ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-6327-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICASSP49357.2023.10094870

NETWORKED POLICY GRADIENT PLAY IN MARKOV POTENTIAL GAMES

Sarper Aydın and Ceyhun Eksin

Texas A&M University, College Station, TX, USA

ABSTRACT

We propose a networked policy gradient play algorithm for solving Markov potential games. In a Markov game, each agent has a reward function that depends on the actions of all the agents and a common dynamic state. A differentiable Markov potential game admits a potential value function that has local gradients equal to the gradients of agents' local value functions. In the proposed algorithm, agents use parameterized policies that depend on the state and other agents' policies. Agents use stochastic gradients and local parameter values received from their neighbors to update their policies. We show that the joint policy parameters converge to a first-order stationary point of a Markov potential game in expectation for general action and state spaces. Numerical results on the lake game exemplify the convergence of the proposed method.

***Index Terms*—** Game theory, reinforcement learning, distributed algorithms

1. INTRODUCTION

In multi-agent reinforcement learning (MARL) settings, multiple agents learn and optimize their actions in dynamic environments without the knowledge of the structure of their rewards and the state transition dynamics [1]. Many real-life applications in the scope of MARL, e.g., autonomous driving [2], electric vehicles [3], and power grids [4], entail competition among agents where agents' rewards and state transition dynamics depend on the joint actions of agents in the system. Markov (stochastic) games model such competitive MARL settings, where agents take actions to maximize individual rewards that depend on other agents' actions and a dynamic state that evolves according to joint actions taken [5]. Here, we focus on solving an important subclass of Markov games known as Markov potential games that admit potential value functions that capture the change in agent's discounted sum of rewards resulting from unilateral policy changes.

The proposed solution algorithm is based on episodic policy gradient methods [6, 7]. In episodic policy gradient, parametrized policies are updated using stochastic gradients that are computed based on rewards collected over roll-out horizons, i.e., episodes. Here, we introduce a new class of policy gradient play algorithms, in which agents include others' policy parameters in their own parametrized policy functions in addition to the state. Given such parametrized policies, agents play against each other in two different episodes with randomly generated lengths to estimate their rewards and gradients induced by the parameterized policies. Agents update their parameters using estimated gradients. In order to be able to execute their policies, agents need access to the policy parameters of other agents which may not be readily available unlike the state. Instead, agents keep and update estimates about others' policy parameters based on information exchanges with their neighbors. We show that agents' policies converge to a stationary point of the potential value function (Theorem 1). This result leverages the facts that random horizon sampling gives unbiased policy gradient estimates (Lemma 2), and individual beliefs on others' parameters converge to true parameter values (Lemma 3).

Initial approaches that use episodic policy gradient play for solving Markov games consider open/closed loop policies for continuous state and action spaces assuming reward functions and state transitions are known [8, 9]. More recent approaches focus only on the direct or softmax parameterization where state and action spaces are finite, and adapt different variations of gradient updates, e.g., projected, natural, for solving Markov potential or general-sum games [10–17]. Our analysis generalizes the setting of these recent studies to continuous state and action pairs given unknown reward and state transition dynamics. Policies that incorporate other agents' policy parameters and local exchange information are additional features of the proposed algorithm that distinguish it from existing MARL algorithms. Numerical experiments on the lake game (a Markov potential game [18]) demonstrate the efficacy of networked policy gradient play.

2. MARKOV POTENTIAL GAMES

A Markov [5] game is played by N agents belonging to the set $\mathcal{N} := \{1, \dots, N\}$. Agent $i \in \mathcal{N}$ can choose its action $a_i \in \mathcal{A}_i \subseteq \mathbb{R}^{\mathcal{K}}$ at a common state $s \in \mathcal{S}$, where the sets $\mathcal{A}_i$ and $\mathcal{S}$ are not necessarily finite. The joint action profile is accordingly defined as $a = (a_1, a_2, \cdots, a_N) \in \mathcal{A}^N := \times_{i\in\mathcal{N}} \mathcal{A}_i$. The joint action profile and the prior state determine the transition probability $\mathcal{P}^a_{s'',s'} = \mathbb{P}(s''|s', a)$, and an initial state s_0 is distributed with $\rho : \mathcal{S} \to [0, 1]$. Agent i receives reward $r_i : \mathcal{S} \times \mathcal{A}^N \to \mathbb{R}$ determined by the action profile and the state. Future rewards are discounted by a discount factor $\gamma \in (0, 1)$ to obtain the cumulative reward. We define the

This work was supported by NSF CCF-2008855.

game by the tuple $\Gamma := (\mathcal{N}, \mathcal{A}^N, \mathcal{S}, \{r_i\}_{i\in\mathcal{N}}, \mathcal{P}, \gamma, \rho)$.

Each agent utilizes a policy function $\pi_i : \mathcal{S} \times \pi_{-i} \rightarrow \Delta(\mathcal{A}_i)$ to sample an action given a state and others agents' policies, where $\Delta(.)$ denotes all probability distributions over the given set, and $-i := \mathcal{N}\backslash\{i\}$ refers to the set of all agents except agent i. When agents follow the joint policy $\Pi = \times_{i\in\mathcal{N}} \pi_i$, each agent has a value function $V_i^{\Pi} : \mathcal{S} \rightarrow \mathbb{R}$ for each state, over an infinite horizon as a discounted sum of rewards,

$$V_i^{\Pi}(s) = \mathbb{E}_{(s,a)\sim\mathcal{P}}\Big[\sum_{t=0}^{\infty} \gamma^t r_{i,t}(s_t, a_t)|s_0 = s\Big], \quad (1)$$

where $\mathcal{P}$ is the distribution of the sequence of states and actions induced by the joint policy[1]. Note here we include time sub-index $t \in \mathbb{N}^+$ in $r_{i,t}$, a_t, and s_t, to indicate agent i's reward, and joint action and common state at decision epoch t. Similarly, we define the Q-function of agent i ($Q_i : \mathcal{S} \times \mathcal{A}^N \rightarrow \mathbb{R}$) for each state $s \in \mathcal{S}$, and joint action pair $a \in \mathcal{A}^N$ given the joint policy Π as below,

$$Q_i^{\Pi}(s, a) = \mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t r_{i,t}(s_t, a_t)|s_0 = s, a_0 = a\Big]. \quad (2)$$

Potential games in static games are an important class of games that assume the existence of a potential function capturing the change in individual utility function values based on unilateral action changes [19]. We define Markov potential games similarly by assuming the existence of a potential value function that mirrors changes in every agent's local value function due to unilateral policy deviations.

Definition 1 (Markov Potential Games) *A game Γ is a Markov potential game, if there exists a potential value function $V^{\Pi}(s) : \Pi \times \mathcal{S} \rightarrow \mathbb{R}$ that is equal to the discounted sum of potential rewards $r_t \in \mathbb{R}$, i.e., $V^{\Pi}(s) = \mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t r_t(s_t, a_t)|s_0 = s, \Pi\Big]$, such that for all $i \in \mathcal{N}$*

$$V_i^{\hat{\Pi}}(s) - V_i^{\Pi}(s) = V^{\hat{\Pi}}(s) - V^{\Pi}(s) \quad \textit{for all } s \in \mathcal{S}, \quad (3)$$

where $\hat{\Pi}$ and Π are two joint policies differing in the policy of agent $i \in \mathcal{N}$ only, i.e., $\hat{\Pi} = (\hat{\pi}_i, \pi_{-i})$ and $\Pi = (\pi_i, \pi_{-i})$.

We assume that agents' joint policies are parametrized by unconstrained and continuous variables $\theta = (\theta_i, \theta_{-i}) \in R^M$ where individual policy parameters $\theta_i \in R^{K_i}$ are such that the following holds $\sum_{i\in\mathcal{N}} K_i = M$. Given the parameterized policies $\Pi_\theta : \mathbb{R}^M \times \mathcal{S} \rightarrow \Delta(\mathcal{A}_i)$, we have differentiable value functions $u_i : \mathbb{R}^M \rightarrow \mathbb{R}$ defined as follows as per (1),

$$u_i(\theta_i, \theta_{-i}) = V_i^{\Pi_\theta}(s) = \mathbb{E}_{\Pi_\theta}\Big[\sum_{t=0}^{\infty} \gamma^t r_{i,t}(s_t, a_t)|s_0 = s\Big]. \quad (4)$$

We further define differentiable Markov potential games as follows.

[1]For brevity, we remove subscript from the expectation in the rest of the paper, unless clarity necessitates it.

Definition 2 (Differentiable Markov Potential Games) *A game Γ is a Markov potential game with differentiable individual value functions u_i, if there exists a potential value function $u : \mathbb{R}^M \rightarrow \mathbb{R}$ such the following holds,*

$$\nabla_i u_i(\theta_i, \theta_{-i}) = \nabla_i u(\theta) \quad \textit{for all } \theta \in \mathbb{R}^M \quad (5)$$

where $\nabla_i(.) = \frac{\partial(.)}{\partial\theta_i}$ denotes the partial derivative of a given function with respect to the agent i's parameters θ_i.

3. POLICY GRADIENT PLAY WITH NETWORKED AGENTS

Given the joint parameterized policy, $\Pi_\theta : \mathbb{R}^M \times \mathcal{S} \rightarrow \Delta(\mathcal{A}^N)$, agent i's policy $\pi_{i,\theta} := \pi_i(a_i|s, \theta)$ is conditionally independent given the state and joint policy parameters, i.e.,

$$\Pi_\theta(a \in \mathcal{A}_q^N|s) = \prod_{i\in\mathcal{N}} \pi_{i,\theta}(a_i \in \mathcal{A}_{i,q}|s) \quad (6)$$

where $\mathcal{A}_q^N = \times_{i\in\mathcal{N}} \mathcal{A}_{i,q}$ and $\mathcal{A}_{i,q}$ are countable measurable partitions over the joint and individual set of actions respectively over which we can define probability distributions such that it holds $\mathcal{A}_i = \bigcup_{q=1}^{\infty} \mathcal{A}_{i,q}$. Note that each agent would like to maximize its cumulative rewards against other agents' policies given the joint action dependent state dynamics.

Next, we provide an expression of the gradient of agent i's value function in terms of the Q-function and sum of log-policies—see Section 7 for proof.

Lemma 1 *Given the parameterized value functions $u_i : \mathbb{R}^M \rightarrow \mathbb{R}$, the gradient of each value function u_i with respect to agent i's parameters θ_i is equal to,*

$$\nabla_i u_i(\theta_i, \theta_{-i}) = \frac{1}{(1-\gamma)} \mathbb{E}\Big[Q_i^{\Pi_\theta}(s, a) \sum_{n\in\mathcal{N}} \nabla_i \log \pi_{n,\theta}(a_n|s)\Big]. \quad (7)$$

In policy gradient play, each agent uses stochastic gradients to update its policy parameters,

$$\theta_{i,t} = \theta_{i,t-1} + \alpha_t \hat{\nabla}_i u_i(\theta_{i,t-1}, \theta_{-i,t-1}), \quad (8)$$

where α_t is a common step size (for the sake of simplicity), and $\hat{\nabla}_i u_i(\theta_{i,t-1}, \theta_{-i,t-1})$ is the stochastic gradient computed based on rewards collected on a roll-out horizon (episode).

We note that the local policies π_i and the stochastic gradients $\hat{\nabla}_i u_i$ depend on other agents' policy parameters. Other agents' parameters θ_{-i} may not be readily available to agent i. Here we assume agent i keeps an estimate of other agents' policy parameters based on information received from its neighbors $\mathcal{N}_i := \{j : (i, j) \in \mathcal{E}\}$ in the communication network $\mathcal{G} = (\mathcal{N}, \mathcal{E})$. Agent i updates its estimate about agent j's policy parameters $\hat{\theta}_{j,t}^i$ locally as follows,

$$\hat{\theta}_{j,t}^i = \sum_{l\in\mathcal{N}_i\bigcup\{i\}} w_{j,l}^i \hat{\theta}_{j,t}^l, \quad (9)$$

where $w^i_{j,l} \geqslant 0$ is the weight that agent i has on agent l's estimate of agent j's parameters.

Assumption 1 *The network $\mathcal{G}$ is strongly connected with weights satisfying* **a)** $w^i_{j,l} \geqslant \eta$ *for* $\eta > 0$ *only if* $l \in \mathcal{N}_i \cup \{i\}$, *otherwise* $w^i_{j,l} = 0$, **b)** $w^i_{i,i} = 1$, *and* **c)** $\sum_{l \in \mathcal{N}_i \bigcup \{i\}} w^i_{j,l} = 1$ *for all* i, j.

The stochastic gradient $\hat{\nabla}_i u_i$ in (8) is computed by estimating the Q-values $\hat{Q}_i$ and gradient of log-policy $\nabla_i \log \pi_\theta$, and by substituting these estimates in (7) along with parameter estimates $\hat{\theta}^i_{-i,t}$. Here, we employ and adapt the random horizon sampling method to devise two episodes over which $\hat{\nabla}_i \log \pi_\theta$ and $\hat{Q}_i$ are computed. As noted in [20], the sequential decision-making structure of the RL setting creates a bias in the gradient estimates in multi-agent settings. The two episodes with random horizons $\mathcal{T}_1$ and $\mathcal{T}_2$ generated from a geometric distribution $Geom(1-\gamma^{0.5})$ such that $\mathbb{P}(\mathcal{T}_k = \tau) = (1-\gamma^{0.5})\gamma^{0.5\times\tau}$ with $k \in \{1,2\}$ ensure unbiased estimates—see Lemma 2. The steps of the episodes and updates are detailed in Algorithm 1.

Algorithm 1 Networked Policy Gradient

1: **Input:** Local estimates $\hat{\theta}^i_{-i,0}$ and $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, initial state s_0 and initial policy $\Pi_{\theta,0}$, and discount factor γ.
2: **for** $t = 1, 2, \cdots$ **do**
3: Draw $\mathcal{T}_1 \sim Geom(1-\gamma^{0.5})$ and reset s_0.
4: Sample actions $a_{i,0} \sim \pi_{i,\hat{\theta}_{t-1}}(.|s_0)$ for all $i \in \mathcal{N}$
5: **for** $\tau = 1, 2 \cdots, \mathcal{T}_1$ **do**
6: Reach state $s_\tau \sim \mathcal{P}^{a_{i,\tau}}_{s_\tau, s_{\tau-1}}$
7: Sample and take actions, $a_{i,\tau+1} \sim \pi_{i,\hat{\theta}_{t-1}}(.|s_{\tau+1})$ for all $i \in \mathcal{N}$
8: **end for**
9: Compute $\nabla_i \log \pi_\theta(a_{\mathcal{T}_1+1}|s_{\mathcal{T}_1+1})$ for all $i \in \mathcal{N}$
10: Draw $\mathcal{T}_2 \sim Geom(1-\gamma^{0.5})$ and set $\hat{Q}_i = 0$.
11: **for** $\tau = 1, 2, \cdots, \mathcal{T}_2$ **do**
12: Receive rewards $r_{i,\tau+\mathcal{T}_1}$ for all $i \in \mathcal{N}$.
13: Collect rewards $\hat{Q}_i = \hat{Q}_i + \gamma^{\tau/2} r_{i,\tau+\mathcal{T}_1}$ for $i \in \mathcal{N}$.
14: Reach state $s_{\tau+\mathcal{T}_1+1} \sim \mathcal{P}^{a_{i,\tau}}_{s_{\tau+\mathcal{T}_1+1}, s_{\tau+\mathcal{T}_1}}$.
15: Sample and take actions $a_{i,\tau+\mathcal{T}_1+1} \sim \pi_{i,\hat{\theta}_{t-1}}(.|s_{\tau+\mathcal{T}_1})$ for all $i \in \mathcal{N}$.
16: **end for**
17: Compute $\hat{Q}_i = \hat{Q}_i + \gamma^{\tau/2} r_{i,\mathcal{T}_1+\mathcal{T}_2+1}$ for $i \in \mathcal{N}$
18: Estimate stochastic gradients by substituting $\hat{Q}_i$ and $\nabla_i \log \pi_\theta$ for the corresponding terms in (7).
19: Update parameters (8) with $\theta_{-i,t-1}$ replaced by $\hat{\theta}_{-i,t-1}$.
20: Update local copies $\hat{\theta}^i_{j,t}$ using (9) for $j \in -i$ and $i \in \mathcal{N}$.
21: **end for**

4. CONVERGENCE OF NETWORKED POLICY GRADIENT PLAY IN MARKOV POTENTIAL GAMES

We make the following assumption on the gradient step size.

Assumption 2 *The step size α_t is in the order of $\alpha = O(1/t)$.*

This assumption is equivalent to the standard assumption of square summable but not summable step-sizes commonly used in convergence of gradient algorithms. The local stochastic gradient obtained by Algorithm 1 is an unbiased estimate of the local gradient in (7).

Lemma 2 *The stochastic estimate $\hat{\nabla}_i u_i(\theta_i, \theta_{-i})$ of the policy gradient in* (7) *is unbiased and bounded for all $i \in \mathcal{N}$.*

Similar to [20], the result follows from the fact that we collect rewards using special discount rates $\gamma^{\tau/2}$ during the two episodes with independent and identically sampled random horizon lengths $\mathcal{T}_1$ and $\mathcal{T}_2$—see steps 13 and 17 in the Algorithm. Next, we state the convergence of beliefs to true policy parameter values.

Lemma 3 *If $\hat{\theta}^i_{j0} = \theta_{j0}$ is satisfied for any pair of agents $(i,j) \in \mathcal{N} \times \mathcal{N} \backslash \{i\}$, then local copies $\hat{\theta}^i_{j,t}$ converges to $\theta_{j,t}$ with the rate $O(\log t/t)$ in expectation, i.e. $\mathbb{E}(||\hat{\theta}^i_{j,t} - \theta_{j,t}|| = O(\frac{\log t}{t})$.*

The result can be proven using the facts that change in parameters is bounded, the step size are such that $\alpha_t = O(1/t)$, and the weights create a row stochastic matrix. Using the unbiased stochastic gradients (Lemma 2) and fast-tracking of the parameter values (Lemma 3), we have the following result, thanks to diminishing stepsizes.

Theorem 1 *Suppose potential game property holds* (5) *for the agents with networked policies defined in* (6). *Let $\{\theta_t\}_{t\geqslant 1}$ be the sequence of policy parameters generated by Algorithm 1. Then, the policy parameters $\{\theta_t\}_{t\geqslant 1}$ converge to a first-order stationary point of the potential function in expectation,*

$$\lim_{t\to\infty} \mathbb{E}(||\nabla_{\theta_t} u(\theta_t)||^2) = 0. \tag{10}$$

This result implies that agents reach a Nash equilibrium (NE), i.e., optimal behavior, of networked policies for convex potential value functions. For non-convex value functions, a stationary point is not necessarily a NE, implying that the convergence is to an approximate NE.

5. NUMERICAL EXPERIMENTS

Each agent determines its rate of phosphorus usage $a_{i,t}$ around a lake. The state $s_t \in \mathbb{R}$, representing the level of phosphorus in the lake, has the following dynamics

$$s_t = bs_{t-1} + \frac{s^c_{t-1}}{s^c_{t-1}+1} + \sum_{i\in\mathcal{N}} a_{i,t-1} \tag{11}$$

where b and c are positive constants. Agent i's reward increases logarithmically with its phosphorus usage, while, at the same time, it prefers to keep the lake phosphorus free,

$$r_{i,t} = log(da_{i,t}) - s^2_t, \tag{12}$$

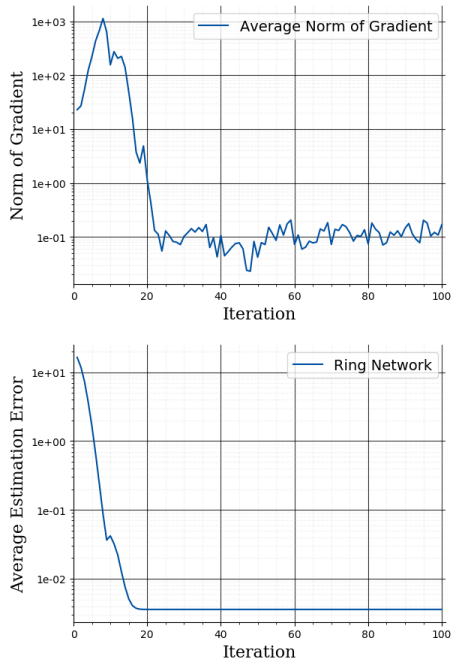

Fig. 1. Networked policy gradient in lake game. (Top) Average norm of gradients of agents $\frac{1}{N}\sum_{i\in\mathcal{N}}||\nabla_i u_i(\theta_i,\hat{\theta})||$ (Bottom) Average estimation error $\frac{1}{N(N-1)}\sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}\setminus\{i\}}||\theta_{i,t}-\hat{\theta}^i_{j,t}||$.

where $d>0$. The lake game is a Markov potential game [18].

We consider $N=5$ agents with discount rate $\gamma=0.9$, and game parameters $b=0.45$, $c=2$, and $d=100$. Agents use a policy with $K_i=2$ real-valued parameters:

$$\pi_{i,\theta}=\text{sigmoid}\left(\theta_{i,s}s+\theta_{i,-i}\left(\frac{1}{(N-1)}\sum_{j\in\mathcal{N}\setminus\{i\}}(\theta_{j,s})\right)\right). \quad (13)$$

We use sigmoid transformation to map unconstrained variables to the range $[0,1]$. We set $s_0=0$ in all episodes and $\alpha_t=0.005/t$. We use a ring communication network with weights $w^i_{i,l}=0.30$ and $w^i_{j,l}=0.70/|\mathcal{N}_i|$ for all $j\in\mathcal{N}_i$.

Fig. 1 (Top) shows the average of individual gradients over 20 runs with randomly initialized policy parameters converges around a stationary point of the value functions. Fig. 1 (Bottom) indicates that the beliefs on other agents' parameters converge to the actual parameter values. The two observations confirm the convergence of the joint policies to a stationary point of a potential value function.

Remark 1 *Note that the policy defined in* (13) *is a deterministic policy, and the analytical definition of deterministic policy gradient is different from the stochastic version. A stochastic policy is equivalent to a deterministic policy when its variance goes to zero [21].*

6. CONCLUSION

We devised a novel class of networked policy gradient play algorithms for solving Markov potential games. The algorithm has two distinct features from existing MARL algorithms: local policies that depend on others' policies (not just the state), and agents exchanging policy parameters over a communication network. We showed that agents beliefs on others' policy parameters convergence to true values. The algorithm includes random roll-out horizons that achieves unbiased estimates of the policy gradients. Given the unbiased gradients and convergence of beliefs on others' policy parameters, we showed convergence of the algorithm to a stationary point. We validated our results with numerical experiments.

7. APPENDIX

7.1. Proof of Lemma 1

By Policy Gradient Theorem [7], we define the gradient in the following,

$$\nabla_i u_i(\theta_i,\theta_{-i})=\int_{\substack{a\in\mathcal{A},\\ s\in\mathcal{S}}} Q_i^{\Pi_\theta}(s,a)d^{\Pi_\theta}\nabla_i\pi_\theta(a|s)\,da\,ds, \quad (14)$$

where $d^{\Pi_\theta}=\sum_{t=0}^{\infty}\gamma^t\rho^a_{s_0,s,t}$ is the discounted sum of density functions $\rho^a_{s_0,s,t}$ of the transition probability function $\mathcal{P}^a_{s_0,s,t}$ from the initial state s_0 to the state s given the joint action a at t steps ahead, and similarly $\pi_\theta(a|s)$ stands for the density function of the joint policy Π_θ.

Then, applying the log-likelihood trick by dividing and multiplying the gradient of $\nabla_i\pi_\theta(a|s)$ by the density $\pi_\theta(a|s)$, it becomes as follows,

$$\nabla_i u_i(\theta_i,\theta_{-i})=\int_{\substack{a\in\mathcal{A},\\ s\in\mathcal{S}}} Q_i^{\Pi_\theta}(s,a)d^{\Pi_\theta}\pi_\theta(a|s)\frac{\nabla_i\pi_\theta(a|s)}{\pi_\theta(a|s)}\,da\,ds \quad (15)$$

$$=\int_{\substack{a\in\mathcal{A},\\ s\in\mathcal{S}}} Q_i^{\Pi_\theta}(s,a)d^{\Pi_\theta}\pi_\theta(a|s)\nabla_i\log\pi_\theta(a|s)\,da\,ds. \quad (16)$$

We divide the integral by $(1-\gamma)$ to have a proper expectation, and using the definition of networked policies (6), the policy gradient becomes,

$$=\int_{\substack{a\in\mathcal{A},\\ s\in\mathcal{S}}} Q_i^{\Pi_\theta}(s,a)d^{\Pi_\theta}\pi^\theta(a|s)\sum_{n\in\mathcal{N}}\nabla_i\log\pi^\theta_n(a_n|s)\,da\,ds \quad (17)$$

$$=\frac{1}{(1-\gamma)}\mathbb{E}_{(s,a)\sim\mathcal{P}}\Big[Q_i^{\Pi_\theta}(s,a)\sum_{n\in\mathcal{N}}\nabla_i\log\pi_{n,\theta}(a_n|s)\Big]. \quad (18)$$

8. REFERENCES

[1] Kaiqing Zhang, Zhuoran Yang, and Tamer Başar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," *Handbook of Reinforcement Learning and Control*, pp. 321–384, 2021.

[2] Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua, "Safe, multi-agent, reinforcement learning for autonomous driving," *arXiv preprint arXiv:1610.03295*, 2016.

[3] Dawei Qiu, Yi Wang, Tingqi Zhang, Mingyang Sun, and Goran Strbac, "Hybrid multi-agent reinforcement learning for electric vehicle resilience control towards a low-carbon transition," *IEEE Transactions on Industrial Informatics*, 2022.

[4] Daner Hu, Zhenhui Ye, Yuanqi Gao, Zuzhao Ye, Yonggang Peng, and Nanpeng Yu, "Multi-agent deep reinforcement learning for voltage control with coordinated active and reactive power optimization," *IEEE Transactions on Smart Grid*, 2022.

[5] Lloyd S Shapley, "Stochastic games," *Proceedings of the national academy of sciences*, vol. 39, no. 10, pp. 1095–1100, 1953.

[6] Ronald J Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, no. 3, pp. 229–256, 1992.

[7] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour, "Policy gradient methods for reinforcement learning with function approximation," *Advances in neural information processing systems*, vol. 12, 1999.

[8] David González-Sánchez and Onésimo Hernández-Lerma, *Discrete–time stochastic control and dynamic potential games: the Euler–Equation approach*, Springer Science & Business Media, 2013.

[9] Sergio Valcarcel Macua, Javier Zazo, and Santiago Zazo, "Learning parametric closed-loop policies for markov potential games," *arXiv preprint arXiv:1802.00899*, 2018.

[10] Runyu Zhang, Zhaolin Ren, and Na Li, "Gradient play in stochastic games: stationary points, convergence, and sample complexity," *arXiv preprint arXiv:2106.00198*, 2021.

[11] Stefanos Leonardos, Will Overman, Ioannis Panageas, and Georgios Piliouras, "Global convergence of multi-agent policy gradient in markov potential games," *arXiv preprint arXiv:2106.01969*, 2021.

[12] Dongsheng Ding, Chen-Yu Wei, Kaiqing Zhang, and Mihailo Jovanovic, "Independent policy gradient for large-scale markov potential games: Sharper rates, function approximation, and game-agnostic convergence," in *International Conference on Machine Learning*. PMLR, 2022, pp. 5166–5220.

[13] David H Mguni, Yutong Wu, Yali Du, Yaodong Yang, Ziyi Wang, Minne Li, Ying Wen, Joel Jennings, and Jun Wang, "Learning in nonzero-sum stochastic games with potentials," in *International Conference on Machine Learning*. PMLR, 2021, pp. 7688–7699.

[14] Angeliki Giannou, Kyriakos Lotidis, Panayotis Mertikopoulos, and Emmanouil-Vasileios Vlatakis-Gkaragkounis, "On the convergence of policy gradient methods to nash equilibria in general stochastic games," *arXiv preprint arXiv:2210.08857*, 2022.

[15] Dingyang Chen, Qi Zhang, and Thinh T Doan, "Convergence and price of anarchy guarantees of the softmax policy gradient in markov potential games," *arXiv preprint arXiv:2206.07642*, 2022.

[16] Roy Fox, Stephen M Mcaleer, Will Overman, and Ioannis Panageas, "Independent natural policy gradient always converges in markov potential games," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2022, pp. 4414–4425.

[17] Weichao Mao, Lin Yang, Kaiqing Zhang, and Tamer Basar, "On improving model-free algorithms for decentralized multi-agent reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2022, pp. 15007–15049.

[18] W Davis Dechert and SI O'Donnell, "The stochastic lake game: A numerical solution," *Journal of Economic Dynamics and Control*, vol. 30, no. 9-10, pp. 1569–1587, 2006.

[19] Dov Monderer and Lloyd S Shapley, "Potential games," *Games and economic behavior*, vol. 14, no. 1, pp. 124–143, 1996.

[20] Kaiqing Zhang, Alec Koppel, Hao Zhu, and Tamer Basar, "Global convergence of policy gradient methods to (almost) locally optimal policies," *SIAM Journal on Control and Optimization*, vol. 58, no. 6, pp. 3586–3612, 2020.

[21] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller, "Deterministic policy gradient algorithms," in *International conference on machine learning*. PMLR, 2014, pp. 387–395.