A Comprehensive Analysis of Game theory on Multi-Agent Reinforcement

Wentao Fan

School of Software engineering, Taiyuan University of Technology, Taiyuan, China

fanwentao7595@link.tyut.edu.cn

Abstract. In recent years, reinforcement learning has gradually emerged as a popular research field in artificial intelligence. However, for multi-agent systems, due to their complex environment and an abundance of agents, the learning target of maximizing the anticipated value of cumulative rewards for specific agents frequently fails to converge. Introducing game theory into reinforcement learning can effectively address the interactions among intelligent agents, provide a rationale for convergence points corresponding to strategies. To address the issue of non-existence of optimal solutions for certain tasks in multi-agent settings, this paper takes a game-theoretic perspective and summarizes classical reinforcement learning algorithms developed in recent years. The basic theory of multi-agent reinforcement learning, essential game theory, categorization of reinforcement learning using multiple agents game strategies, primary worries are addressed in the study. It also analyzes the challenges that game-theoretic multi-agent reinforcement learning algorithms may encounter in the future, along with the relevant optimization directions.

Keywords: Multi-Agent, game theory, reinforcement, Nash Equilibrium.

1. Introduction

Multi-Agent Reinforcement Learning (MARL) is an area that has received a lot of interest. in recent years, an interest that stems from its potential applications in areas such as robotics, autonomous vehicles, and multi-intelligence systems. MARL involves multiple intelligences interacting and interacting with their environment to learn optimal strategies through a trial-and-error process. Game theory is a fundamental framework used in MARL that provides the mathematical basis to model and analyze strategic interactions between intelligences [1].

Game theory provides a powerful set of tools for studying the dynamics and outcomes of interactions among rational decision makers. It provides a framework for analyzing strategic decisions by considering the actions and strategies of multiple intelligences and their effects on each other's outcomes. By applying game theory to the field of reinforcement learning, researchers can better understand the complex dynamics that arise when multiple intelligences interact and learn simultaneously.

The basic theory of multi-agent reinforcement learning, essential game theory, categorization of reinforcement learning using multiple agents game strategies, primary worries, and potential future developments are addressed in the study. In Section 2, the fundamental principles of reinforcement learning and games are clarified briefly; Section 3 classifies and introduces the algorithms related to multi-intelligence games and typical game training algorithms; Section 4 analyzes the key problems and challenges of game training, and finally the prospects for the development of multi-intelligence reinforcement learning game training.

2. Basic Game theory

2.1. Reinforcement Learning Basic Theory

This paragraph provides an explanation of the basics of reinforcement learning algorithms and game theory, as well as an introduction to the development of game theory

2.1.1 Markov Decision

The MDP stems from of a five-tuple, which generally includes S, A, P, R, γ, etc. S illustrates the intelligence state space, A denotes the intelligence action space, and P is the intelligence state transfer function, which gets characterized as

$$P: S \times A \times S \rightarrow [0,1] \tag{1}$$

P represents the likelihood distribution of state transfer to the next state's' ∈ S for an agent body in states ∈ S using a given action a ∈ A. The instantaneous payoff function R of the intelligence is given by:

$$R: S \times A \times S \rightarrow R \tag{2}$$

R denotes the immediate payoff obtained by the state transfer to the next state's' when the intelligence adopts the action in the case of states. Summing all the immediate returns, the total gain of the intelligence Rt is obtained:

$$R_t = \sum_{t'=t}^{T} \gamma_{t-t} r_i \tag{3}$$

where γ ∈ [0, 1] is the discount rate used to balance the impact of the instantaneous and long-term returns of the intelligences on the total returns 2. In the Markov decision process, when an intelligence moves from one state to another, it only needs to consider the current state and behavior, not the prior actions taken and the environment in which it is located.

2.1.2 Reinforcement learning

Reinforcement learning is the way in which an intelligence learns by interfacing with its environment in order to obtain the maximum benefit. By taking optimal actions and interacting with the environment, the intelligence is rewarded and learns the best strategies to guide problem solving (Fig.1). In layman's terms, reinforcement learning is the process by which an intelligence adapts to its environment through trial and error, driven by the need to maximize the reward. The interaction process between an agent body and its environment can be identified by three elements: state s, action a, and reward r. The agent body interacts with the environment subject to the initial state s. Perform action a and participate in the environment to obtain reward r and move to the next state s

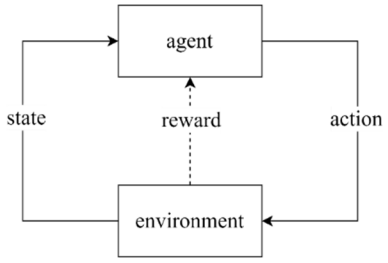

Fig. 1. Reinforcement learning basic process

The learning outcome of an intelligence is to acquire a strategyπ that is appropriate for the environment, π is the probability that the intelligence may choose a certain behavior, denoted as

$$\pi: S \times A \rightarrow [0,1] \tag{4}$$

Strategies can be divided into deterministic and stochastic strategies 3. Deterministic strategies are those in which an intelligence chooses a certain action when it encounters the same state at different moments or episodes. The stochastic strategy is a probability distribution, i.e., the probability of selecting a certain action for the input of a state and the output of an intelligence. Odds of picking a specific mode of action is the output of a state input to the intelligence. The stochastic strategy can be showcased as follows

$$\pi(a|s) = p(a_t = a|s_t = s) \tag{5}$$

As the intelligence interacts with its environment, it continuously optimizes its currently used strategies to make them better and better, a process known as strategy updating. Strategy updating is performed iteratively in reinforcement learning, so that the intelligence can obtain an optimal strategy. In order to determine the superiority or inferiority of an intelligence's strategy at a certain state s, Define the state value function $V_\pi(s)$:

$$V_\pi(s) = E[G_t|s_t = s, \pi] \tag{6}$$

Similarly, in order to determine the superiority of the execution of action a by an intelligence at a state s, specify the state action value function: $Q_\pi(s,a)$:

$$Q_\pi(s,a) = E[G_t|s_t = s, a_t = a, \pi] \tag{7}$$

where G_t is the total gain R, or cumulative return, obtained by the intelligence from the current state to the conclusion of the engagement procedure. In the reinforcement learning procedure, the intelligence evaluates the value function to determine the strengths and weaknesses of its own strategy and to improve it.

2.1.3 Multi-agent Reinforcement Learning

Real-world scenarios often have multiple agents interacting with each other, and multi-agent systems are a way to model real-world multi-agent interaction scenarios 4. Multi-agent reinforcement learning is an artificial intelligence method that uses reinforcement learning methods to train multi-agent systems, following a stochastic game process 6.

In multi agent reinforcement learning, action space of agent A (i =1 , 2, …, n) Interact to form a joint action space A:

$$A = A_1 \times A_2 \times \cdots \times A_n \tag{8}$$

Algorithms for multi-agent reinforcement learning can be divided into fully collaborative, fully competitive, and mixed approaches. [5]. By employing multi-agent reinforcement learning, complex tasks can be accomplished, and algorithm efficiency can be improved

2.2. Game Theory

Game theory is the formal subject of conflict or cooperation models between rational entities with interests. The formal formulation of game theory is generally composed of the elements of Player, Strategy, Payoff, and Rationality [8-9].

There are various classifications of game theory (Fig.2). According to whether the players make decisions simultaneously in the game process, they can be classified into standardization games (static games) and extended games (dynamic games); according to whether the players know the information of the game process, they can be categorized as perfect information games and imperfect information games, etc. For a general standardization game, the following formal definition is made:

$$G =< N, \{a_i\}_{i=1}^n, \{u_i\}_{i=1}^n > \tag{9}$$

where N is an array of gamers.; n denotes the number of players participating in the game; ai denotes the player's gathering of approaches; ui denotes the ith player's payout mechanism. Define a ∗ = (a1∗ , a2∗ , ..., a) to be a collection of game outcomes consisting of strategies, and ai to be the set of strategies other than player i, i.e., the opponent strategies of player *i*:

$$a_{-i} = (a_1, \cdots, a_{i-1}, a_{i+1}, \cdots, a_n) \tag{10}$$

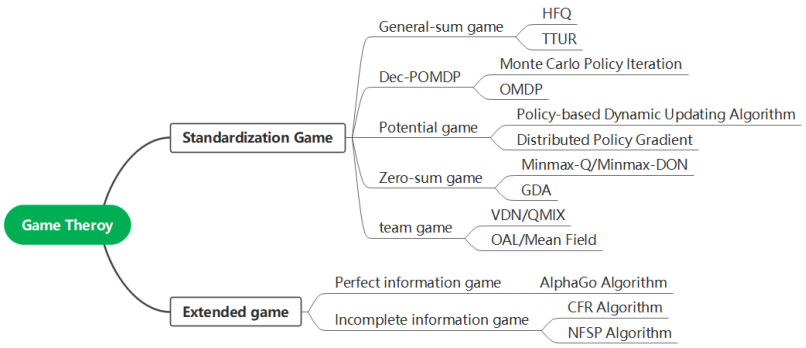

Fig. 2. The basic classification of game theory algorithms

2.2.1Nash Equilibrium

In game theory, a Nash equilibrium is an assortment of strategies:$(\sigma_*^1, \sigma_*^2, \cdots, \sigma_*^n)$.This strategy allows each player to have no reduction in that player's award while other players' strategies remain the same, i.e.$\forall s \in S, i = 1,2, \cdots, n$, both have the following inequality 10:

$$R^i(s, \sigma_n^1, \sigma_n^2, \cdots, \sigma_n^n) \geq R^i\left(s, \sigma_n^1, \sigma_n^2, \cdots, \sigma_n^{i-1}, \sigma^i, \sigma_n^{i+1}, \cdots, \sigma_n^n\right) \quad (11)$$

Among them,σ^i is the set of all possible strategies for player *i.* Nash equilibrium is an important theoretical foundation of game theory. Nash proved the fact of equilibrium points in standard games and created the theoretical basis for the application of game theory.

3. Standardization Game

In standard games, the game players are divided into games of common interest and games of different interests by whether their objectives are the same.

3.1. Common Interest Game

3.1.1Team Game

Team games have been extensively explored in part because they are crucial for developing distributed AI. In a team game, each intelligence only needs to maintain its unique the function of value, and the function of value depends only on the current state and action, thus avoiding the problems of environmental non-smoothness and dimensional explosion when considering joint actions. Therefore, many single-intelligence algorithms can be applied to this problem. If there are multiple Nash equilibria in the game, even if the learning goals do not conflict among each intelligence, this can lead to the problem that the intelligence will not eventually learn the optimal strategy. Sandholm et al. propose an optimal adaptive learning algorithm using preferred action selection and incomplete history sampling, and show that the method converges to the optimal Nash equilibrium for team games with multiple Nash equilibria, but the convergence of the method is not guaranteed in general stochastic games 11. Therefore, based on the OAL algorithm, Arslanet al. propose an algorithm for decentralized Q-learning for stochastic games and verify the rate at which the algorithm converges to the ideal Nash equilibrium 12. Table 1 is the Advantages and disadvantages of main the algorithms under team game.

Table 1. Advantages and disadvantages of main algorithms under team game

Algorithm	Advantage	Disadvantage
ATT-MADDPG (The Packet Routing Environment)	Embedding an attention mechanism within a centralized critic allows for an adaptive way to model the dynamic joint strategies of teammates.	The increasing number of agents is difficult to model and converge
G2ANet (Predator-Prey)	Modeling the relationships between agents using a complete graph simplifies the learning process	Due to the edge weights being either 0 or 1, useful edges are removed, and it is not possible to converge to the global optimum.
OAL (The Virtual Game)	The proposal of preference-based action and incomplete history sampling can lead to convergence towards the optimal policy.	In general-sum stochastic games, it is not possible to converge to the optimal strategy.
Partaker-Sharer (Predator-Prey)	It can address the issue with cooperative games failing to converge to the Nash equilibrium under the teacher-student framework.	Adjusting the recommendation based on confidence level leads to slow convergence speed.
VDN (2D Gridworld)	The joint value function is the linear sum of the value functions of each agent.	Efficiency is low, and there are few games that meet the prerequisite conditions.
QMIX (Two-Step Game)	Using neural networks to roughly represent the joint value function, which improves the efficiency	It is required that the joint value function is strictly monotonic with respect to individual value functions.
QTRAN (Multi-domain Gau ssian Squeeze and Modified Predator Prey)	By using the VDN (Value Decomposition Networks) approach to obtain the sum of joint value functions, and then employing neural networks to approximate the deviation of the joint value function from the total of joint value functions, it blends the positive aspects of the VDN and QMIX algorithms.	It failed to overcome the drawbacks of both VDN and QMIX algorithms, and the convergence conditions are too strict.

3.1.2 Stochastic Game

A game is recognized as a potential game if each player's change of goal or strategy choice can be mapped to some global function, which is called a potential function. In general, a potential game can be considered as an " individual agent component " of a multi-agent game, because the interests of all the intelligences in the SPG are clarified as a single potential function. The complexity of the problem increases when the potential game is extended to a stochastic potential game. Macuaet al studied the general form of potential games where the strategy is linked to the state 13. The moment of the action and proved the existence of a Nash equilibrium under this premise, while theoretically proving that one may determine the Nash equilibrium in a purely strategic potential game by addressing the MDP [14]. Mazumdar et al. proposed a strategy-based dynamic update algorithm for potential games and applied it to the Morse-Smale game, proving that the algorithm can converge to a local Nash equilibrium 15. In complex multi-agent systems, effective learning requires a high degree of coordination among all participating intelligences, but coordination and communication among intelligences is often inefficient. Chen proposed decentralized implementation and centralized

training and exploration by policy distillation to facilitate coordination and effective learning among intelligences 16.

3.2. Diverse interests at play

Different-interest games refer to situations where players have different learning objectives. Based on whether the sum of payoffs for all players is zero, these games can be categorized as finite zero-sum games and general-sum games [17-23].

3.2.1 finite zero-sum game.

Finite zero-sum games refer to games in which the number of players is limited, and there exists a strictly competitive relationship. In these games, the total payouts to all participants and expenditures are zero. In two-player games of zero-sum, due to the completely opposing learning objectives of the two players, the Nash equilibrium solution is essentially a saddle point. Adler proved the equivalence between zero-sum games for two participants and optimization by linearity [2121. This means that it is possible to specify a two-player zero-sum game as a linear optimization problem, and any linear optimization issue can be simplified as a zero-sum game problem.

However, this theorem does not apply to games with non-concave and non-convex payoff functions. For zero-sum games with discrete action spaces, Shapley proposed the first value iteration method and proved that a contraction mapping called HShapley is used in two-player zero-sum games., and it can converge to a Nash equilibrium. In contrast to the Shapley's model-based value iteration approach, for two-player zero-sum games, Littman devised the model-free Minimax-Q method., where "Minimax" refers to using the minimax theorem to construct linear programming for finding the Nash equilibrium strategy for each specific state, and "Q" refers to utilizing the TD method from Q-learning to iterate the state value function or action-state value function. Subsequently, Littman proved that the Minimax-Q algorithm's Bellman operator is a contraction mapping and converges to a single fixed point under identical circumstances as Q-learning. However, the aforementioned algorithm is still based on a Q-table and cannot be applied to scenarios with large action and state spaces.

Fan et al. combined DQN with the minimax theorem and proposed the Minimax-DQN algorithm to solve two-player zero-sum games, quantifying the difference between the policies obtained by this method and the Nash equilibrium. In 2014, Goodfellow et al. introduced Generative Adversarial Networks (GANs), which use neural networks to make solving such problems possible 22. In GANs, two parameterized neural network models (a generator G and a discriminator D) engage in a zero-sum game. The discriminator is used to assess if the input samples are real or fake, while the generator creates "fake" samples using random variables. Through competitive training with one other, they enhance each other's performance.

The above methods still cannot address issues such as the instability of training in multi-agent algorithms, sensitivity of agents to environmental changes, and susceptibility to local optima. Li et al. proposed the M3DDPG algorithm, with the core objective of training each agent to perform well, even when the opponent reacts in the worst possible manner 23.

In the search for Nash equilibrium, using the same step size to implement Gradient-Descent-Ascent (GDA) is equivalent to applying policy gradient methods to both agents in one algorithm. By fine-tuning the step size, the GDA method can effectively address certain two-player zero-sum games.

3.2.2 finite general sum game

In contrast to zero-sum games, which have a computational complexity of P-complete, general-sum stochastic games contain a computational complexity of PPAD, so the intricacy of solving general-sum stochastic games is much more difficult than zero-sum stochastic games. Herings et al. invoke the idea of homomorphisms in topology, and propose an algorithm that can find homomorphic paths between equilibrium solutions for solving multiple independent MDPs and multi-player stochastic games, thus transforming the problem of solving general-sum stochastic games into a problem of solving multiple independent MDPs. path algorithm, which transforms the multi-player

stochastic game into a problem of solving multiple independent MDPs, but this method can only be applied to the two-player game with a small state set.

Subsequently, an assortment of algorithms with values were proposed to solve general and stochastic games. Most of these methods use classical Q-learning as the central controller, with the difference being the equilibrium applied by the central controller to guide the intelligences to converge gradually over the iterations. For example, the Nash-Q algorithm employs a Nash equilibrium, while the correlated-Q algorithm employs a correlated equilibrium (Nash).

4. Extended Gaming

If the utility function is shared knowledge, such a game is said to be a complete information game, such as Go and Xiangqi, otherwise it is an incomplete information game, such as Texas Hold'em. Poker.

4.1. Extended games with complete information

Nash's main contribution in game theory is to prove that there must exist a Nash equilibrium with combined approaches under a finite-player finite-substandard type game. However, this Nash equilibrium assumes that the strategies of the other players do not change when the players are making decisions, but in the extended game the first-decision player cannot know the strategies of the later-decision player, so it will lead to the existence of an unbelievable Nash equilibrium.

4.2. Extended games with incomplete information

The truth is that games frequently take the shape of lengthy simulations containing faulty information, such as real-time strategy games and military confrontations, and extended games with incomplete information are considered to be one of the major difficulties in the next generation of artificial intelligence. There are three main difficulties in solving extended games with incomplete information, one is that the subgames are interrelated, the second is that there are state-indivisible information sets, which makes the state-based value estimation method in reinforcement learning no longer applicable, and the third is that the solution scale of the game is relatively large, for example, the number of information sets in bridge and Texas hold'em are 1067 and 10162, respectively. In recent years, for example, in the case of Texas hold'em, extended games with incomplete information have made good progress, and the third is that the solution scale of the game is relatively large. In recent years, for example, Texas Hold'em, the extended game with incomplete information has made good progress.

In the incomplete information extended game, it is difficult to directly model the joint strategies due to the interactions between the strategies. Tian et al. proposed the joint strategy search (JPS) method based on the density of strategy variations, which can iteratively improve the joint strategies of the collaborating intelligences in the incomplete information game without reevaluating the whole game, and the algorithmic performance is verified in the grid world. Validation. The main algorithms for solving incomplete information games are the CFR series and the NFSP series of algorithms under the self-game, and the validation environments for the two series of algorithms are mainly Texas Hold'em.

4.2.1 Counterfactual regret minimization algorithm（CFR）

The CFR algorithm combines the regret-value minimization algorithm and the average strategy to achieve the goal of minimizing the global regret value by minimizing the regret value on a single set of information, which ultimately makes the average strategy in the game process converge to the Nash equilibrium (Fig.3) 19. Thus, the need to traverse the entire game tree, time complexity and slow convergence are the principal flaws in the algorithm. In order to reduce the time complexity of the CFR algorithm, Lazy-CFR proposed by Zhou et al. adopts the inert correction strategy for the disadvantage of the original CFR that the whole game tree must be traversed in each round, and

achieves the same efficiency as CFR under the condition that only some of the nodes of the game need to be accessed. Other improvements to the CFR algorithm include Best-Response Pruning, Brown et al. demonstrated that adding BRP to the CFR algorithm reduces the risk of convergence to Nash. Brown et al. showed that adding BRP to the CFR algorithm reduces the number of moves that do not contribute to convergence to a Nash equilibrium, thereby accelerating convergence and saves space. Table 2 shows the main advantages and disadvantages of the CFR.

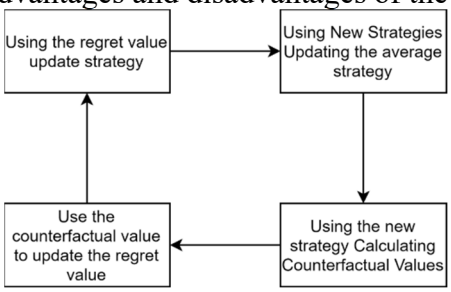

Fig. 3. Iteration steps of CFR

Table 2. Analysis of the main advantages and disadvantages of the CFR

Algorithm	Advantages	Disadvantage
CFR	The algorithm combines the regret-value minimization algorithm and averaging strategy	Requires a priori knowledge, agent with perfect recall; to traverse entire game tree
Lazy-CFR	An inert update strategy is used, which does not need to traverse the entire game tree	A priori knowledge is still required; agent need to have perfect recall
MCCFR (Abstract poker games)	To make the R approach less time-consuming, the algorithm employs Monte Carlo sampling.	Require a priori knowledge and perfect recall; large variance
VR-MCCFR (Leduc hold'em)	Take the average utility value of the nodes without visits as the baseline mean utility value without visiting nodes as baseline mitigates the problem of high variance	Require a priori knowledge and perfect recall
Regression CFR (Leduc hold'em)	Using regression trees as function approximator	need travelling through the whole game tree and a great deal of prior knowledge
DNCRM (One-Card-Poker)	Based on dual neural networks, there is no need large amount of prior knowledge and accelerates convergence	Agent is required a perfect recall
Deep-CFR (heads-upflop hold'empoker)	Utilizing the fitting ability of the neural network power to act as a function approximator	Agent is required a perfect recall
SD-CFR (Leduc Poker)	Extracting the average policy from the iterative Q-value network buffer has a lower error and improves the convergence speed of Deep- CFR	Inability to Solve Incomplete Information game problem
DREAM (Leduc Poker)	Keeping the variance low Converging in incomplete information to a Nash equilibrium	Need travelling through the whole game tree, time complexity is high
LONR (NoSDE Markov Game)	Without the need for perfect recall Conditions can still converge	Updated rules similar to Q-learning Convergence time to be improved

4.2.2 Self-playing algorithm.

Self-gaming is a training algorithm in the context of multi-intelligence, the essence of self-gaming is to generate data through simulation of one's own behavior, and to learn and improve based on that

data, the implementation of self-gaming has successfully challenged the classical view that an expert adversary is necessary to achieve good performance 20.

Samuel designed self-gaming in a checkers program and found that this model worked particularly well early on. Epstein notes that programs trained in self-gaming methodically identify and explore their own paths through the search space Path. Tesauro et al. designed TD-Gammon to reach the expert level in its ability to play backgammon.

AlphaGo Zero The training data for AlphaGo Zero comes entirely from self-game training. Kaplanis et al. show that the experience distribution of each intelligence is affected by the changing strategies of the opponents. Balduzzi et al. described the self-game training algorithm, noting that the self-game is suitable for games modeled by transfer games. Hernandez et al. define a general framework for SP using formal notation, under which the definition of popular SP algorithms is harmonized, and further show that self-games exhibit periodic strategy evolution. Table 3. is the typical self-play training methods.

Table 3. Typical self-play training methods

Name	Traits	Application
Classic Self-Play Method	"Engage in a battle with the latest version of myself."	Game of heckers
Uniform Historical Training Method	"Uniformly random sampling opponents across all historical versions."	AlphoGO
Interval Uniform Training Method	"Uniformly random sampling opponents from the recent (δM, M) interval."	Humanoid
Interval Constraint Training Method	"Avoiding unequal sampling opportunities between old and new policies."	RirRPS
Backtracking Proportional Training Method	"Play against the latest version of myself with a certain ratio, and the rest against past versions."	OpenAI Five

5. Optimization of game theory reinforcement learning algorithms.

5.1. Convergence

Most of the current game reinforcement learning algorithms are mainly based on experimental based, due to the powerful fitting properties of deep neural networks, which make many algorithms in converge well in most cases, but there is no theoretically given but there is no theoretical proof of convergence. Meanwhile, for algorithms with convergence proofs, the convergence conditions are often too harsh. Meanwhile, for algorithms with proof of convergence, the convergence conditions are often too strict, such as the Nash-Q Learning algorithm, Mean Field algorithm requires the existence of saddle points or global optimal points at each stage of the game, or even does not require the existence of a global optimal point. The global optimal point is not even allowed to appear alternately, which makes it difficult to satisfy the above convergence conditions.

This results in few games satisfying the above convergence conditions, so the algorithms are limited in their ability to solve the problem. Therefore, the algorithm is limited in the number of problems it can solve.

5.2. Solution rule

Roughgarden, a leading scholar in the study of algorithmic game theory, proved that the Nash equilibrium is an NP-hard problem. Shish equilibrium solution is an NP-hard problem. Some traditional mathematical analysis algorithms, such as Lemke-Howso algorithm, global Newton algorithm, distributed primal-dual algorithms, projected gradient algorithms, etc. are used to resolve the game's Nash equilibrium. The basic idea is to convert the Nash equilibrium computational problem into a certain type of optimization problem under certain assumptions. The basic idea is to

convert the computational problem of Nash equilibrium into a certain type of optimization problem, variational inequality problem or mutual complement problem under certain assumptions.

However, most of these methods require However, most of these methods require a high degree of smoothness in the payoff functions of the participants, and these algorithms often rely on the selection of initial points. However, most of these methods require a high degree of smoothness in the payment function of the game participants, and these algorithms often rely on the selection of the initial point, which makes the solution complex and sometimes, the solution time is too long, resulting in failure to find the correct Nash equilibrium value. These algorithms limit the scope of application of these algorithms to a certain extent.

6. Game Theory Reinforcement Learning Algorithmic Heavyweights

6.1. Dimensional explosion

Dimensional explosion refers to the significant increase in the action space, state space, and parameter count in multi-agent systems, leading to exponentially growing computational complexity. There are numerous key strategies for solving this issue:

Hybrid training mechanisms: One approach is to use a hybrid training mechanism, specifically centralized training with decentralized execution (CTDE). In CTDE, the training process is centralized, allowing for efficient coordination and learning across agents, while the execution phase is distributed, enabling decentralized decision-making.

Exploration algorithms based on pseudo-counts: These algorithms utilize a density model designed to evaluate frequencies satisfying certain properties, known as pseudo-counts. By calculating pseudo-counts that generalize well in continuous spaces, exploration efficiency is improved, thereby mitigating the issue of dimensional explosion.

6.2. Unsteady environment

With regarding multi-agent reinforcement learning, the environment's state transition function depends on the joint actions of the agents, resulting in non-stationarity of the environment. There are several main approaches to address this problem:

Using the Actor-Critic (AC) framework: By obtaining information and actions from other agents during the training process, agents can anticipate and adapt to environmental dynamics, minimizing the effects of unexpected occurrences.

Opponent modeling: By simulating the strategies of other agents, opponent modeling can stabilize the training process of agents and provide a more predictable environment for learning.

Utilizing meta-learning: Meta-learning techniques can enable agents to quickly adapt to non-stationary environments by learning to learn. These methods allow agents to acquire knowledge and strategies that facilitate faster adaptation and decision-making in dynamic environments.

6.3. Credit assignment

Credit assignment is an inevitable issue when multi-agent systems contain more agents than before. There are several main methods to address this problem:

Average allocation: This method distributes credit equally among all participating agents, regardless of their individual contributions or performance.

Differential reward assignment: Differential reward assignment allocates credit based on the difference between an agent's actual performance and the average performance of all agents. Agents with above-average performance receive positive credit, while those with below-average performance receive negative credit.

Advantage function allocation: Advantage function allocation assigns credit based on the advantage function, which compares an agent's performance to that of the other agents in terms of relative performance. Agents with a higher advantage receive more credit. TVT algorithm based on episodic memory retrieval by DeepMind: DeepMind proposed the TVT (Temporal Value Transport)

algorithm, which utilizes episodic memory retrieval to assign credit. This algorithm retrieves relevant memories from the past to estimate the impact of an agent's actions on future rewards, allowing for more accurate credit assignment.

7. Conclusion

This paper takes a game-theoretic perspective to review multi-agent reinforcement learning algorithms. It begins with a brief introduction to the relevant background of multi-agent reinforcement learning, aimed at addressing the complexities of entity relationships and non-optimal solutions in multi-agent systems. Subsequently, game theory concepts are introduced, and the algorithms are categorized based on standard-form and extensive-form games, including common-interest games, different-interest games, perfect-information games, and imperfect-information games. Through a horizontal comparison, the strengths and weaknesses of each algorithm are highlighted. Furthermore, the paper summarizes the current optimization directions of game-based reinforcement learning algorithms, focusing on their convergence and equilibrium solution techniques. Finally, the paper analyzes the major challenges in multi-agent reinforcement learning, such as the dimensionality curse, environment instability, and credit assignment. Game-theoretic reinforcement learning algorithm for multiple agents demonstrate excellent performance in complex scenarios, particularly in areas such as vehicle autonomy, robot control, and financial risk management. showing promising prospects for development and practical applications. With further research, it is believed that game training will continuously make progress in interpretability, strategy iteration, exploration, and breakthroughs.

Reference

[1] Sun Yu, CAO Lei, Chen Xiliang, et al. Review of multi-agent deep reinforcement learning [J]. Computer Engineering and Applications, 2020, 56(5):12.

[2] Wang Jun, CAO Lei, Chen Xi liang. Review on reinforcement learning in multi-agent game [J]. Computer Engineering and Applications, 2021, 57(21):13.

[3] Liu Hao, Research on reinforcement Learning Algorithm and its Equilibrium in Multi-Agent Game [D]. Xi 'an University of Science and Technology

[4] Dorri A, Kanhere S S, Jurdak R. Multi-agent systems: A survey [J]. Ieee Access, 2018, 6: 28573-28593.

[5] Buşoniu L, Babuška R, De Schutter B. Multi-agent reinforcement learning: An overview[J]. Innovations in multi-agent systems and applications-1, 2010: 183-221.

[6] Solan E, Vieille N. Stochastic games [J]. Proceedings of the National Academy of Sciences, 2015, 112(45): 13743-13746.

[7] Gronauer S, Diepold K. Multi-agent deep reinforcement learning: a survey[J]. Artificial Intelligence Review, 2022: 1-49.

[8] Morgenstern O, Von Neumann J, Kuhn H W, et al. Theory of games and economic behavior [M]. J. Wiley and Sons, 1964.

[9] Hu Yujing. Game, Equilibrium and Knowledge Transfer in Multi-agent reinforcement Learning [D]. Nanjing University.

[10] Nash Jr J F. Equilibrium points in n-person games [J]. Proceedings of the national academy of sciences, 1950, 36(1): 48-49.

[11] Wang X, Sandholm T. Reinforcement learning to play an optimal Nash equilibrium in team Markov games [J]. Advances in neural information processing systems, 2002, 15.

[12] Arslan G, Yüksel S. Decentralized Q-learning for stochastic teams and games [J]. IEEE Transactions on Automatic Control, 2016, 62(4): 1545-1558.

[13] Macua S V, Zazo J, Zazo S. Learning parametric closed-loop policies for markov potential games [J]. arXiv preprint arXiv:1802.00899, 2018.

[14] Leslie D S, Collins E J. Generalised weakened fictitious play [J]. Games and Economic Behavior, 2006, 56(2): 285-298.

[15] Mazumdar E, Ratliff L J, Sastry S. On the convergence of gradient-based learning in continuous games [J]. arXiv preprint arXiv:1804.05464, 2018.

[16] Chen G. A New Framework for Multi-Agent Reinforcement Learning--Centralized Training and Exploration with Decentralized Execution via Policy Distillation [J]. arXiv preprint arXiv:1910.09152, 2019.

[17] Adler I. The equivalence of linear programs and zero-sum games[J]. International Journal of Game Theory, 2013, 42: 165-177.

[18] Chen X, Deng X. Settling the Complexity of Two-Player Nash Equilibrium [C]// IEEE Annual Symposium on Foundations of Computer Science. 2006, 6: 261-272.

[19] Zinkevich M, Johanson M, Bowling M, et al. Regret minimization in games with incomplete information [J]. Advances in neural information processing systems, 2007, 20.

[20] Bai Y, Jin C. Provable self-play algorithms for competitive reinforcement learning[C]//International conference on machine learning, 2020: 551-560.

[21] Adler I. The equivalence of linear programs and zero-sum games [J]. International Journal of Game Theory, 2013, 42: 165-177.

[22] Goodfellow I. Pouget-Abadie J, Mirza M, et al [J]. Generative adversarial nets, 2014, 2672: 2680.

[23] Li S, Wu Y, Cui X, et al. Robust multi-agent reinforcement learning via minimax deep deterministic policy gradient[C]//Proceedings of the AAAI conference on artificial intelligence. 2019, 33(01): 4213-4220.