Exploration Analysisc in Finite-Horizon Turn-based Stochastic Games

Paper #1091

ABSTRACT

Exploration and exploitation trade-off is one of the key concerns in Reinforcement Learning. Previous work on one-player Markov Decision Processes has reached near-optimal results for PAC and high probability regret guarantees. However, such an analysis is lacking for the more complex stochastic games where multi-players are involved and all players aim to find an approximate Nash Equilibrium. In this work, we concentrate on the exploration issue for the N-player finite-horizon turn-based stochastic games (FTSG). We propose a framework, *Upper Bounding the Values for Players* (UBVP), to guide exploration in FTSGs. The key insight for UBVP is that players choose the optimal policy conditioning on the policies of the others simultaneously. Thus players can explore *in the face of uncertainty* and get close to the Nash Equilibrium. Finally we propose two algorithms based on UBVP. One is *Uniform*-PAC and has a sample complexity of $\tilde{O}(1/\epsilon^2)$ to get an ϵ-Nash Equilibrium for arbitrary $\epsilon > 0$. The other has a regret of $\tilde{O}(\sqrt{T})$ with a high probability.

KEYWORDS

Multi-agent Learning; Reinforcement Learning

1 INTRODUCTION

Sequential decision-making processes among multi-players are quite common in practice, such as board games [21] and computer networks [15]. Many of these processes can be formalized as stochastic games [20] where players interact with the environment and get their rewards. If the dynamics of the environment is known, players can solve the game to find a Nash Equilibrium (NE), at which no player is willing to change its current policy individually. When the environment is unknown to players, i.e. the Reinforcement Learning (RL) setting, finding the NE can be much harder. Each player needs to exploit its best policies as much as possible while ensuring enough exploration to avoid being trapped in sub-optimal policies. At the same time, players need to take others into consideration and finally converge to some approximate NE [16]. The exploration-exploitation trade-off among all players is thus an essential issue for stochastic games.

Substantial progress has been made on solving the exploration-exploitation trade-off for Markov Decision Process (MDP), which models the interaction between a single player and the environment. Some [1, 9] have reached near optimal regrets of order $\tilde{O}(\sqrt{T})$ for T time steps, while some [3, 4] follow Probably Approximately Correct (PAC) or Uniform-PAC frameworks to provide a sample complexity of $\tilde{O}(1/\epsilon^2)$ for an ϵ-optimal policy[1]. These works handle the exploration-exploitation trade-off with the *Optimism in the Face of Uncertainty* (OFU) principle, i.e. choosing the policy that is optimal under the current uncertainty estimation.

Although stochastic games can be considered as an extension of MDPs to multi-player scenarios, the OFU principle cannot be directly applied. The first challenge is that the optimal policy under uncertainty is not a clear concept due to the involvement of multiple players. When other players change their policies, the estimation of the rewards for one player also changes. This leads to the second challenge: players need to explore the environment properly such that they can identify an NE. It is non-trivial to apply the OFU principle to multi-agent problems since players usually cannot reach optima simultaneously in the face of uncertainty. More specifically, players may gain their maximum rewards under different environment parameters. Previous work [8, 13, 27] on finding the equilibrium for general stochastic games extends the methods from MDPs and mostly lacks a non-asymptotic analysis for exploration. UCSG [24] follows the OFU principle for exploration in two-player zero-sum stochastic games and gives a sample complexity bound of order $\tilde{O}(\text{poly}(1/\epsilon))$. However, this algorithm is asymmetric for both players and is not suitable to extend to the more challenging N-player general-sum cases.

In this work, we focus on the exploration-exploitation trade-off for Finite-horizon Turn-based Stochastic Games (FTSG), where there are N players, S states and A actions. Finite horizon indicates that the total number of time steps of one game is limited to a finite number H and turn-based games mean that there is only one player that takes an action at each time step. Many traditional games like Go [22] and computer games like Civilization series [19] fall into this category.

Finding the NEs is the core target for solving FTSGs. Although FTSGs with a known environment are well studied in game theory as extensive games [16], there is little work on reinforcement learning in general FTSGs. We first define PAC and high probability regret as performance measurements in FTSGs, based on the concept of approximate NEs. Then we propose a framework, named *Upper Bounding the Values for Players* (UBVP), to identify NEs for FTSG. To the best of our knowledge, our work is the first to address the exploitation-exploration trade-off for FTSGs with theoretical guarantees. UBVP applies the OFU principle in a way that all players are optimal conditioning on others. Thus they gradually converge to the best responses of each other. In this way, we find an approximate NE.

Based on the framework of UBVP, we propose two algorithms. The first one has a polynomial sample complexity under the Uniform-PAC framework [4], a strengthening of the classical PAC framework. Specifically, we show that with a probability at least $1 - \delta$, for all

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 2020, Auckland, New Zealand

https://doi.org/doi

[1] In this work, $\tilde{O}$ indicates the order ignoring poly-logarithmic terms. Here we ignore other parameters and we leave detailed analysis to latter sections.

$\epsilon > 0$, the number of games using policies that are not ϵ-NE is bounded by $\tilde{O}(\frac{H^4S^2A}{\epsilon^2}\text{polylog}(\frac{N}{\delta}))$. The bound holds for arbitrary $\epsilon > 0$. This result of UBVP on FTSG matches Uniform-PAC algorithm on MDPs, i.e. UBEV. The second algorithm we propose has a regret upper bounded by $\tilde{O}(NLH\sqrt{SAT} + NH^3S^2AL^2 + NH\sqrt{TL})$, which is N times that of UCB-VI [6] on MDPs.

The rest paper is organized as follows. Section 2 reviews related work. Then Section 3 presents our problem formulation of FTSG and Section 4 gives some insight ideas for FTSG. Next, we present our framework UBVP in Section 5 and propose two instantitions in section 6. Finally, we give some discussion and our conclusion.

2 RELATED WORK

Markov Decision Process Much work has been done on the exploration and exploitation trade-off on finite-horizon MDPs. There are mainly two kinds of performance measurements for such algorithms. The regret measures the culminated rewards difference between the optimal policy and the algorithm's policies. UCBVI [1] and vUCQ [10] reach the near optimal regret of $\tilde{O}(H\sqrt{SAT})$[2]. On the other hand, sample complexity follows the Probably Approximately Correct (PAC) framework and measures the times needed to get an approximately optimal policy. UCFH [3] proves to guarantee a sample complexity upper bound of $\tilde{O}(H^3S^2A/\epsilon^2)$. However, [4] indicates that both measurements have limitations and proposes a new framework of Uniform-PAC, which can achieve both near-optimal regret and the PAC guarantee. Our work follows the Uniform-PAC framework and concentrates on the exploration issue under FTSG.

Multi-Objective MDP If the immediate reward is a d-dimensional vector, an MDP turns to be a Multi-Objective MDP (MOMDP) [18]. MOMDPs only involve one agent, and the agent just considers how to trade-off the combination of these rewards. In contrast, in a FTSGs, each agent only aims to maximize its own reward and these agents can be adversarial.

Stochastic games In general, solving the (both finite-horizon and infinite-horizon) N player simultaneous stochastic games is a hard problem even when the parameters of the environment are given [20]. Things are much more complicated under the RL setting, where the environment is unknown. Much work under the RL setting concentrates on two-player zero-sum games [12, 17, 23]. R-MAX [2] firstly gives a PAC guarantee for two-player zero-sum games. Recently UCSG [24] extends UCRL2 [9] in MDP to the two-player zero-sum stochastic games and gives a $\tilde{O}(\text{poly}(1/\epsilon))$ sample complexity. We restrict the games to finite-horizon turn-based games and give a provable framework, which enjoys comparable performances to algorithms on MDP.

Monte Carlo Tree Search Some work [6, 11] analyzes the exploration in MCTS as best arm identification problems under the bandit framework. They focus on the problem of searching in the game trees constructed by the two-player zero-sum turn-based games and try to identify the optimal actions and solve it with the OFU principle from the bandit point of view. By this way they provide a problem-dependent sample complexity bound for this kind of game trees. Our setting differs from their work in two aspects. Firstly, transition functions in stochastic games involve randomness, which make the exploration more complicated. Secondly, we extend the games to more general cases that are not restricted to two-player and zero-sum.

[2]Notice that in this work, we follow [4] and consider the time-dependent dynamics. The results we list here are the results on our setting.

3 PROBLEM FORMULATION

In this section, we first introduce the formal definition for Finite-horizon Turn-based Stochastic Games (FTSG). Then we introduce the Nash Equilibrium (NE) for FTSGs. Finally we introduce the performance measurements in FTSGs.

3.1 Finite-horizon Turn-based Stochastic games

We concentrate on games with a reset action. That is, the environment will reset to an initial state after a fixed number of time steps. We use *episode* to describe the steps between one initial state and its next reset state. Further, we use *depth* to describe the steps from the initial state of the current episode.

Formally, a *Finite-horizon Turn-based Stochastic game* (FTSG) is a six-tuple $\mathcal{G} = \langle N, \mathcal{S}, \mathcal{A}, R, P, H\rangle$, where

- N is the number of players. We use $[N] = \{1, 2, ..., N\}$ to denote the player set and players are distinguished by number $i \in [N]$.
- $\mathcal{S}$ denotes the state space, and $\mathcal{S}_i \subset \mathcal{S},\ i \in [N]$, denotes the state space for player i. Since the games we concern are turn-based, we have $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset$ if $i \neq j$. We use a player function $Player(s)$ to indicate the player that takes an action at s. We denote $|\mathcal{S}| = S$.
- $\mathcal{A}_i$ is the action space for player i and $\mathcal{A} = \cup_{i\in[N]}\mathcal{A}_i$. We denote $|\mathcal{A}| = A$.
- R is a reward function that maps each state-action pair $(s, a) \in \mathcal{S}\times\mathcal{A}$ and a depth h to a probability distribution over $[0, 1]^N$. With a bit abuse of notations, we also use $R(s, a, h)$ to represent one sampled vector from the distribution. Here $R_i(s, a, h)$ represents the sample reward for player i. We denote the expectation of $R(s, a, h)$ as vector $r(s, a, h)$.
- $P(\cdot|s, a, h)$ is the transition probability over $\mathcal{S}$ from state $s \in \mathcal{S}$, a chosen action $a \in \mathcal{A}$ and current horizon. Here we consider a general time-dependent dynamics and thus P has a dependence on h. We further denote $P(s, a, h)$ as the transition vector for state s, action a and depth h.
- H is the horizon of the game.

We have no more assumptions on the game. In general, a state s is able to transit to any state $s' \in \mathcal{S}$. Therefore, the FTSG class we study here is broad and includes a large number of problems. For instance, if $\mathcal{S}_i = \mathcal{S}$, i.e. there is no other player except i in the game, FTSG becomes a Markov Decision Process (MDP). For convenience, we assume that the game begins from a specific state $s_1 \in \mathcal{S}$. The extension to random initial states is straightforward.

The policy of player i ($i \in [N]$), denoted as π_i, maps each state $s \in \mathcal{S}_i$ and its current depth h to an action $a \in \mathcal{A}_i$. We use Π_i to denote the set of all possible policies for π_i. The policies we define here are deterministic[3]. For FTSG, there exists at least one tuple $(\pi_1, \pi_2, ...\pi_N)$ that is the Nash Equilibrium (NE) of the game, and thus we can only consider deterministic policies. Note that the existence of pure strategy NE may fail if the game is not turn-based

[3]They are usually called pure strategies in game theory.

or if it is not finite horizon. In one episode of the game, players follow $\pi = (\pi_1, \ldots \pi_N)$. Further, we denote π_{-i} as the policy tuple which removes π_i from π. To make notations neat, for depth $h \in [H]$, we make $\pi(s,h) = \pi_i(s,h)$ if $s \in \mathcal{S}_i$. In our work below, for kth episode, we denote the policy tuple we use as $\pi^k = (\pi_1^k, \pi_2^k, ..., \pi_N^k)$.

Following the notations of MDP, we use V and Q values to estimate the expected rewards for states and state-action pairs. For player $i \in [N]$, depth $h \in [H]$, state $s \in \mathcal{S}$ and action a for s, the V and Q values are defined as:

$$V_{i,h}^{\pi}(s) := \mathbb{E}\left[\sum_{h'=h}^{H} r_i(s_{h'}, \pi(s_{h'}, h'), h') | s_h = s\right],$$

$$Q_{i,h}^{\pi}(s,a) := \mathbb{E}\left[r_i(s_h, a_h, h) + \sum_{h'=h+1}^{H} r_i(s_{h'}, \pi(s_{h'}, h'), h') | s_h = s, a_h = a\right],$$

where $s_{h'}$ is the state at depth h'. Note that even if $s \notin S_i$, we also define V and Q for player i. Further, we use $V_{i,h}^{\pi}$ without indicating the state to represent the vector for all states of horizon h.

The Bellman equation still holds for the FTSG and can be formalized as

$$V_{i,h}^{\pi}(s) = Q_{i,h}^{\pi}(s, \pi(s,h)) = r_i(s, \pi(s,h), h) + P(s, \pi(s,h), h)^{\top} V_{i,h+1}^{\pi},$$

for $h \in [H]$. Specifically, we define $V_{i,H+1}^{\pi}(s) = 0$ for all $i \in [N]$, $s \in \mathcal{S}$ and any π. We also define s_{H+1} as a terminal state for the convenience of notation.

In the Reinforcement Learning (RL) setting, the dynamics of the environment is unknown to players. That is, the reward function R and the transition function P are unknown. Thus, players can only estimate both functions with the given observations during the interactions with the environment. Therefore, insufficient knowledge about the environment might lead to sub-optimal solutions. Efficient and enough exploration for the environment is one of the key issue for solving games.

3.2 Nash Equilibrium for Stochastic games

In this section we formally introduce Nash Equilibrium, a key concept in games, and the related ϵ-Nash Equilibrium.

Definition 1. *A policy tuple $\pi^* = (\pi_1^*, \pi_2^*, ..., \pi_N^*)$ where $\pi_i^* \in \Pi_i$, $i \in [N]$, is a **Nash Equilibrium** (NE) of FTSG $\mathcal{G}$ if for all i and any $\pi_i' \in \Pi_i$,*

$$V_{i,1}^{\pi^*}(s_1) \geq V_{i,1}^{(\pi_i', \pi_{-i}^*)}(s_1).$$

Definition 2. *A policy tuple $\pi = (\pi_1, \pi_2, ..., \pi_N)$ where $\pi_i \in \Pi_i$, $i \in [N]$, is an ϵ-**Nash Equilibrium** (ϵ-NE) of FTSG $\mathcal{G}$ if for any $\pi_i' \in \Pi_i$,*

$$V_{i,1}^{\pi}(s_1) \geq V_1^{(\pi_i', \pi_{-i})}(s_1) - \epsilon.$$

If we have full information of the game dynamics, we can solve for the NE through dynamic programming. That is, we can work out the NE policy for the state at horizon h and then propagate their rewards to get the NE policy for depth $h-1$. However, in the RL setting, players must explore the environment and then identify an NE. We expect players to converge to an approximate NE after a finite number of games for exploration.

3.3 Performance Measurements

Recall that in MDP, there usually exist some different kinds of standards to measure the performance of an algorithm. Here we first introduce PAC and regret guarantees for MDP and then we extend them for FTSGs.

Without loss of generality, we then assume that $\mathcal{S}_1 = \mathcal{S}$ and this reduce an FTSG to an MDP. This MDP has an optimal policy π_1^* and optimal value function $V_{1,1}^{\pi_1^*}$. Here we use $\mathbb{I}\{\cdot\}$ as the indicator function. Below we list two important measures for MDP [4]:

- PAC: With a probability at least $1-\delta$, there exits a function $f^{PAC}(S, A, H, \epsilon, \delta)$, such that

$$\sum_{t=1}^{\infty} \mathbb{I}\left\{V_{1,1}^{\pi_1^*}(s_1) - V_{1,1}^{\pi_1^t}(s_1) \geq \epsilon\right\} \leq f^{PAC}(S, A, H, \epsilon, \delta).$$

- High probability regret: With a probability at least $1-\delta$, there exits a function $f^{HPR}(S, A, H, T, \delta)$, such that at episode T,

$$TV_{1,1}^{\pi_1^*}(s_1) - \sum_{t=1}^{T} V_{1,1}^{\pi_1^t}(s_1) \leq f^{HPR}(S, A, H, T, \delta).$$

These standards cannot be extended to FTSGs directly since there exits more than one agent and the learning goal here is to find the approximate NE. Therefore we define new standards for FTSGs.

We define the number of episodes during which the learning algorithm does not choose ϵ-NE, which can be denoted as:

$$L^{\epsilon} = \sum_{k \in \mathbb{N}} \mathbb{I}\left\{\pi^k \text{ is not an } \epsilon\text{–NE}\right\}.$$

Then upper bounding L^{ϵ} can lead to a PAC guarantee for FTSGs.

Recall that regret describes the distance between the maximum rewards one agent can get with a hindsight policy and its actual rewards. Thus for each player i, we define its regret as

$$Reg_i(T) = \max_{\pi_i} \sum_{t=1}^{T} V_{i,1}^{(\pi_i, \pi_{-i}^k)}(s_1) - \sum_{t=1}^{T} V_{i,1}^{\pi^k}(s_1).$$

This definition of regret reduce to that in MDP when there is only one player. Then we can define the total regret for our algorithm as

$$Reg(T) = \sum_{i \in [N]} Reg_i(T).$$

Therefore we can also measure the performance of an algorithm for FTSGs with PAC or high probability regret bound.

4 EXPLORATION IN FTSG

Players gain information about the environment by playing games. In each game, players together influence the trajectory. One player might fail to gain its desired information, since other players take actions according to their own purposes. Hence the exploration for FTSG is much more complicated than that of MDP. In this section, we use a simple case to introduce the core insight for exploration in FTSG.

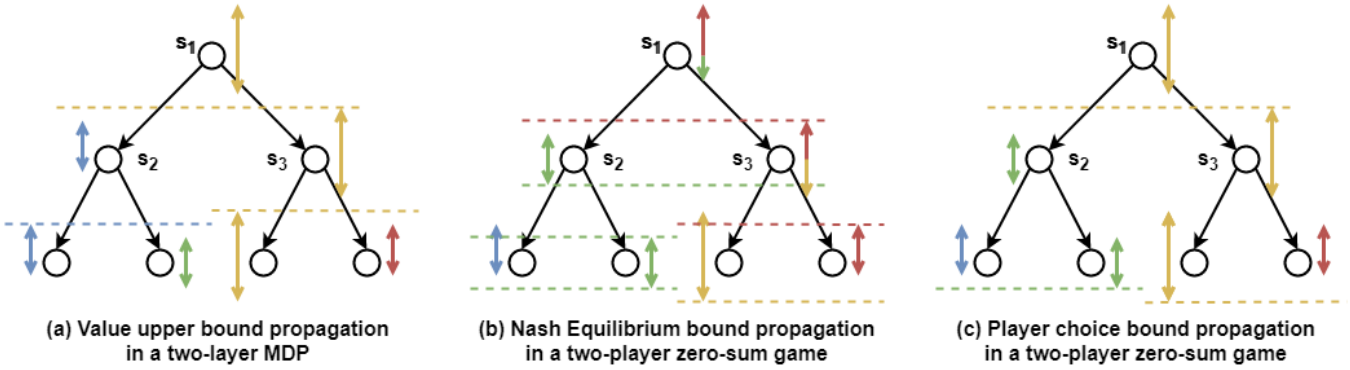

Figure 1: This is a simple example with 3 states, 2 actions and 2 layers to illustrate the exploration. For convenience, we assume the transitions are deterministic and the rewards are 0 unless a terminal node is reached. The color bar in figures indicate the high probability bound for estimated values. In (a), there is only one player and we propagate backward the upper confidence bound of V values. In (b) and (c), s_1 is for player 1 and s_2, s_3 are for player 2. The rewards for player 1 are shown with color bars and player 2 has opposite rewards. In (b), We propagate backward the bounds for NE. In (c), we propagate backward the value bounds of chosen actions.

4.1 Exploration in MDP

The key to efficiently find approximate NEs for FTSG is the exploration-exploitation trade-off. This trade-off has been handled for the special case of MDP. In particular, previous work [1, 9] proposes efficient exploration algorithms following the *Optimism in the Face of Uncertainty* (OFU) principle. The essential idea of OFU is to choose the policy with maximum expected rewards under the current estimation for the true model. We illustrate this principle in Fig. 1(a). Since there is only one player involved, we just need to concentrate on the upper confidence bounds of the rewards. For each state, we choose the action with the maximum upper bound and propagate back this upper bound. Then we find a policy that is optimal under current uncertainty.

To make this clear, we consider a simple case where there are only 3 states and 2 actions, as shown in Fig. 1(a). Rewards are only given at terminal nodes. We use colorful bars to indicate the confidence set for the optimal V values of nodes. Then the OFU principle chooses the actions with high upper bounds. In Fig. 1(a), the chosen policy for this iteration follows the trajectory of the yellow bar.

4.2 NE uncertainty estimation

In stochastic games, we can still estimate the uncertainty of reward and transition functions. However, it is nontrivial to identify the optimal policy for one player under current uncertainty since its final cumulative rewards is also influenced by other players. The optimal policy for this player can vary if the policies of others change. A straightforward idea is to estimate the uncertainty of the NE values and choose the optimal policy tuple according to the estimation. However, the chosen policy under this NE uncertainty estimation fails to explore efficiently since players cannot reach optimality simultaneously.

We still use the simple case for inspiration. As shown in Fig. 1 (b), we consider a two-player zero-sum FTSG, where all NEs have equal reward values. Player 1 takes actions at the first layer and player 2 acts at the second layer. Then a sampled reward for player 1 is returned. The color bars also indicate the confidence bounds for the rewards for player 1 under NE. The uncertainty estimation for the four terminal reward functions (i.e. the four color bars) can be calculated with concentration bounds. Then we back-propagate the confidence bound for the NE values with current rewards estimation. As indicated in the figure, for state s_3, player 2 here always chooses the minimum value of both actions, and thus the NE value for s_3 is bounded by the minimum of upper and lower bounds of its children. Then the confidence bound for s_3 is bounded by the upper bound of red bar and the lower bound of yellow bar.

However, after bounding the NE values, we cannot assign proper actions for both players. Following the OFU principle, it is easy to see that player 1 should choose the action on the right for the largest NE value and then player 2 should choose the left for s_3. However, we can see that the yellow bar does not directly contribute to the NE value estimation of s_1. That is, the trajectory in this episode is not the one we aim to explore, leading to inefficient exploration in games. This happens because the two players cannot reach their optima at the same time in the face of uncertainty. As shown in Fig 1(b), player 1 takes action according to the upper bound value while player 2 follows the lower bound. This mismatch cannot guarantee efficient convergence to NE. Further, when we extend to more general N player FTSGs, there might exist more than one NE in a game and they usually have different rewards. Using one NE as the target can result in bad complexity if the algorithm converges to another NE.

4.3 Reaching optimality simultaneously

As observed above, we require all players to explore collaboratively and converge to an NE. Under the setting of FTSGs, the player at some depth can exactly infer the best choices for nodes below it. Thus this player can choose its optimal action conditioning on the choices of below states, rather than the NE estimation. Then we propagate backward the confidence bound of the chosen action.

Algorithm 1 Upper Bounding the Values for Players

Input: $N, \mathcal{S}, \mathcal{A}, H, \mathcal{H}^1 = \emptyset, \delta$
for episode $k = 1, 2, \ldots$ **do**
  for $h = H, H-1, \ldots, 1$ **do**
    for $s \in \mathcal{S}$ **do**
      for All possible action a for s such that $n^k(s,a) > 0$ and all player $i \in [N]$ **do**
        Compute $\bar{r}_i^k(s,a,h)$ with Eq. (1)
        Compute $\bar{P}^k(s'|s,a,h)$ with Eq. (2) for possible s'
        Compute $\tilde{Q}_{i,h}^k(s,a) = ComputeQ$
      end for
      $j = Player(s)$
      $\pi^k(s,h) = \pi_j^k(s,h) = \arg\max_a Q_{j,h}^k(s,a)$
      for All $i \in [N]$ **do**
        Compute $V_{i,h}^k(s) = Q_{i,h}^k(s, \pi^k(s,h))$
      end for
    end for
  end for
  for step $h = 1, \ldots, H$ **do**
    Choose action $a_h^k = \pi^k(s_h^k, h)$
    Get to state s_{h+1}^k and get reward vector R_h^k
  end for
  Update $\mathcal{H}^{k+1} = \mathcal{H}^k \cup \{(s_h^k, a_h^k, R_h^k, s_{h+1}^k)\}_{h=1}^H$
end for

As shown in Fig. 1 (c), player 2 takes actions with minimum lower bounds. That is, if player 2 chooses the right action at s_2 and chooses the left action at s_3, then player 1 considers the green bar and the yellow bar as the value confidence sets for s_2 and s_3. With this estimation, player 1 chooses the right action. By operating like this, players reach optima conditioning on the choices of other players. Hence they gradually converge to the best response of the others and this leads to an NE. Furthermore, this idea has no assumption on NEs and can be easily extended to N-player FTSGs. Based on this insight, we propose our algorithm, and we give the sample complexity analysis in the next section.

5 UPPER BOUNDING THE VALUES FOR PLAYERS

We now present our framework, *Upper Bounding the Values for Players* (UBVP). We first give the procedure for UBVP in this section. Then we give an analysis outline for UBVP. Notice that there is a key bonus function in the procedure and we do not specify it in this section. In next section, we design specific bonus functions with Uniform-PAC or high probability regret guarantees.

5.1 UBVP procedure

Our work is mainly motivated by the *optimism in the face of uncertainty* (OFU) principle in MDP and our goal now is to conduct efficient exploration to find an approximate NE. The key part for exploration in stochastic games is to estimate the uncertainty of the values and design proper policies to interact with the environment.

Following the OFU principle, the player at some state should have estimations of the expected rewards for each of its actions and it chooses the optimal one. As proposed in Section 4, we upper bound values of states based on the actions of states from deeper depths. We build the value upper bounds by backward induction. From depth H, players can choose optimal actions without the influence of others. Then we calculate the reward upper bounds of all players based on these actions. Next we propagate these upper bounds back to depth $H-1$. Through backward induction, we can work out the value bounds for each depth. Then players choose the actions with maximum values. Following our calculation, with high probability, each player in the game chooses its optimal policy conditioning on the policies of other players in the face of current uncertainty.

Algorithm 2 ComputeQ

Input: $V_{i,h+1}^k, \bar{P}^k(s,a,h), \bar{r}_i^k(s,a,h), n^k(s,a,h), \{n^k(s,a,s',h)\}_{s'}, H, S, \delta$
$b_h^k(s,a) = \phi(n^k(s,a,h), \{n^k(s,a,s',h)\}_{s'}, \bar{P}(s,a,h), V_{i,h+1}^k, \delta)$
$Q_{i,h}^k(s,a) = \min\{Q_{i,h}^{k-1}(s,a), H, \bar{r}_i(s,a,h) + \bar{P}^k(s,a,h)^\top V_{i,h+1}^k + b_h^k(s,a)\}$
Output: $Q_{i,h}^k(s,a)$

Below we give the framework of UBVP. This framework is applicable to different implementations for the value estimations in order to satisfy different learning goals.

For convenience, we use $s_h^{k'}$, $a_h^{k'}$ and $R_h^{k'}$ to respectively denote the reached state, corresponding action and immediate reward at depth h of episode k'. Before episode k, we have a set of observed data $\mathcal{H}^k$, including $(s_h^{k'}, a_h^{k'}, R_h^{k'}, s_{h+1}^{k'})$, $h \in [H]$ and $k' \in [k-1]$. With $\mathcal{H}^k$, we calculate the count of visiting the state-action-depth tuple (s,a,h), denoted by $n^k(s,a,h)$, and the count of transiting to state s' immediately from (s,a,h), denoted by $n^k(s,a,s',h)$. For (s,a,h) at episode k with $n^k(s,a,h) > 0$, we estimate its immediate reward and transition probability by

$$\bar{r}_i^k(s,a,h) = \sum_{k' \in [k-1]} R_h^{k'} \mathbb{I}(s_h^{k'} = s, a_h^{k'} = a)/n^k(s,a,h), \quad (1)$$

$$\bar{P}^k(s'|s,a,h) = n^k(s,a,s',h)/n^k(s,a,h), \quad (2)$$

where s' is any possible next state.

Then for state-action pair (s,a) at h, we can calculate the upper bound of Q and V values from depth H to 1. More specifically, for each player i, with calculated V value upper bounds at depth $h+1$, denoted as $V_{i,h+1}^k$, we can work out the upper bound value $Q_{i,h}^k(s,a)$ by adding $\bar{r}_i(s,a,h) + \bar{P}^k(s,a,h)^\top V_{i,h+1}^k$ with an extra bonus term $b_h^k(s,a) = \phi(n^k(s,a,h), \{n^k(s,a,s',h)\}_{s'}, \bar{P}(s,a,h), V_{i,h+1}^k, \delta)$. Here b_h^k is the bonus to bound the uncertainty of the estimation for current Q value and we call ϕ the bonus function. The exact forms of ϕ is not given here. In fact ϕ can be defined in different ways to satisfy different performance measurements. We require it to satisfy the below property:

Property 1. *The bonus function ϕ for UBVP should satisfy that with a probability at least $1-\delta$, for all $i \in [N]$, $k > 0$, $h \in [H]$ and*

$(s,a) \in \mathcal{S} \times \mathcal{A}$:

$$|(r_i^k(s,a,h) - \bar{r}_i^k(s,a,h)) + \left(P^k(s,a,h) - \bar{P}^k(s,a,h)\right)^\top V_{i,h+1}^{*,k}| \le b_h^k(s,a), \tag{3}$$

where $V_{i,h+1}^{*,k}(s) = \max_{\pi_i \in \Pi_i} V_{i,h+1}^{(\pi_i, \pi_{-i}^k)}(s)$.

This property of ϕ can ensure that our calculated $Q_{i,h}^k(s,a)$ is a proper upper bound for $\max_{\pi_i} Q_{i,h}^{(\pi_i,\pi_{-i}^k)}$, as we will soon prove in Lemma 5.2. This is the key that UBVP can guide proper exploration and converge to the NE.

The pseudo code of the UBVP algorithm is given in Alg. 1. From line 3 to line 15, we calculate the upper bounds of V values for all N players through backward induction. At the same time, we work out the policy tuple π^k by letting each player choose the optimal action. This tuple is then used to play the game of this episode and collect data. The upper bounds calculation for Q values are given in Alg. 2.

5.2 Analysis for UBVP

Above we have proposed the framework for UBVP, except the exact form for ϕ. Here based on the property of ϕ, we give a general analysis outline.

Firstly, we define the **best response distance** to describe how close a policy tuple is to the NE.

Definition 3. *For a policy tuple $\pi = (\pi_1, \pi_2, ..., \pi_N)$ where $\pi_i \in \Pi_i$, $i \in [N]$, the **best response distance** (Bsd) of player i for π is defined as*

$$Bsd_i(\pi) := \max_{\pi_i' \in \Pi_i} V_{i,1}^{(\pi_i', \pi_{-i})}(s_1) - V_{i,1}^{(\pi)}(s_1).$$

Intuitively, $Bsd_i(\pi)$ measures the difference between the largest expected value that player i can get against π_{-i} and the actual value with π_i. Recall the definition of NE, it is natural to connect this value with ϵ-NE.

Lemma 5.1. *For policy tuple $\pi = (\pi_1, \pi_2, ..., \pi_N)$ where $\pi_i \in \Pi_i$, $i \in [N]$, if $Bsd_i(\pi) \le \epsilon$ for all player i, then π is an ϵ-NE.*

Although the proof is straightforward, we give it in the Appendix A for completeness[4].

With Bsd_i defined above, we now bound L^ϵ and $R(T)$ with

$$L^\epsilon \le \sum_{k \in \mathbb{N}^+} \mathbb{I}\left[\exists i \in [N], Bsd_i(\pi^k) > \epsilon\right],$$
$$Reg(T) \le \sum_{i \in [N]} \sum_{k \in [T]} Bsd_i(\pi^k),$$

where $\mathbb{N}^+ := \{1, 2, ...\}$ is the set of positive integers. Therefore, we can turn to analyze Bsd_i to measure the performance of UBVP.

It is easy to see that in UBVP, players are symmetric, and thus the result for one can be adapted to the others. Now we fix our attention on player i, $i \in [N]$. We first give the key lemma that connects the upper bound V and Bsd_i.

[4]Our appendix is as an anonymous file at url: https://github.com/anonymous-file/aamas_appendix.git

Lemma 5.2. *With a probability at least $1-\delta$, for any $i \in [N]$, $k \in \mathbb{N}$, $h \in [H]$ and $s \in \mathcal{S}$,*

$$\max_{\pi_i \in \Pi_i} V_{i,h}^{(\pi_1, \pi_{-i}^k)}(s) \le V_{i,h}^k(s).$$

Proof. This lemma can be proved by induction on h. Below discussion is based on the event that inequality(3) holds for all $i \in [N]$, $k \in \mathbb{N}$ and $h \in [H]$.

First we consider $h = H$ and any $k \in \mathbb{N}$. For any $s \notin \mathcal{S}_i$, the choice of π_i has no influence on $V_{i,H}^{\pi_i, \pi_{-i}^k}(s)$ and we always have $\max_{\pi_i} V_{i,H}^{(\pi_i, \pi_{-i}^k)}(s) = V_{i,H}^{\pi^k}(s) \le V_{i,H}^k(s)$. For any $s \in \mathcal{S}_i$, we first denote that $\pi_i' = \arg\max_{\pi_i} V_{i,H}^{(\pi_i, \pi_{-i}^k)}(s)$. Then we have that $V_{i,H}^{(\pi_i', \pi_{-i}^k)}(s) \le Q_{i,H}^k(s, \pi_i'(s,H)) \le V_{i,H}^k(s)$.

Then we assume that for $h+1$, $h \in [H-1]$, any $k \in \mathbb{N}$ and $s \in \mathcal{S}$, we assume that

$$\max_{\pi_i \in \Pi_i} V_{i,h+1}^{(\pi_i, \pi_{-i}^k)}(s) \le V_{i,h+1}^k(s).$$

Now we turn to depth h. If $s \in \mathcal{S}_i$, we denote $\pi_i' = \arg\max_{\pi_i \in \Pi_i} V_{i,h}^{(\pi_i, \pi_{-i}^k)}(s)$ and $a' = \pi_i'(s,h)$ for convenience. Then we have

$$\begin{aligned}
\max_{\pi_i \in \Pi_i} V_{i,h}^{(\pi_i, \pi_{-i}^k)}(s) \le & r_i(s,a',h) + \sum_{s' \in \mathcal{S}} P(s'|s,a',h) \max_{\pi_i \in \Pi_i} V_{i,h+1}^{(\pi_i, \pi_{-i}^k)}(s') \\
\le & \bar{r}_i^k(s,a',h) + \bar{P}^k(s,a',h)^\top V_{i,h+1}^k + b_h^k(s,a) \\
\le & Q_{i,h}^k(s,a') \\
\le & V_{i,h}^k(s).
\end{aligned}$$

The second inequality holds by using Property 3 of ϕ and the induction assumption.

If $s \notin \mathcal{S}_i$, it is simple to have

$$\begin{aligned}
& \max_{\pi_i \in \Pi_i} V_{i,h}^{(\pi_i, \pi_{-i}^k)}(s) \\
\le & r_i(s, \pi^k(s,h), h) + \sum_{s' \in \mathcal{S}} P(s'|s, \pi^k(s,h), h) \max_{\pi_i \in \Pi_i} V_{i,h+1}^{(\pi_i, \pi_{-i}^k)}(s') \\
\le & \bar{r}_i^k(s, \pi^k(s,h), h) + \bar{P}^k(s, \pi^k(s,h), h)^\top V_{i,h+1}^k + b_h^k(s, \pi^k(s,h)) \\
\le & V_h^k(s).
\end{aligned}$$

The second inequality holds using inequality(3). Therefore, we finish our proof with induction. □

This is the key that UBVP can conduct exploration and converge to an NE. It should be emphasized that this inequality holds for all states. That is, even if state $s \notin \mathcal{S}_i$,the inequality still holds for player i. Intuitively, this is because we construct $V_{i,h}^k$ such that it is the upper bound of the optimal V for player i conditioning on other player's policy. Besides, $V_{i,h}^{(\pi_i, \pi_{-i}^k)}(s)$ has the same action with π^k on states not in $\mathcal{S}_i$ and thus we ensure the upper bound property. Furthermore, the inequalities hold for all players at the same time, and this ensures them to converge to the best response of other players. This is exactly why we can converge to NEs.

Now we define $\Delta_i^k := V_{i,1}^k(s_1) - V_{i,1}^{\pi^k}(s_1)$. Using Lemma 5.2, we have that with high probability, we have

$$Bsd_i(\pi^k) \le \Delta_i^k.$$

Then we turn to upper bound the sample complexity of Δ_i^k. Notice that the two terms in Δ_i^k follows the same policy tuple π^k but are calculated under different FTSGs. Inspired by techniques in MDP, we can decompose Δ_i^k and upper bound the separated terms.

For clarity, we denote x as a state-action pair (s, a). Before episode $k \in \mathbb{N}$, we are given the history $\mathcal{H}^k$. Then we can calculate the empirical mean of rewards and transition functions for all x and depth h, as shown in the line 6-7 of Alg. 1. For episode k and depth h, denote $w_h^k(x)$ as the probability of reaching state-action pair x at depth h following policy π^k. Now we can decompose Δ_i^k as

$$\begin{aligned}
&V_{i,1}^k(s_1) - V_{i,1}^{\pi^k}(s_1)\\
=&\bar{r}_i^k(s_1, \pi(s_1,1),1) - r_i(s_1,\pi(s_1,1),1) + b_1^k(s_1,\pi(s_1,1))\\
&+ \bar{P}^k(s_1,\pi(s_1,1),1)^\top V_{i,2}^k - P(s_1,\pi(s_1,1),1)^\top V_{i,2}^{\pi^k}\\
=&(\bar{r}_i^k(s_1,\pi(s_1,1),1) - r_i(s_1,\pi(s_1,1)),1)\\
&+ \left(\bar{P}^k(s_1,\pi(s_1,1),1) - P(s_1,\pi(s_1,1),1)\right)^\top V_{i,2}^k\\
&+ P(s_1,\pi(s_1,1),1)^\top \left(V_{i,2}^k - V_{i,2}^{\pi^k}\right) + b_1^k(s_1,\pi(s_1,1))\\
=&\sum_{h=1}^{H}\sum_{x\in\mathcal{S}\times\mathcal{A}} w_h^k(x)\left((\bar{r}_i^k - r_i)(x,h) + \left(\bar{P}^k - P\right)(x,h)^\top V_{i,h+1}^k + b_h^k(x)\right).
\end{aligned} \tag{4}$$

Therefore, we can design proper bonus function ϕ such that Eq. (4) can be bounded. This bound for Δ_i^k is also the bound for $Bsd_i(\pi^k)$. And finally, we can get the upper bounds for either $R(T)$ or L^ϵ.

The specific theoretical analysis can vary for different forms of ϕ and different performance measurement. Fortunately, exiting work on MDPs has proposed various value-based algorithms in UCB-type. Many of them can be be adapted to UBVP.

6 TWO ALGORITHMS BASED ON UBVP

As analyzed above, we can design different kinds of ϕ to satisfy different kinds of performance measurement. Below we give two kinds of instantation for UBVP. The first algorithm ensures a Uniform-PAC guarantee with a sample complexity $\tilde{O}(1/\epsilon^2)$ and the second has a regret of $\tilde{O}(\sqrt{T})$ with high probability.

6.1 A Uniform-PAC algorithm

Uniform-PAC[4] is a variant of PAC, which does not require a known ϵ as input of the algorithm. Formally, an algorithm is Uniform-PAC if for with a probability at least $1-\delta$, for all $\epsilon > 0$, L^ϵ is upper bounded by some function $f^{UPAC}(N, S, A, H, \epsilon, \delta)$. Here we follow the design of UBEV in [4] and choose the bonus term ϕ^{UPAC} as:

$$\phi^{UPAC} = (H+1)\sqrt{\frac{2\ln\ln(\max\{e, n^k(s,a,h)\}) + \ln((24N+30)SA/\delta)}{n^k(s,a,h)}}.$$

It is easy to verify that ϕ^{UPAC} satisfies Property 1 following the proof of Corollary E.1 in [4]. We give more formal proof in the Appendix B.

Note that all players share the same bonus, so at each (s, a) pair, the bonus term $b_h^k(s,a)$ only needs to be calculated once for all players.

We now show that our algorithm UBVP with ϕ^{UPAC} is Uniform-PAC and we give the upper bound of its sample complexity below.

Theorem 6.1. *(Uniform-PAC bound) If UBVP chooses the bonus function as ϕ^{UPAC}, then for $\delta \in (0,1)$ and $\epsilon > 0$, with a probability at least $1-\delta$, the sample complexity of UBVP for an FTSG is upper bounded by*

$$O\left(\frac{H^4SA}{\epsilon^2}\min\{S, N+\epsilon(N+S^2A)\}\text{polylog}\left(N,H,S,A,\frac{1}{\delta},\frac{1}{\epsilon}\right)\right).$$

Recall that if for some player i, $\mathcal{S}_i = \mathcal{S}$, the FTSG is in fact an MDP and L^ϵ for FTSG is also the related sample complexity for MDPs. Indeed, the complexity of FTSG is comparable with the related work on MDPs. Specifically, the previous Uniform-PAC algorithm for MDPs, UBEV, achieves the sample complexity upper bound of $O(\frac{H^4SA}{\epsilon^2}\min\{S, 1+\epsilon S^2A\}\text{polylog}\left(H,S,A,\frac{1}{\delta},\frac{1}{\epsilon}\right))$. We can directly apply the analysis for UBEV to get a sample complexity of $O(\frac{NH^4SA}{\epsilon^2}\min\{S, 1+\epsilon S^2A\}\text{polylog}\left(H,S,A,\frac{N}{\delta},\frac{1}{\epsilon}\right))$ for UBVP. However, careful analysis can improve the first term of the sample complexity to $O(\frac{H^4S^2A}{\epsilon^2}\text{polylog}\left(H,S,A,\frac{N}{\delta},\frac{1}{\epsilon}\right))$ and we give a proof sketch of this part in section 6.1.1. Therefore, for relatively large ϵ, UBVP solves FTSG with a sample complexity the same as that of UBEV, except an extra N in logarithmic terms. Considering that UBVP works out a policy tuple with N policies as the solution for FTSG, the extra cost on N is indeed cheap. For a fixed $\epsilon > 0$, the lower bound of the sample complexity for MDPs is $\tilde{\Omega}(\frac{H^3SA}{\epsilon^2}\ln(\frac{SA}{\delta}))$ [4] for sufficiently small ϵ. It is still unknown whether the FTSG games have larger lower bounds.

6.1.1 Proof Sketch. State-action pair x will contribute much to Δ_i^k only if it has a large w_h^k. Thus we use set $\mathcal{E}_h^k$ as $\{x \in \mathcal{S}\times\mathcal{A} : w_h^k \geq \epsilon/(2H^2S)\}$. For those x outside of $\mathcal{E}_h^k$, their contribution can be bounded by a small term. Then we apply the Chernoff-Hoeffding's bound [7] and its variant [25] to give that

$$\begin{aligned}
&\sum_{h=1}^{H}\sum_{x\in\mathcal{E}_h^k} w_h^k(x)\left((\bar{r}_i^k - r_i)(x,h) + \left(\bar{P}^k - P\right)(x,h)^\top V_{i,h+1}^k + b_h^k(x)\right)\\
\leq&(2H\sqrt{S}+2)\sum_{h=1}^{H}\sum_{s\in\mathcal{E}_h^k} w_h^k(x)\sqrt{\frac{2\text{llnp}(n^k(x,h)) + \ln(\frac{(24H+30)SA}{\delta})}{n^k(x,h)}},
\end{aligned} \tag{5}$$

where δ is the input parameter that shows with what probability that we cannot tolerate the error. It worth a notification here that Eq. (5) does not include specific paramters for player i. That is, if Eq. (5) holds for player i, it also holds for all other players. This is why we can remove the term N in the sample complexity.

Now we just need to upper bound Eq. (5). For a state-action pair x, its visit count $n^k(x)$ can be considered as samples from its reachability distribution. Thus we are able to connect $n^k(x)$ with $\sum_{k,h} w_h^k(x)$. Then notice for each episode k, x either satisfies $w_h^k(x) < \epsilon/(2H^2S)$ for all $h \in [H]$ or accumulate at least $\epsilon/(2H^2S)$ for $\sum_{k,h} w_h^k(x)$. In this way, we can upper bound the number of episodes that have large Δ_i^k. With a more careful analysis, we give the below lemma for player i.

Lemma 6.2. *Following UBVP, with a probability $1-\delta$, for any $\epsilon > 0$ and any $i \in [N]$, the number of episodes that $\Delta_i^k > \epsilon$ is at most*

$$O\left(\frac{H^4S^2A}{\epsilon^2}\text{polylog}\left(N,H,S,A,\frac{1}{\delta},\frac{1}{\epsilon}\right)\right).$$

The detailed proof for lemma 6.2 is given in the appendix B.

With the result for player i, we then turn to L^ϵ. Since $Bsd_i(\pi^k) \leq \Delta_i^k$, we always have $L^\epsilon \leq \sum_{k\in\mathbb{N}} \mathbb{I}(\exists i \in [N], s.t.\ \Delta_i^k > \epsilon)$. However, close inspection of lemma 6.2 reminds us that the sample complexity is mainly bounded in the form of Eq. (5), which is same for all players. That is, the result given in lemma 6.2 holds for all N players at the same time. Therefore, we upper bound the sample complexity of L^ϵ and finish the proof for the $\tilde{O}(\frac{H^4S^2A}{\epsilon^2})$ term in theorem 6.1. The complete proof can refer to the Appendix B.

6.2 An algorithm with High Probability Regret Bound

For the high probability regret guarantee, we can choose two kinds of bonus functions:

$$\phi_1^{HPR} = 8HL\sqrt{1/n^k(s,a,h)},$$

$$\phi_2^{HPR} = \sqrt{\frac{8LVar_{s'\sim\bar{P}(s,a,h)}V_{i,h+1}^k(s')}{n^k(s,a,h)}} + \frac{14HL}{3n^k(s,a,h)} + HL\sqrt{\frac{1}{n^k(s,a,h)}} + \sqrt{\frac{8\sum_{s'}\bar{P}(s,a,s',h)C(s')}{n^k(s,a,h)}},$$

where $C(s') = \min\{10^4H^3S^2AL^2/n^k(s,a,s',h), H^2\}$ and $L = \ln(5HSATN/\delta)$. The two ϕ functions are designed by extending UCB-VI [6]. We add $HL\sqrt{1/n^k(s,a,h)}$ to bonuses of UCBVI to design ϕ. The extra term is designed to upper bound reward functions. The two kinds of ϕ satisfy Property 1 using the analysis of Lemma 18 and Sec.5.2 in [6]. Then we can give below theorem for UBVP:

Theorem 6.3. *(High Probability Regret Bound) If UBVP use ϕ_1^{HPR} as its bonus function, then with a probability at least $1-\delta$, the regret of UBVP has an order of:*

$$R(T) = O\left(NHL\sqrt{HSAT} + H^3S^2AL^2\right).$$

If UBVP use ϕ_2^{HPR} as its bonus function, then with a probability at least $1-\delta$, the regret of UBVP has an order of:

$$R(T) = O\left(NLH\sqrt{SAT} + NH^3S^2AL^2 + NH\sqrt{TL}\right).$$

The proof is straightforward by combining Lemma 5.2 with the analysis in [6]. Notice that the setting in [6] is time-independent dynamics. Thus we only need to replace S with SH when pigeonhole principle is used.

6.3 Other extensions

We have listed two kinds of instantations for UBVP. In fact, UBVP can further adapts to many other algorithms on MDP to solve FTSGs. Recent work provides tighter bounds by considering the lower bounds for Q values [5, 26]. That is, we can extra update below two values during the process of UBVP:

$$\underset{\sim}{Q}_{i,h}^k(s,a) = \min\{\underset{\sim}{Q}_{i,h}^{k-1}(s,a), H, \bar{r}_i(s,a,h) + \bar{P}^k(s,a,h)^\top \underset{\sim}{V}_{i,h+1}^k - b_h^k(s,a)\},$$

$$\underset{\sim}{V}_{i,h+1}^k(s) = \underset{\sim}{Q}_{i,h}^k(s,\pi^k(s,h)).$$

Then we can still choose ϕ following existing work [5, 26] to give the corresponding regret bounds or PAC sample complexity. Notice that these designs of ϕ highly rely on the estimated Q values and thus the regret bound or sample complexity should be N times that in MDP.

7 DISCUSSION

As analyzed above, we propose a Uniform-PAC algorithm UBVP for FTSG. Here in this section, we give some further discussion for UBVP.

Multiple NEs: As is known in game theory, it is possible for a game to have more than one NE. In general cases, players gain different rewards at different NEs. If players converge to different NEs, their policy tuple is hardly an NE. Our method UBVP can indeed guarantee players to converge to the same NE, since we ensure that players simultaneously reach optimality in each episode. That is, after playing k games where k is large enough, we would expect π^k is an ϵ-NE. However, UBVP cannot ensure which NE it is.

More specifically, in FTSG, multiple NEs appear if at some state $s \in S_i$, player i gets the same optimal rewards for more than one actions. In UBVP, players act according to the upper bound of the Q value estimations. Therefore, UBVP would explore all the NEs. Simply consider a state $s \in \mathcal{S}_i$ at depth H. If the optimal expected reward for player i corresponds to two actions. Then we can expect the two actions would be chosen about the same number of times. Thus, we can expect UBVP would try to reduce the uncertainty for all NEs.

[14] defines some kinds of criteria for different NEs. As we analyzed above, with enough number of episodes, UBVP should have explored all the NEs. Therefore it is possible for UBVP to recommend a specific NE with some extra designs for choosing policy. We left it as a future work.

Decentralized cases: Our algorithm ensures the convergence to approximate NE only if all players follow UBVP. The exploration of the unknown environment requires all players to cooperate. Otherwise it is possible that some potential policies are not identified. Therefore if the learning goal is to efficiently identify the Nash Equilibrium, it is necessary that all players can explore together. In the decentralized cases, each player aims rewards without considering other players. UCSG [24] provides a solution for two-player zero-sum stochastic games. Intuitively, we think it might be possible that a centralized method can also perform well, e.g. low regret, in the decentralized case.

8 CONCLUSIONS

To sum up, we extend the OFU principle for exploration in MDP to N-player FTSGs and propose a framework named UBVP that can guide efficient exploration to converge to approximate NEs. UBVP is the first method that gives a non-asymptotic analysis of NEs for FTSG in the Reinforcement Learning setting. We also proposes two algorithms based on UBVP, which have Uniform-PAC and high probability regret guarantees respectively. Our analysis shows that these algorithm match the results on finite-horizon MDPs expect the N terms.

Stochastic games that are not turn-based or infinite-horizon are more complicated cases and the exploration for such games is still an open challenge. Our work could provide some insights for solving this challenge. Essentially, UBVP mainly provides a way for players to choose actions that can be optimal against others. We believe that this is also one of the key issues for these more general cases and is worth of a systematic investigation.

REFERENCES

[1] Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. 2017. Minimax Regret Bounds for Reinforcement Learning. In *Proceedings of the 34th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Doina Precup and Yee Whye Teh (Eds.), Vol. 70. PMLR, International Convention Centre, Sydney, Australia, 263–272. http://proceedings.mlr.press/v70/azar17a.html

[2] Ronen I Brafman and Moshe Tennenholtz. 2002. R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research* 3, Oct (2002), 213–231.

[3] Christoph Dann and Emma Brunskill. 2015. Sample complexity of episodic fixed-horizon reinforcement learning. In *Advances in Neural Information Processing Systems*. 2818–2826.

[4] Christoph Dann, Tor Lattimore, and Emma Brunskill. 2017. Unifying PAC and regret: Uniform PAC bounds for episodic reinforcement learning. In *Advances in Neural Information Processing Systems*. 5713–5723.

[5] Christoph Dann, Lihong Li, Wei Wei, and Emma Brunskill. 2019. Policy Certificates: Towards Accountable Reinforcement Learning. In *International Conference on Machine Learning*. 1507–1516.

[6] Aurélien Garivier, Emilie Kaufmann, and Wouter M Koolen. 2016. Maximin action identification: A new bandit framework for games. In *Conference on Learning Theory*. 1028–1050.

[7] Wassily Hoeffding. 1994. Probability inequalities for sums of bounded random variables. In *The Collected Works of Wassily Hoeffding*. Springer, 409–426.

[8] Junling Hu and Michael P Wellman. 2003. Nash Q-learning for general-sum stochastic games. *Journal of machine learning research* 4, Nov (2003), 1039–1069.

[9] Thomas Jaksch, Ronald Ortner, and Peter Auer. 2010. Near-optimal Regret Bounds for Reinforcement Learning. *Journal of Machine Learning Research* 11 (2010), 1563–1600.

[10] Sham M Kakade, Mengdi Wang, and Lin F Yang. 2018. Variance Reduction Methods for Sublinear Reinforcement Learning. *arXiv: Artificial Intelligence* (2018).

[11] Emilie Kaufmann and Wouter M Koolen. 2017. Monte-carlo tree search by best arm identification. In *Advances in Neural Information Processing Systems*. 4897–4906.

[12] Michail G Lagoudakis and Ronald Parr. 2002. Value function approximation in zero-sum markov games. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 283–292.

[13] Michael L Littman. 1994. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*. Elsevier, 157–163.

[14] Michael L Littman, Nishkam Ravi, Arjun Talwar, and Martin Zinkevich. 2006. An efficient optimal-equilibrium algorithm for two-player game trees. In *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 298–305.

[15] Kien C Nguyen, Tansu Alpcan, and Tamer Basar. 2009. Stochastic games for security in networks with interdependent nodes. In *2009 International Conference on Game Theory for Networks*. IEEE, 697–703.

[16] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V Vazirani. 2007. *Algorithmic game theory*. Cambridge university press.

[17] Julien Perolat, Bruno Scherrer, Bilal Piot, and Olivier Pietquin. 2015. Approximate dynamic programming for two-player zero-sum markov games. In *International Conference on Machine Learning (ICML 2015)*.

[18] Diederik M Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. 2013. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research* 48 (2013), 67–113.

[19] Richard Rouse III. 2010. *Game design: Theory and practice*. Jones & Bartlett Learning.

[20] Lloyd S Shapley. 1953. Stochastic games. *Proceedings of the national academy of sciences* 39, 10 (1953), 1095–1100.

[21] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. 2017. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815* (2017).

[22] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. 2017. Mastering the game of go without human knowledge. *Nature* 550, 7676 (2017), 354.

[23] Csaba Szepesvári and Michael L Littman. 1996. Generalized markov decision processes: Dynamic-programming and reinforcement-learning algorithms. In *Proceedings of International Conference of Machine Learning*, Vol. 96.

[24] Chen-Yu Wei, Yi-Te Hong, and Chi-Jen Lu. 2017. Online reinforcement learning in stochastic games. In *Advances in Neural Information Processing Systems*. 4987–4997.

[25] Tsachy Weissman, Erik Ordentlich, Gadiel Seroussi, Sergio Verdu, and Marcelo J Weinberger. 2003. Inequalities for the L1 deviation of the empirical distribution. *Hewlett-Packard Labs, Tech. Rep* (2003).

[26] Andrea Zanette and Emma Brunskill. 2019. Tighter Problem-Dependent Regret Bounds in Reinforcement Learning without Domain Knowledge using Value Function Bounds. In *International Conference on Machine Learning*. 7304–7312.

[27] Kaiqing Zhang, Zhuoran Yang, Han Liu, Tong Zhang, and Tamer Başar. 2018. Finite-sample analyses for fully decentralized multi-agent reinforcement learning. *arXiv preprint arXiv:1812.02783* (2018).

A PROOF OF LEMMA 5.1

For any $\pi_i' \in \Pi_i$, we have

$$V_{i,1}^{(\pi_i',\pi_{-1})}(s_1) - V_{i,1}^{\pi}(s_1) \le \max_{\pi_i' \in \Pi_i} V_{i,1}^{(\pi_i',\pi_{-i})}(s_1) - V_{i,1}^{\pi}(s_i)$$
$$\le Expl(\pi) \le \epsilon.$$

This bound holds for all players and thus we finish the proof.

B SUPPLEMENTARY RESULTS FOR SECTION 6.1

B.1 Failure events

For episode k, we denote $w_h^k(s)$ to be the probability of reaching state s at depth h following policy π^k. Also, we denote $n_h^k(s,a)$ denote the count of visiting state-action pair (s,a) at depth h.Then we define some notations for the proof:

$$w_{\min} = \frac{\epsilon}{2H^2S}$$
$$\mathcal{E}_h^k = \{x \in \mathcal{S} \times \mathcal{A} : w_h^k(x) \ge w_{\min}\}$$
$$\text{llnp}(n) = \ln(\ln(n)).$$

Notice that ϵ only appears in our analysis. Thus our method finds approximate NE for arbitrary $\epsilon > 0$, like UBEV does.

Then we define some failure episodes that happen with a small probability. For $\delta' \in (0,1)$, we define

$$F_1^k = \left\{\exists x \in \mathcal{S} \times \mathcal{A},\ h \in [H] : n^k(x,h) < \frac{1}{2}\sum_{k'<k} w_h^{k'}(x) - \ln(\frac{SAH}{\delta'})\right\}$$

$$F_2^k = \left\{\exists x \in \mathcal{S} \times \mathcal{A}, h \in [H] :\ ||(\bar{P}^k(x,h) - P(x,h))||_1 > \sqrt{\frac{4}{n^k(x,h)}\left(2\text{llnp}(n^k(x,h)) + \ln\left(\frac{3SAH(2^S-2)}{\delta'}\right)\right)}\right\}$$

$$F_3^k = \left\{\exists x \in \mathcal{S} \times \mathcal{A}, h \in [H], i \in [N] :\ |\bar{r}_i(x,h) - r_i(x,h)| > \sqrt{\frac{1}{n^k(x,h)}\left(2\text{llnp}(n^k(x,h)) + \ln\left(\frac{3SAH}{\delta'}\right)\right)}\right\}$$

$$F_4^k = \left\{\exists x, x' \in \mathcal{S} \times \mathcal{A},\ h \in [H], u < h : n^k(x,h) < \frac{1}{2}n^k(x',u)\sum_{i<k} w_{u,h}^k(x|x') - \ln\left(\frac{S^2A^2H^2}{\delta'}\right)\right\}$$

$$F_5^k = \left\{\exists x \in \mathcal{S} \times \mathcal{A},\ h \in [H], i \in [N] : \left|(\bar{P}_k(x,h) - P(x,h))^\top V_{i,h+1}^{*,k}\right| \ge \sqrt{\frac{(H-h)^2}{n^k(x,h)}\left(2\text{llnp}(n^k(x,h)) + \ln\frac{3SAH}{\delta'}\right)}\right\}$$

$$F_6^k = \Bigg\{\exists x \in \mathcal{S} \times \mathcal{A},\ s' \in \mathcal{S},\ h \in [H] : |\bar{P}^k(s'|x,h) - P(s'|x,h)| \ge \sqrt{\frac{2P(s'|x)}{n^k(x,h)}\left(2\text{llnp}(n^k(x,h)) + \ln\frac{3S^2AH}{\delta'}\right)}$$
$$+ \frac{1}{n^k(x,h)}\left(2\text{llnp}(n^k(x,h)) + \ln\frac{3S^2AH}{\delta'}\right)\Bigg\},$$

where $V_{i,h+1}^{*,k}(s) = \max_{\pi_i \in \Pi_i} V_{i,h+1}^{(\pi_i,\pi_{-i}^k)}(s)$ and $w_{u,h}^k(x|x')$.

Lemma B.1. *(Probabilities for failure events) For $\delta' \in (0,1)$, the bellow inequalities hold*

$$\mathbb{P}(\cup_{k=1}^{\infty}F_1^k) \le \delta',\ \mathbb{P}(\cup_{k=1}^{\infty}F_2^k) \le \delta',\ \mathbb{P}(\cup_{k=1}^{\infty}F_3^k) \le 2N\delta', \mathbb{P}(\cup_{k=1}^{\infty}F_4^k) \le \delta', \mathbb{P}(\cup_{k=1}^{\infty}F_5^k) \le 2N\delta', \mathbb{P}(\cup_{k=1}^{\infty}F_6^k) \le 2\delta'.$$

Proof. This lemma is highly relative to the appendices E.2 of UBEV. Below corollaries are all presented in [4].

Specifically, F_1^k here is exactly the F_k^N in UBEV. Thus we can get $\mathbb{P}(\cup_{k=1}^{\infty}F_1^k) \le \delta'$ by directly apply Corollary E.4.

For F_2^k, we can directly refer to Corollary E.3 and get $\mathbb{P}(\cup_{k=1}^{\infty}F_2^k) \le \delta'$.

Our definition of F_3^k is almost the same as F_k^R in UBEV except that we need to bound the immediate rewards for all players now. Thus we can prove $\mathbb{P}(\cup_{k=1}^{\infty}F_3^k) \le 2N\delta'$ with Corollary E.1 and a union bound over all players.

Then F_4^k corresponds to F_k^{CN} and $\mathbb{P}(\cup_{k=1}^{\infty}F_4^k) \le \delta'$ with Corollary E.4.

F_5^k extends F_k^V to FTSGs. Notice that in MDPs, V_{t+1}^* is a fixed vector, while in FTSGs, this does not holds. Therefore we need carefully consider the relationship of $P(x,h)$ and $V_{i,h+1}^{*,k}$. Since we consider the time-dependent dynamics, we always have $P(x,h)$ and $V_{i,h+1}^{*,k}$ independent for all players. Therefore, we can apply Corollary E.1 in [4] to get that $\mathbb{P}(\cup_{k=1}^{\infty}F_5^k) \le 2N\delta'$.

Finally for F_6^k, it corresponds to F_k^P and has $\mathbb{P}(\cup_{k=1}^{\infty} F_6^k) \leq 2\delta'$ with Corollary E.2. □

Then we define the union of all these failure events as

$$F = \cup_k [F_1^k \cup F_2^k \cup F_3^k \cup F_4^k \cup F_5^k \cup F_6^k].$$

We can get the conclusion that $\mathbb{P}(F) \leq \delta$ with lemma B.1 and letting $\delta' = \delta/(4N+5)$. That is the supplement set of F, denoted as F^c, happens with a probability at least $1-\delta$.

B.2 Property 1 requirement

We choose the bonus function as

$$\phi^{UPAC} = (H+1)\sqrt{\frac{2\ln\ln(\max\{e, n^k(s,a,h)\}) + \ln((24N+30)SA/\delta)}{n^k(s,a,h)}}.$$

On event F^c, we have that

$$\begin{aligned}
&|(r_i^k(s,a,h) - \bar{r}_i^k(s,a,h)) + \left(P^k(s,a,h) - \bar{P}^k(s,a,h)\right)^\top V_{i,h+1}^{*,k}| \\
\leq &|(r_i^k(s,a,h) - \bar{r}_i^k(s,a,h))| + |\left(P^k(s,a,h) - \bar{P}^k(s,a,h)\right)^\top V_{i,h+1}^{*,k}| \\
\leq &b_h^k(s,a),
\end{aligned}$$

where the last inequality holds using the definition of F_3^k and F_5^k.

Therefore, our design of ϕ^{UPAC} satisfies Property 1.

B.3 Nice and Friendly episodes

Then we define the nice episodes:

Definition 4. *Episode k is called **nice episode** if $\forall\, x \in \mathcal{S}\times\mathcal{A}$, at least one of the below two conditions holds:*

$$(1)\ w_h^k(x) < w_{\min},\ \forall\, h \in [H];$$

$$(2)\ \frac{1}{4}\sum_{i<k}\sum_{h'=1}^{H} w_{h'}^i(x) \geq \ln(\frac{SAH}{\delta'}).$$

Next we define friendly episodes:

Definition 5. *Episode k is called **friendly episode** if $\forall\, x, x' \in \mathcal{S}\times\mathcal{A}$, at least one of the below two conditions holds:*

$$(1)\ w_{u,h}^k(x|x') < w_{\min},\ \forall\, h \in [H];$$

$$(2)\ \frac{1}{4}\sum_{i<k}\sum_{h'=1}^{H} w_{u,h}^k(x|x') \geq H\ln(\frac{S^2A^2H^2}{\delta'}).$$

This is exactly the same as the definition in UBEV.

With Lemma E.2 in [4] we can bound the number of episodes that are not nice or friendly on F^c.

Lemma B.2. *(Sample complexity for non-nice episodes) On F^c, the number of episodes which are not nice is no more than*

$$\frac{6H^3S^2A}{\epsilon}\ln\left(\frac{HSA}{\delta'}\right).$$

Lemma B.3. *(Sample complexity for non-friendly episodes) On F^c, the number of episodes which are not friendly is no more than*

$$\frac{48H^4S^3A^2}{\epsilon}\ln\left(\frac{H^2S^2A^2}{\delta'}\right).$$

Then we can concentrate on nice and friendly episodes.

Lemma B.4. *(Property of nice episodes) Let $r \geq 1$. Let $D \geq 1$ be a poly-logarithmic function of relevant parameters. For $\epsilon' > 0$ there are at most*

$$\frac{8H^rSA}{\epsilon'^r}\text{polylog}(S, A, H, \delta'^{-1}, \epsilon'^{-1})$$

nice episodes such that

$$\sum_{h=1}^{H}\sum_{s\in\mathcal{E}_h^k} w_h^k(x)\left(\frac{\text{llnp}(n^k(x,h)) + D}{n^k(x,h)}\right)^{1/r} > \epsilon'.$$

This lemma can be directly proved with Lemma E.3 of [4].

Lemma B.5. *(Property of friendly episodes) On good event F^c, there are at most*

$$(\frac{9216}{\epsilon} + 417S)\text{polylog}(S, A, H, \delta'^{-1}, \epsilon'^{-1})$$

friendly episodes such that

$$V_{i,1}^{*,k} - V_{i,1}^{\pi^k} \geq \epsilon,$$

if $\delta' \leq \frac{3AS^2H}{e^2}$

This lemma can be derived from Lemma E.8 of [4].

B.4 The proof of lemma 6.2

Now we give the complete proof for lemma 6.2. For arbitrary player $i \in [N]$, we can decompose $\Delta_i^k = V_{i,1}^k(s_1) - V_{i,1}^{\pi^k}(s_1)$ to get

$$\begin{aligned}
&V_{i,1}^k(s_1) - V_{i,1}^{\pi^k}(s_1) \\
=&\bar{r}_i^k(s_1, \pi(s_1,1),1) - r_i(s_1, \pi(s_1,1),1) + \bar{P}^k(s_1, \pi(s_1,1),1)^\top V_{i,2}^k - P(s_1, \pi(s_1,1),1)^\top V_{i,2}^{\pi^k} + b_1^k(s_1, \pi(s_1,1)) \\
=&(\bar{r}_i^k(s_1, \pi(s_1,1),1) - r_i(s_1, \pi(s_1,1)),1) + \left(\bar{P}^k(s_1, \pi(s_1,1),1) - P(s_1, \pi(s_1,1),1)\right)^\top V_{i,2}^k \\
&+ P(s_1, \pi(s_1,1),1)^\top \left(V_{i,2}^k - V_{i,2}^{\pi^k}\right) + b_1^k(s_1, \pi(s_1,1)) \\
=&\sum_{h=1}^H \sum_{x \in \mathcal{S}\times\mathcal{A}} w_h^k(x) \left((\bar{r}_i^k(x,h) - r_i(x,h)) + \left(\bar{P}^k(x,h) - P(x,h)\right)^\top V_{i,h+1}^k + b_h^k(x)\right) \\
=&\sum_{h=1}^H \sum_{x \in \mathcal{E}_h^k} w_h^k(x) \left((\bar{r}_i^k(x,h) - r_i(x,h)) + \left(\bar{P}^k(x,h) - P(x,h)\right)^\top V_{i,h+1}^k + b_h^k(x)\right) \\
&+ \sum_{h=1}^H \sum_{x \notin \mathcal{E}_h^k} w_h^k(x) \left((\bar{r}_i^k(x,h) - r_i(x,h)) + \left(\bar{P}^k(x,h) - P(x)\right)^\top V_{i,h+1}^k + b_h^k(x,h)\right) \\
\leq&\sum_{h=1}^H \sum_{x \in \mathcal{E}_h^k} w_h^k(x) \left((\bar{r}_i^k(x,h) - r_i(x,h)) + \left(\bar{P}^k(x,h) - P(x,h)\right)^\top V_{i,h+1}^k + b_h^k(x)\right) + \sum_{h=1}^H \sum_{x \notin \mathcal{E}_h^k} H w_{\min}.
\end{aligned}$$

Recall that we consider deterministic policies here. Thus the second term can be bounded by $H^2 S w_{\min} = \frac{1}{2}\epsilon$.
Thus we just need to upper bound the first term. Recall that

$$b_h^k(x) = (H+1)\sqrt{\frac{1}{n^k(x,h)}\left(2\text{llnp}(n^k(x,h)) + \ln\left(\frac{6SAH}{\delta'}\right)\right)}.$$

We also have that on good event F^c

$$\begin{aligned}
\left(\bar{P}^k(x,h) - P(x,h)\right)^\top V_{i,h+1}^k \leq& ||\bar{P}^k(x,h) - P(x,h)||_1 ||V_{i,h+1}^k||_\infty \\
\leq& \sqrt{\frac{4H^2}{n^k(x,h)}\left(2\text{llnp}(n^k(x,h)) + \ln\left(\frac{3SAH(2^S-2)}{\delta'}\right)\right)} \\
\leq& \sqrt{\frac{4H^2S}{n^k(x)}\left(2\text{llnp}(n^k(x)) + \ln\left(\frac{6SAH}{\delta'}\right)\right)} \\
\bar{r}_i^k(x,h) - r_i(x,h) \leq& \sqrt{\frac{1}{n^k(x,h)}\left(2\text{llnp}(n^k(x,h)) + \ln\left(\frac{3SAH}{\delta'}\right)\right)}.
\end{aligned}$$

Combine them and we get

$$\sum_{h=1}^{H}\sum_{x\in\mathcal{E}_h^k} w_h^k(x)\left((\bar{r}_i^k(x,h)-r_i(x,h))+\left(\bar{P}^k(x,h)-P(x,h)\right)^\top V_{i,h+1}^k+b_h^k(x)\right)$$
$$\leq(2H\sqrt{S}+H+2)\sum_{h=1}^{H}\sum_{s\in\mathcal{E}_h^k} w_h^k(x)\sqrt{\frac{1}{n^k(x,h)}\left(2\text{llnp}(n^k(x,h))+\ln\left(\frac{6SAH}{\delta'}\right)\right)}. \tag{6}$$

Then we let $r=2$, $D=\ln(6SA/\delta')$ and $\epsilon'=\epsilon/(4H\sqrt{S}+2)$. By applying lemma B.4, there are at most

$$\frac{32HSA(2H\sqrt{S}+1)^2}{\epsilon^2}\text{polylog}(S,A,H,\delta'^{-1},\epsilon^{-1})$$

nice episodes such that

$$\sum_{h=1}^{H}\sum_{x\in\mathcal{E}_h^k} w_h^k(s)\left((\bar{r}^k(x,h)-r(x,h))+\left(\bar{P}^k(x,h)-P(x,h)\right)^\top V_{h+1}^{k,U}+b_h^{k,U}(x)\right)>\epsilon.$$

Therefore, by choosing $\delta'=\delta/(4N+5)$, we finish our proof.

B.5 Proof of theorem 6.1

Here we give the proof of our main theorem. Our target is give the sample complexity of L^ϵ. Since all players are involved in L^ϵ, we solve this problem by bounding each player separately.

Recall our definition for the the *best response distance* of player i for π as

$$Bsd_i(\pi):=\max_{\pi_i'\in\Pi_i} V_{i,1}^{(\pi_i',\pi_{-i})}(s_1)-V_{i,1}^{(\pi)}(s_1).$$

We rewrite L^ϵ with $Bsd_i(\pi^k)$ where $\pi^k=(\pi_1^k,\pi_2^k,...,\pi_N^k)$ is the policy tuple used during episode k. We now can bound L^ϵ with

$$L^\epsilon\leq\sum_{k\in\mathbb{N}}\mathbb{I}\left[\exists i\in[N],Bsd_i(\pi^k)>\epsilon\right].$$

It is easy to see that in UBVP, players are symmetric, and thus the result for one can be adapted to the others. Now we consider player i, $i\in[N]$. Now we define $\Delta_i^k:=V_{i,1}^k(s_1)-V_{i,1}^{\pi^k}(s_1)$. Now we give the key lemma that connect Δ_i^k and Bsd_i. Our design for UBVP exactly ensure such a good connection. Recall Lemma 5.2, we have that on good events F^c, for any $i\in[N]$, $k\in\mathbb{N}$, $h\in[H]$ and $s\in\mathcal{S}$,

$$\max_{\pi_i\in\Pi_i} V_{i,h}^{(\pi_1,\pi_{-i}^k)}(s)\leq V_{i,h}^k(s).$$

Lemma 5.2 can lead to the result that on F^c, for all $i\in[N]$,

$$Bsd_i(\pi^k)=\max_{\pi_i\in\Pi_i} V_{i,1}^{(\pi_i,\pi_{-i}^k)}(s_1)-V_{i,1}^{\pi^k}(s_1)\leq\Delta_i^k.$$

Naturally, on good events F^c,

$$L^\epsilon\leq\sum_{i\in[N]}\sum_{k\in\mathbb{N}}\mathbb{I}\left[\Delta_i^k>\epsilon\right].$$

Further we notice that applying lemma B.4 to the right hand side of δ_i^k. This process in fact is independent of the choice of players. That is, this holds for all players at the same time. Hence the sample complexity in lemma 6.2 is exactly the sample complexity of L^ϵ. It might be strange that the number of player N only appears in the polylog term. A rethinking can remind us that the number of states S in fact include the complexity of player numbers implicitly, because S is the number of states for all players.

Therefore we finish the proof of first part of theorem 6.1.

For the second part of the theorem, since the analysis is relies on the exact values of player i. We cannot removes the term N. Thus from Lemma B.5 and Lemma B.3, we can get that on good event F^c:

$$L^\epsilon\leq O((NS+\frac{N+H^4S^3A^2}{\epsilon})\text{polylog}(N,S,A,H,\delta^{-1},\epsilon^{-1})).$$

Therefore we finish the proof.

C SUPPLEMENTARY RESULTS FOR SECTION 6.2

Recall that we design two ϕ functions as:

$$\phi_1^{HPR} = 8HL\sqrt{1/n^k(s,a,h)},$$

$$\phi_2^{HPR} = \sqrt{\frac{8LVar_{s'\sim\bar{P}(s,a,h)}V_{i,h+1}^k(s')}{n^k(s,a,h)}} + \frac{14HL}{3n^k(s,a,h)} + HL\sqrt{1/n^k(s,a,h)} + \sqrt{\frac{8\sum_{s'}\bar{P}(s,a,s',h)C(s')}{n^k(s,a,h)}},$$

where $C(s') = \min\{10^4H^3S^2AL^2/n^k(s,a,s',h), H^2\}$ and $L = \ln(5HSATN/\delta)$.

We add a term $HL\sqrt{1/n^k(s,a,h)}$ to bonus 1 and 2 in [6] to get the two bonus functions. We add this extra term because we assume reward functions are random functions. Thus we need extra term to upper bound the gap caused by r_i.