Learning stationary strategies in large-population multi-agent systems with exchangeable agents

Paper Id: 6496

Abstract

In this paper, reinforcement learning for large population multi-agent systems with exchangeable agents is presented. A multi-agent system is said to have exchangeable agents if permuting the index of agents does not impact the dynamics and rewards. Such systems are also called systems with symmetric or homogeneous agents. A key feature of exchangeable multi-agent systems is that the dynamic and reward coupling between the agents is through the mean-field (i.e., the empirical distribution). The planning solution for such systems—both for the case when the agents are strategic (i.e., mean-field games) and when the agents are co-operative (mean-field teams) have been considered in the literature. In this paper, we present off-line policy gradient based reinforcement learning algorithms that converge to a local stationary mean-field equilibrium (in case of games) or a local stationary mean-field team optimal solution (in case of teams). The algorithms are demonstrated using a stylized model of malware spread in networks. The example also illustrates that the team and game solution may, in general, differ.

1 Introduction

Multi-agent systems with large number of agents arise in many modern technological systems such as Internet of Things (IoT), swarm robotics, taxi supply-demand matching, smart grids, and algorithmic trading. A salient feature of many such systems is that agents are exchangeable i.e., if we permute the index of the agents, the system dynamics and rewards do not change. We start by presenting a model that makes this statement precise.

1.1 System model

Consider a multi-agent system with n agents indexed by the set $N = \{1, \dots, n\}$. For any agent i, $i \in N$, let $X_t^i \in \mathcal{X}$ and $A_t^i \in \mathcal{A}$ denote the state and action of agent i at time t. Note that the state space $\mathcal{X}$ and action space $\mathcal{A}$ are the same for all agents. For ease of exposition, we assume that $\mathcal{X}$ and $\mathcal{A}$ are finite. However, most of the results extend to continuous $\mathcal{X}$ and $\mathcal{A}$ under appropriate technical assumptions.

Let $\mathbf{X}_t = (X_t^1, \dots, X_t^n)$ and $\mathbf{A}_t = (A_t^1, \dots, A_t^n)$ denote the state and action profile of all agents in the system. The system evolves in a controlled Markovian manner, i.e., for any realization $(\mathbf{x}_{1:t}, \mathbf{a}_{1:t})$ of $(\mathbf{X}_{1:t}, \mathbf{A}_{1:t})$, we have

$$\mathbb{P}(\mathbf{X}_{t+1} | \mathbf{X}_{1:t} = \mathbf{x}_{1:t}, \mathbf{A}_{1:t} = \mathbf{a}_{1:t}) = \mathbb{P}(\mathbf{X}_{t+1} | \mathbf{X}_t = \mathbf{x}_t, \mathbf{A}_t = \mathbf{a}_t).$$

At time t, a vector reward $\mathbf{R}_t = (R_t^1, \dots, R_t^n)$ is generated, where R_t^i corresponds to the reward for agent i. The rewards are random variables which depend only on the current state and action, i.e.,

$$\mathbb{E}[\mathbf{R}_t | \mathbf{X}_{1:t}, \mathbf{A}_{1:t}] = \mathbb{E}[\mathbf{R}_t | \mathbf{X}_t, \mathbf{A}_t].$$

Following Arabneydi and Mahajan (2016), we say that the agents are *exchangeable* if permuting the states and actions of the agents permutes the next state and the rewards.[1] We call such systems *exchangeable* multi-agent systems (E-MAS).

Definition 1 (Exchangeable agents) The agents of a Markovian multi-agent system are called *exchangeable* if for any permutation[2] σ of N and any $\mathbf{x}_t$, $\mathbf{x}_{t+1}$, and $\mathbf{a}_t$:

1. the dynamics are exchangeable, i.e.,

$$\mathbb{P}(\mathbf{X}_{t+1} = \mathbf{x}_{t+1} | \mathbf{X}_t = \mathbf{x}_t, \mathbf{A}_t = \mathbf{a}_t) = \mathbb{P}(\mathbf{X}_{t+1} = \sigma\mathbf{x}_{t+1} | \mathbf{X}_t = \sigma\mathbf{x}_t, \mathbf{A}_t = \sigma\mathbf{a}_t).$$

2. the rewards are exchangeable, i.e.,

$$\mathbb{E}[\mathbf{R}_t | \mathbf{X}_t = \mathbf{x}_t, \mathbf{A}_t = \mathbf{a}_t] = \mathbb{E}[\sigma\mathbf{R}_t | \mathbf{X}_t = \sigma\mathbf{x}_t, \mathbf{A}_t = \sigma\mathbf{a}_t].$$

□

Planning for E-MAS has been extensively investigated. For static games, the results go back to results of symmetric games by Nash (1951); Maynard Smith and Price (1973). For dynamic games, they go back to results on anonymous games by Bergin and Bernhardt (1995); Jovanovic and Rosenthal (1988). Starting from Huang et al. (2006); Lasry and Lions (2007), there is an enormous literature on mean-field games. Similar ideas have also been used in mean-field teams (Arabneydi and Mahajan, 2014, 2016) and $\mathbb{C}$Dec-POMDPs (Nguyen et al., 2017a,b).

[1] In the game theory literature, exchangeable multi-agent systems are called *symmetric games* (Nash, 1951). The term *exchangeable multi-agent systems* is motivated from the connection with exchangeable random variables in probability theory (Diaconis, 1988).

[2] For any permutation σ of N, $\sigma(x^1, \dots, x^n)$ denotes the vector $(x^{\sigma(1)}, \dots, x^{\sigma(n)})$.

However, the literature on reinforcement learning for E-MAS is much sparser. Recently, Hüttenrauch et al. (2018) considered reinforcement learning for a swarm of homogeneous agents with local observations. In this paper, we investigate reinforcement learning for multi-agent systems with exchangeable agents with a completely decentralized information structure.

1.2 Mean-field coupling

Let $\Delta(\mathcal{X} \times \mathcal{A})$ denote the space of probability distributions on $\mathcal{X} \times \mathcal{A}$. Given any vector $\mathbf{x} = (x^1, \dots, x^n)$ and $\mathbf{a} = (a^1, \dots, a^n)$, let $\bar{z} = \xi(\mathbf{x}, \mathbf{a})$, $\bar{z} \in \Delta(\mathcal{X} \times \mathcal{A})$, denote the mean-field (or empirical distribution) of $(\mathbf{x}, \mathbf{a})$, i.e.,

$$\bar{z}(x, a) = \frac{1}{n} \sum_{i \in N} \mathbb{1}\{x^i = x, a^i = a\}, \quad \forall (x, a) \in \mathcal{X} \times \mathcal{A}.$$

For an E-MAS, we use $\bar{Z}_t = \xi(\mathbf{X}_t, \mathbf{A}_t)$ to denote the mean-field of states and actions.

A characteristic feature of E-MAS is that the dynamic and reward coupling between the agents is only through the mean-field (or the empirical distribution). In particular, we have the following.

Proposition 1 *In E-MAS, for any permutation σ of N,*

1. *the dynamics are coupled through the mean-field, i.e.,*

$$\begin{aligned}\mathbb{P}(X^i_{t+1} | \mathbf{X}_t = \mathbf{x}_t, \mathbf{A}_t = \mathbf{a}_t) \\ = \mathbb{P}(X^i_{t+1} | X^i_t = x^i_t, A^i_t = a^i_t, \bar{Z}_t = \xi(\mathbf{x}_t, \mathbf{a}_t));\end{aligned}$$

2. *the rewards are coupled through the mean-field, i.e.,*

$$\begin{aligned}\mathbb{E}[R^i_t | \mathbf{X}_t = \mathbf{x}_t, \mathbf{A}_t = \mathbf{a}_t] \\ = \mathbb{E}[R^i_t | X^i_t = x^i_t, A^i_t = a^i_t, \bar{Z}_t = \xi(\mathbf{x}_t, \mathbf{a}_t)].\end{aligned}$$

□

PROOF This is an immediate consequence of exchangeability of dynamics and reward and the fact that for any permutation σ of N, $\xi(\mathbf{x}, \mathbf{a}) = \xi(\sigma\mathbf{x}, \sigma\mathbf{a})$. ■

For ease of notation, we use the following:

- $P(x^i_{t+1} | x^i_t, a^i_t, z_t)$ denotes $\mathbb{P}(X^i_{t+1} = x^i_{t+1} | X^i_t = x^i_t, A^i_t = a^i_t, \bar{Z}_t = \bar{z}_t)$.
- $r(x^i_t, a^i_t, \bar{z}_t)$ denotes $\mathbb{E}[R^i_t | X^i_t = x^i_t, A^i_t = a^i_t, \bar{Z}_t = \bar{z}_t]$.
- Z_t denotes the mean-field of states, i.e.,

$$Z_t(x) = \frac{1}{n} \sum_{i \in N} \mathbb{1}\{X^i_t = x\}, \quad \forall x \in \mathcal{X}.$$

Note that due to exchangeability both P and r are the same for all agents.

As an example of E-MAS, consider the following model from Huang and Ma (2016, 2017).

Example 1 (Malware spread in networks) Let X^i_t denote the health of agent i belonging to $\mathcal{X} = [0, 1]$, where $X^i_t = 0$ is the most healthy state and $X^i_t = 1$ is the least healthy state. The action space $\mathcal{A} = \{0, 1\}$, where $A^i_t = 0$ means DO NOTHING and $A^i_t = 1$ means REPAIR. The dynamics are given by

$$X^i_{t+1} = \begin{cases} X^i_t + (1 - X^i_t)\omega_t, & \text{for } A^i_t = 0, \\ 0, & \text{for } A^i_t = 1, \end{cases}$$

where $\{\omega_t\}_{t \geq 1}$ is a $[0, 1]$-valued i.i.d. process with probability density f. The above dynamics means that if the agent takes the DO NOTHING action, then its state deteriorates to a worse condition in the interval $[1 - X^i_t, 1]$; if the agent takes the REPAIR action, then its state resets to the most healthy state.

In this example, dynamics are decoupled. But the rewards are coupled through the mean-field Z_t of states. The reward of agent i at time t is:

$$r(X^i_t, A^i_t, Z_t) = -(k + \langle Z_t \rangle) X^i_t - \lambda A^i_t,$$

where $\langle Z_t \rangle$ is the mean of Z_t (i.e., equal to $\int_0^1 x Z_t(x) dx$. The reward function indicates that an agent incurs a cost of λ for taking the REPAIR action. In addition, the agent incurs a cost $k + \langle Z_t \rangle$ times its state X^i_t. □

In the above example with n agents, the state space is $[0, 1]^n$ and the action space is $\{0, 1\}^n$. Thus, the size of the state and action spaces increases exponentially with the number of agents. However, as the number of agents become large, it is possible to use mean-field approximation (as explained below) to decouple the agents and obtain a solution whose complexity does not depend on the number of agents.

1.3 Mean-field approximations

When the number n of agents is large, each agent has a negligible influence on the mean-field and, hence, on the dynamics and cost of other agents. So, it is reasonable to assume that each agent ignores the state of other agents. More precisely:

(A1) All agents follow a time-homogeneous *oblivious* strategy $\pi \colon \mathcal{X} \to \Delta(\mathcal{A})$, i.e., each agent chooses action $A^i_t \sim \pi(X^i_t)$.

Such strategies are called oblivious because each agent only uses it's current state to choose it's current action and is oblivious to the states of other agents.

The second assumption that we make is that the mean-field of the state becomes stationary.

(A2) When all agents follow strategy π, the mean-field of states $\{Z_t\}_{t \geq 0}$ converges almost surely to a constant limit z^* (which we call the stationary mean-field).

When the population is large, the mean-field of states and actions is related to the mean-field of the states as follows:

$$\bar{z}^*(x, a) = z^*(x)\pi(a|x), \quad \forall (x, a) \in \mathcal{X} \times \mathcal{A},$$

which we denote by $\bar{z}^* = z^* \odot \pi$. For the mean-field of states to be stationary, it must satisfy

$$z^*(y) = \sum_{x \in \mathcal{X}} \sum_{a \in \mathcal{A}} z^*(x)\pi(a|x) P(y|x, a, z^* \odot \pi). \quad (1)$$

We use the notation $z^* = \Phi(z^*, \pi)$ to indicate that z^* is the stationary mean-field and satisfies (1).

Following Weintraub et al. (2005, 2008); Adlakha et al. (2015), we assume that under Assumptions (A1) and (A2), agents evaluate their performance by assuming that the mean-field takes its stationary value at all times. In particular, given

a strategy $\pi\colon \mathcal{X} \to \Delta(\mathcal{A})$ and a candidate stationary mean-field distribution $z^* \in \Delta(\mathcal{X})$, agent i's evaluates its performance starting from initial state $x \in \mathcal{X}$ as

$$V_{\pi,z^*}(x) = \mathbb{E}_{\substack{A^i_t \sim \pi(X^i_t) \\ X^i_{t+1} \sim P_{\pi,z^*}(\cdot|X_t,A_t)}} \Bigg[\sum_{t=1}^{\infty} \gamma^{t-1} r(X^i_t, A^i_t, z^* \odot \pi) \Bigg],$$

where $\gamma \in (0,1)$ is the discount factor and $P_{\pi,z^*}(y|x,a) = P(y|x,a,z^* \odot \pi)$. Such a *mean-field approximation* may be written as the solution of the following Bellman fixed-point equation.

$$V_{\pi,z^*}(x) = \sum_{a\in\mathcal{A}} \pi(a|x) \Big[r(x,a,z^* \odot \pi) + \gamma \sum_{y\in\mathcal{X}} P_{\pi,z^*}(y|x,a) V_{\pi,z^*}(y) \Big].$$

1.4 Solution concepts

We consider two setups: games and teams. In games, each agent is interesting in maximizing its individual discounted total reward, i.e., agent i's reward is

$$\mathbb{E}\Bigg[\sum_{t=1}^{\infty} \gamma^t R^i_t \Bigg].$$

In teams, all agents are interested in maximizing the sum of discounted total rewards across all agents, i.e., the team reward is

$$\frac{1}{n} \mathbb{E}\Bigg[\sum_{t=1}^{\infty} \gamma^t \sum_{i\in N} R^i_t \Bigg].$$

where we have added a normalizing factor of $1/n$ so that the team reward does not blow up as the number of agents become large.

For games, we use the following refinement of Nash equilibrium, which was proposed by Adlakha et al. (2015), as a solution concept.

Definition 2 (Stationary mean-field equilibrium (SMFE)) A stationary mean-field equilibrium (SMFE) is a pair of strategy $\pi\colon \mathcal{X} \to \Delta(\mathcal{A})$ and mean-field $z^* \in \Delta(\mathcal{X})$ which satisfy the following two properties:

1. *Sequential rationality:* For any other stationary strategy $\tilde{\pi}\colon \mathcal{X} \to \Delta(\mathcal{A})$,
$$V_{\pi,z^*}(x) \geq V_{\tilde{\pi},z^*}(x), \quad \forall x \in \mathcal{X}.$$
2. *Consistency:* The mean-field z^* is stationary under strategy π, i.e.,
$$z^* = \Phi(z^*, \pi).$$

□

For teams, we use the following refinement of mean-field team optimal strategy (Arabneydi and Mahajan, 2014) as the solution concept.

Definition 3 (Stationary mean-field team optimal strategy (SMFTO)) A strategy $\pi\colon \mathcal{X} \to \Delta(\mathcal{A})$ is stationary mean-field team optimal (SMFTO) if it satisfies the following property:

- *Optimality:* For any other stationary strategy $\tilde{\pi}\colon \mathcal{X} \to \Delta(\mathcal{A})$,
$$V_{\pi,z^*}(x) \geq V_{\tilde{\pi},\tilde{z}^*}(x), \quad \forall x \in \mathcal{X},$$
where z^* and $\tilde{z}^*$ are the stationary mean-field distributions corresponding to π and $\tilde{\pi}$, respectively, i.e., satisfy $z^* = \Phi(z^*, \pi)$ and $\tilde{z}^* = \Phi(\tilde{z}^*, \tilde{\pi})$.

□

Remark 1 The definitions of sequential rationality and optimality are different. Sequential rationality is defined with respect to the mean-field z^*; while considering the performance of an alternative strategy $\tilde{\pi}$ it is assumed that the mean-field does not change. In contrast, optimality is a property of a strategy; while considering the performance of an alternative strategy $\tilde{\pi}$, the mean-field approximation of the performance is with respect to the stationary mean-field corresponding to $\tilde{\pi}$. Thus, in general, SMFE and SMFTO are different. □

1.5 Locally optimal solutions

The notions of sequential rationality and optimality used in Definitions 2 and 3 are global concepts: they seek a property that holds for *all* other strategies $\tilde{\pi}\colon \mathcal{X} \to \Delta(\mathcal{A})$. As such, verifying sequential rationality or optimality suffers from the curse of dimensionality. One way to reduce the computational complexity is to look for *local* solution concepts, i.e., seek properties that hold for all strategies $\tilde{\pi}$ in a *local neighborhood* (with respect to some topology on the space of strategies) of π.

To make this idea precise, we assume that the strategies are parameterized by θ belonging to some closed convex subset Θ of the Euclidean space. A strategy parameterized by θ is denoted by π_θ. Assuming that the value function is differentiable with respect to the parameter θ, we get the following generalizations of SMFE and SMFTO.

Definition 4 (Local Stationary mean-field equilibrium (LSMFE)) A local stationary mean-field equilibrium (LSMFE) is a pair of strategy $\pi\colon \mathcal{X} \to \Delta(\mathcal{A})$ and mean-field $z^* \in \Delta(\mathcal{X})$ which satisfy the following two properties:

1. *Local sequential rationality:* For all $x \in \mathcal{X}$, we have $\partial V_{\pi_\theta,z^*}(x)/\partial\theta = 0$.
2. *Consistency:* $z^* = \Phi(z^*, \pi_\theta)$.

□

Definition 5 (Local stationary mean-field team optimal strategy (LSMFTO)) A strategy $\pi\colon \mathcal{X} \to \Delta(\mathcal{A})$ is local stationary mean-field team optimal (LSMFTO) if it satisfies the following property:

- *Optimality:* For all $x \in \mathcal{X}$, we have $dV_{\pi_\theta,z^*_\theta}(x)/d\theta = 0$, where z^*_θ is the stationary mean-field distribution corresponding to π_θ, i.e., satisfies $z^*_\theta = \Phi(z^*_\theta, \pi_\theta)$.

□

Remark 2 From the chain rule of derivatives, we have

$$\frac{dV_{\pi,z}(x)}{d\theta} = \frac{\partial V_{\pi,z}(x)}{\partial \pi} \frac{\partial \pi}{\partial \theta} + \frac{\partial V_{\pi,z}(x)}{\partial z} \frac{\partial z}{\partial \theta}.$$

The first term is equal to $\partial V_{\pi_\theta,z}(x)/\partial\theta$. In general, $\partial V_{\pi,z}(x)/\partial z \neq 0$ and $\partial z/\partial\theta \neq 0$. Thus, local optimality is not the same as local sequential rationality. This is also illustrated by the numerical results for Example 1 presented in Sec. 3. □

2 Reinforcement learning for E-MAS

There is a vast literature on multi-agent reinforcement learning (MARL) and we refer the reader to Shoham et al. (2003, 2007); Buşoniu et al. (2010) for an overview. There are some recent papers on deep RL (Foerster et al., 2016, 2017; Lowe et al., 2017; Palmer et al., 2017; Perolat et al., 2017), which consider agents running deep RL algorithms in strategic (and, therefore, non-stationary) environments. The focus of these papers is on the evolution of agents' behavior and not necessarily on the convergence to any pre-specified solution concept.

In this paper, we present an off-line reinforcement learning for E-MAS where the system dynamics are not known but a system simulator is available that returns the per-step rewards and the mean-field. For such models, we develop policy gradient based reinforcement learning algorithms that converge to LSMFE and LSMFTO solutions.

We assume that the initial state is distributed according to $\xi_0 \in \Delta(\mathcal{X})$. Given a policy π, define the mean-field approximation of an agent's performance as

$$J_{\pi,z^*} = \mathbb{E}_{X\sim\xi_0}[V_{\pi,z^*}(X)] = \sum_{x\in\mathcal{X}} V_{\pi,z^*}(x)\xi_0(x).$$

Then, we have the following.

Proposition 2 *Sequential rationality and optimality can be characterized in terms of J_{π,z^*}. In particular:*

1. *For any stationary mean-field $z^* \in \Delta(\mathcal{X})$, a strategy π_θ, $\theta \in \Theta$, is sequentially rational with respect to z^* if and only if for all initial distributions ξ_0, $\partial J_{\pi_\theta,z^*}/\partial\theta = 0$.*
2. *A strategy π_θ, $\theta \in \Theta$, is LSMFTO if and only if for all initial distributions ξ_0, $dJ_{\pi_\theta,z^*_\theta}/d\theta = 0$, where z^*_θ is the stationary mean-field corresponding to π_θ.* □

PROOF We prove the first part. The proof of the second part is similar. From the definition of J_{π,z^*} we have

$$\frac{\partial J_{\pi_\theta,z^*}}{\partial\theta} = \sum_{x\in\mathcal{X}} \frac{\partial V_{\pi_\theta,z^*}(x)}{\partial\theta}\xi_0(x). \tag{2}$$

Now suppose π_θ is locally sequential rational, which implies that $\partial V_{\pi_\theta,z^*}(x)/\partial\theta = 0$ for all $x \in \mathcal{X}$. Hence, by Eq. (2), we get that $\partial J_{\pi_\theta,z^*}/\partial\theta = 0$ for all initial distributions ξ_0.

Now suppose that $\partial J_{\pi_\theta,z^*}/\partial\theta = 0$ for all initial distributions ξ_0. Given an $x \in \mathcal{X}$, pick ξ_0 such that $\xi_0(x) = 1$ and $\xi_0(y) = 0$ for all $y \neq x$. Then, from Eq. (2), $\partial J_{\pi_\theta,z^*}/\partial\theta = 0$ implies that $\partial V_{\pi_\theta,z^*}(x)/\partial\theta = 0$. Since the choice of x was arbitrary, we have that $\partial V_{\pi_\theta,z^*}(x)/\partial\theta = 0$ for all $x \in \mathcal{X}$. Hence, π_θ is locally sequential rational. ■

Now we present reinforcement learning algorithms for both the game and the team setups.

2.1 RL for stationary MF games

For the game setup, we make the following assumption.

(A3) An agent-level system simulator is available which generates samples of the next state X^i_{t+1} and reward R^i_t for any value of current state X^i_t, current action A^i_t, and mean-field Z_t.

The key idea behind reinforcement learning for mean-field games is as follows. Suppose $G_{\theta,z}$ is an unbiased estimator of $\partial J_{\pi_\theta,z}/\partial\theta$. Then, we can start with an initial guess $\theta_0 \in \Theta$ and $z_0 \in \Delta(\mathcal{X})$ and at each step of the iteration, update the guess (θ_k, z_k) using two-timescale stochastic gradient ascent (Borkar, 1997):

$$z_{k+1} = z_k + \beta_k\big[\hat{\Phi}(z_k, \pi_{\theta_k}) - z_k\big], \tag{3a}$$

$$\theta_{k+1} = \big[\theta_k + \alpha_k G_{\theta_k,z_k}\big]_\Theta, \tag{3b}$$

where $[\,\cdot\,]_\Theta$ denotes projection on to the set Θ and $\hat{\Phi}(z,\pi)$ is an unbiased approximation of $\Phi(z,\pi)$ which is generated as follows: generate a mini-batch of m samples $(X^j, A^j, Y^j)_{j=1}^m$ where $X^j \sim z$, $A^j \sim \pi(\cdot|X^j)$, and $Y^j \sim P(\cdot|X^j, A^j, z\odot\pi)$ and set

$$\hat{\Phi}(z,\pi)(y) = \frac{1}{m}\sum_{j=1}^m \mathbb{1}\{Y^j = y\}.$$

The learning rates $\{\alpha_k, \beta_k\}_{k\geq 0}$ are chosen according to the standard conditions for two-timescale algorithms: $\sum\alpha_k = \infty$, $\sum\beta_k = \infty$, $\sum(\alpha_k^2 + \beta_k^2) < \infty$, and $\sum\beta_k/\alpha_k = 0$. Then, we have the following:

Proposition 3 *If the iterations* (3) *converge to a limit (θ^*, z^*) along any sample path, then (π_{θ^*}, z^*) is LSMFE.* □

PROOF Since $\{(z_k,\theta_k)\}_{k\geq 0}$ converges to (z^*,θ^*) we have that $\hat{\Phi}(z^*,\pi_{\theta^*}) - z^* = 0$ and $G_{\theta^*,z^*} = 0$. The first equality implies that z^* is the stationary under strategy π^* (and is, therefore consistent). The second inequality implies that $\partial J_{\pi_{\theta^*},z^*}/\partial\theta = 0$; thus, (π_{θ^*}, z^*) satisfies local sequential rationality. Hence (π_{θ^*}, z^*) is LSMFE. ■

General conditions for convergence of two-timescale stochastic approximation given in iteration (3) are presented in Borkar (1997); Leslie (2004). These conditions can be verified for specific models as illustrated by the next result.

Proposition 4 *Consider the model of Example 1. Suppose the strategies are parameterized by $\theta \in [0,1]$ such that*[3]

$$\pi_\theta(x) = \begin{cases} 0, & \text{if } x < \theta, \\ 1, & \text{if } x \geq \theta. \end{cases} \tag{4}$$

Then, iteration (3) *converges almost surely to a LSMFE (π^*, z^*).* □

The proof is omitted due to space.

Remark 3 In theory, two-timescale algorithms are nice because they are amenable to a proof of convergence. However, in practice, the convergence is slow and there are no good methods to adapt the learning rates. So, in practice, rather than running a two-scale algorithm, it is better to run a large but fixed number of iterations of variable running at the faster timescale for every iteration of variable running at the slower timescale. For iteration (3) this is equivalent to running multiple iterations of (3a) (with a fixed learning rate β) for every iteration of (3b). In the sequel, we run B iterations of (3b)

[3] It is shown in Huang and Ma (2016, 2017) that such a parameterization is without loss of optimality.

Algorithm 1: `MF_Update`

input : θ : Policy parameter, z_0 : Initial mean-field
ξ_0 : Initial state distribution
B : Iteration count, m : Batch size
output : z : Final mean-field
Let $\bar{z}_0 = z_0 \odot \pi_\theta$
for $j = 0 : m$ **do**
  Initialize $X_0^j \sim \xi_0$
  for $t = 0 : B$ **do**
    Sample $A_t^j \sim \pi_\theta(X_t^j)$, $X_{t+1}^j \sim P(\cdot|X_t^j, A_t^j, \bar{z}_0)$
for $x \in \mathcal{X}$ **do**
  $z(x) = \frac{1}{m} \sum_{j=1}^{m} \mathbb{1}\{X_{B+1}^j = x\}$
return z

Algorithm 2: Likelihood ratio based algorithm to compute LSMFE

input : θ_0 : Initial policy, z_0 : Initial mean-field
ξ_0 : Initial state distribution
K : Iteration count
B : Iterations for mean-field update
m : Batch size for mean-field update
output : (θ^*, z^*) : Estimated LSMFE solution
for *iterations* $k = 0 : K$ **do**
  $z_{k+1} = \texttt{MF\_Update}(\theta_k, z_k, \xi_0, B, m)$
  $G_{\theta_k, z_{k+1}} = \texttt{PolicyGradient}(\theta_k, \xi_0, z_{k+1})$
  $\theta_{k+1} = [\theta_k + \alpha_k G_{\theta_k, z_{k+1}}]_\Theta$
return θ_{K+1}

with $\beta_k = 1$, which is shown in Algorithm 1 and is equivalent to a particle based Monte Carlo computation of the generated mean-field of the system. □

To convert iteration (3) to a complete algorithm, we need an algorithm that computes an unbiased estimator $G_{\theta,z}$ for $\partial J_{\pi_\theta,z}/\partial\theta$ for a given z. Since z is fixed, $\partial J_{\pi_\theta,z}/\partial\theta$ may be computed using any of the standard policy gradient based approaches for reinforcement learning: likelihood ratio based gradient estimators (Sutton et al., 2000; Konda and Tsitsiklis, 2003) or simultaneous perturbation based gradient estimators (Spall, 1992; Maryak and Chin, 2008; Katkovnik and Kulchitsky, 1972; Bhatnagar et al., 2013).

2.1.1 Likelihood ratio based gradient estimation One approach to estimate the performance gradient is to use likelihood radio based estimates (Rubinstein, 1989; Glynn, 1990; Williams, 1992). Suppose the policy $\pi_\theta(X)$ is differentiable with respect to θ. For any time t, define the likelihood function $\Lambda_\theta^t = \nabla_\theta \log[\pi_\theta(A_t \mid X_t)]$, where with a slight abuse of notation $\pi_\theta(A_t|X_t)$ denotes the probability of choosing action A_t in state X_t under policy π_θ. Then from Williams (1992); Sutton et al. (2000); Baxter and Bartlett (2001) we know that:

$$\frac{\partial V_{\theta,z}(x)}{\partial\theta} = \mathbb{E}_{A_t\sim\pi_\theta(X_t)}\Big[\sum_{\sigma=0}^{\infty} \Lambda_\theta^\sigma V_{\pi_\theta,z}(X_\sigma) \;\Big|\; X_0 = x\Big].$$

Thus,

$$\frac{\partial J_{\theta,z}}{\partial\theta} = \mathbb{E}_{X\sim\xi_0}\left[\frac{\partial V_{\theta,z}(X)}{\partial\theta}\right].$$

An algorithm to compute LSMFE based on the likelihood ratio approach is given in Algorithm 2. The `PolicyGradient` function in Algorithm 2 can be obtained by an actor only method such as Monte Carlo (Sutton and Barto, 1998) or Renewal Monte Carlo (Subramanian and Mahajan, 2018), or using an actor critic method such as SARSA (Sutton and Barto, 1998). Additionally, variance reduction techniques such as subtracting a baseline or using mini-batch averaging may also be used.

Remark 4 As explained in Remark 3, instead of using the two-timescale gradient ascent, we use a B step rollout of (3a) with a constant learning rate of $\beta_k = 1$ in Algorithm 2. In addition, when computing the gradient, we use the updated value z_{k+1} while computing the gradient estimate $G_{\theta_k,z_{k+1}}$ instead of z_k as specified in (3b). This may be viewed as a Gauss-Seidel method for the two-timescale algorithm, which generally converges faster. □

2.1.2 Simultaneous perturbation based gradient estimation Another approach to estimate the performance gradient is to use simultaneous perturbation based methods (Spall, 1992; Maryak and Chin, 2008; Katkovnik and Kulchitsky, 1972; Bhatnagar et al., 2013). This approach is useful when the policy π_θ is not a differentiable function of its parameters θ. Now, given any distribution ξ_0, we can estimate $J_{\pi_\theta,z}$ using $V_{\pi_\theta,z}$ as:

$$J_{\pi_\theta,z} = \mathbb{E}_{X\sim\xi_0}[V_{\pi_\theta,z}(X)].$$

To generate the two-sided form of simultaneous perturbation based estimate, we generate two random parameters $\theta^+ = \theta + c\,\eta$ and $\theta^- = \theta - c\,\eta$, where η is a random variable with the same dimension as θ and c is a small constant. Let $\pi^+ = \pi_{\theta^+}$ and $\pi^- = \pi_{\theta^-}$. Then, the two-sided simultaneous perturbation estimate is given by

$$G_{\theta,z} = \frac{\eta}{2c}(J_{\pi^+,z} - J_{\pi^-,z})$$

When $\eta_i \sim$ Rademacher(± 1), the above method is called simultaneous perturbation stochastic approximation (SPSA) (Spall, 1992; Maryak and Chin, 2008); when $\eta_i \sim$ Normal$(0, I)$ it is called smoothed functional stochastic approximation (SFSA) (Katkovnik and Kulchitsky, 1972; Bhatnagar et al., 2013).

An algorithm to compute LSMFE using the simultaneous perturbation approach is given in Algorithm 3. As in the case of the likelihood ratio based approach, the `PolicyEvaluation` function in Algorithm 3 may be obtained by an actor only method such as Monte Carlo (Sutton and Barto, 1998) or Renewal Monte Carlo (Subramanian and Mahajan, 2018), or using an actor critic method such as SARSA (Sutton and Barto, 1998).

2.2 RL for stationary MF teams

For the team setup, we make the following assumption.

Algorithm 3: Simultaneous perturbation based algorithm to compute LSMFE

input : θ_0 : Initial policy, z_0 : Initial mean-field
ξ_0 : Initial state distribution
K : Iteration count, c : Perturbation size
B : Iterations for mean-field update
m : Batch size for mean-field update
output : (θ^*, z^*) : Estimated LSMFE solution
for *iterations* $k = 1 : K$ **do**
$z_{k+1} = \texttt{MF\_Update}(\theta_k, z_k, \xi_0, B, m)$
Let $\eta \sim \texttt{Rademacher}(\pm 1)$ or $\eta \sim \mathcal{N}(0,1)$
$\theta_k^+ = \theta_k + \eta\beta$ and $\theta_k^- = \theta_k - \eta\beta$.
$\widehat{J}_k^+ = \texttt{PolicyEvaluation}(\theta_k^+, \xi_0, z_{k+1})$
$\widehat{J}_k^- = \texttt{PolicyEvaluation}(\theta_k^-, \xi_0, z_{k+1})$
$G_{\theta_k, z_{k+1}} = \frac{\eta}{2c}(\widehat{J}_k^+ - \hat{J}_k^-)$
$\theta_{k+1} = [\theta_k + \alpha_k G_{\theta_k, z_{k+1}}]_\Theta$
return θ_{K+1}

(A4) A system level simulator is available which generates samples of the next system state $\mathbf{X}_{t+1}$ and system rewards $\mathbf{R}_t$ for any value of the current system state $\mathbf{X}_t$ and system actions $\mathbf{A}_t$.

The key idea behind reinforcement learning for mean-field teams is as follows. Suppose T_θ is an unbiased estimator for $dJ_{\pi_\theta, z_\theta}/d\theta$, where z_θ is the fixed point of $z = \Phi(z, \pi_\theta)$. Then, we can start with an initial guess $\theta_0 \in \Theta$, and at each step of the iteration, update the guess using stochastic gradient ascent:

$$\theta_{k+1} = \big[\theta_k + \alpha_k T_{\theta_k}\big]_\Theta, \tag{5}$$

where $\{\alpha_k\}_{k\geq 0}$ is a sequence of learning rates that satisfies the standard conditions: $\sum \alpha_k = \infty$ and $\sum \alpha_k^2 < \infty$. Then, we have the following:

Proposition 5 *If the iteration* (5) *converges to a limit* θ^* *along any sample path, then* π_{θ^*} *is LSMFTO.* □

The proof is similar to Proposition 3 and is omitted.

General conditions for convergence of stochastic approximation given in iteration (5) are presented in Borkar (2008); Kushner and Yin (2003). These conditions can be verified for specific models.

To convert iteration (5) to a complete algorithm, we need an algorithm that computes an unbiased estimator $T_{\theta,z}$ of $dJ_{\pi_\theta, z_\theta}/d\theta$. Likelihood ratio based gradient estimators do not work in this case because, in order to compute $d\mathbb{E}[r(X_t^i, A_t^i, z_\theta)]/d\theta$, we need to compute $dz_\theta/d\theta$ and there are no good methods to do so. There are some results in the literature on the sensitivity of the stationary distribution of a Markov chain to its transition probability (e.g., Funderlic and Meyer (1986) and references therein), but these results only provide loose bounds on $dz_\theta/d\theta$. However, it is possible to adapt simultaneous perturbation based methods to generate estimators of $dJ_{\pi_\theta, z_\theta}/d\theta$. We present one such estimator in the next section.

Algorithm 4: `Stationary_MF`

input : θ : Policy parameter, ξ_0 : Initial state distribution
B : Iteration count, m : Batch size
output : z : Final mean-field
for $j = 1 : m$ **do**
for $i \in N$ **do**
Sample $X_0^{i,j} \sim \xi_0$
for $t = 0 : B$ **do**
for $i \in N$ **do**
Sample $A_t^{i,j} \sim \pi(X_t^{i,j})$
Sample $\mathbf{X}_{t+1}^j \sim P(\cdot|\mathbf{X}_t^j, \mathbf{A}_t^j)$
for $x \in \mathcal{X}$ **do**
$z^j(x) = \frac{1}{n}\sum_{i\in N} \mathbb{1}\{X_{B+1}^{i,j} = x\}$
$z = \frac{1}{m}\sum_{j=1}^m z^j$
return z

Algorithm 5: Simultaneous perturbation based algorithm to compute LSMFTO

input : θ_0 : Initial policy
ξ_0 : Initial state distribution
K : Iteration count, c : Perturbation size
B : Iterations for mean-field update
m : Batch size for mean-field update
output : θ^* : Estimated LSMFTO solution
for *iterations* $k = 1 : K$ **do**
Let $\eta \sim \texttt{Rademacher}(\pm 1)$ or $\eta \sim \mathcal{N}(0,1)$
$\theta_k^+ = \theta_k + \eta\beta$ and $\theta_k^- = \theta_k - \eta\beta$.
$z_k^+ = \texttt{Stationary\_MF}(\theta^+, \xi_0, B, m)$
$z_k^- = \texttt{Stationary\_MF}(\theta^-, \xi_0, B, m)$
$\widehat{J}_k^+ = \texttt{PolicyEvaluation}(\theta_k^+, \xi_0, z_k^+)$
$\widehat{J}_k^- = \texttt{PolicyEvaluation}(\theta_k^-, \xi_0, z_k^-)$
$T_{\theta_k} = \frac{\eta}{2c}(\widehat{J}_k^+ - \hat{J}_k^-)$
$\theta_{k+1} = [\theta_k + \alpha_k T_{\theta_k}]_\Theta$
return θ_{K+1}

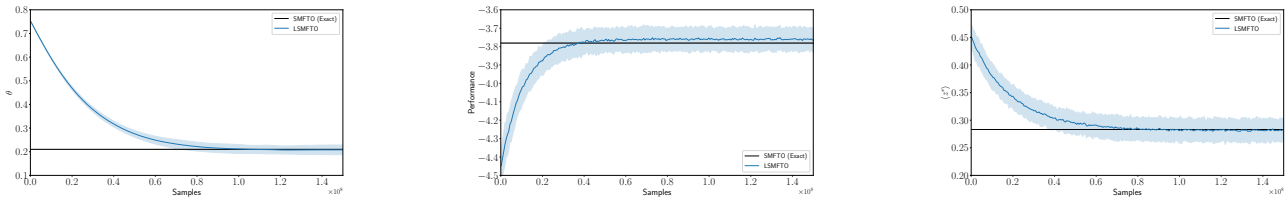
2.3 Simultaneous perturbation based gradient estimation

We first consider estimating z_θ for a given π_θ. Suppose the system is such that if it starts from any initial distribution and each agent follows strategy π_θ, the mean-field converges to the stationary distribution z_θ. Then, we can estimate z_θ by simply running the system for a sufficiently long time. An algorithm based on this idea is shown in Algorithm 4.

Then, to generate the two-sided simultaneous perturbation based estimate of $dJ_{\pi_\theta, z_\theta}/d\theta$, we generate two random parameters $\theta^+ = \theta + c\,\eta$ and $\theta^- = \theta - c\,\eta$, where η and c are as in Sec. 2.1.2. Let $\pi^+ = \pi_{\theta^+}$ and $\pi^- = \pi_{\theta^-}$. Generate $z^+ = z_{\pi^+}$ and $z^- = z_{\pi^-}$ using Algorithm 4. Then, the two-sided simultaneous perturbation estimate is given by

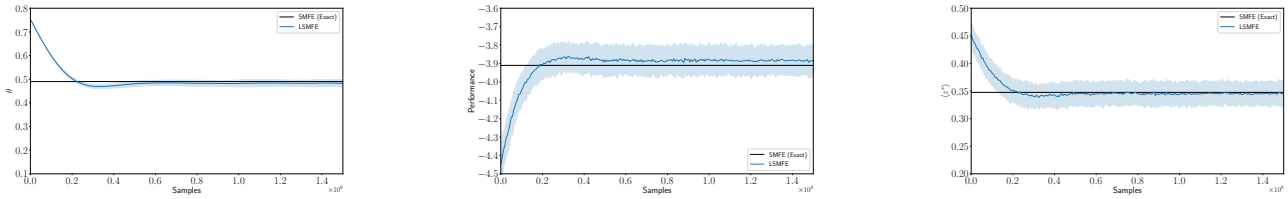
$$T_\theta = \frac{\eta}{2c}\big(J_{\pi^+, z^+} - J_{\pi^-, z^-}\big).$$

An algorithm to compute LSMFTO using simultaneous



(a) Parameter θ versus samples (b) Performance versus samples (c) $\langle z^* \rangle$ versus samples

Figure 1: RL algorithm converging to LSMFTO for Example 1. The solid line shows the mean value and the shaded region shows the $\pm$ two standard deviation region over 100 runs.



(a) Parameter θ versus samples (b) Performance versus samples (c) $\langle z^* \rangle$ versus samples

Figure 2: RL algorithm converging to LSMFE for Example 1. The solid line shows the mean value and the shaded region shows the $\pm$ two standard deviation region over 100 runs.

perturbation approach is given in Algorithm 5. As was the case for Algorithm 3, the `PolicyEvaluation` function in Algorithm 5 may be obtained by an actor only method such as Monte Carlo (Sutton and Barto, 1998) or Renewal Monte Carlo (Subramanian and Mahajan, 2018), or using an actor critic method such as SARSA (Sutton and Barto, 1998).

Remark 5 There is a subtle difference between Algorithms 1 and 4. Algorithm 1 is computing B rollouts of the iteration $z_{k+1} = \Phi(z_k, \pi_\theta)$ while Algorithm 4 is computing the fixed point of $z = \Phi(z, \pi_\theta)$ by starting at an arbitrary distribution and rolling out the system dynamics for B iterations. □

3 Numerical Experiments

In this section, we present numerical results for the model of Example 1 with $n = 1000$ agents where $f = \text{Uniform}[0, 1]$, $k = 0.2$, and $\lambda = 0.5$. We discretize the continuous state space $\mathcal{X} = [0, 1]$ into 101 uniformly sized cells $\{0, 0.01, \dots, 1\}$, use a discount factor $\gamma = 0.9$, and consider strategies of the form (4) parameterized by $\theta \in [0, 1]$.

The parameterized strategies of the form (4) are not differentiable with respect to θ, so we estimate the gradient using simultaneous perturbation methods (Algorithms 3 and 5). In both algorithms, policy evaluation is done using Monte Carlo with m trajectories of length H. For both cases, the parameters of the algorithms are chosen as follows: $\theta_0 = 0.9$, $z_0 = \xi_0 = \text{Uniform}(\mathcal{X})$, $K = 200$, $c = 0.01$, $\eta \sim \text{Rademacher}(\pm 1)$, $B = 200$, $H = 200$, $m = 1000$, and choose the learning rate using ADAM (Kingma and Ba, 2014) with the α parameter of ADAM equal to 0.01 and all other ADAM parameters equal to their default values. For both setups, we repeat the experiment 100 times.

The values of the parameter θ, the performance J, and average $\langle z^* \rangle$ of the stationary mean field versus samples for both the team and the game setups are shown in Figures 1 and 2. For comparison, the exact SMFTO and SMFE solutions are also plotted. The SMFTO solution is computed by a brute force search over all $\theta \in [0, 1]$. The SMFE solution is computed using the method described in Huang and Ma (2016). These exact solutions are also shown in Figures 1 and 2. The plots show that the convergence of the RL algorithm is fairly fast (less than 10^8 samples or 17 iterations[4]) and the variation across multiple runs is small. It is worth highlighting that the game and team solutions differ.

4 Conclusion

We present off-line reinforcement learning algorithms for exchangeable multi-agent systems. Exchangeability implies that the dynamics and the rewards are coupled only through the mean-field; evaluating performance under the mean-field approximation (assuming that the mean-field is stationary and not influenced by a single agent) decouples the n-agent problem (where n is large) into a single agent problem.

For games, we present a two-timescale gradient ascent algorithm that converges to LSMFE. For teams, we present a standard gradient ascent algorithm that converges to LSMFTO. The results do not explicitly depend on the number of agents. In the planning literature there are various results that state that the finite population solution is $\mathcal{O}(1/\sqrt{n})$-Nash or $\mathcal{O}(1/n)$-team optimal. Thus, the results can be used in multi-agent systems with moderate number of agents.

[4]Note that each iteration has $mB + 2mH = 6 \times 10^5$ samples.

References

Adlakha, S.; Johari, R.; and Weintraub, G. Y. 2015. Equilibria of dynamic games with many players: Existence, approximation, and market structure. *Journal of Economic Theory* 156:269–316.

Arabneydi, J., and Mahajan, A. 2014. Team optimal control of coupled subsystems with mean-field sharing. In *CDC*.

Arabneydi, J., and Mahajan, A. 2016. Linear quadratic mean field teams: Optimal and approximately optimal decentralized solutions. arXiv:1609.00056v2.

Baxter, J., and Bartlett, P. L. 2001. Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research* 15:319–350.

Bergin, J., and Bernhardt, D. 1995. Anonymous sequential games: Existence and characterization of equilibria. *Economic Theory* 5(3):461–489.

Bhatnagar, S.; Prasad, H.; and Prashanth, L. 2013. *Stochastic Recursive Algorithms for Optimization: Simultaneous Perturbation Methods*, volume 434. Springer.

Borkar, V. S. 1997. Stochastic approximation with two time scales. *Systems & Control Letters* 29(5):291–294.

Borkar, V. S. 2008. *Stochastic Approximation: A Dynamical Systems Viewpoint*. Cambridge University Press.

Buşoniu, L.; Babuška, R.; and De Schutter, B. 2010. Multi-agent reinforcement learning: An overview. In *Innovations in multi-agent systems and applications*. Springer.

Diaconis, P. 1988. Recent progress on de Finetti's notions of exchangeability. *Bayesian statistics* 3:111–125.

Foerster, J.; Assael, Y. M.; et al. 2016. Learning to communicate with deep multi-agent reinforcement learning. In *NIPS*, 2137–2145.

Foerster, J.; Nardelli, N.; et al. 2017. Stabilising experience replay for deep multi-agent reinforcement learning. *arXiv:1702.08887*.

Funderlic, R., and Meyer, C. 1986. Sensitivity of the stationary distribution vector for an ergodic markov chain. *Linear Algebra and its Applications* 76:1 – 17.

Glynn, P. 1990. Likelihood ratio gradient estimation for stochastic systems. *Comm. ACM* 33:75–84.

Huang, M., and Ma, Y. 2016. Mean field stochastic games: Monotone costs and threshold policies. In *CDC*.

Huang, M., and Ma, Y. 2017. Mean field stochastic games with binary actions: Stationary threshold policies. In *CDC*.

Huang, M.; Malhamé, R. P.; and Caines, P. E. 2006. Large population stochastic dynamic games: closed-loop Mckean-Vlasov systems and the Nash certainty equivalence principle. *Comm. in Information & Systems* 6(3):221–252.

Hüttenrauch, M.; Šošić, A.; and Neumann, G. 2018. Deep reinforcement learning for swarm systems. *arXiv:1807.06613*.

Jovanovic, B., and Rosenthal, R. W. 1988. Anonymous sequential games. *J. of Math. Economics* 17(1):77 – 87.

Katkovnik, V., and Kulchitsky, Y. 1972. Convergence of a class of random search algorithms. *Automation and Remote Control* 33(8):1321–1326.

Kingma, D., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv:1412.6980*.

Konda, V. R., and Tsitsiklis, J. N. 2003. On actor-critic algorithms. *SIAM J. on Cont. and Optim.* 42(4):1143–1166.

Kushner, H., and Yin, G. G. 2003. *Stochastic approximation and recursive algorithms and applications*. Springer.

Lasry, J.-M., and Lions, P.-L. 2007. Mean field games. *Japanese journal of mathematics* 2(1):229–260.

Leslie, D. S. 2004. *Reinforcement learning in games*. Ph.D. Dissertation, The University of Bristol.

Lowe, R.; Wu, Y.; et al. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. *arXiv:1706.02275*.

Maryak, J. L., and Chin, D. C. 2008. Global random optimization by simultaneous perturbation stochastic approximation. *IEEE Trans. Autom. Control* 53(3):780–783.

Maynard Smith, J., and Price, G. 1973. The logic of animal conflict. *Nature* 246(5427):15–18.

Nash, J. 1951. Non-cooperative games. *Annals of Mathematics* 54(2):286–295.

Nguyen, D. T.; Kumar, A.; and Lau, H. C. 2017a. Collective multiagent sequential decision making under uncertainty. In *AAAI*.

Nguyen, D.; Kumar, A.; and Lau, H. 2017b. Policy gradient with value function approximation for collective multiagent planning. In *NIPS*.

Palmer, G.; Tuyls, K.; Bloembergen, D.; and Savani, R. 2017. Lenient multi-agent deep reinforcement learning. *arXiv:1707.04402*.

Perolat, J.; Leibo, J. Z.; et al. 2017. A multi-agent reinforcement learning model of common-pool resource appropriation. *arXiv:1707.06600*.

Rubinstein, R. Y. 1989. Sensitivity analysis and performance extrapolation for computer simulation models. *Operations Research* 37(1):72–81.

Shoham, Y.; Powers, R.; and Grenager, T. 2003. Multi-agent reinforcement learning: a critical survey. Technical report, Stanford University.

Shoham, Y.; Powers, R.; and Grenager, T. 2007. If multi-agent learning is the answer, what is the question? *Artificial Intelligence* 171(7):365–377.

Spall, J. C. 1992. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Trans. Autom. Control* 37(3):332–341.

Subramanian, J., and Mahajan, A. 2018. Renewal Monte Carlo: Renewal theory based reinforcement learning. *ArXiv:1804.01116*.

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement learning: An introduction*. MIT Press.

Sutton, R. S.; McAllester, D. A.; et al. 2000. Policy gradient methods for reinforcement learning with function approximation. In *NIPS*, 1057–1063.

Weintraub, G. Y.; Benkard, C. L.; and Van Roy, B. 2005. Oblivious equilibrium: A mean field approximation for large-scale dynamic games. In *NIPS*, 1489–1496.

Weintraub, G. Y.; Benkard, C. L.; and Van Roy, B. 2008. Markov perfect industry dynamics with many firms. *Econometrica* 76(6):1375–1411.

Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8(3-4):229–256.