoothly and become flat gradually. The results in the final period are consistent with the average payoffs in Figure 1. Since Q-learning is not an equilibrium learner, it fluctuates a little throughout periods while the payoffs of EWL finally become very smooth because most agents converge to pure strategies of equilibria.

4.4. The Impacts of Parameters. Since the parameters ρ and Φ affect the results of learning, it is important to evaluate and optimize them to improve the performance of agents. But the adaptive learning in multiagent learning is too complicated to find an arithmetic analytical solution; hence, we use simulation to test the impacts of the parameters in 4 different games in Table 1. We set 2 experiments to test the parameters ρ and Φ , respectively. We initially set $\rho = 0.8$ and $\Phi = 0.9$ and the iteration number $L = 20$. We repeat each experiment 10 times and calculate the average payoffs. In the first group, we change the value of ρ from 0 to 1 with an interval of 0.1, and then we observe the average profit of one population. Likewise, we change the value of Φ in the second group. The average payoffs in 4 games of 2 groups are illustrated in Figures 4 and 5 with the 4 lines, where battle represents the battle of the sexes, prison represents prison dilemma, hawk represents hawk dove, and stag represents stag hunt. We find that the impact of ρ is mild when Φ remains at a high level. The impacts of Φ are different for the 4 games, in which it affects the hawk dove game more than the others. Generally, the impact of parameters is limited and the performance of EWL is relatively stable with varying parameters. But there is a potential to improve its performance by parameter optimization.

5. Summary and Discussion

We introduce a novel learning algorithm to model the strategy adjustment of MARL agents. EW learning integrates reinforcement learning as well as fictitious play to efficiently utilize historical information for policy evaluation and update. We have conducted several simulation experiments to test the performance of EW learning in different games. The results demonstrate that the EWL agents outperform Q-learning in all 4 games. It is shown that EW learning is effective and more profitable in most cases of our experiments. We find that most agents converge to pure strategy and form equilibria from a



(a)

(b)

(c)

(d)

Figure 3: The learning processes: (a) Hawk dove; (b) battle of the sexes; (c) prison dilemma; (d) stag hunt.

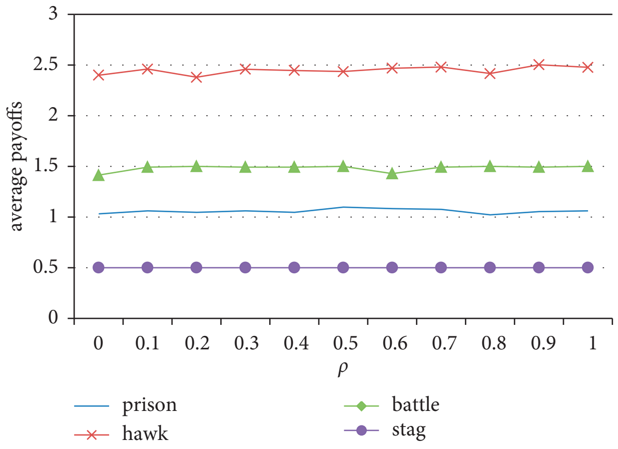


Figure 4: Average payoffs in 4 games varying with ρ.

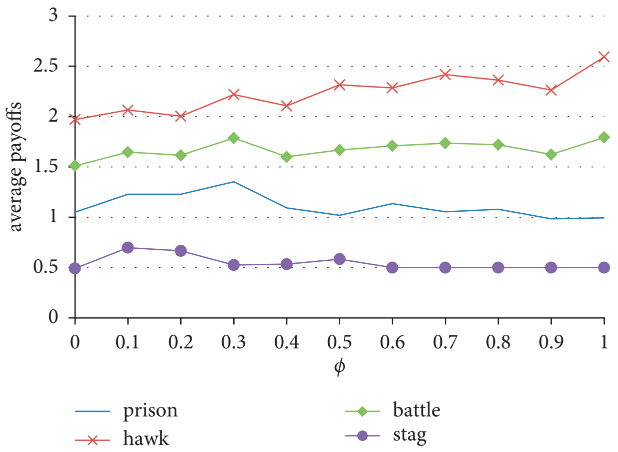


Figure 5: Average payoffs in 4 games varying with φ.

population perspective in all the games, which demonstrates the stability of EWL. In addition, we have observed the learning process and found that the payoffs change smoothly and become flat gradually through periods. Furthermore, the impacts of 2 important parameters are evaluated to test the efficiency of EWL and search for optimal values. But our experiment settings still have some limitations. For example, the agent amount of 40 is medium size, and we did not test the performance in a sparser or denser environment. Due to the time constraint of this research, we only examined the impact of the single parameter instead of parameter combinations. We are going to fill these gaps and extend our research to a wider range of game models in the future.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

We gratefully acknowledge the financial support received from the Humanities and Social Sciences Youth Foundation of the Ministry of Education of China (grant no. 18YJC630278), the Department of Education of Zhejiang Province (grant no. Y201737950), and the National Natural Science Foundation of China (grant no. 72172111).

References

[1] H. R. Leea and T. Leea, "Multi-agent reinforcement learning algorithm to solve a partially-observable multi-agent problem in disaster response," *European Journal of Operational Research*, vol. 291, no. 1, pp. 296–308, 2020.

[2] Y. S. Nasir and D. Guo, "Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 10, pp. 2239–2250, 2019.

[3] J. Hu and M. P. Wellman, "Nash Q-learning for general-sum stochastic games," *Journal of Machine Learning Research*, vol. 4, no. 11, pp. 1039–1069, 2003.

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

[5] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Basar, "Fully decentralized multi-agent reinforcement learning with networked agents," in *Proceedings of the International Conference on Machine Learning*, pp. 5872–5881, Stockholm, Swedan, July 2018.

[6] L. Xue, C. Sun, D. Wunsch, Y. Zhou, and F. Yu, "An adaptive strategy via reinforcement learning for the prisoner's dilemma game," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 1, pp. 301–310, 2017.

[7] Y. Hou, Y. S. Ong, L. Feng, and J. M. Zurada, "An evolutionary transfer reinforcement learning framework for multiagent systems," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 4, pp. 601–615, 2017.

[8] L. Liu and X. Chen, "Evolutionary game dynamics in multiagent systems with prosocial and antisocial exclusion strategies," *Knowledge-Based Systems*, vol. 188, Article ID 104835, 2020.

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

[10] K. Tuyls, T. Lenaerts, K. Verbeeck, B. Manderick, and S. Maes, "Towards a relation between learning agents and evolutionary dynamics," *Proceedings of BNAIC, Citeseer*, pp. 21-22, 2002.

[11] J. Z. Leibo, V. Zambaldi, and M. Lanctot, "Multi-agent reinforcement learning in sequential social dilemmas," in *Proceedings of the Sixteenth International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 464–473, Richland, SC, May 2017.

[12] H. Ding, G. S. Zhang, S. H. Wang, J. Li, and Z. Wang, "Q-learning boosts the evolution of cooperation in structured population by involving extortion," *Physica A: Statistical Mechanics and Its Applications*, vol. 536, Article ID 122551, 2019.

[13] N. Kamra, U. Gupta, K. Wang, F. Fang, F. Liu, and M. Tambe, "Deep fictitious play for games with continuous action spaces," in *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 2042–2044, Montreal QC, Canada, May 2019.

[14] Y. Viossat and A. Zapechelnyuk, "No-regret dynamics and fictitious play," *Journal of Economic Theory*, vol. 148, no. 2, pp. 825–842, 2013.

[15] C. Camerer and T. H. Hua, "Experience-weighted attraction learning in normal form games," *Econometrica*, vol. 67, no. 4, pp. 827–874, 1999.

[16] Z. Shan, S. Yongduan, F. L. Lewis, and A. Davoudi, "Optimal robust output containment of unknown heterogeneous multiagent system using off-policy reinforcement learning," *IEEE Transactions on Cybernetics*, vol. 48, no. 11, pp. 3197–3207, 2017.

[17] E. Nisioti and N. Thomos, "Robust coordinated reinforcement learning for MAC design in sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 10, pp. 2211–2224, 2019.

[18] W. Zhang, L. Ma, and X. Li, "Multi-agent reinforcement learning based on local communication," *Cluster Computing*, vol. 22, no. 6, Article ID 15366, 2019.

[19] M. J. Matarić, "Reinforcement learning in the multi-robot domain," *Robot Colonies*, vol. 4, no. 1, pp. 73–83, 1997.

[20] K. Yang, W. Wang, and B. Hu, "Evolutionary game models on multiagent collaborative mechanism in responsible innovation," *Scientific Programming*, vol. 2020, Article ID 8875099, 11 pages, 2020.

[21] T. Rashid, M. Samvelyan, C. Schroeder, G. Farquhar, J. Foerster, and S. Whiteson, "Qmix: monotonic value function factorisation for deep multi-agent reinforcement learning," in *Proceedings of the International Conference on Machine Learning, PMLR*, pp. 4295–4304, Stockholm, Sweden, July 2018.

[22] P. Sunehag, G. Lever, A. Gruslys et al., "Value-decomposition networks for cooperative multi-agent learning," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 2085–2087, Stockholm, Sweden, July 2018.

[23] Y. Yang, J. Hao, and B. Liao, "Qatten: a general framework for cooperative multiagent reinforcement learning. Working paper," 2020, https://arxiv.org/abs/2002.03939.

[24] K. Son, D. Kim, W. J. Kang, D. Hostallero, and Y. Yi, "QTRAN: learning to factorize with transformation for cooperative multi-agent reinforcement learning," in *Proceedings of the 2019 International Conference on Machine Learning*, pp. 5887–5896, Long Beach, California, USA, June 2019.

[25] S. Sen, M. Sekaran, and J. Hale, "Learning to coordinate without sharing information," in *Proceedings of the Twelfth National Conference on Artificial Intelligence, Seattle*, pp. 426–431, Washington, California, August 1994.

[26] C. Claus and C. Boutilier, "The dynamics of reinforcement learning in cooperative multiagent systems," in *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pp. 746–752, Madison, Wisconsin, July 1998.

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

[28] Z. Zhang, D. Wang, and J. Gao, "Learning automata-based multiagent reinforcement learning for optimization of cooperative tasks," *IEEE transactions on neural networks and learning systems*, pp. 1–14, 2020.

[29] H. Liu, Z. Zhang, and D. Wang, "WRFMR: a multi-agent reinforcement learning method for cooperative tasks," *IEEE Access*, vol. 8, Article ID 216331, 2020.

[30] Y. Yang, X. Wang, Y. Xu, and Q. Huang, "Multiagent reinforcement learning-based taxi predispatching model to balance taxi supply and demand," *Journal of Advanced Transportation*, vol. 2020, Article ID 8674512, 12 pages, 2020.

[31] J. Hao, H. F. Leung, and Z. Ming, "Multiagent reinforcement social learning toward coordination in cooperative multiagent systems," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 9, no. 4, pp. 1–20, 2015.

[32] H. Tang, L. Wang, Z. Wang et al., "An optimal rewiring strategy for reinforcement social learning in cooperative multiagent systems. Working paper," 2018, http://arxiv.org/abs/1805.08588v1.

[33] A. Greenwald and K. Hall, "Correlated-Q learning," in *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, pp. 242–249, Washinton DC, August 2003.

[34] M. L. Littman, "Value-function reinforcement learning in Markov games," *Cognitive Systems Research*, vol. 2, no. 1, pp. 55–66, 2001.

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

[36] M. M. Alipour, S. N. Razavi, M. R. F. Derakhshi, and M. A. Balafar, "A hybrid algorithm using a genetic algorithm and multiagent reinforcement learning heuristic to solve the traveling salesman problem," *Neural Computing & Applications*, vol. 30, no. 9, pp. 2935–2951, 2018.