https://doi.org/10.1093/comjnl/bxac027

Improvement of MADRL Equilibrium Based on Pareto Optimization

Zhiruo Zhao, Lei Cao*, Xiliang Chen, Jun Lai and Legui Zhang

Command & Control Engineering College, Army Engineering University of PLA, Nanjing, CO 210000 China

**Corresponding author: caolei.nj@foxmail.com*

In order to solve the incalculability caused by the issue of inconsistent objective functions in multi-agent deep reinforcement learning, the concept of Nash equilibrium is introduced. However, a Marko game may have multiple equilibriums, how to filter out a stable and optimal one is worth studying. Besides solution concept, how to keep the balance between exploration and exploitation is another key issue in reinforcement learning. On basis of the methods, which can converge to Nash equilibrium, this paper makes improvement through Pareto optimization. In order to alleviate the problem of over fitting caused by Pareto optimization and non-convergence caused by strategy change, we use stratified sampling in place of random sampling as assistance. What's more, our methods are trained through fictitious self-play to make full of self-learning experiences. By analyzing the experiment carried out on MAgent platform, the proposed methods are not only far better than traditional methods, but also reaching or even surpassing the state of art MADRL methods.

Keywords: MADRL; Nash equilibrium; pareto optimization; stratified sampling; fictitious self-play

Received 25 August 2021; Revised 8 December 2021; Editorial Decision 27 January 2022
Handling editor: Domenico Rosaci

1. INTRODUCTION

In multi-agent system, agents not only interact with the environment, but also affect each other, thus a stable strategy is hard to be learnt. In order to train agents who can directly learn from scratch, reinforcement learning is introduced. With the help of neural network, multi-agent reinforcement learning has tremendous successful applications in robotics, planning and real-time strategy games. On the other hand, considering multi-agent is devoted to analyzing the interaction between agents, even human beings under different incentives and economic constraint, it is natural to find inspirations in game theory, which is born to study the decisions made by multiple parties with individual preferences, capabilities and information.

Minimax [1] is the very first method in game theory applied in multi-agent reinforcement learning, however, its only appropriate for two-person zero-sum game. In particular, perfectly competitive zero-sum game is only a special model in game theory, there has much more general-sum games in which agents can either cooperate or compete with others to obtain their own maximum returns. In general-sum games, the goal of each agent is different, thus the return functions are no longer all the same. Due to lacking a unified objective function, there is no optimal solution in many cases. Spontaneously, Nash equilibrium is introduced [2]. With assistance of the new solution concept, multi-agent reinforcement learning methods solve more complex problems.

However, multi-agent reinforcement learning based on Nash equilibrium is only suitable for two-person games. When the number of agents increasing, the state space, together with action space exploding, Nash Q value cannot be calculated by traditional methods. Some mathematical methods such as mean field theory, is adopted to alleviate the issue of dimension explosion. Let us see back to Nash equilibrium, according to the definition, there may be multiple Nash equilibrium with mixed strategy in a finite game [3], which means the quality of equilibrium is different. Some of the equilibriums are weakly strategy dominated, whereas the others may be strongly strategy dominated, how to eliminate the poor solution and filter out the better one is of great value.

Exploration and exploitation are one of the key issues in reinforcement learning. In reinforcement learning, the purpose of exploration is to find more information about the environment, whereas exploitation is to use the known environment information to maximize the reward. Different from the traditional exploration strategies, whose goal is to look at the problem from a long-term perspective and find the global optimal solution. In multi-agent reinforcement learning game with Nash equilibrium, the goal of exploration is to make refinement in order to obtain more stable and optimal equilibrium.

Pareto optimization is also called pareto improvement. The main idea is to make at least one person better without making anyone worse on the premise of pareto optimal change. In other words, pareto optimization is to increase one's own payoff without harming others' interests. Although pareto improvement can quickly make the Q value close to the possible optimal, it can also lead to over estimation easily. In order to solve this problem, we adopt stratified sampling in place of random sampling to sample the experience data according to a fixed proportion, which not only retains the stability, but also alleviate the variance.

Fictitious self-play [4] is a sample-based variant of fictitious play [5], in which, the player repeats a game to select the best response to the opponent's average strategy in each iteration. In some particular type of game, the average strategy distribution of fictitious players can converge to Nash equilibrium. The algorithm laid the foundation for self-learning experience, had a great impact on multi-Agent reinforcement learning algorithm, and combined with sufficient computing power, produced many exciting results including Alpha Zero.

In this paper, we propose Pareto Q-learning and Pareto-AC based on Mean Field Multi Agent Reinforcement Learning (MFMARL) [6] methods with trembling hand perfect improvement. Owing to pairwise [7] and mean field theory [8], the interaction of groups is simplified as the interaction between a single agent and its neighbors. The joint actions between agents are simplified by the calculated average actions and their own actions. The joint action space, together with the complexity of the problem are both greatly reduced. Although pairwise significantly reduces the complexity of interaction between agents, it reserves the global interaction between any pair of agents implicitly. Through trembling hands perfect strategy exploration, the weakly dominated strategy is eliminated and dominated strategy with stable Nash equilibrium is reserved. Our experiment is carried out on MAgent platform [9]. During the training stage, the network parameters are trained and saved through fictitious self-play. During the test, the trained model is used to battle with each other in pairs. The experimental results show that the algorithm proposed in this paper is far better than the traditional algorithm, reaching or even surpassing the latest MADRL algorithm in winning rate, while it takes longer training times.

Our contributions are listed as follows:

1. Pareto improvement is applied in exploration and exploitation to filter out the optimal Nash equilibrium.
2. Stratified sampling instead of random sampling is introduced to solve the overfitting and alleviate the greater variance problem to some degree.
3. The proposed methods are trained through fictitious self-play to make full of self-learning experiences.

This paper is organized as follows: Section 2 sketches the related work; Section 3 introduces the theoretical basis involved in this paper and our proposed algorithms in form of pseudocode and frame diagram; Section 4 elaborates our experiments on MAgent platform and in Section 5, we make a summary and put forward the future research direction.

2. RELATED WORK

In Littman's groundbreaking paper, Minimax Q-learning [1] are proposed to study two agents who have opposite goals, which adopts a single return function to maximize one's rewards while minimizing the other's rewards. Hu and Wellman first proposed Nash Q-learning [2], expanding Minimax Q from zero-sum game to general-sum game. Solving Nash Q in general-sum game is mainly Lemke Howson algorithm and its extended algorithm, which is suitable for two-person game [10]. For general-sum Markov games, Littman also proposes a Friend-or-Foe Q-learning (FFQ) [11], in which the agent classifies other players as friends or foes. When all states and behaviors can be accessed infinitely, FFQ algorithm is proved to converge to Nash equilibrium. With the combination of reinforcement learning and neural network, the equilibrium of deep reinforcement learning has become the focus of research. Mean field multi-agent reinforcement learning [6] is mainly dedicated to solving the interaction and computational difficulties between large-scale agents. Combined with the mean field theory, it proposes two main algorithms MF-Q and MF-AC. This paper gives the convergence proof in detail to prove that the proposed method can converge to the Nash equilibrium point. MFMARL algorithm can be applied both in competitive and cooperative environment.

In the issues of exploration and exploitation in reinforcement learning, Noise is widely used. Plappert [12] proposes that adding noise should not only be limited to the output of the network, but also try to add noise to the network parameters θ to increase exploration. In Fortunato's paper [13], the parameters of noise are learned by gradient descent based on reinforcement learning loss, rather than adjusted by heuristic method. This can realize end-to-end learning, but increases the number of parameters and computational overhead. Beside noise, count-based exploration is another research direction. In Choi's paper [14] the region related to the action are extracted from the original image observation, and then use count-based to calculate the internal excitation. Based on the above count-based theory, Ostrovski [15] puts forward some practical suggestions on the specific use of probability model $\rho(x)$, and obtains good results.

Pareto Optimization is of great importance as well. Ngatchou [16] provides basic knowledge for solving multi-objective optimization problems. The key is the intelligent meta heuristic method (evolutionary algorithm or group-based technology), and the key is the technology to effectively generate the Pareto optimal frontier. Lin [17] is among the first to provide a Pareto efficient framework for multi-objective recommendation with theoretical guarantees. Moreover, the framework can be applied to any other objectives with differentiable formulations and any model with gradients, which shows its strong scalability.

There are many methods to reducing high variance issues. Common random numbers are often used to compare the merits between two systems. The core idea of the antithetic variates is to generate another new random number based on one generated random number, which changes in the opposite direction. The idea of stratified sampling is to divide the definition field of the random variable into unsected regions, then sample it in each region and use all the values used to estimate expectations, where the weights of each section are the ratio of the regions to the defined domain measurements like length, area and volume.

3. METHOD

3.1. Nash Q-learning

The equilibrium strategy of Multi Agent Reinforcement Learning (MARL) is generally presented by a specific joint strategy: $\pi_* \triangleq \left[\pi_*^1, \cdots \pi_*^N\right]$, for all $s\epsilon S, j\epsilon\left\{1, \cdots, N\right\}$ and all valid π^j, it satisfies:

$$v^j\left(s; \pi_*\right) = v^j\left(s; \pi_*^j; \pi_*^{-j}\right) \geq v^j\left(s; \pi^j; \pi^{-j}\right) \# \tag{1}$$

In which, $\pi_*^{-j} \triangleq \left[\pi_*^1, \cdots, \pi_*^{j-1}, \pi_*^{j+1}, \cdots, \pi_*^N\right]$. π represents a certain strategy, π_* represents strategy sets, π_*^j represents a strategy set of j, s represents a certain state, Srepresents state set and $-j$ represents all the other agents besides j.

In Nash equilibrium, agents choose actions according to their best response π_*^j on condition that other agents follow π_*^{-j}. It has been proved that there is at least one Nash equilibrium solution in a stochastic n-person game [3].

Given a fixed Nash strategy π_*, all agents start from the initial state s and follow the policy π_*, Nash Q value $v^{nash}(s)$ is calculated as below:

$$v^{nash}(s) \triangleq \left[v_{\pi_*}^1(s), \cdots, v_{\pi_*}^N(s)\right] \# \tag{2}$$

And the corresponding Nash Q function $Q^{nash}\left(s, a\right)$ is:

$$Q^{nash}\left(s, a\right) = E\left[r\left(s, a\right) + \gamma v^{nash}\left(s'\right)\right] \# \tag{3}$$

Among which, a represents action and γ is discount factor, $\gamma \in \left[0, 1\right]$.

3.2. MFMARL with trembling hand perfect improvement

With the increasing number of agents and joint action space, agent still has to learn its own strategy together with the actions of other agents, leading the calculating of Q functions becomes more complex. To solve this problem, the Q function of agent i is factorized by pairwise interactions:

$$Q^i\left(s, a\right) = \frac{1}{N^i} \sum_{j\in\mathcal{N}(i)} Q^i\left(s, a^i, a^j\right) \# \tag{4}$$

Among which, $N^i = \mid \mathcal{N}(i) \mid$,$\mathcal{N}(i)$ is a set of neighbors of agent i. After taking advantage of mean field approximation, the interaction between agents i and its adjacent agents, which is represented by j, can be simplified as interaction between agents i and the mean value of its neighbors.

Mean field Q function is updated as below:

$$Q_{t+1}^i\left(s, a^i, \overline{a}^i\right) = \left(1-\alpha_t\right) Q_t^i\left(s, a^i, \overline{a}^i\right) + \alpha_t\left[r_t^i + \gamma V_t^i\left(s'\right)\right] \# \tag{5}$$

where α_t denotes the learning rate, and $\overline{a}^i$ is the mean action of other agents. The mean field value function for agent i at time t is:

$$v_t^i\left(s'\right) = \sum_{a^i} \pi_t^i\left(a^i|s', \overline{a}^i\right) \mathbb{E}_{\overline{a}^i\left(a^{-i}\right)\sim\pi_t^{-i}}\left[Q_t^i\left(s, a^i, \overline{a}^i\right)\right] \# \tag{6}$$

The mean action $\overline{a}^i$ of all i's neighbors is:

$$\overline{a}^i = \frac{1}{N^i} \sum_k a^k, a^k \sim \pi_t^k\left(\bullet|s, \overline{a}^k\right) \# \tag{7}$$

After calculating $\overline{a}^i$, the new Boltzmann policy is:

$$\pi_t^i\left(a^i|s, \overline{a}^i\right) = \frac{\exp\left(-\beta Q_t^i\left(s, a^i, \overline{a}^i\right)\right)}{\sum_{a^{i'}\epsilon\mathcal{A}^i} \exp\left(-\beta Q_t^i\left(s, a^i, \overline{a}^i\right)\right)} \# \tag{8}$$

According to the above function, mean field value is:

$$v^{MF}(s) \triangleq \left[v^1(s), \cdots, v^N(s)\right] \# \tag{9}$$

And the Q function is:

$$Q^{MF}\left(s, a\right) = E\left[r\left(s, a\right) + \gamma v^{MF}\left(s'\right)\right] \# \tag{10}$$

The state-action value function is fitted by neural network, and the loss function is used to reduce the error between the fitted Q value and the real Q value:

$$L\left(\varnothing_j\right) = \left(y_j - Q_{\varnothing_j}\left(s, a_j, \overline{a}_j\right)\right)^2 \# \tag{11}$$

Among which, $y_j = r_j + \gamma v_{\overline{\varnothing}_j}^{MF}\left(s'\right)$ is the value function of MF-Q,$\overline{\varnothing}_j$ is the parameters of target networks in DQN. By deriving from the above equation, the parameter gradient direction can be obtained as follows:

$$\nabla_{\varnothing_j} L\left(\varnothing_j\right) = \left(y_j - Q_{\varnothing_j}\left(s, a_j, \overline{a}_j\right)\right) \nabla_{\varnothing_j} Q_{\varnothing_j}\left(s, a_j, \overline{a}_j\right) \# \quad (12)$$

MF-AC is similar to Deep Policy Gradient (DPG), in which the strategy gradient equation of Actor network can be written as:

$$\nabla_{\theta_j} L\left(\theta_j\right) = \nabla_{\theta_j} \log \pi_{\theta_j}(s) Q_{\varnothing_j}\left(s, a_j, \overline{a}_j\right)\Big|_{a=\pi_{\theta_j}(s)} \# \quad (13)$$

The updating method of Critic network is similar to MF-Q.

Trembling hand perfect equilibrium is a stable Nash equilibrium, which was proposed by German economist Selten in 1975. He believes that there is a possibility that game participants make mistakes in decision-making. Assuming that participants deviate from the original equilibrium with a small probability and choose other strategies, if the original equilibrium can still be maintained, the equilibrium is stable, which is called the trembling hand perfect equilibrium. In Eric Rasmussen's classic book *Game and Information: an Overview of Game Theory* [18], the definition of the trembling hand perfect equilibrium in a limited action set game is as follows:

Definition 1. *The policy set π^* is a trembling hand perfect equilibrium, if for anyε,there exists a positive vector $\delta_1, \cdots, \delta_n$ and a fully mixture policy vector $\sigma_1, \cdots, \sigma_n$ in [0,1]. After each original policy replaced by a new policy $\left(1-\delta_i\right)\pi_i + \delta_i\sigma_i$, the new game still has a Nash equilibrium, and the distance between new policy and π^* is less than ε.*

According to Equation (8) and the definition of Trembling Hand Perfect (THP) equilibrium, the new policy is:

$$\pi_t^i(THP) = \left(1-\delta_i\right)\pi_t^i\left(a^i|s, \overline{a}^i\right) + \delta_i\sigma_i \# \quad (14)$$

In order to distinguish the mean field value function represented by Equation (9), after selecting the action according to the trembling hand perfect strategy, the THP value function is:

$$v^{THP}(s) \triangleq \left[v^1(s), \cdots, v^N(s)\right] \# \quad (15)$$

The mean field state-action value function represented by Equation (10) is modified as:

$$Q^{THP}(s,a) = E\left[r(s,a) + \gamma v^{THP}\left(s'\right)\right] \# \quad (16)$$

3.3. Pareto improvement

Pareto optimization is to increase one's own payment without harming the interests of others. It is the ideal kingdom of efficiency. From the perspective of optimization goals, Pareto optimality looks at the whole situation, whereas Nash equilibrium only focuses on the part. From the perspective of constraints, Pareto optimality can find the optimal solution in the global range, and Nash equilibrium can only change its own dimension. In order to distinguish whether Nash equilibrium has Pareto, it needs to be specified. Here, we consider n-person finite random games [19].

Suppose participant p has m_p actions, $1 \le m_p \le \infty, p = 1, 2, \ldots, n$. The payoff function of n-person finite game is composed of $n+1$ dimensional array A: $m_1 \times \cdots \times m_n \times n$. The last dimension of the array corresponds to the set of participants: the payment function $A(\cdot, \ldots, \cdot, p)$ of player p is represented by the sub array $m_1 \times \cdots \times m_n$ of A. If a pure strategy Nash equilibrium $i^* = \left(i_1^*, \ldots, i_n^*\right)$ exists, the number of pure strategies $i = i_1, \ldots, i_n$ that will dominate it is defined as his degree $d\left(i^*\right)$. When $d\left(i^*\right) = 0$, the pure strategy Nash equilibrium is the pure strategy Pareto optimal. When $d\left(i^*\right) > 0$, pure strategy Nash equilibrium is Pareto invalid. $d\left(i^*\right)$ is used to measure the Pareto efficiency of pure strategy Nash equilibrium (but this index ignores any mixed strategy Nash equilibrium that may dominate pure strategy Nash equilibrium).

In well-defined games, the conclusion on the Pareto effectiveness of Nash equilibrium is generally recent. In the game induced by market mechanism, if there are continuous participants and other conditions are met, the Nash equilibrium is Pareto optimal. When the number of participants is limited, Nash equilibrium is usually not Pareto optimal. In the n-person non cooperative game with smooth payment function, the action set of each participant is a finite dimensional simplex, and the Nash equilibrium is usually not Pareto optimal.

Definition 2. *Hypothetical strategy game $\Gamma = \left(N, \left(S_i\right)_{i\in N}, \left(u_i\right)_{i\in N}\right)$ has a finite player set $N = \{1, \ldots, n\}$ [20]. Participants $i \in N$ has a set of strategies S_i, state $s \in S = S_1 \times \cdots \times S_n$ is sometimes referred to as a policy summary or summary. The cost function of participant i is $c_i : S \to R$, map each state $s \in S$ to a real number. For other participants, $s_{-i} = \left(s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n\right)$。If there is no subset $I \subseteq U$, $\mid I \mid \le K$,can jointly benefit from the new strategy deviating from the equilibrium strategy, then state $s \in S$ is a K-Strong Equilibrium. Formally, there is no tuple $\left(s', I\right) \in S \times 2^N$, in which $s' \ne s$ and $\mid I \mid \le K, \forall i \in I, c_i\left(s'\right) < c_i(s), \forall i \in I$ covers all $s_i = s_i'$. N-SE is called strong equilibrium, 1-SE is called Nash equilibrium. Pareto optimal Nash equilibrium is a new equilibrium in which there is no state s' satisfies $c_i\left(s'\right) < c_i(s)$, for all $i \in N$.*

Supposing the strategy set π is a Nash equilibrium strategy. At time t, select joint action $a = \left[a^1, \ldots, a^N\right]$ according to π and get the reward set $r = \left[r^1, \ldots, r^N\right]$. If for any ε,there is a positive vector $\delta_1, \cdots, \delta_n$ on [0,1] and a completely mixed strategy vector $\sigma_1, \cdots, \sigma_n$, making the new game π^*replaced by the strategy $\left(1-\delta_i\right)\pi_i + \delta_i\sigma_i$ has a Nash equilibrium. At the same time t, select the joint action $a^* = \left[a^{*1}, \ldots, a^{*N}\right]$ according to π^*, and get the reward set$r^* = \left[r^{*1}, \ldots, r^{*N}\right]$. If

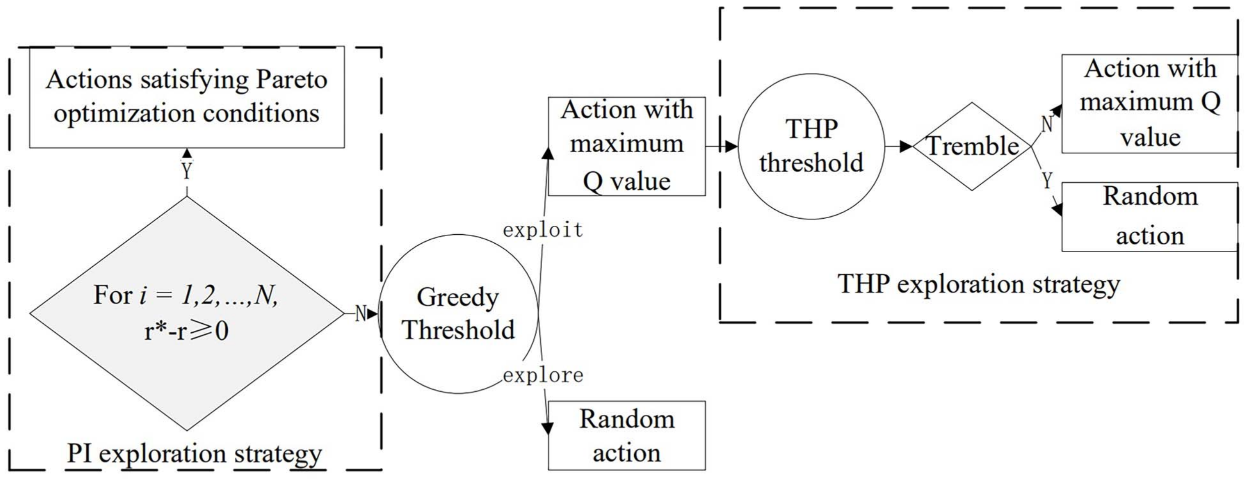
Actions satisfying Pareto optimization conditions
Y
For i = 1,2,...,N, r*-r⩾0
N
PI exploration strategy
Greedy Threshold
exploit
explore
Action with maximum Q value
Random action
THP threshold
Tremble
N
Y
Action with maximum Q value
Random action
THP exploration strategy

FIGURE 1. Pareto improvement strategy selection method.

TABLE 1. Basic parameters of the environment.

Amounts of agents	64 vs 64
Map size	40
Action	Move, attack
Rewards	−0.005 move; 0.2 attack an enemy; 5 kill an enemy; −0.1 attack an empty grid; −0.1 be attacked or killed; 1 attack an enemy with your teammates at the same time

each reward value in r^* is no less than r, then strategy set π^* is a Pareto improvement of strategy set π. That is,for $i = 1,2,...,N$:

$$r^{*i}_t - r^i_t \geq 0\# \tag{17}$$

According to the principle of Pareto improvement, the value function is:

$$v^{pareto}(s) \triangleq \left[v^{*1}(s), \cdots, v^{*N}(s)\right]\# \tag{18}$$

The state-action function is:

$$Q^{pareto}(s,a) = E\left[r(s,a) + \gamma v^{pareto}(s')\right]\# \tag{19}$$

This paper first pre trained a model, which can converge to the stable Nash equilibrium solution through the trembling hand perfect improved MFMARL method. The significance of pre training is that the calculation of Pareto improvement is complex and needs to occupy a lot of time and space. The time cost of Pareto optimization at the beginning is too high, which reduces the efficiency and feasibility of the algorithm. After pre training, the strategy will be adjusted in the exploration stage according to the Pareto improvement constraints. When the model converges again, the Pareto improved Nash equilibrium solution is obtained. Figure 1 shows the strategy selection method of Pareto improvement.

Figure 2 shows the framework of the improvement MADRL equilibrium methods based on Pareto improvement in detail. Starting from the initial state, agent first selects its own action through the improved exploration strategy of pareto improvement, and then calculates the average action a of its neighbors according to the average field approximation $\overline{a}$. Then, by interacting with the environment, the objective function calculates the loss update Q value, the reward function calculates the reward value obtained by each agent and then enters the next state according to the state transition function. Finally, the state set generated by interacting with the environment is saved in the replay buffer pool for later training. In order to alleviate the over fitting problem in training, stratified sampling is applied in replace of random sampling according to health point and rewards.

The pseudocodes of Pareto-Q and Pareto-AC are as below.

4. EXPERIMENT

4.1. Environment

We choose the mixed cooperative-competitive battle game in Magent [6] as our scenario, where two armies fighting against each other in a grid world. Each army adopt a reinforcement learning algorithm on their own and the goal of each army is to

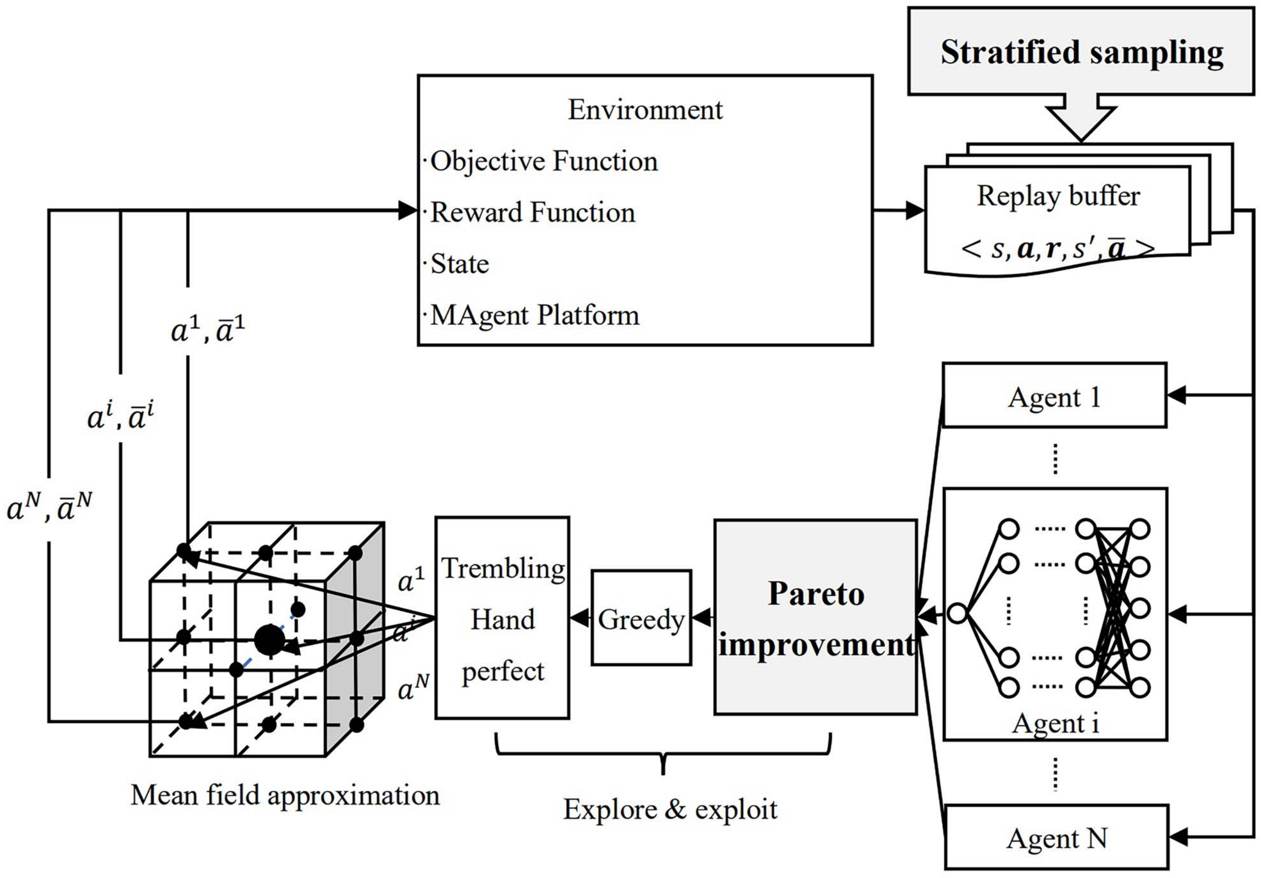

FIGURE 2. The frame diagram of MADRL equilibrium methods based on Pareto improvement.

get more rewards by collaborating to destroy all the opponents while minimizing its own loss and battle times.

Table 1 shows the basic parameters of the environment.

In order to prove the merits of our approach, we also increase the amounts of agents to 128 and enlarge the map size to 60 without changing action and rewards to make further analysis.

4.2. Baselines

In order to verify our algorithm, we set the traditional IL, AC, THP-Q, THP-AC and the latest algorithms Multi Agent Deep Deterministic Policy Gradient (MADDPG) and Multi Agent Proximal Policy Optimization (MAPPO) as baselines for comparison. For each baseline, we pre train the model through fictitious self-play and save it in local. After the training of the algorithm proposed in this paper, Pareto-Q and Pareto-AC will fight 10 groups with the pre trained baseline respectively, each group containing 50 rounds of combat. The total numbers of casualties of the two models in each round are recorded, and the win rate after each battle is calculated.

IL: Every agent in an army applies DQN [21] algorithm without communicating or cooperating with others.

AC: Every agent in an army applies advantageous AC [22] algorithm without communicating or cooperating with others.

MDDPG [23]: It is an AC architecture algorithm and can be regarded as an extension of DDPG in multi-agent environment. The experience replay pool of each agent is responsible for storing the training experience tuple:

$$D_i = \left(o_1, \ldots, o_N, a_1, \ldots, a_N, r_1, \ldots, r_N, o'_1, \ldots, o'_N\right) \# \tag{20}$$

Experience replay pool provides network training by randomly sampling small samples, which reduces the correlation between training experiences and improves the efficiency of network training. The algorithm updates the actor network of each agent by gradient descent:

$$\nabla_{\theta_i} J(\mu_i) = \mathbb{E}_{o,a\sim D_i} \times \left[\nabla_{\theta_i} \mu_i (a_i | o_i) \nabla_{a_i} Q_i^{\mu} (o_1, a_1, \ldots, o_N, a_N)\Big|_{a_i=\mu_i(o_i)} \right] \# \tag{21}$$

where D_i represents the ER of agent i. The critical network of each agent is updated iteratively by minimizing the loss

Algorithm 1 Pareto-Q-Learning (Pareto-Q)

```
Initialize $Q_{\emptyset^i}, Q_{\emptyset_-^i}, \bar{a}^i$, i = 1, 2, …, N
While training
  For each agent $i$, sample action $a^i$ based on THP strategy and compute new mean action $\bar{a}^i$
  Execute joint action $\boldsymbol{a} = [a^1, \dots, a^N]$, receive joint reward $\boldsymbol{r} = [r^1, \dots, r^N]$ and next state $s'$
  Store $< s, \boldsymbol{a}, \boldsymbol{r}, s', \bar{\boldsymbol{a}} >$ in replay buffer $\mathcal{D}$
  For i = 1, …, N do
    Sample a minibatch of $K$ experiences from $\mathcal{D}$
    Sample action $a_-^i$ from $\emptyset_-^i$ with $a_-^i \leftarrow \bar{a}^i$
    Update the Q-network according to equation (12)
  End for
  Update the parameters of the target network for each agent: $\emptyset_-^i \leftarrow \tau\emptyset^i + (1-\tau)\emptyset_-^i$
End while
While Pareto improvement
  For each agent $i$, sample action $a^i$ based on Pareto strategy and compute new mean action $\bar{a}^i$
  Execute joint action $\boldsymbol{a} = [a^1, \dots, a^N]$, receive joint reward $\boldsymbol{r} = [r^1, \dots, r^N]$ and next state $s'$
  Store $< s, \boldsymbol{a}, \boldsymbol{r}, s', \bar{\boldsymbol{a}} >$ in replay buffer $\mathcal{D}$
  For i = 1, …, N do
    Using stratified sampling to sample a minibatch of $K$ experiences from $\mathcal{D}$
    Sample action $a_-^i$ from $\emptyset_-^i$ with $a_-^i \leftarrow \bar{a}^i$
    Update the Q-network according to equation (12)
  End for
  Update the parameters of the target network for each agent: $\emptyset_-^i \leftarrow \tau\emptyset^i + (1-\tau)\emptyset_-^i$
End while
```

function:

$$L(\theta_i) = \mathbb{E}_{o,a,r,o'}\left[\left(Q_i^{\mu}(o_1, a_1, \dots, o_N, a_N) - y\right)^2\right]\# \quad (22)$$

$$y = r_i + \gamma Q_i'^{\mu'}\left(o_1', a_1', \dots, o_N', a_N'\right)\Big|_{a_i' = \mu_i'(o_i')}\# \quad (23)$$

MAPPO [24]: MAPPO adopts centralized training and distributed execution. When the algorithm inputs critical, it is not only its own state-action information, but also the information of other agents for joint update training.

In MAPPO algorithm, the policy parameter is $\theta = \{\theta_1, \cdots, \theta_N\}$ and the set of all agent policies is $\pi = \{\pi_1, \cdots, \pi_N\}$. Then we can write the gradient of agent i's expected return $J(\theta_i) = E[R_i]$ as:

$$\nabla_{\theta_i} J(\theta_i) = E_{s\sim p^{u}, a_i \sim \pi_i}\left[\nabla_{\theta_i} \log \pi_i(a_i|o_i)\, Q_i^{\pi}(X, a_1, \cdots, a_N)\right]\# \quad (24)$$

where $Q_i^{\pi}(X, a_1, \cdots, a_N)$ is a centralized action value function, which combines the actions of $(a_1, \cdots, a_N)$ all agents, adding some status information X as input and then take the Q value of Agent i as output.

THP-Q and **THP-AC**, as elaborated in Section 3, are no longer described in detail.

4.3. The setting of hyperparameter

Tables 2 and 3 show the hyperparameters of Pareto-Q and Pareto-AC, respectively, and when the number of agents and map size change, the hyperparameters remain unchanged.

4.4. Results and discussion

4.4.1. Training results

Figure 3 shows the cumulative rewards scatter diagrams of value-based Q-learning algorithm and policy-based AC algorithm before and after Pareto improvement when the map size is 40, whereas Fig. 4 shows the cumulative rewards scatter diagrams when the map size is 60. Among them, red dots represent the cumulative rewards before Pareto improvement,

Algorithm 2 Pareto-Actor-Critic (Pareto-AC)

```
Initialize $Q_{\emptyset^i}, Q_{\emptyset^i_-}, \pi^i_\theta, \pi^i_{\theta^-}, \bar{a}^i$, i = 1, 2, …, N
While training
  For each agent i, sample action $a^i$ based on THP strategy and compute new mean action $\bar{a}^i$
  Execute joint action $\boldsymbol{a} = [a^1, \ldots, a^N]$, receive joint reward $\boldsymbol{r} = [r^1, \ldots, r^N]$ and next state $s'$
  Store $< s, \boldsymbol{a}, \boldsymbol{r}, s', \bar{\boldsymbol{a}} >$ in replay buffer $\mathcal{D}$
  For i = 1, …, N do
    Sample a minibatch of $K$ experiences from $\mathcal{D}$
    Sample action $a^i_-$ from $\emptyset^i_-$ with $a^i_- \leftarrow \bar{a}^i$
    Update the Critic network according to equation (12)
    Update the Actor network according to equation (13)
  End for
  Update the parameters of the target network for each agent:
      $\emptyset^i_- \leftarrow \tau_\emptyset \emptyset^i + (1 - \tau_\emptyset)\emptyset^i_-$
      $\theta^i_- \leftarrow \tau_\theta \theta^i + (1 - \tau_\theta)\theta^i_-$
End while
While Pareto improvement
  For each agent i, sample action $a^i$ based on Pareto strategy and compute new mean action $\bar{a}^i$
  Execute joint action $\boldsymbol{a} = [a^1, \ldots, a^N]$, receive joint reward $\boldsymbol{r} = [r^1, \ldots, r^N]$ and next state $s'$
  Store $< s, \boldsymbol{a}, \boldsymbol{r}, s', \bar{\boldsymbol{a}} >$ in replay buffer $\mathcal{D}$
  For i = 1, …, N do
    Using stratified sampling to sample a minibatch of $K$ experiences from $\mathcal{D}$
    Sample action $a^i_-$ from $\emptyset^i_-$ with $a^i_- \leftarrow \bar{a}^i$
    Update the Critic network according to equation (12)
    Update the Actor network according to equation (13)
  End for
  Update the parameters of the target network for each agent:
      $\emptyset^i_- \leftarrow \tau_\emptyset \emptyset^i + (1 - \tau_\emptyset)\emptyset^i_-$
      $\theta^i_- \leftarrow \tau_\theta \theta^i + (1 - \tau_\theta)\theta^i_-$
End while
```

.

and black squares represent the cumulative rewards after Pareto improvement. In this paper, Boltzmann fitting is used to fit the cumulative rewards convergence curves before and after Pareto improvement, in which the black curve represents the results after Pareto improvement and the blue curve represents the results before Pareto improvement. Because the Pareto improvement is not carried out before the experimental results converge, the red dots and black squares almost coincide at the beginning of training.

According to Fig. 3a, it is easy to see that the algorithm tends to converge after 700 episodes of training. According to the fitted convergence curve, the cumulative rewards of Pareto improved Q-learning is slightly higher than the original THP-Q. According to the position distribution of scatter points, although the maximum value of cumulative rewards is increased after Pareto improvement, its distribution becomes more dispersed. According to Fig. 3b, the AC based algorithm needs a longer exploration time, and the algorithm tends to

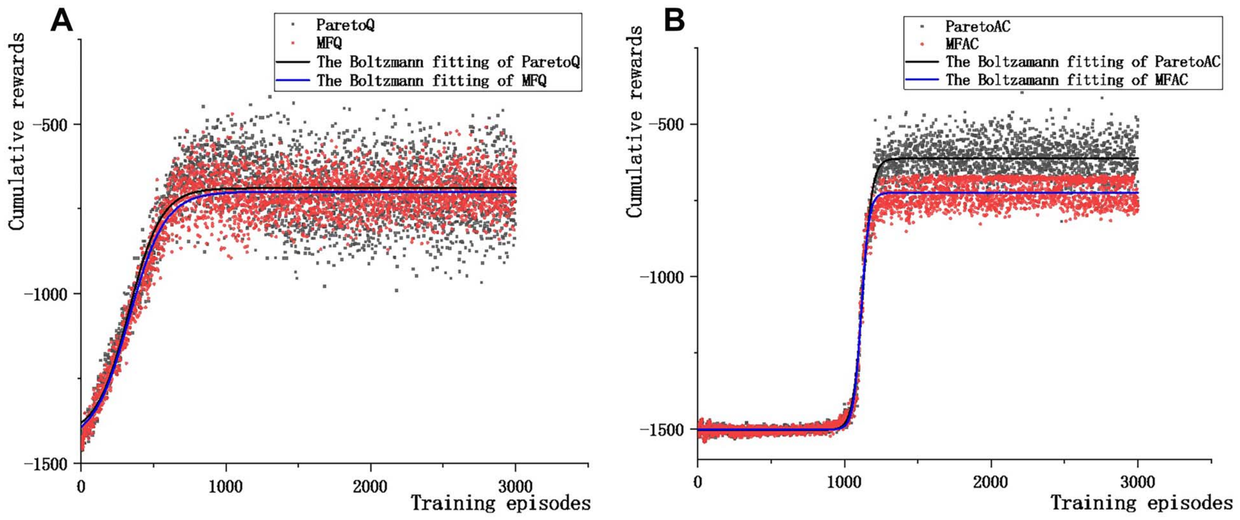

FIGURE 3. The cumulative rewards when map size is 40. (**a**) Value-based (**b**) Policy-based.

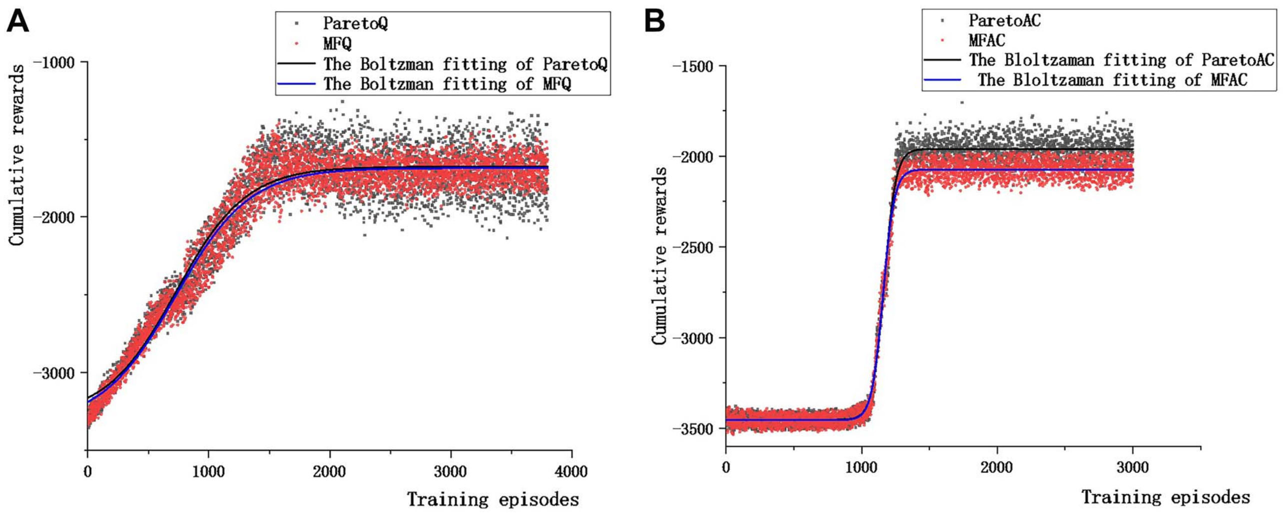

FIGURE 4. The cumulative rewards when map size is 60. (**a**) Value-based (**b**) Policy-based.

TABLE 2. The hyperparameter of Pareto-Q.

Algorithm	Pareto-Q
Learning rate	0.0001
Discount	0.95
Tau	0.005
Batch size	64
Update steps	5

TABLE 3. The hyperparameter of Pareto-AC.

Algorithm	Pareto-AC
Learning rate	0.0001
Discount	0.95
Batch size	64
Value_coef	0.1
Ent_coef	0.08

TABLE 4. Descriptive statistics of algorithms when map size is 40.

	Pareto-Q	THP-Q	Pareto-AC	THP-AC
Training episodes	3000	3000	3000	3000
Mean value	−770.18	−783.81	−946.48	−1013.55
Standard deviation	202.55	192.01	422.82	368.94
Training times	28 223.82	23 519.69	48 415.47	38 679.56

TABLE 5. Descriptive statistics of algorithms when map size is 60.

	Pareto-Q	THP-Q	Pareto-AC	THP-AC
Training episodes	3800	3800	3000	3000
Mean value	−1993.69	−2010.28	−2539.30	−2605.68
Standard deviation	496.21	489.30	706.90	653.92
Training times	48 973.34	42 385.64	73 454.32	63 434.79

converge after 1200 rounds. According to the fitted convergence curve, the cumulative rewards of Pareto improved algorithm is significantly improved, and the improvement effect based on AC framework is slightly higher than that based on Q value.

As is shown vividly in Fig. 4, when the number of agents increases and the map size enlarges, the results stay the same.

In order to further analyze the differences before and after Pareto improvement, we have made descriptive statistics on the experimental data, as shown in Tables 4 and 5.

Table 4 shows the differences in three dimensions of each algorithm before and after Pareto improvement when the training episodes are the same. After Pareto improvement, the mean value tends to decrease, but the standard deviation increases. The reason may be that Pareto improvement improves the payoff of at least one agent without harming the interests of other participants, so the cumulative rewards and standard deviation increases. However, the essence of Pareto improvement is to adjust the strategy that has reached equilibrium, which may lead to the instability of the algorithm in the training process and large standard deviation. Meanwhile, it is not difficult to see that Pareto improvement requires longer training time.

Table 5 shows the same results when the number of agents increases and the map size enlarges.

4.4.2. Testing results

The average win rate of Pareto improved algorithm against each baseline is as follows:

Table 6 shows the average win rate of Pareto improved algorithm against baselines. We can see that Pareto-AC has an advantage over all baselines. Pareto-Q only has an absolute advantage over traditional algorithms, but it still can achieve the similar effect as the latest algorithm. Broadly speaking, it can be seen that Pareto improves the win rate to a large extent. When the number of agents increases and the map size enlarges, the results stay the same as Table 7 shows.

Next, we will show the comparison between Pareto improved algorithm and baselines more intuitively in the form of graph.

According to Fig. 5a and b, it can be seen that the win rate of the improved Pareto-AC is much higher than that of the traditional AC algorithm and can also defeat the original THP-AC, but the robustness is relatively poor, which may be related to the increased variance after Pareto improvement.

According to Fig. 5c and d, the win rates of the improved Pareto-AC are higher than the latest MADRL algorithms MAPPO and MADDPG. It is quite clear that our algorithm perform better in the battle scenario of MAgent platform.

According to Fig. 6a and b, it can be seen that the win rate of the improved Pareto-Q is much higher than that of traditional IL algorithm, and the win rate is slightly higher than original THP-Q. But the advantage is not as obvious as AC improved methods. This can also be seen from the difference in the convergence curve fitted before and after Pareto improvement in cumulative rewards.

According to Fig. 6c and d, the win rate of the improved Pareto-Q is slightly lower than that of the latest MADRL algorithms MAPPO and MADDPG, but the gap is small. It can be seen that the effect of our proposed algorithm is equivalent to that of the latest methods, and Pareto improvement has achieved a certain effect in enhancing the win rate.

According to Fig. 7, it can be seen that compared with the value-based reinforcement learning framework, the Pareto improvement has a more significant effect under the AC framework, and the win rate of Pareto-AC in battle is much higher than that of Pareto-Q.

TABLE 6. The average win rate when map size is 40.

Win rate	IL	AC	THP-Q	THP-AC	MADDPG	MAPPO	Pareto-Q	Pareto-AC
Pareto-Q	5.58	_____	1.08	_______	0.86	0.89	_______	0.57
Pareto-AC	_____	16.86	______	1.97	1.25	1.08	1.74	_______

TABLE 7. The average win rate when map size is 60.

Win rate	IL	AC	THP-Q	THP-AC	MADDPG	MAPPO	Pareto-Q	Pareto-AC
Pareto-Q	5.44	_____	1.10	_______	0.77	0.92	_______	0.63
Pareto-AC	_____	15.58	______	1.89	1.34	0.99	1.59	_______

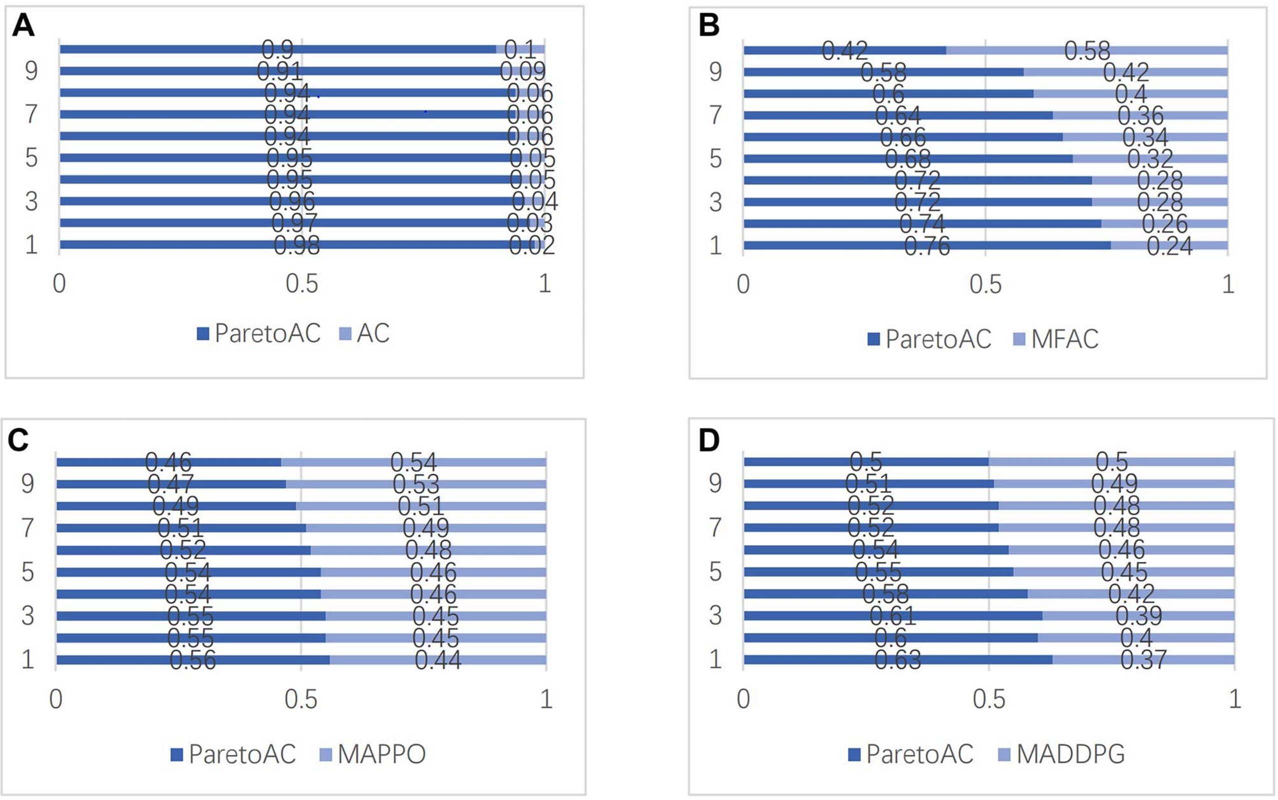

FIGURE 5. The win rate of Pareto-AC vs baselines when map size is 40. (**a**) Pareto-AC vs AC (**b**) Pareto-AC vs MFAC, (**c**) Pareto-AC vs MAPPO and (**d**) Pareto-AC vs MADDPG.

In the experiment, besides win rate, the difference between random sampling and stratified sampling is also compared. When the experience reply buffer is sampled randomly, the algorithm may not only have the problem of over fitting, but also cannot converge because the strategy is unstable. In stratified sampling, we can choose whether to carry out Pareto improvement in probability, which can not only alleviate the over fitting issue caused by Pareto improvement, but also avoid the problem of non-convergence caused by unstable strategy to a certain extent, making the result more stable.

5. CONCLUSION

In this paper, improvement of MADRL equilibrium solution based on Pareto optimization method is proposed, which refines the Nash equilibrium to a large extent. The stratified

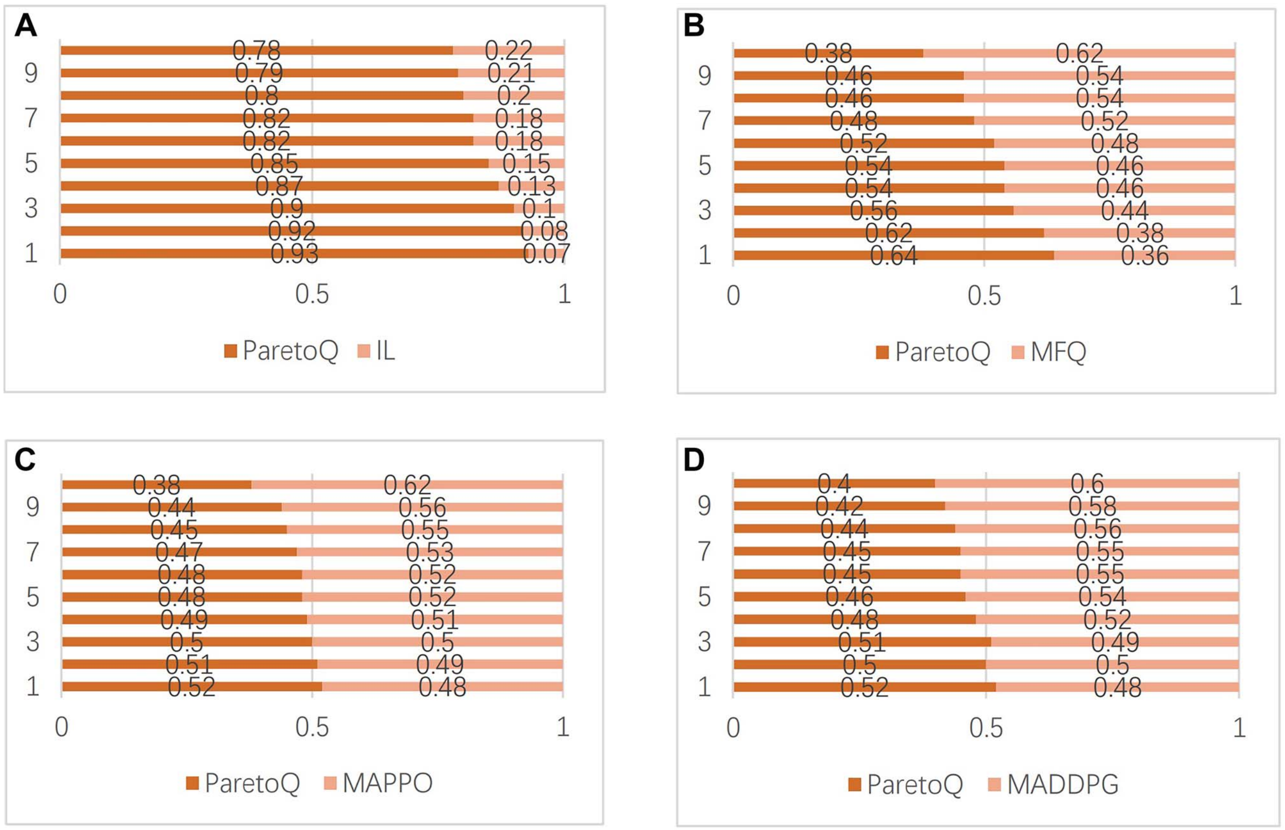

FIGURE 6. The win rate of Pareto-Q vs baselines when map size is 40. (**a**) Pareto-Q vs IL (**b**) Pareto-Q vs MFQ (**c**) Pareto-Q vs MAPPO and (**d**) Pareto-Q vs MADDPG.

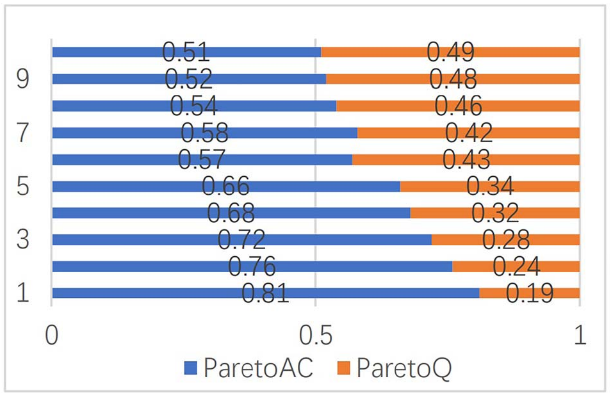

FIGURE 7. The win rate of Pareto-AC vs Pareto-Q when map size is 40.

sampling is used to prevent the non-convergence problem caused by over fitting and strategy instability after Pareto improvement. After training, a satisfactory multi-agent deep reinforcement learning algorithm model is obtained. The experimental results show that after Pareto improvement, the win rate is enhanced, especially the Pareto-AC algorithm improved on basis of AC framework, whose win rate in battle exceeds all baselines. Although the proposed methods have achieved outstanding results, this paper still has some shortcomings. Pareto improvement needs longer training time and leads greater variance, thus its training efficiency and robustness still have room to be improved. In the future, we will continue to focus on the refinement of Nash equilibrium and its combination with multi-agent reinforcement learning.

DATA AVAILABILITY

All data are incorporated into the article and its online supplementary material.

ACKNOWLEDGMENTS

National Natural Science Foundation of China (61806221).

REFERENCES

[1] Littman, M.L. (1994) *Markov Games as a Framework for Multi Agent Reinforcement Learning*. Morgan Kauffman Publishers, Inc.

[2] Hu J. (1998) Multiagent reinforcement learning: theoretical framework and an algorithm, *The 15th International Conference on Machine Learning*. Madison, Wisconsin, USA, July 24–27, 1998, 242–250.

[3] Fink, A.M. (1964) Equilibrium in a stochastic N-person game. *Journalof Science of the Hiroshima University*, Series AI (mathematics), vol. 28, no. 1, pp. 89–93, 1964.

[4] Heinrich J, Lanctot M, Silver D. (2015) Fictitious Self-Play in Extensive-Form Games *The 32nd International Conference on Machine Learning*, Lille, France, 805–813.

[5] Berger, U. (2005) Brown's original fictitious play. *Game Theory Inf., 2005*, 135, 572–578.

[6] Yang Y, Rui L, Li M, *et al.* (2018) Mean Field Multi Agent Reinforcement Learning *The 35th International Conference on Machine Learning*. 2018.

[7] Blume, L.E. (1993) The statistical mechanics of strategic interaction. *Games Econ. Action*, 5, 387–424.

[8] Stanley, E.H. (1979) Phase transitions and critical phenomena. *Am. J. Phys. 1979*, 40, 927.

[9] Zheng L, Yang J, Cai H, *et al.* (2018) MAgent: A Many-Agent Reinforcement Learning Platform for Artificial Collective Intelligence. *The 32nd AAAI Conference on Artificial Intelligence*, New Orleans, Louisiana, USA, February 2–7, 2018, 8222–8223.

[10] Hu, J. and Wellman, M.P. (2003) Nash Q-learning for general-sum stochastic games. *J. Mach. Learn. Res.*, 4, 1039–1069.

[11] Littman M L. (2001) Friend-or-Foe Q-Learning in General-Sum Games. *The 18th International Conference on Machine Learning*, Williams College, Williamstown, MA, USA, 322–328.

[12] Plappert M, Houthooft R, Dhariwal P, *et al.* (2018) Parameter Space Noise for Exploration. *The 6th International Conference on Learning Representations*, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings.

[13] Fortunato M, Azar M G, Piot B, *et al.* (2018) Noisy Networks for Exploration. *The 6th International Conference on Learning Representations*, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings.

[14] J Choi, Guo Y, Moczulski M, *et al.* (2019) Contingency-Aware Exploration in Reinforcement Learning. *The 7th International Conference on Learning Representations*. New Orleans, LA, USA, May 6–9, 2019.

[15] Ostrovski, Georg, *et al.* (2017) Count-Based Exploration with Neural Density Models. *The 34th International Conference on Machine Learning*, Sydney, NSW, Australia, 2721–2730.

[16] Ngatchou P, Zarei A, El-Sharkawi A. (2005) Pareto Multi Objective Optimization, Intelligent Systems Application to Power Systems. *Proceedings of the 13th International Conference on. IEEE*, pp. 84–91, doi: 10.1109/ISAP.2005.1599245.

[17] Lin X, Chen H, Pei C, *et al.* (2019) A Pareto-Efficient Algorithm for Multiple Objective Optimization in E-commerce Recommendation. *Proceedings of the 13th ACM Conference on Recommender Systems, RecSys 2019*, Copenhagen, Denmark, September 16–20, 2019. ACM 2019, ISBN 978-1-4503-6243-6.

[18] Rasmusen, E. (2001) *Games and Information: An Introduction to Game Theory*. Blackwell, 2001.

[19] Cohen, J.E. (1998) Cooperation and self-interest: Pareto-inefficiency of Nash equilibria in finite random games. *Proceedings of the National Academy of Sciences of the United States of America*, 1998, 95, 9724–31. doi: 10.1073/pnas.95.17.9724.

[20] Hoefer, M. and Skopalik, A. (2013) On the complexity of Pareto-optimal Nash and strong equilibria. *Theory Comput. Syst., 2013*.

[21] Volodymyr, M., Koray, K., David, S. *et al.* (2015) Human-level control through deep reinforcement learning. *Nature*, 518, 529–533.

[22] Bayiz, Y.E. (2014) *Multi Agent Actor-Critic Reinforcement Learning for Cooperative Tasks*. http://resolver.tudelft.nl/uuid:033c4238-62e8-4ef7-b43d-b43f3b97856f.

[23] Lowe R, Wu Y, Tamar A, *et al.* (2017) Multi agent Actor-Critic for Mixed Cooperative-Competitive Environments. *The 31st Conference on Neural Information Processing Systems*, Long Beach, CA, USA, 6379–6390.

[24] Yu, C., Velu, A., Vinitsky, E. *et al.* (2021) The surprising effectiveness of MAPPO in cooperative. *Multi Agent Games*. CoRR abs/2103.01955.