

# Markov Games with Decoupled Dynamics: Price of Anarchy and Sample Complexity

Runyu (Cathy) Zhang, Yuyang Zhang, Rohit Konda, Bryce Ferguson, Jason Marden, Na Li

**Abstract**—This paper studies the finite-time horizon Markov games where the agents’ dynamics are decoupled but the rewards can possibly be coupled across agents. The policy class is restricted to local policies where agents make decisions using their local state. We first introduce the notion of smooth Markov games which extends the smoothness argument for normal form games ([1], [2]) to our setting, and leverage the smoothness property to bound the price of anarchy of the Markov game. For a specific type of Markov game called the Markov potential game, we also develop a distributed learning algorithm, multi-agent soft policy iteration (MA-SPI), which provably converges to a Nash equilibrium. Sample complexity of the algorithm is also provided. Lastly, our results are validated using a dynamic covering game.

## I. INTRODUCTION

Multiagent Markov Decision Processes (MDPs) have found numerous applications, such as autonomous vehicles [3], swarm robotics [4], collaborative manufacturing [5], decentralized energy management [6], and social network analysis [7], among many others. In these problems, a critical question is how individual agents can learn effective strategies collectively in complex multi-agent environments.

Given the success of reinforcement learning (RL) for MDPs, many studies have focused on the empirical and theoretical performance of RL in multi-agent settings (e.g., [8]–[15]). The study of multi-agent RL is inherently more complex than that of single-agent RL due to the (strategic) interactions between agents. Consequently, most provable algorithms in multi-agent RL (e.g., [10]–[13]) only consider convergence to Nash equilibria (NEs) or their extensions (e.g., coarse correlated equilibria (CCE)) instead of global optimality. However, in many cases, NE may have poor performance because it only considers the individual rationality of each agent and hence does not necessarily result in the best overall outcome for the system as a whole. Therefore, in addition to understanding the convergence to a NE, it is also crucial to investigate the quality of the NEs compared to the global optimal solution, where all agents in the system coordinate and cooperate to achieve a common objective.

Runyu (Cathy) Zhang, Yuyang Zhang and Na Li are with the John A. Paulson School of Engineering and Applied Sciences, Harvard University (e-mail: runyuzhang@fas.harvard.edu, yuyangzhang@g.harvard.edu, nali@seas.harvard.edu). Rohit Konda, Bryce Ferguson, Jason Marden are with the Department of Electrical and Computer Engineering, University of California, Santa Barbara (e-mail: rkonda@ucsb.edu, blferguson@ece.ucsb.edu, jrmarden@ece.ucsb.edu).

Due to the difficulty in characterizing which specific NE (or CCE) an algorithm converges to, one way to characterize the quality of solutions is to study the worst-case performance of NEs, such as Price of Anarchy (PoA, [16]), which is defined as the ratio between the performance of the worst NE and the global optimum. The PoA has been well studied in the *static game* settings where there is only a single stage/state. In many applications such as traffic routing [17], resource allocation [18]–[20], and auctions [21], PoA bounds have been established. Moreover, [1] introduces a general approach for generating the PoA bounds for a class of games called smooth games. However, it is unclear how to extend these results to the Markov settings where there are many states and stages.

Several recent works, such as [22] and [23], have extended the smoothness property to the Markov game settings. However, one limitation of their approach is that they define smoothness directly on the value functions, which are long-term accumulative rewards, of the Markov game. Value functions are typically hard to compute, making it difficult to verify the smoothness condition. These studies suggest that analyzing the PoA for Markov games may be fundamentally more challenging than for static games due to the increased complexity of the problem structure.

**Our Contributions:** Given the difficulty of studying PoA for general Markov games, this paper focuses on a specific type of Markov games, where one agent’s state transition only depends on its own state and action but everyone’s stage reward is coupled with each other’s state and action. This setting finds many applications such as multi-robots, traffic systems, and sensor networks.

The first part of this paper is dedicated to addressing the challenges associated with the study of PoA, where we provide both positive and negative results. In the case of the former, we extend the smoothness arguments presented in previous work [1], [2] for normal form static games, to identify a sufficient condition for a lower bound on the PoA. This is accomplished by taking advantage of the unique structure of our problem, which allows for verification of smoothness solely for the stage rewards, as opposed to the value functions. Compared with existing conditions presented in [22], [23], our smoothness condition is significantly easier to verify. In terms of negative results, we provide a counterexample to illustrate that the PoA of the stage rewards alone fails to lower-

bound the PoA of the Markov game, thus emphasizing the importance of our proposed smoothness condition.

Furthermore, we develop a novel distributed learning algorithm - multi-agent soft policy iteration (MA-SPI) - for a specific type of Markov game called the Markov potential game (MPG), which provably converges to a Nash equilibrium. Notably, MA-SPI can be implemented in a fully decentralized manner, where agents only require local information to update their policies. We also proved that it takes  $\tilde{O}\left(\frac{n^4 H^7}{c^6 \epsilon^4} + \frac{n^2 \max_i |S_i|^2 \max_i |A_i|^2 H^4}{c^2 \epsilon^2}\right)$  samples to learn a policy that is  $\epsilon$  close to a NE. Here  $n$  is the number of agents,  $H$  is the horizon length,  $|S_i|, |A_i|$  is the size of  $i$ -th agent's local state and action space and the constant  $c$  represents the level of sufficient exploration of the system states (rigorously defined in Assumption 1).

In addition to theoretical analysis, we also evaluate our results using a dynamic covering game as an example, where a group of collaborative agents work together to cover as many rewards as possible. The game's smoothness and price of anarchy were analyzed for three different reward designs: identical-interest, marginal-contribution, and utility-sharing. Additionally, we apply the MA-SPI algorithm to the dynamic covering game and numerically show that MA-SPI converges to a Nash equilibrium.

## II. PROBLEM SETTINGS AND PRELIMINARIES

### A. Markov games with decoupled dynamics

In this paper, we consider a Markov game model with decoupled dynamics

$$\mathcal{M} := \{\{S_i, \mathcal{A}_i, P_i, r_i, \rho_i\}_{i=1}^n, v, H\}, \quad (1)$$

where there are  $n$  agents, and  $H$  is the horizon of the Markov game.  $s_{i,h} \in S_i$  is the local state of agent  $i$  at horizon  $h$  and  $a_{i,h} \in \mathcal{A}_i$  is the local action. The dynamics of agents are decoupled in the sense that agent  $i$ 's next state  $s_{i,h+1}$  is fully determined by the local action and the current local state  $s_{i,h+1} \sim P_{i,h}(\cdot | s_{i,h}, a_{i,h})$ . We denote the global state and global action as the concatenation of local states and actions, i.e.,  $s_h = \{s_{1,h} \dots, s_{n,h}\}, a_h = \{a_{1,h} \dots, a_{n,h}\}$ , and thus the state/action space is a product space of the local state/action spaces  $\mathcal{S} = S_1 \times \dots \times S_n, \mathcal{A} = \mathcal{A}_1 \times \dots \times \mathcal{A}_n$ . Agent  $i$ 's stage reward at horizon  $h$  is  $r_{i,h} : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ , which can possibly depend on other agents' states and actions. Apart from individual rewards, there's a social welfare function  $v_h : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$  that measures the social welfare of the system at horizon  $h$ .  $\rho_i$  is the initial local state distribution for agent  $i$ . A stochastic policy  $\pi = \{\pi_h : \mathcal{S} \rightarrow \Delta(\mathcal{A})\}_{h=1}^H$  (where  $\Delta(\mathcal{A})$  is the probability simplex over  $\mathcal{A}$ ) specifies a strategy in which agents choose their actions based on the current state in a stochastic fashion. Throughout the paper, we consider the *local policy class*  $\Pi^{\text{local}}$ , where agents need to take actions independently based on their own local state, i.e.,  $a_{i,h} \sim \pi_{i,h}(\cdot | s_{i,h})$ . Within the local policy class, the *joint* policy can be written as the product of

individual policies, i.e.  $\pi_h(a_h | s_h) = \prod_{i=1}^n \pi_{i,h}(a_{i,h} | s_{i,h})$ . For a given policy agent  $i$ 's total reward is denoted as

$$J_i(\pi) := \mathbb{E}_{s_{j,1} \sim \rho_j}^{\pi} \sum_{h=1}^H r_{i,h}(s_h, a_h),$$

and the total social welfare is

$$W(\pi) := \mathbb{E}_{s_{i,1} \sim \rho_i}^{\pi} \sum_{h=1}^H v_h(s_h, a_h). \quad (2)$$

Here the notation  $\mathbb{E}^{\pi}$  denotes the expectation taken over the trajectory by implementing policy  $\pi$ . Agent  $i$ 's objective is to maximize its own total reward  $J_i$ .

**Remark 1** (Justification and Limitation of the setting). *The settings described above can find many applications such as sensor coverage, autonomous vehicles, and multi-robotics. In these applications, individual agents (i.e., sensors, vehicles, and robots) have their own dynamics which depend on their own state and action. In Section VI, we provide details on modeling a dynamic covering game using our framework. Nevertheless, one major limitation in our current setting is that the policy class is constrained to fully localized policy whereas in practice, agents could decide their actions based on more information such as neighboring agents' states. We leave it as our important future work to generalize the policy class.*  $\square$

Throughout the paper, we use the index notation  $-i$  to denote all the agents other than agent  $i$ , for example  $\pi_{-i}$  is used to denote  $(\pi_1, \dots, \pi_{i-1}, \pi_{i+1}, \dots, \pi_n)$ .

**Definition 1** (Nash equilibrium (NE)). *A policy  $\pi^{\text{NE}} = \{\pi_1^{\text{NE}}, \dots, \pi_n^{\text{NE}}\}$  is called a Nash equilibrium (NE) if for any  $i \in [n]$ ,*

$$J_i(\pi^{\text{NE}}) \geq J_i(\pi_i, \pi_{-i}^{\text{NE}})$$

*We also denote the set of Nash equilibria as  $\Pi^{\text{NE}}$ .*

*Given a policy  $\pi$ , we denote the Nash gap for the policy by:*

$$\text{NE-Gap}_i(\pi) := \max_{\pi'_i} J_i(\pi'_i, \pi_{-i}) - J_i(\pi_i, \pi_{-i}).$$

$$\text{NE-Gap}(\pi) := \sum_{i=1}^n \text{NE-Gap}_i(\pi)$$

*A policy  $\pi$  is called an  $\epsilon$ -NE if  $\text{NE-Gap}(\pi) \leq \epsilon$ .*

**Definition 2** (Price of anarchy (PoA)). *For a Markov game  $\mathcal{M}$  defined as Equation (1),*

$$\text{PoA}(\mathcal{M}) := \frac{\min_{\pi^{\text{NE}} \in \Pi^{\text{NE}}} W(\pi^{\text{NE}})}{\max_{\pi \in \Pi^{\text{local}}} W(\pi)}$$

From the definition of PoA, we can see that for any  $\pi^{\text{NE}}$ , the performance can be lower bounded by PoA, that is,

$$W(\pi^{\text{NE}}) \geq \text{PoA}(\mathcal{M}) \cdot \max_{\pi \in \Pi^{\text{local}}} W(\pi).$$

In this paper, we mainly focus on two goals: i) firstly, to provide a lower bound for PoA for a class of Markov

games so that the bound could be used to bound the performance ratio for any  $\pi^{\text{NE}}$ ; ii) secondly, to design a distributed reinforcement learning algorithm that enables agents to reach a NE by only using their local information (local states  $s_i$ , actions  $a_i$  and local rewards  $r_i$ ), even without knowing the Markov model  $\mathcal{M}$ .

### B. Other preliminaries

Throughout this paper, we will need the following notations and definitions.

We define the state distribution of agent  $i$  as  $d_{i,h}^\pi : \mathcal{S}_i \rightarrow [0, 1]$  such that  $d_{i,h}^\pi(s_i)$  denotes the probability of state  $s_i$  being visited at horizon  $h$  by implementing policy  $\pi$ , i.e.  $d_{i,h}^\pi(s_i) = \Pr\{s_{i,h} = s_i | s_{i,1} \sim \rho_i, a_{i,\tau} \sim \pi_i(\cdot | s_\tau)\}$ . Because the transition probabilities are decoupled and local policies are considered, we have that the total state distribution can be written as:  $d_h^\pi(s) = \Pr\{s_h = s\} = \prod_i d_{i,h}^\pi(s_i)$ .

Agent  $i$ 's  $Q$ -functions is defined as

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

Similar to [12], we define the following ‘averaged’  $Q$ -functions apart from the standard definition of  $Q$ -functions, which will play an important role in later algorithm design.

$$\bar{Q}_{i,h}^\pi(s_i, a_i) := \mathbb{E}_{\substack{j \neq i \\ s_j \sim d_{j,h}^{\pi_j}(\cdot)}} \mathbb{E}_{\substack{j \neq i \\ a_j \sim \pi_j(\cdot | s_j)}} Q_{i,h}^\pi(s_i, s_{-i}, a_i, a_{-i}) \quad (3)$$

Note that our definition of the ‘‘averaged’’  $Q$ -function differs from that in [12]. Specifically, we not only take a weighted average over other agents’ actions but also their local states.

### III. SMOOTH MARKOV GAMES AND PRICE OF ANARCHY

Characterizing the exact PoA can be challenging even in the normal form game setting, as it compares the performance ratio of the *worst* NE against the global optimal. The smoothness argument [1] serves as a useful tool for PoA analysis which can give a canonical PoA lower bound. In this section, we first identify a class of Markov games named  $(\lambda, \mu)$ -generalized smooth Markov games, which is an extension of the smoothness argument in [1], [2] for normal form games, and show that  $(\lambda, \mu)$ -generalized smooth Markov games have a uniform lower bounds on the PoA.

**Definition 3** ( $(\lambda, \mu)$ -generalized smoothness). For  $\mu \geq 0, \lambda > 0$ , a Markov game  $\mathcal{M}$  defined as in Equation (1) is called  $(\lambda, \mu)$ -generalized smooth if the following inequality holds for any state and action pairs  $(s_h, a_h), (s_h^*, a_h^*)$

$$\begin{aligned} & \sum_{i=1}^n r_{i,h}(s_h, a_h) - r_{i,h}(s_h^*, a_h^*) \\ & \leq (1 + \mu)v_h(s_h, a_h) - \lambda v_h(s_h^*, a_h^*), \quad \forall h \in [H], \end{aligned}$$

Definition 3 is an extension of the generalized smoothness introduced in [2] to the Markov game setting. The definition implies that at each horizon  $h$ , the normal form game formed induced by stage reward  $\{r_{i,h}\}_{i=1}^n$  and social welfare function  $v_h$  satisfies the generalized smoothness argument in [2]. Notably, this condition is much easier to verify than the smoothness conditions in [22], [23] as it only requires checking the reward and welfare functions at each horizon without any value function computation. The following theorem shows that we can obtain a similar bound on PoA for the extension to the Markov game setting.

**Theorem 1.** Any  $(\lambda, \mu)$ -generalized smooth Markov game  $\mathcal{M}$  satisfies the following PoA bound:

$$\text{PoA}(\mathcal{M}) \geq \frac{\lambda}{(1 + \mu)}$$

*Proof.* We use  $\pi^* = (\pi_1^*, \dots, \pi_n^*)$  to denote the globally optimal policy in the local policy class. From the definition of  $(\lambda, \mu)$ -generalized smoothness,

$$0 \leq \sum_{i=1}^n (J_i(\pi^{\text{NE}}) - J_i(\pi_i^*, \pi_{-i}^{\text{NE}})) \quad (4)$$

$$\begin{aligned} &= \sum_{i=1}^n \mathbb{E}_{\pi^{\text{NE}}} \sum_{h=1}^H r_{i,h}(s_h, a_h) - \\ & \mathbb{E}_{(\pi_i^*, \pi_{-i}^{\text{NE}})} \sum_{h=1}^H r_{i,h}(s_{i,h}^*, s_{-i,h}, a_{i,h}^*, a_{-i,h}) \end{aligned} \quad (5)$$

$$\begin{aligned} &= \mathbb{E}_{s \sim \pi^{\text{NE}}, s^* \sim \pi^*} \sum_{h=1}^H \left( \sum_{i=1}^n (r_i(s_h, a_h) \right. \\ & \left. - r_i(s_{i,h}^*, s_{-i,h}, a_{i,h}^*, a_{-i,h})) \right) \end{aligned} \quad (6)$$

$$\leq \mathbb{E}_{s \sim \pi^{\text{NE}}, s^* \sim \pi^*} \sum_{h=1}^H ((\mu + 1)v(s_h, a_h) - \lambda v(s_h^*, a_h^*)) \quad (7)$$

$$= (\mu + 1)W(\pi^{\text{NE}}) - \lambda W(\pi^*),$$

where the inequality (4) is derived by the definition of NE, (5) to (6) holds because the trajectory  $\{s_{i,h}, a_{i,h}\}_{h=1}^H$  of agent  $i$  is independent of any other agents  $\{s_{-i,h}, a_{-i,h}\}_{h=1}^H$ , (6) to (7) holds because of the smoothness condition. Thus we have

$$\frac{W(\pi^{\text{NE}})}{W(\pi^*)} \geq \frac{\lambda}{\mu + 1}, \implies \text{PoA}(\mathcal{M}) \geq \frac{\lambda}{\mu + 1}. \quad \square$$

**Remark 2.** The proof of Theorem 1 can be easily extend to  $\epsilon$ -NEs and thus a performance lower bound can also be derived for  $\epsilon$ -NEs:

$$\begin{aligned} & -\epsilon \leq (\mu + 1)W(\pi^{\epsilon\text{-NE}}) - \lambda W(\pi^*) \\ \implies & W(\pi^{\epsilon\text{-NE}}) \geq \frac{\lambda}{\mu + 1} W(\pi^*) - \frac{\epsilon}{\mu + 1} \end{aligned}$$

### A. A counter-example for general Markov games

Theorem 1 suggests that as long as the stage rewards of the Markov game form a  $(\lambda, \mu)$ -generalized smooth game for every horizon  $h$ , then the PoA bound analysis can be generalized from the static normal form game to the Markov game. Given this positive result, a natural follow-up question is, whether the statement can be generalized to general Markov games in the sense that stage-wise PoA lower bound can also bound the PoA for the Markov game which is not necessarily a smooth game. Unfortunately, this statement is not necessarily true.

We can construct a simple counter-example with  $n = 2, H = 3$ . In this counter-example, the reward functions  $r_{i,h}$ 's only depend on states (but not actions), and for any  $1 \leq h \leq H$ ,  $\{r_{i,h}(\cdot)\}_{i=1}^n$  forms a normal form game on the state space with  $\text{PoA}(\{r_{i,h}(\cdot)\}_{i=1}^n) \geq 1/2$ , yet the PoA for the Markov game  $\text{PoA}(\mathcal{M}) = 7/16 < 1/2$ . The counter-example is constructed as follows: both agent 1 and 2 follow the (local) state transition dynamics as shown in Figure 1. The reward table for  $h = 2$  and  $h = 3$  is shown as in Table I, II. It can be verified that the PoA is  $1/2$  for Table I (the worst NE is a mixed NE where both agents play  $X$  and  $Y$  with probability  $1/2$ ) and  $1$  for Table II (there's a unique NE which is also the global optimal). From another perspective, the Markov game is equivalent to a normal form game on the action space at  $h = 1$ , whose reward table is shown in Table III. Thus the PoA for the dynamic game is the same as Table III, which is  $\text{PoA}(\mathcal{M}) = 7/16 < 1/2$  (the worst NE is a mixed NE where both agents play  $A$  with probability  $3/4$  and  $Y$  with probability  $1/4$ ).

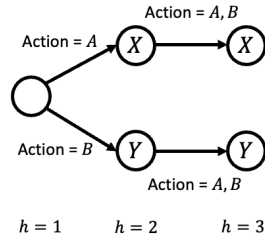


Fig. 1: State Transition

	X	Y
X	2	0
Y	0	2

TABLE I: Reward table for  $h = 2$

	X	Y
X	0	1
Y	1	2

TABLE II: Reward table for  $h = 3$

	A	B
A	2	1
B	1	4

TABLE III: Equivalence to a normal form game

## IV. LEARNING THE NASH EQUILIBRIUM FOR MARKOV POTENTIAL GAMES

The PoA lower bound proved by the previous section suggests that as long as an algorithm finds a NE for a  $(\lambda, \mu)$ -generalized smooth Markov game, it is at most  $\lambda/(\mu + 1)$  suboptimal in terms of the performance ratio. However, finding a NE is known to be intractable for general sum games [24], [25], thus as a starting point, in this paper we mainly focus on an important subclass, namely the Markov potential game, and propose a multi-agent soft policy iteration (MA-SPI) algorithm that learns a NE efficiently using samples.

**Definition 4.** (Markov Potential Game (MPG, [12], [13], [26], [27])) A Markov game  $\mathcal{M}$  defined as Equation (1) is a Markov potential game if there exists stage potential functions  $\{\phi_h : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}\}_{h=1}^H$  such that for all agents' rewards and state-action pairs at all horizon  $1 \leq h \leq H$ ,

$$r_{i,h}(s'_{i,h}, s_{-i,h}, a'_{i,h}, a_{-i,h}) - r_{i,h}(s_{i,h}, s_{-i,h}, a_{i,h}, a_{-i,h}) = \phi_h(s'_{i,h}, s_{-i,h}, a'_{i,h}, a_{-i,h}) - \phi_h(s_{i,h}, s_{-i,h}, a_{i,h}, a_{-i,h}). \quad (8)$$

From the definition of MPG, we can define the total potential function as a function on the policy space

$$\Phi(\pi) := \mathbb{E}_{s_{i,1} \sim \rho_i}^{\pi} \sum_{h=0}^H \phi_h(s_h, a_h), \quad (9)$$

and it is not hard to verify that for any  $(\pi_i, \pi_{-i}), (\pi'_i, \pi_{-i}) \in \Pi^{\text{local}}$  the total potential function satisfies:

$$J_i(\pi_i, \pi_{-i}) - J_i(\pi'_i, \pi_{-i}) = \Phi(\pi_i, \pi_{-i}) - \Phi(\pi'_i, \pi_{-i}). \quad (10)$$

The notion of the Markov potential game is first proposed for continuous dynamical systems in [26], [27]. The idea is then generalized to the Markov game setting in [12], [13], where Equation (10) is directly used as the definition of MPG. However, Equation (10) is hard to check for general Markov games. In contrast, thanks to the decoupled dynamics and local policy class considered in this paper, the condition can be further simplified as Definition 4, whose condition is defined over stage rewards, much easier to verify.

For the learning algorithm, we also make the following assumption on the MPG.

**Assumption 1** (Sufficient Exploration). For any policy  $\pi$ , there exists a uniform constant  $c > 0$  such that  $d_{i,h}(s_i) > c, \forall i, h, s_i$ .

Assumption 1 requires that any state at any horizon  $h$  has a positive probability of being visited. Similar assumptions are standard for proving convergence for sample-based RL algorithms (e.g., [28], [29]). We would like to note that Assumption 1 is admittedly stronger than assumptions in [28] (they only require a certain discounted state visitation distribution to be positive, yet we require the state visitation distribution at every horizon  $h$  to be positive). We leave it to future work to extend this assumption.

### A. Algorithm design: multi-agent soft policy iteration

Our algorithm can be summarized as the following soft policy iteration update:

$$\pi_{i,h}^{(t+1)}(a_i | s_i) = (1 - \eta_t) \pi_{i,h}^{(t)}(a_i | s_i) + \eta_t \mathbf{1}\{a_i = \underset{a'_i}{\operatorname{argmax}} \widehat{Q}_{i,h}^{(t)}(s_i, a'_i)\}. \quad (11)$$

Here  $\widehat{Q}_{i,h}^{(t)}$  is an estimation of the averaged-Q function  $\overline{Q}_{i,h}^{(t)}$  defined in Equation (3). Note that when the stepsize

$\eta_t = 1$ , the algorithm is equivalent to each agent executing the policy iteration algorithm, i.e., choosing greedy actions that maximize the current estimation of its own averaged-Q functions. However, for multi-agent systems, the greedy update with  $\eta_t = 1$  might cause miscoordination and thus fail to converge, hence we introduce the soft policy iteration by setting the next iteration's policy as a convex combination of the current policy and the greedy policy.

Given Equation (11), the key part of the algorithm lies in how to estimate the averaged-Q functions. To start with, we present the following property of the averaged Q-functions (the proof is deferred to the online version of this paper [30]):

**Lemma 1.** *The averaged Q-functions  $\bar{Q}_{i,h}^\pi$  in Equation (3) satisfies the following Bellman equation:*

$$\bar{Q}_{i,h}^\pi(s_i, a_i) = \bar{r}_{i,h}^\pi(s_i, a_i) + \sum_{s'_i, a'_i} P_{i,h}(s'_i | s_i, a_i) \pi_{i,h+1}(a'_i | s'_i) \bar{Q}_{i,h+1}^\pi(s'_i, a'_i), \quad (12)$$

where

$$\bar{r}_{i,h}^\pi(s_i, a_i) := \mathbb{E}_{\substack{j \neq i \\ s_j \sim d_{j,h}^\pi(\cdot)}} \mathbb{E}_{\substack{j \neq i \\ a_j \sim \pi_j(\cdot | s_j)}} r_{i,h}(s_i, s_{-i}, a_i, a_{-i})$$

The key idea of our sample-based algorithm is to estimate  $\bar{Q}_{i,h}^\pi(s_i, a_i)$  and then perform soft policy iteration (Equation (11)). Given Equation (12), we start by estimating  $\bar{r}_{i,h}^\pi(s_i, a_i)$  and  $P_{i,h}(s'_i | s_i, a_i)$  and then calculate the estimated averaged-Q functions using Equation (12).

In our algorithm, there are two different types of data collection processes, namely the pre-data collection and the on-policy data collection. In the pre-data collection step, the dataset  $D_K$  is collected by setting each agent's policy the uniform random policy. The on-policy data collection step is collected by running a specific policy, for each policy  $\pi^{(t)}$ , the dataset  $D_J^{(t)}$  is collected by implementing  $\pi^{(t)}$ .

The estimated state transition probability  $\hat{P}_{i,h}(s'_i | s_i, a_i)$  uses the dataset  $D_K$  from the pre-data collection which contains  $T_K$  samples  $D_K = \{\{s_{i,h}^{(k)}, a_{i,h}^{(k)}\}_{h=1}^H\}_{k=1}^{T_K} \in D_K$ , i.e.,

$$\hat{P}_{i,h}(s'_i | s_i, a_i) := \begin{cases} \frac{\sum_{k=1}^{T_K} \mathbf{1}\{s_{i,h+1}^{(k)} = s'_i, s_{i,h}^{(k)} = s_i, a_{i,h}^{(k)} = a_i\}}{\sum_{k=1}^{T_K} \mathbf{1}\{s_{i,h}^{(k)} = s_i, a_{i,h}^{(k)} = a_i\}}, & \text{if } \sum_{k=1}^{T_K} \mathbf{1}\{s_{i,h}^{(k)} = s_i, a_{i,h}^{(k)} = a_i\} \geq 1; \\ \mathbf{1}\{s'_i = s_i\}, & \\ \text{if } \sum_{k=1}^{T_K} \mathbf{1}\{s_{i,h}^{(k)} = s_i, a_{i,h}^{(k)} = a_i\} = 0. & \end{cases} \quad (13)$$

The averaged reward  $\bar{r}_{i,h}^\pi(s_i, a_i)$  for each given policy  $\pi$  is estimated using the on-policy dataset  $D_J$  which contains  $T_J$  samples  $D_J = \{\{s_{i,h}^{(k)}, a_{i,h}^{(k)}\}_{h=1}^H\}_{k=1}^{T_J}$ , i.e.

$$\hat{r}_{i,h}^\pi(s_i, a_i) := \begin{cases} \frac{\sum_{k=1}^{T_J} r_{i,h}(s_i, a_i, s_{-i,h}^{(k)}, a_{-i,h}^{(k)}) \mathbf{1}\{s_{i,h}^{(k)} = s_i\}}{\sum_{k=1}^{T_J} \mathbf{1}\{s_{i,h}^{(k)} = s_i\}}, & \text{if } \sum_{k=1}^{T_J} \mathbf{1}\{s_{i,h}^{(k)} = s_i\} \geq 1; \\ 0, & \text{if } \sum_{k=1}^{T_J} \mathbf{1}\{s_{i,h}^{(k)} = s_i\} = 0. \end{cases} \quad (14)$$

Then we can replace  $P_{i,h}$  and  $\bar{r}_{i,h}^\pi$  in Equation (12) with  $\hat{P}_{i,h}$  and  $\hat{r}_{i,h}^\pi$  and get the equation for the estimated averaged-Q functions which can be solved by backward dynamic programming.

$$\begin{aligned} \hat{Q}_{i,h}^\pi(s_i, a_i) &= \hat{r}_{i,h}^\pi(s_i, a_i) + \\ &\sum_{s'_i, a'_i} \hat{P}_{i,h}(s'_i | s_i, a_i) \pi_{i,h+1}(a'_i | s'_i) \hat{Q}_{i,h+1}^\pi(s'_i, a'_i), \quad (15) \\ \hat{Q}_{i,H+1}^\pi(s_i, a_i) &= 0. \end{aligned}$$

Lastly, the policy update is given by Equation (11). A full description of the multi-agent soft policy iteration (MA-SPI) algorithm is in Algorithm 1

---

#### Algorithm 1 Multi-Agent Soft Policy Iteration (MA-SPI)

---

- 1: **Pre-Data Collection:** Get dataset  $D_K := \{\{s_{i,h}^{(k)}, a_{i,h}^{(k)}\}_{h=1}^H\}_{k=1}^{T_K}$  by sampling  $s_{i,1}^{(k)} \sim \rho_i, a_{i,h}^{(k)} \sim \text{Unif}(\mathcal{A}_i), 1 \leq h \leq H$ .
  - 2: **Estimation of  $P_{i,h}(s'_i | s_i, a_i)$ 's:** Estimate the state transition probabilities  $P_{i,h}(s'_i | s_i, a_i)$ 's using Equation (13) and dataset  $D_K$ .
  - 3: **for**  $t = 1, 2, \dots, T_G + 1$  **do**
  - 4:   **for Agent**  $i = 1, 2, \dots, n$  **do**
  - 5:     **On-policy Data Collection:** Get dataset  $D_J^{(t)} := \{\{s_{i,h}^{(k)}, a_{i,h}^{(k)}\}_{h=1}^H\}_{k=1}^{T_J}$  by sampling  $s_{i,1}^{(k)} \sim \rho_i, a_{i,h}^{(k)} \sim \pi_{i,h}^{(t)}(\cdot | s_{i,h}^{(k)}), 1 \leq h \leq H$ .
  - 6:     **Estimation of  $\bar{r}_{i,h}^\pi(s_i, a_i)$ 's:** Calculate  $\hat{r}_{i,h}^\pi(s_i, a_i)$ 's by Equation (14).
  - 7:     **Estimation of  $\bar{Q}_{i,h}^\pi(s_i, a_i)$ 's:** Calculate  $\hat{Q}_{i,h}^\pi(s_i, a_i)$ 's by Equation (15)
  - 8:     **Policy Update:** Update policy  $\pi_i^{(t+1)}$  by Equation (11).
  - 9:   **end for**
  - 10: **end for**
- 

#### B. Sample Complexity

**Theorem 2.** *Under Assumption 1, for any  $\epsilon \leq 1$  and any  $\delta < 1$ , by running Algorithm 1 with*

$$\begin{aligned} T_G &\geq \frac{64n^2 H^3 \left( (\Phi_{\max} - \Phi_{\min} + 1)/2 + \ln(8nH^{\frac{3}{2}} c^{-1} \epsilon^{-1}) \right)^2}{c^2 \epsilon^2} \\ T_J &\geq \frac{2048n^2 H^4 \ln(8\delta^{-1} nHT_G \sum_i |S_i| |A_i|)}{c^4 \epsilon^2} \\ T_K &\geq \frac{2048n^2 \max_i |S_i|^2 \max_i |A_i|^2 H^6 \ln(8\delta^{-1} nHT_G \sum_i |S_i|^2 |A_i|)}{c^4 \epsilon^2}, \end{aligned}$$

*it can be guaranteed that the output policies  $\pi^{(t)}$ 's from Algorithm 1 satisfy:*

$$\sum_{t=1}^{T_G} \frac{1}{\sqrt{t}} \text{NE-Gap}(\pi^{(t)}) \leq \frac{\epsilon \sqrt{T_G}}{2} + \frac{\epsilon \sum_{t=1}^{T_G} \frac{1}{\sqrt{t}}}{2}, \quad (16)$$

which implies that:

$$\min_{1 \leq t \leq T_G} \text{NE-Gap}(\pi^{(t)}) \leq \epsilon. \quad (17)$$

The sample complexity of Algorithm 1 is given by  $T_G T_J + T_K$ . From Theorem 2, by setting

$$T_G \sim \tilde{O}\left(\frac{n^2 H^3}{c^2 \epsilon^2}\right), T_J \sim \tilde{O}\left(\frac{n^2 H^4}{c^4 \epsilon^2}\right),$$

$$T_K \sim \tilde{O}\left(\frac{n^2 \max_i |S_i|^2 \max_i |A_i|^2 H^4}{c^2 \epsilon^2}\right),$$

where  $\tilde{O}$  hides the log factors, running Algorithm 1 can find an approximate NE with NE gaps smaller than  $\epsilon$ . Thus the complexity for finding an  $\epsilon$ -NE is  $\tilde{O}\left(\frac{n^4 H^7}{c^6 \epsilon^4} + \frac{n^2 \max_i |S_i|^2 \max_i |A_i|^2 H^4}{c^2 \epsilon^2}\right)$ .

## V. PROOF SKETCHES FOR THEOREM 2

This section provides a brief proof sketch of our sample complexity result. Due to space limitations, we defer the detailed proofs into the online version of this paper [30]. The proof of the theorem can be decomposed into the following three major steps.

**Step 1: iteration complexity of MA-SPI.** The proof of Theorem 2 is based on the following main lemma on the iteration complexity of MA-SPI.

**Lemma 2.** *Suppose*

$$\max_{s_i, a_i} \left| (\widehat{Q}_{i,h}^{\pi^{(t)}} - \overline{Q}_{i,h}^{\pi^{(t)}})(s_i, a_i) \right| \leq \epsilon_Q,$$

$$\forall 1 \leq i \leq n, 1 \leq h \leq H, 1 \leq t \leq T_G$$

then running Equation (11) with  $\eta_t = \frac{1}{\sqrt{4n^2 H^3 t}}$  will guarantee that

$$\sum_{t=1}^{T_G} \frac{1}{\sqrt{t}} \text{NE-Gap}(\pi^{(t)}) \leq$$

$$\frac{\sqrt{4n^2 H^3}}{c} (\Phi_{\max} - \Phi_{\min} + 1 + \ln(T_G)) + \frac{2nH\epsilon_Q}{c} \sum_{t=1}^{T_G} \frac{1}{\sqrt{t}},$$

which implies that

$$\min_{1 \leq t \leq T_G} \text{NE-Gap}(\pi^{(t)}) \leq$$

$$\underbrace{\frac{\sqrt{4n^2 H^3} (\Phi_{\max} - \Phi_{\min} + 1 + \ln(T_G))}{2c\sqrt{T_G}}}_{\text{Part I}} + \underbrace{\frac{2nH\epsilon_Q}{c}}_{\text{Part II}}.$$

Lemma 2 suggests that the minimum NE-gap depends on two terms. The first term (Part I) diminishes to zero at rate  $\tilde{O}\left(\frac{1}{\sqrt{T_G}}\right)$ , whereas the second term (Part II) is a bias term that is caused by the estimation error  $\epsilon_Q$  of the averaged Q-functions. Thus, in order to reach an  $\epsilon$ -NE, in the proof we set both Part I and II to be smaller than  $\frac{\epsilon}{2}$ , and this gives a lower bound for  $T_G$  and an upper bound for  $\epsilon_Q$ . Lemma 2 is proved by a sufficient ascent lemma whose detail is in the supplementary material [30].

**Step 2: sensitivity of  $\epsilon_Q$  to the estimation errors.** The rest of the proof focuses on the analysis of  $\epsilon_Q$ . In this step, we first bound  $\epsilon_Q$  by the estimation errors of  $\widehat{r}_{i,h}^{\pi}(s_i, a_i)$  and  $P_{i,h}(s'_i | s_i, a_i)$ , which is stated as follows:

**Lemma 3.** *Suppose*

$$\left| \widehat{r}_{i,h}^{\pi}(s_i, a_i) - \overline{r}_{i,h}^{\pi}(s_i, a_i) \right| \leq \epsilon_r, \forall s_i \in \mathcal{S}_i, a_i \in \mathcal{A}_i, h,$$

$$\left| \widehat{P}_{i,h}(s'_i | s_i, a_i) - P_{i,h}(s'_i | s_i, a_i) \right| \leq \epsilon_P, \forall s'_i, s_i \in \mathcal{S}_i, a_i \in \mathcal{A}_i, h,$$

then  $\widehat{Q}_{i,h}^{\pi}$  calculated from Equation (15) satisfies:

$$\max_{s_i, a_i} \left| (\widehat{Q}_{i,h}^{\pi} - \overline{Q}_{i,h}^{\pi})(s_i, a_i) \right| \leq \epsilon_r (H+1-h) + \epsilon_P H (H+1-h) |S_i|.$$

Lemma 3 is proved by induction on the Bellman equation (Equation 12).

**Step 3: bound the estimation errors of  $\widehat{r}_{i,h}^{\pi}(s_i, a_i)$  and  $P_{i,h}(s'_i | s_i, a_i)$**  The last step is bounding the estimation errors of the averaged-reward  $\overline{r}_{i,h}^{\pi}(s_i, a_i)$  and transition probability  $P_{i,h}(s'_i | s_i, a_i)$ .

**Lemma 4.** *Under Assumption 1, fix  $s'_i, s_i, a_i, h$ , for  $\epsilon \leq 1$ ,*

$$\Pr\left(\left|\widehat{r}_{i,h}^{\pi}(s_i, a_i) - \overline{r}_{i,h}^{\pi}(s_i, a_i)\right| \geq \epsilon\right) \leq 4 \exp\left(-\frac{\epsilon^2 c^2 T_J}{32}\right).$$

$$\Pr\left(\left|\widehat{P}_{i,h}(s'_i | s_i, a_i) - P_{i,h}(s'_i | s_i, a_i)\right| \geq \epsilon\right) \leq 4 \exp\left(-\frac{\epsilon^2 c^2 T_K}{32 |A_i|^2}\right).$$

Finally, by combining the above three steps and setting  $\epsilon_Q, \epsilon_r, \epsilon_P$  to appropriate values, we can calculate lower bounds for  $T_G, T_J, T_K$  in order to get an  $\epsilon$ -NE.

## VI. APPLICATION TO THE DYNAMIC COVERING GAME

In this section, we use a simple dynamic covering game to illustrate our results. In a dynamic covering game, at each time, the agents' local states form a static covering game and the transition of the local states is governed by some local transition. While the PoA bounds for covering games in the static setting are well-established (e.g.

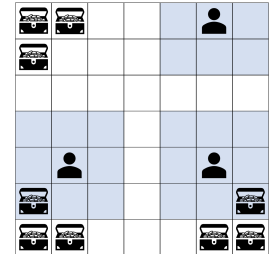


Fig. 2: Multi-agent Dynamic Covering Game.

[1], [19]), not much is known for the dynamic setting. In this section, we look into three most commonly used reward functions (namely the identical-interest, marginal-contribution and utility-sharing), analyze their PoA bounds, and simulate the proposed soft policy iteration algorithm.

Specifically, we consider a group of  $n$  collaborative agents in a covering game on a 2D grid of size  $N$  ( $[N] \times [N]$ ). For a grid  $(j, k)$ , there is a reward of value  $w_{jk}$ . The local state of each agent is its grid position,  $s_{i,h} \in [N] \times [N]$ . The local action  $a_{i,h}$  represents the actions that an agent can take (such as moving up, down, left, right) on the grid. Each agent can have its own local station transition  $P(s_{i,h+1} | s_{i,h}, a_{i,h})$ , either deterministic or stochastic. Given a position  $s_{i,h}$  for agent  $i$ , the local area it can cover is denoted by  $\hat{s}_{i,h} \subset [N] \times [N]$  which includes the grid  $s_{i,h}$  and neighboring grids, e.g., the blue

area in Figure 2.<sup>1</sup> The social welfare function  $v$  at time step  $h$  is how much treasure the agents cover at time step  $h$ :

$$v(s_h) := \sum_{(j,k) \in \cup_{i=1}^n \hat{s}_{i,h}} w_{jk}$$

For different choices of stage reward function  $r_{i,h}$ , agents might converge to different NEs. Here we mainly look into the performance of the following three different types of reward functions:

- 1) Identical-interest (II):  $r_{i,h}(s_h) = v(s_h)$ . The most natural idea might be setting all agents' stage rewards to be the same as the social welfare  $v$ , since this aligns with the ultimate objective.
- 2) Marginal-contribution (MC).  $r_{i,h}(s_h) = v(s_h) - v(\emptyset, s_{-i,h})$ . This type of reward design is also known as 'marginal contribution' [31] in static covering games/submodular games. The reward of agent  $i$  is the change in the total reward if agent  $i$  is removed. It is easy to see that this design is equivalent to setting  $r_i(s_h) = \sum_{(j,k) \in \hat{s}_{i,h}} w_{jk} \mathbf{1}\{\text{Only } i \text{ covers } (j,k)\}$ .
- 3) Utility-sharing (US) [19].  $r_i(s_h) = \sum_{(j,k) \in \hat{s}_{i,h}} w_{jk} \cdot f(\#\text{agents that cover } (j,k))$ , where  $f$  is a monotonically decreasing function. Here the reward for covering  $(j,k)$  is  $w_{jk} f(\#\text{agents that cover } (j,k))$ . The agents will get a smaller reward for covering a grid that is already covered by many other agents. This type of reward assignment is also closely related to Shapley value and its variants. More discussions are available in [32]–[34].

#### A. Smoothness and PoA Bounds

In this section, we establish the smoothness and PoA bounds for all the above discussed reward functions.

**Lemma 5.** 1) *The sets of NEs for identical-interest and marginal-contribution are the same.*

- 2) *For marginal-contribution, the Markov game is  $(1,1)$ -smooth, and thus the PoA for both marginal-contribution and identical-interest is bounded by  $\text{PoA}(\mathcal{M}) \geq \frac{1}{2}$*
- 3) *For utility-sharing, the Markov game is  $(1, \mu_f)$ -smooth, where  $\mu_f = \sup_n n f(n) - f(n+1)$ . More specifically, for*

$$f(n) = \frac{(n-1)! \sum_{i=n}^{+\infty} \frac{1}{i!}}{e-1}, \quad (18)$$

$\mu_f = \frac{1}{e-1}$  and thus the price of anarchy is bounded by  $\text{PoA}(\mathcal{M}) \geq 1 - \frac{1}{e}$ .

*Proof.* The first statement is trivial from the fact that II and MC share the same stage potential function, which can be chosen as the social welfare i.e.,  $\phi = v$ . The smoothness condition for the second statement and third statement follows naturally from the smoothness analysis for the static covering game (see, e.g., [1] for the second

<sup>1</sup>Without causing confusion, we use the notation  $s_{i,h}$  to denote the local state of agent  $i$  and  $\hat{s}_{i,h}$  to denote the local area it can cover from the position.

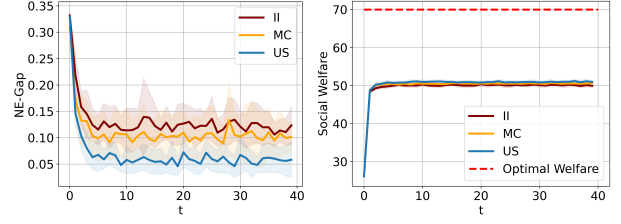


Fig. 3: **Left:** Social NE-Gap of Algorithm 1; **Right:** Social Welfare of Algorithm 1;

statement and [19] for the third statement). And the PoA bounds follow from Theorem 1.  $\square$

#### B. Numerical Simulations

In this section, we test the MA-SPI algorithm (Algorithm 1) in a dynamic covering game on a 2D grid of size 7. The state space of any agent  $i$  consists of the 49 grids. Treasure  $w = 1$  is set in grids  $(0,0), (0,1), (1,0), (0,5), (0,6), (1,6), (5,0), (6,0), (6,1)$ , i.e. in three corners of the grid, as is shown in Figure 2.

There are 3 agents in the game, whose initial positions are randomly selected uniformly among the 49 grids. Each agent has 4 actions, i.e. up, right, down and left. When any agent takes any action, the exact action is executed with probability  $2/3$ , a random action of the four is executed with probability  $1/3$ . When an action is executed, the agent transits to the corresponding grid or stays still if the corresponding grid is outside of the 2D plane.

Every agent interacts with the environment for  $H = 10$  time steps. At every time step, every agent covers an area with size 3 centered at its state, and receives reward according to the reward function and the treasures in its area of coverage.

We first verify that a covering game forms a MPG with any of the three types of reward functions. For II and MC reward designs, it is obvious that the welfare function  $v$  can serve as the stage potential function  $\phi$  in Equation (8). For utility-sharing, the stage potential function can be chosen as [19]:

$$\phi(s_h, a_h) = \sum_{(j,k) \in \hat{s}_h} w_{j,k} \sum_{l=1}^{\#\text{agents that cover } (j,k)} f(l),$$

which satisfies the condition in Equation (8). For any reward design, function  $\phi$  gives the potential function  $\Phi$  following Equation (9), and therefore proves that the game is an MPG.

Moreover, it can be verified by induction that this game satisfies Assumption 1 with  $c = \frac{1}{49} \times (\frac{1}{6})^{10}$ . We simulate Algorithm 1 in the environment described above. In the actual training process, the algorithm parameters are set to  $T_G = 40, T_J = 800, T_K = 50000$ , which is sufficient to show convergence for our algorithm. For the US reward design, the utility function  $f$  is set to Equation (18).

Figure 3 shows the NE-Gap and the social welfare in the training process for all three types of reward functions respectively. For all the reward designs, the NE-gap converges to almost zero, and the social welfare achieves around 50, with the PoA being approximately  $\text{PoA} \approx 50/70 > 1 - \frac{1}{e}$ , which aligns with Lemma 5.



## VII. CONCLUSIONS AND FUTURE WORKS

This paper studies the price of anarchy (PoA) and efficient learning of a specific type of Markov games with decoupled dynamics. Firstly, the generalized smoothness condition is proposed to obtain a unified lower bound of PoA. For a specific type of Markov games known as the Markov potential games, we propose a distributed learning algorithm, multi-agent soft policy iteration (MA-SPI), which provably converges to a Nash equilibrium. Our results are also validated using a dynamic covering game example. As future works, it would be valuable to investigate how to achieve efficient learning and PoA analysis for broader equilibrium notions beyond Nash equilibrium, such as coarse correlated equilibria. In addition, we plan to explore algorithm design for more general policy classes where agents have access not only to their local states but also to their neighbors' information.

### ACKNOWLEDGEMENT

Runyu Zhang and Yuyang Zhang are supported by NSF AI institute: 2112085 and NSF CNS: 2003111. Rohit Konda and Bryce Ferguson are supported by ONR: N00014-20-1-2359 and AFOSR: FA95550-20-1-0054 and FA9550-21-1-0203.

### REFERENCES

- [1] T. Roughgarden, "Intrinsic robustness of the price of anarchy," *Journal of the ACM (JACM)*, vol. 62, no. 5, pp. 1–42, 2015.
- [2] R. Chandan, D. Paccagnan, and J. R. Marden, "When smoothness is not enough: Toward exact quantification and optimization of the price-of-anarchy," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 4041–4046, 2019.
- [3] D. Isele, R. Rahimi, A. Cosgun, K. Subramanian, and K. Fujimura, "Navigating occluded intersections with autonomous vehicles using deep reinforcement learning," in *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 2034–2039, IEEE, 2018.
- [4] M. Hüthenrauch, S. Adrian, G. Neumann, et al., "Deep reinforcement learning for swarm systems," *Journal of Machine Learning Research*, vol. 20, no. 54, pp. 1–31, 2019.
- [5] K. Xia, C. Sacco, M. Kirkpatrick, C. Saidy, L. Nguyen, A. Kircaliali, and R. Harik, "A digital twin to train deep reinforcement learning agent for smart manufacturing plants: Environment, interfaces and intelligence," *Journal of Manufacturing Systems*, vol. 58, pp. 210–230, 2021.
- [6] P. Kofinas, A. Dounis, and G. Vouros, "Fuzzy q-learning for multi-agent decentralized energy management in microgrids," *Applied energy*, vol. 219, pp. 53–67, 2018.
- [7] Y. Zhang, B. Song, and P. Zhang, "Social behavior study under pervasive social networking based on decentralized deep reinforcement learning," *Journal of Network and Computer Applications*, vol. 86, pp. 72–81, 2017.
- [8] R. Pina, V. De Silva, J. Hook, and A. Kondo, "Residual q-networks for value function factorizing in multi-agent reinforcement learning," 2022.
- [9] J. Wang, J. Wang, Y. Zhang, Y. Gu, and T.-K. Kim, "Shaq: Incorporating shapley value theory into multi-agent q-learning," *arXiv preprint arXiv:2105.15013*, 2021.
- [10] Z. Song, S. Mei, and Y. Bai, "When can we learn general-sum markov games with a large number of players sample-efficiently?," *CoRR*, vol. abs/2110.04184, 2021.
- [11] C. Jin, Q. Liu, Y. Wang, and T. Yu, "V-learning - A simple, efficient, decentralized algorithm for multiagent RL," *CoRR*, vol. abs/2110.14555, 2021.
- [12] R. Zhang, Z. Ren, and N. Li, "Gradient play in multi-agent markov stochastic games: Stationary points and convergence," *CoRR*, vol. abs/2106.00198, 2021.
- [13] S. Leonardos, W. Overman, I. Panageas, and G. Piliouras, "Global convergence of multi-agent policy gradient in markov potential games," *CoRR*, vol. abs/2106.01969, 2021.
- [14] S. H. Li, D. Calderone, and B. Açıkmeşe, "Congestion-aware path coordination game with markov decision process dynamics," *IEEE Control Systems Letters*, vol. 7, pp. 431–436, 2022.
- [15] D. Calderone and S. Shankar, "Infinite-horizon average-cost markov decision process routing games," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–6, IEEE, 2017.
- [16] E. Koutsoupias and C. Papadimitriou, "Worst-case equilibria," in *Stacs*, vol. 99, pp. 404–413, Springer, 1999.
- [17] H. Youn, M. T. Gastner, and H. Jeong, "Price of anarchy in transportation networks: efficiency and optimality control," *Physical review letters*, vol. 101, no. 12, p. 128701, 2008.
- [18] D. Paccagnan, R. Chandan, and J. R. Marden, "Utility design for distributed resource allocation—part i: Characterizing and optimizing the exact price of anarchy," *IEEE Transactions on Automatic Control*, vol. 65, no. 11, pp. 4616–4631, 2019.
- [19] M. Gairing, "Covering games: Approximation through non-cooperation," in *International Workshop on Internet and Network Economics*, pp. 184–195, Springer, 2009.
- [20] V. Gkatzelis, K. Kollias, and T. Roughgarden, "Optimal cost-sharing in general resource selection games," *Operations Research*, vol. 64, no. 6, pp. 1230–1238, 2016.
- [21] J. Hartline, D. Hoy, and S. Taggart, "Price of anarchy for auction revenue," in *Proceedings of the fifteenth ACM conference on Economics and computation*, pp. 693–710, 2014.
- [22] G. Radanovic, R. Devidze, D. Parkes, and A. Singla, "Learning to collaborate in markov decision processes," in *International Conference on Machine Learning*, pp. 5261–5270, PMLR, 2019.
- [23] D. Chen, Q. Zhang, and T. T. Doan, "Convergence and price of anarchy guarantees of the softmax policy gradient in markov potential games," *arXiv preprint arXiv:2206.07642*, 2022.
- [24] X. Chen, X. Deng, and S.-H. Teng, "Settling the complexity of computing two-player nash equilibria," *Journal of the ACM (JACM)*, vol. 56, no. 3, pp. 1–57, 2009.
- [25] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou, "The complexity of computing a nash equilibrium," *Communications of the ACM*, vol. 52, no. 2, pp. 89–97, 2009.
- [26] S. V. Macua, J. Zazo, and S. Zazo, "Learning parametric closed-loop policies for markov potential games," *CoRR*, vol. abs/1802.00899, 2018.
- [27] D. González-Sánchez and O. Hernández-Lerma, *Discrete-time stochastic control and dynamic potential games: the Euler–Equation approach*. Springer Science & Business Media, 2013.
- [28] A. Agarwal, S. M. Kakade, J. D. Lee, and G. Mahajan, "On the theory of policy gradient methods: Optimality, approximation, and distribution shift," *The Journal of Machine Learning Research*, vol. 22, no. 1, pp. 4431–4506, 2021.
- [29] G. Qu and A. Wierman, "Finite-time analysis of asynchronous stochastic approximation and q-learning," in *Proceedings of Thirty Third Conference on Learning Theory (J. Abernethy and S. Agarwal, eds.)*, vol. 125 of *Proceedings of Machine Learning Research*, pp. 3185–3205, PMLR, 09–12 Jul 2020.
- [30] R. Zhang, Y. Zhang, R. Konda, B. Ferguson, J. Marden, and N. Li, "Markov games with decoupled dynamics: Price of anarchy and sample complexity," *arXiv preprint arXiv:2304.03840*, 2023.
- [31] A. Vetta, "Nash equilibria in competitive societies, with applications to facility location, traffic routing and auctions," in *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.*, pp. 416–425, IEEE, 2002.
- [32] L. S. Shapley, "Notes on the n-person game—ii: The value of an n-person game.(1951)," *Lloyd S Shapley*, 1951.
- [33] A. Ghorbani and J. Zou, "Data shapley: Equitable valuation of data for machine learning," in *International Conference on Machine Learning*, pp. 2242–2251, PMLR, 2019.
- [34] Y. Kwon and J. Zou, "Beta shapley: a unified and noise-reduced data valuation framework for machine learning," *arXiv preprint arXiv:2110.14049*, 2021.