An Independent Learning Algorithm for a Class of Symmetric Stochastic Games

Bora Yongacoglu
Department of Mathematics and Statistics
Queen’s University
Kingston, ON K7L 3N6
1bmy@queensu.ca

Gürdal Arslan
Department of Electrical Engineering
University of Hawaii at Manoa
Honolulu, HI 96822
gurdal@hawaii.edu

Serdar Yüksel
Department of Mathematics and Statistics
Queen’s University
Kingston, ON K7L 3N6
yuksel@queensu.ca

Abstract

In multi-agent reinforcement learning, independent learners are those that do not access the action selections of other learning agents in the system. This paper investigates the feasibility of using independent learners to find approximate equilibrium policies in non-episodic, discounted stochastic games. We define a property, here called the $\epsilon$-revision paths property, and prove that a class of games exhibiting symmetry among the players has this property for any $\epsilon \geq 0$. Building on this result, we present an independent learning algorithm that comes with high probability guarantees of approximate equilibrium in this class of games. This guarantee is made assuming symmetry alone, without additional assumptions such as a zero sum, team, or potential game structure.

1 Introduction

Reinforcement learning (RL) algorithms use experience and feedback information to improve one’s performance in a control task. In recent years, the field of RL has advanced tremendously both in terms of fundamental theoretical contributions and successful applications. With improvements in computer hardware, these advances have lead to the deployment of RL algorithms in large scale engineering systems in which many learning agents act, observe, and learn in a shared environment. Multi-agent reinforcement learning (MARL) is the study of emergent behaviour in such complex, strategic environments, and is one of the important frontiers in modern artificial intelligence and automatic control research.

The literature on MARL is relatively small when compared to that of single-agent RL, and this owes largely to the inherent challenges of multi-agent settings: decentralized information and non-stationarity. The first such challenge is that some relevant information will be unavailable to some of the players. This may occur due to strategic considerations, as competing agents may wish to hide their actions or knowledge from their rivals, or it may occur simply because of obstacles in communicating, observing, or storing large quantities of information in decentralized systems.

The second challenge inherent to MARL comes from the non-stationarity of the environment from the point of view of any individual agent. As one agent learns how to improve its performance, it will alter its behaviour. This can have a destabilizing effect on the learning processes of the remaining
agents, who may change their policies in response to outdated strategies. Notably, this issue arises when one tries to apply single-agent RL algorithms—which typically rely on state-action value estimates or gradient estimates that are made using historical data—in multi-agent settings. A number of studies have reported non-convergent play when single-agent algorithms using local information are employed (without modification) in multi-agent settings, e.g. [2], [3].

Designing decentralized learning algorithms with desirable convergence properties is a task of great practical importance that lies at the intersection of the two challenges above. The notion of decentralization considered in this paper involves agents that observe a global state variable but do not observe the actions of other agents. Learning algorithms suitable for this information structure are called independent learners in the machine learning literature [3, 4, 5, 6, 7, 8]; they have also been called payoff-based and radically uncoupled in the control and game theory literatures, respectively, [9] [10] [11].

In this paper, our goal is to provide a MARL algorithm that is suitable for independent learners in a complex system, requires little coordination among agents, and comes with provable guarantees for its long-run performance. For our theoretical framework, we consider many player symmetric stochastic games with a discounted cost criterion. In this setting, we propose an independent learning algorithm and prove that, under mild conditions, this algorithm leads to approximate equilibrium policies in self-play.

The algorithm presented in this paper builds on the exploration phase technique of [12] for policy evaluation, but differs considerably in how players update their policies and explore their policy spaces. Here, players discretize their policy space with a uniform quantizer and use an “ε-satisficing” rule to explore this quantized set, occasionally using random search when unsatisfied. Our proof of convergence exploits a structural property enjoyed by all symmetric games (defined formally in §2.2), and this allows us to avoid assuming additional structure in the game, such as the existence of a potential function or zero-sum costs.

Contributions:

(i) For any stochastic game and ε ≥ 0, we define ε-revision paths and a related ε-revision paths property;

(ii) In Theorem[1] we prove that all symmetric games (as defined in Definition[3]) have the ε-revision paths property, for all ε ≥ 0;

(iii) In Algorithm[4] we provide an algorithm for playing symmetric stochastic games and, in Theorem[2] we prove that self-play drives the baseline policy to ε-equilibrium.

1.1 Related Work

The study of learning in games, beginning with Brown’s fictitious play algorithm [13] and Robinson’s analysis thereof [14], is nearly as old as game theory itself. There is a large literature on fictitious play and its variants, with most works in this line considering a different information structure than the decentralized one studied here. The bulk of work on fictitious play focuses on settings with perfect monitoring of the actions of other players, but some recent works consider various decentralized information structures, e.g. [15] [16].

A number of empirical works in the 1990s studied the behaviour resulting from independent RL agents coexisting in various shared environments, e.g. [2, 17, 3]. Contemporaneously, [18] popularized using stochastic games as the framework for MARL. Several joint action learners (learners that require access to the past actions of all other agents) were then proposed for playing stochastic games and proven to converge to equilibrium under various assumptions. A representative sampling of this stream of algorithms includes the Minimax Q-learning algorithm of [18], the Nash Q-learning algorithm of [19], and the Friend-or-Foe Q-learning algorithm of [20].

Early work on independent learners includes the following: [3], which popularized the terminology of joint action learners and independent learners and stated conjectures; [21], which presented an independent learner for teams with deterministic state transitions and cost realizations and proved its convergence to optimality in that (restricted) setting; and [22], which proposed the WoLF-Policy Hill Climbing algorithm for general-sum stochastic games and conducted simulation studies.
Due in part to the challenges posed by non-stationarity and decentralized information, most contributions to the literature on independent learners focused either on the stateless case of repeated games and produced formal guarantees, such as the works of [11, 23, 24, 9, 10, 25], or otherwise studied the stateful setting and presented only empirical results, such as the works [26, 6, 8].

More recently, there has been a number of papers that study independent learners for games with non-trivial state dynamics while still presenting rigorous guarantees. These include [7] and [27], which study two-player, zero-sum games, and [28] and [12], which study teams, potential games, and their generalizations. In each of those works, the algorithm studied takes advantage of structural properties of the games considered.

In this paper, we study symmetric stochastic games with $N \geq 2$ players. This class of games is distinct from the classes listed above: an $N$-player symmetric stochastic game need not be a zero-sum game, admit a potential function, or be weakly acyclic under best-replies. As such, existing algorithms such as those studied in [7, 27, 28] and [12] do not come with convergence guarantees when applied to symmetric stochastic games. Nevertheless, games in this class exhibit structural properties that do not require further assumptions on the cost function. This paper shows how to exploit this structure.

### Organization

The remainder of the paper is organized as follows: Section 2 describes the stochastic games model and covers background results on Q-learning and value learning; Section 3 defines $\epsilon$-revision paths and proves that symmetric games have the $\epsilon$-revision paths property for all $\epsilon \geq 0$. A number of approximation results, concerning quantization of the policy sets and perturbations of policies, are presented in Section 4. The architecture and paradigm of the main algorithm is presented in Section 5. Our algorithm and main theorem is presented in Section 6. A simulation study is summarized in Section 7. Some discussion about the limitations and consequences of this work is provided in the technical appendices.

### Notation

$\mathbb{R}$ denotes the real numbers, $\mathbb{Z}_{\geq 0}$ and $\mathbb{N}$ denote the nonnegative and positive integers, respectively. $\Pr(\cdot)$ and $E(\cdot)$ denote the probability and the expectation, respectively. For a finite set $S$, $\mathcal{P}(S)$ denotes the set of probability distributions over $S$. For finite sets $S, S'$, we let $\mathcal{P}(S'|S)$ denote the set of stochastic kernels on $S'$ given $S$. An element $T \in \mathcal{P}(S'|S)$ is a collection of probability distributions on $S'$, with one distribution for each $s \in S$, and we write $T(s|s)$ for $s \in S$ to make this distributional dependence on $s$ explicit. We write $Y \sim f$ to denote that the random variable $Y$ has distribution $f$. If the distribution of $Y$ is a mixture of other distributions, say with mixture components $f_i$ and weights $p_i$ for $1 \leq i \leq n$, we write $Y \sim \sum_{i=1}^n p_i f_i$. The Dirac distribution concentrated at $x \in \mathbb{R}$ is denoted $\delta_x$. For a finite set $S$, $\text{Unif}(S)$ denotes the uniform distribution over $S$ and $2^S$ denotes the set of subsets of $S$.

### 2 Model

#### 2.1 Stochastic games with discounted costs

A finite, discounted stochastic game $G$ is described by the list

$$G = (\mathcal{N}, \mathcal{X}, \{U^i\}_{i \in \mathcal{N}}, \{e^i\}_{i \in \mathcal{N}}, \{\beta^i\}_{i \in \mathcal{N}}, P, \nu_0).$$

The components of $G$ are the following: $\mathcal{N}$ is a finite set of $N \in \mathbb{N}$ players/agents. $\mathcal{X}$ is a finite set of states. For agent $i \in \mathcal{N}, U^i$ is a finite set of actions, and we write $U := \bigtimes_{i \in \mathcal{N}} U^i$. An element $u \in U$ is called a joint action. For agent $i, e^i : \mathcal{X} \times U \to \mathbb{R}$ is a stage cost function, and $\beta^i \in [0, 1)$ is a discount factor. A random initial state $x_0 \in \mathcal{X}$ is given by $x_0 \sim \nu_0$. $P$ is a Markov transition kernel, which describes state transition probabilities through equation (2), below.

At time $t \in \mathbb{Z}_{\geq 0}$, the state variable is denoted by $x_t \in \mathcal{X}$, and the action selected by agent $i \in \mathcal{N}$ is denoted by $u^i_t \in \mathcal{U}^i$, while the joint action is denoted by $u_t = (u^i_t)_{i \in \mathcal{N}} \in U$. For all $t \in \mathbb{Z}_{\geq 0}$, the
A policy for agent $i \in \mathcal{N}$ is a rule for selecting a sequence of actions $(u_t^i)_{t \geq 0}$ based on information that is locally available at the time of each decision. The action $u_t^i \in U^i$ is chosen according to a (possibly random) function of agent $i$’s observations up to time $t$. In this paper, we focus on independent learners, which are agents that do not use/cannot access the complete joint action $u_t$, for any time $t$. Instead, at time $t$ an independent learner may use only the history of observations of states $(x_0, \ldots, x_t)$, its own local actions $(u_0^i, \ldots, u_{t-1}^i)$, and its numerical cost realizations. Independent learners are contrasted with joint action learners, which are learners that have access also to the actions of other agents.

The set of stationary policies for player $i$ is identified with the set $\Delta^i := \mathcal{P}(U^i|\mathcal{X})$ of probability distributions on $U^i$ given $\mathcal{X}$. When agent $i$ uses policy $\gamma^i \in \Delta^i$, it selects its action $u_t^i \sim \gamma^i(\cdot|x_t)$. For ease of notation, we denote the collected policies of all agents $j \neq i$ by using $-i$ in the agent index and we use boldface characters to refer to joint objects, e.g. $\gamma^{-i} = (\gamma^j)_{j \neq i}$. We let $\Delta^{-i} := \times_{j \neq i} \Delta^j$ and $\Delta := \times \Delta^i$. Using these conventions, we can re-write a joint policy $(\gamma^1, \ldots, \gamma^N)$ as $(\gamma^i, \gamma^{-i})$, a joint action can be re-written as $u = (u^i, u^{-i})$, and so on.

Given a joint policy $\gamma$, we use $\Pr^{\gamma}$ to denote the resulting probability measure on trajectories $\{(x_t, u_t)\}_{t \geq 0}$ and we use $E^{\gamma}$ to denote the associated expectation. The objective of agent $i \in \mathcal{N}$ is to find a policy that minimizes the expectation of its series of discounted costs, given by

$$J^i_x(\gamma^i, \gamma^{-i}) = E^{\gamma} \left[ \sum_{t \geq 0} (\beta^t)^{\epsilon} c^i(x_t, u_t^i, u_t^{-i}) \mid x_0 = x \right]$$

for all $x \in \mathcal{X}$. Note that agent $i$ controls only its own policy, $\gamma^i$, but its cost is affected by the actions of the remaining agents. Since agents have possibly different cost functions, we adopt the notion of equilibrium to represent those policies that are person-by-person optimal and stationary.

**Definition 1** Let $i \in \mathcal{N}$, $\epsilon \geq 0$. For $\gamma^{-i} \in \Delta^{-i}$, a policy $\gamma^{*i} \in \Delta^i$ is called an $\epsilon$-best-response to $\gamma^{-i}$ if

$$J^i_x(\gamma^{*i}, \gamma^{-i}) \leq \inf_{\pi^i \in \Delta^i} J^i_x(\pi^i, \gamma^{-i}) + \epsilon, \quad \forall x \in \mathcal{X}. \quad (4)$$

**Definition 2** A joint policy $\gamma^* = (\gamma^{*i})_{i \in \mathcal{N}} \in \Delta$ constitutes a (Markov perfect) $\epsilon$-equilibrium if $\gamma^{*i}$ is an $\epsilon$-best-response to $\gamma^{*-i}$ for each agent $i$.

For the special case of $\epsilon = 0$, a 0-best-response is simply called a best-response and a 0-equilibrium is called an equilibrium. Let $\Delta^\epsilon_{eq}$ denote the set of $\epsilon$-equilibrium policies, for $\epsilon \geq 0$. For any stochastic game, we have $\Delta^0_{eq} \neq \emptyset$ (see [29]), and since a 0-best-response is, a fortiori, an $\epsilon$-best-response, this implies that $\Delta^\epsilon_{eq} \neq \emptyset$.

### 2.2 Symmetric Games

In some applications, the strategic environment being modelled exhibits symmetry in the agents. To model such settings, we define a class of symmetric games with the following properties: (1) each agent has access to the same set of actions; (2) the state dynamics depend only on the profile of actions taken by all players, without special dependence on the identities of the agents. That is, permuting the agents’ actions in a joint action leaves the conditional probabilities for the next state unchanged; (3) such a permutation results in a corresponding permutation of costs incurred. We formalize and clarify these points in the definition below. First, we introduce additional notation: if $U^i = U^j$ for all $i, j \in \mathcal{N}$, given a permutation $\sigma : \mathcal{N} \to \mathcal{N}$ and joint action $a = (a^i)_{i \in \mathcal{N}}$, we define $\sigma(a) \in U$ to be the joint action in which $i$’s component is given by $a^{\sigma(i)}$. That is, player $i$’s action in $\sigma(a)$ is given by player $\sigma(i)$’s action in $a$.

\footnote{Agent $i$ does not know the function $c^i$ but observes the scalars $c^i(x_0, u_0^{-i}), \ldots, c^i(x_{t-1}, u_{t-1}^{-i}, u_{t-1}^i).$}
Definition 3 A discounted stochastic game $G$ is called symmetric if the following holds:

- $\mathbb{U}^i = \mathbb{U}^j$ and $\beta^i = \beta^j$ for any $i, j \in \mathcal{N}$;
- For any $i \in \mathcal{N}$, permutation $\sigma$ on $\mathcal{N}$, and $a \in \mathbb{U}$, we have
  \[ c^i(x, \sigma(a)) = c^{\sigma(i)}(x, a), \quad \text{and} \quad P(\cdot|x, a) = P(\cdot|x, \sigma(a)) \, . \]

Observe the following useful fact about symmetric games:

Lemma 1 Let $G$ be a symmetric game and let $\gamma$ be a joint policy. For $i, j \in \mathcal{N}$, if $\gamma^i = \gamma^j$, then
  \[ i \text{ is } \epsilon\text{-best-responding to } \gamma^{-i} \iff j \text{ is } \epsilon\text{-best-responding to } \gamma^{-j} \]

Proof: By symmetry, $\gamma^{-i} = \gamma^{-j}$ and the proof is immediate. \hfill \Box

2.3 Learning in MDPs

In online independent learning, pertinent information for policy updating is not available to the players. Player $i$ does not know the policy used by players $-i$, the value of its current policy against those of the other players, or whether its current policy is an $\epsilon$-best-response. We now recall some background on Q-learning and summarize how it can be used to address these uncertainties.

Markov decision processes (MDPs) can be viewed as a stochastic game with one player, i.e. $|\mathcal{N}| = 1$. In standard Q-learning [30], a single agent interacts with its MDP environment using some policy and maintains a vector of Q-factors, the $i$th iterate denoted $Q^i_t \in \mathbb{R}^{\mathbb{X} \times \mathbb{U}}$. Upon selecting action $u_t$ at state $x_t$ and observing the subsequent state $x_{t+1} \sim P(\cdot|x_t, u_t)$ and cost $c(x_t, u_t)$, the Q-learning agent updates its Q-factors as follows:

\[ Q_{t+1}(x_t, u_t) = Q_t(x_t, u_t) + \theta_t(x_t, u_t) \left( c(x_t, u_t) + \beta \min_{a \in \mathbb{U}} Q_t(x_{t+1}, a) - Q_t(x_t, u_t) \right) \tag{5} \]

where $\theta_t(x_t, u_t) \in [0, 1]$ is a random step-size parameter and $Q_{t+1}(s, a) = Q_t(s, a)$ for all $(s, a) \neq (x_t, u_t)$.

Under mild conditions, $Q_t \rightarrow Q^*$ almost surely as $t \rightarrow \infty$, where $Q^* \in \mathbb{R}^{\mathbb{X} \times \mathbb{U}}$ is called the optimal Q-factor. [31], [32]. The value $Q^*(x, u)$ represents the expected discounted cost-to-go from the initial state $x$, assuming that the agent initially chooses action $u$ and follows an optimal policy thereafter. The vector $Q^*$ can be used to construct an optimal policy $\pi^*$ by selecting

\[ \pi^*(x) \in \left\{ u \in \mathbb{U} : Q^*(x, u) = \min_{a \in \mathbb{U}} Q^*(x, a) \right\} \, . \]

2.4 Learning in Stochastic Games

In the single-agent literature, the MDP is fixed and the $Q^*$ notation is used, but in principle one could introduce a notation to specify the relevant MDP. Returning to the game setting, if all agents except $i$ follow a stationary policy $\gamma^{-i} \in \Delta^{-i}$, agent $i$ faces an environment that is equivalent to an MDP that depends on $\gamma^{-i}$. We denote agent $i$'s $t$th Q-factor iterate by $Q^i_t$ and $i$'s optimal Q-factors when playing against $\gamma^{-i}$ by $Q^{*i}_{\gamma^{-i}} \in \mathbb{R}^{\mathbb{X} \times \mathbb{U}}$. With this notation, $Q^{*i}_{\gamma^{-i}}(x, u')$ represents agent $i$'s expected discounted cost-to-go from the initial state $x$ assuming that agent $i$ initially chooses $u'$ and uses an optimal policy thereafter while the other agents use $\gamma^{-i}$, a fixed stationary policy. We note that an optimal policy for $i$ is guaranteed to exist since $i$ faces a finite, discounted MDP, and that any optimal policy for $i$ in this MDP is a 0-best-response to $\gamma^{-i}$ in the underlying game $G$. More generally, we have the following fact, which will be leveraged in the design of our main algorithm in Section 6. For any $i \in \mathcal{N}$, $\gamma^{-i} \in \Delta^{-i}$,

\[ \pi^i \text{ is an } \epsilon\text{-best-response to } \gamma^{-i} \iff J^i_\epsilon(\pi^i, \gamma^{-i}) \leq \min_{u' \in \mathbb{U}} Q^{*i}_{\gamma^{-i}}(x, u') + \epsilon, \quad \forall x \in \mathbb{X} \, . \]

\[ Q^* \text{ is also called the state-action value function and the action value function.} \]
3 \( \epsilon \)-revision processes in games

The idea of “satisficing” refers to becoming satisfied and halting search when a sufficiently good input has been found in an optimization problem \([33]\). Satisficing has a long history in both single-agent decision theory (e.g. \([34, 35]\)) and also multi-agent game theory (e.g. \([36]\)). Recently, there has been some interest in studying learning dynamics in games where agents only change their policy when they are not \( \epsilon \)-best-responding. For example, see \([37]\) Section 5]. Other works in the same vein include the aspiration learning algorithms of \([24]\) and \([28]\).

It is natural to ask the following: what assumptions must be made on a game in order to guarantee that some type of satisficing dynamics can lead to \( \Delta^\epsilon_{eq} \)? With this question in mind, we give our next definition.

**Definition 4** A (possibly finite) path of joint policies \((\gamma_k)_{k \geq 0}\) is called an \( \epsilon \)-revision path if, for every \( k \geq 0 \), \( \gamma_k \in \text{BR}_\epsilon^i(\gamma_{k-1}^{-i}) \) implies \( \gamma_{k+1}^i = \gamma_k^i \).

**Definition 5** For \( \epsilon \geq 0 \), a game \( G \) is said to have the \( \epsilon \)-revision paths property if for every \( \gamma_0 \in \Delta \), there exists an \( \epsilon \)-revision path of finite length, say \( \gamma_0, \gamma_1, \ldots, \gamma_L \), such that \( \gamma_L \in \Delta^\epsilon_{eq} \).

We note that the definitions above are not attached to any particular dynamical system on \( \Delta \). They do not require that a player must switch a best-response when not already \( \epsilon \)-best-responding. As such, one may (loosely) interpret the \( \epsilon \)-revision paths property as a necessary condition for convergence to \( \Delta^\epsilon_{eq} \) when players employ an arbitrary \( \epsilon \)-satisficing rule for updating their policy. (We say loosely here because you may fail to satisfy the \( \epsilon \)-satisficing paths property but still admit infinite sequences converging to \( \Delta^\epsilon_{eq} \).

It is easy to see that teams and exact potential games both have the \( \epsilon \)-revision paths property for any \( \epsilon \geq 0 \). In this paper, we choose to not focus on these classes of games, as there are a number of existing independent learning algorithms that take advantage of some structural properties of games in those classes that do not hold for symmetric games in general. For a sampling of some such algorithms, we cite \([28]\) and \([12]\) for weakly acyclic games and teams; for two-player, zero-sum stochastic games we cite \([7]\) and \([27]\); and for potential games we cite \([38]\).

**Theorem 1** If \( G \) is a symmetric game, then \( G \) has the \( \epsilon \)-paths property for all \( \epsilon \geq 0 \).

**Proof:** We prove the claim by explicitly constructing a valid \( \epsilon \)-revision path into \( \Delta^\epsilon_{eq} \). Intuitively, beginning from an arbitrary policy, unsatisfied players (i.e. players not \( \epsilon \)-best-responding) can change policies to match the policy of other players. We create a cohort of players using the same policy and progressively grow the cohort until either we stop because we have found an \( \epsilon \)-equilibrium or because no player is satisfied, which allows us to move in one step to an arbitrary \( \epsilon \)-equilibrium.

Let \( \pi_0 \in \Delta \) be an initial policy. Let \( S_0 := \{ i \in \mathcal{N} : \pi_0^i \notin \text{BR}_\epsilon^i(\pi_0^{-i}) \} \) be the set of players not \( \epsilon \)-best-responding at \( \pi_0 \). If \( S_0 = \emptyset \) then \( \pi_0 \in \Delta^\epsilon_{eq} \) and \( (\pi_0) \) is a valid \( \epsilon \)-revision path from \( \pi_0 \) into \( \Delta^\epsilon_{eq} \). If \( S_0 = \emptyset \), then \( (\pi_0, \pi^*) \) with any \( \pi^* \in \Delta^\epsilon_{eq} \subset \Delta^\epsilon_{eq} \) is a valid \( \epsilon \)-revision path.

Suppose now that \( 1 \leq |S_0| \leq N - 1 \). Select a distinguished player \( j(1) \notin S_0 \), and construct a successor policy \( \pi_1 \) as follows:

\[
\pi_1^i := \begin{cases} 
\pi_0^i, & \text{if } i \notin S_0, \\
\pi_0^{j(1)}, & \text{if } i \in S_0.
\end{cases}
\]

Now, define \( S_1 := \{ i \in \mathcal{N} : \pi_1^i = \pi_0^{j(1)} \} \) to be the set of all players whose policy matches \( j(1) \)'s policy under \( \pi_1 \), and note that \( S_0 \cup \{ j(1) \} \subseteq S_1 \). We have thus constructed a valid \( \epsilon \)-revision path \( (\pi_0, \pi_1) \) and a sequence of sets \( (S_0, S_1) \) such that the following three properties hold for \( k = 1 \):

(I) All players in \( S_0 \) use the same policy, i.e. \( \pi_k^i = \pi_k^j \) for all \( i, j \in S_k \);

(II) \( |S_k| \geq |S_{k-1}| + 1 \).

(III) If player \( j \notin S_k \), then \( \pi_k^i \neq \pi_k^j \) for any \( i \in S_k \).
For \( n \geq 1 \), suppose \( (\pi_0, \pi_1, \ldots, \pi_n) \) is a valid \( \epsilon \)-revision path, and suppose \( S_0, S_1, \ldots, S_n \) is a sequence of subsets of \( \mathcal{N} \) such that for any \( k \in \{1, \ldots, n\} \) we have (I) \( \pi_k^i = \pi_k^j \) for any \( i, j \in S_k \); (II) \( |S_k| \geq |S_{k-1}| + 1 \); (III) if \( i \in S_k, j \notin S_k \), then \( \pi_k^i \neq \pi_k^j \).

If \( \pi_n \in \Delta_{eq} \), then \((\pi_0, \ldots, \pi_n)\) is an \( \epsilon \)-revision path from \( \pi_0 \) into \( \Delta_{eq} \). On the other hand, if \( \pi_n \notin \Delta_{eq} \), we proceed in cases.

**Case 1:** \((|S_n| = N \text{ and } \pi_n \notin \Delta_{eq})\) By (I), \( \pi_n^i = \pi_n^j \) for every \( i, j \in S_n = \mathcal{N} \). By Lemma 1, no agent is \( \epsilon \)-best-responding at \( \pi_n \). (Otherwise some player is \( \epsilon \)-best-responding, and therefore all are and so \( \pi_n \in \Delta_{eq} \), which we have ruled out.) Then, for any \( \pi^* \in \Delta_{eq} \), we have that \((\pi_0, \ldots, \pi_n, \pi^*)\) is a valid \( \epsilon \)-revision path from \( \pi_0 \) into \( \Delta_{eq} \).

**Case 2:** \((|S_n| \leq N - 1 \text{ and } \pi_n \notin \Delta_{eq})\) Again by Lemma 1, either (2a) all agents in \( S_n \) are \( \epsilon \)-best-responding at \( \pi_n \) or (2b) none are. We treat Case 2a first. Since \( \pi_n \) is not an \( \epsilon \)-equilibrium and each player in \( S_n \) is \( \epsilon \)-best-responding, there must exist a player \( j(n+1) \in \mathcal{N} \setminus S_n \) such that \( j(n+1) \) is not \( \epsilon \)-best-responding at \( \pi_n \), and we construct a policy \( \pi_{n+1} \) as

\[
\pi_{n+1}^i = \begin{cases} 
\pi_n^i, & \text{if } i \neq j(n+1) \\
\pi_s^i, & \text{if } i = j(n+1),
\end{cases}
\]

where \( s \in S_n \) is any player in \( S_n \). Thus, we have that (I) for all \( i, j \in S_{n+1} := S_n \cup \{j(n+1)\} \),

\[
\pi_{n+1}^i = \pi_{n+1}^j; \quad (II) \ |S_{n+1}| \geq |S_n| + 1; \quad \text{and (III) if } j \notin S_{n+1}, \text{ then } \pi_{n+1}^j \neq \pi_{n+1}^i \text{ for any } i \in S_{n+1}.
\]

Note also that \((\pi_0, \ldots, \pi_n, \pi_{n+1})\) is a valid \( \epsilon \)-revision path.

In Case 2b, players in \( S_n \) are not \( \epsilon \)-best-responding at \( \pi_n \). We select \( j(n+1) \in \mathcal{N} \setminus S_n \) and construct \( \pi_{n+1} \) as

\[
\pi_{n+1}^i := \begin{cases} 
\pi_{n}^{j(n+1)}, & \text{if } i \in S_n, \\
\pi_n^i, & \text{if } i \notin S_n.
\end{cases}
\]

We define \( S_{n+1} := \{ i \in \mathcal{N} : \pi_{n+1}^i = \pi_{n}^{j(n+1)} \} \). Once again, properties (I)–(III) hold and \((\pi_0, \ldots, \pi_n, \pi_{n+1})\) is a valid \( \epsilon \)-revision path.

Note that this process—of producing \( S_{n+1} \) and \( \pi_{n+1} \) out of \( S_n \) and \( \pi_n \)—can be repeated only finitely many times before stopping. This is because \( 1 + n \leq |S_0| + n \leq |S_n| \leq N \), and so we are constrained to \( n \leq N - 1 \). In every case, we use this process to produce an \( \epsilon \)-revision path of length at most \( N + 1 \) from \( \pi_0 \) to \( \Delta_{eq} \).

With this important structural property of symmetric games now established, we will proceed by developing algorithms for playing such games.

### 4 Approximation results

#### 4.1 Quantized policies

For ease of analysis and for algorithm design, we will restrict player \( i \)'s policy selection from the uncountable set \( \Delta^i \) to a finite subset \( \Pi^i \subset \Delta^i \). The set \( \Pi^i \) is obtained via uniform quantization of \( \Delta^i \), and restriction to \( \Pi^i \) is justified using the following result.

**Lemma 2** For every \( i \in \mathcal{N}, x \in \mathbb{X}, \) the cost functional \( J_x^i : \Delta \to \mathbb{R} \) is continuous.

The proof of Lemma 2 can be found on page 15.

Since \( \Delta \) is compact, Lemma 2 implies that each cost functional \( J_x^i \) is also uniformly continuous.

From this we have that for any \( \epsilon > 0 \), there exists \( \xi(\epsilon) > 0 \) such that if two joint policies \( \gamma, \pi \) are \( \xi \)-close, then \( |J_x^i(\gamma) - J_x^i(\pi)| < \epsilon \), for any \( i \in \mathcal{N}, x \in \mathbb{X} \).

A quantization \( \Pi \) of \( \Delta \) into bins of radius less than \( \xi(\epsilon) \) has the desirable property that player \( i \) always has an \( \epsilon \)-best-response in \( \Pi^i \) to any policy \( \gamma^{-i} \in \Delta^{-i} \) for the remaining players. Moreover, as there is at least one equilibrium in \( \Delta \), say \( \gamma^* \), we are guaranteed at least one \( \epsilon \)-equilibrium in \( \Pi \).

---

That is, the distance \( d(\gamma, \pi) := \max \{ |\gamma^i(a^i|s) - \pi^i(a^i|s)| : i \in \mathcal{N}, a^i \in \Pi^i, s \in \mathbb{X} \} < \xi \)
We now outline our algorithmic approach to finding \( \epsilon \)-equilibrium in a symmetric stochastic game. The approach taken here builds on a technique presented in [12], which decouples learning and adaptation. This decoupled design is used to mitigate the challenges relating to learning in a non-stationary environment.

During a learning phase, each agent follows a fixed perturbed policy and estimates whether it is \( \epsilon \)-best-responding to the (unobserved) joint policy of the remaining agents. At the end of a learning phase, the agents synchronously update their policies, which will then be followed for the subsequent learning phase. At its core, this approach consists of four parts:

### 4.2 Perturbed policies

We now introduce perturbed policies, which play an important role in the design of our algorithm in the subsequent sections.

**Definition 6** Let \( j \in \mathcal{N}, \rho^j \in (0,1), \) and \( \gamma^j \in \Delta^j \). We define a policy \( \tilde{\gamma}^j \in \Delta^j \) as

\[
\tilde{\gamma}^j (\cdot|x) = (1 - \rho^j) \gamma^j (\cdot|x) + \rho^j \text{Unif}(U^j), \quad \forall x \in \mathcal{X},
\]

and we refer to \( \tilde{\gamma}^j \) as the \( \rho^j \)-perturbation of \( \gamma^j \).

The dependence on \( \rho^j \) above is implicit but important. The quantity \( \rho^j \) can be interpreted as the frequency with which player \( j \) experiments with uniform random action selection, while following a baseline policy \( \pi^j \) with frequency \( 1 - \rho^j \). If \( \rho \in (0,1)^{\mathcal{N}} \), an analogous construction \( \tilde{\pi} \) will be called the \( \rho \)-perturbation of the joint policy \( \pi \).

We now state two results about the approximation when players jointly switch from a particular policy in \( \Pi \) to its perturbation, where \( \Pi \) is the restricted set of joint policies of Assumption [1].

**Lemma 3** For any \( \psi > 0 \), there exists \( \rho(\psi) > 0 \) such that, if \( \rho^j \leq \rho(\psi) \) for all \( j \in \mathcal{N} \), we have

\[
\| Q^{\pi, i} - Q^{\tilde{\pi}} \|_\infty \leq \psi, \quad \forall i \in \mathcal{N} \text{ and } \pi \in \Pi,
\]

where \( \tilde{\pi} \) is the \( \rho \)-perturbed policy associated to \( \pi \).

**Proof:** See [12, Lemma 3].

**Lemma 4** For any \( \psi > 0 \), there exists \( \tilde{\rho}(\psi) > 0 \) such that if \( \rho^j \leq \tilde{\rho}(\psi) \) for every \( j \in \mathcal{N} \), then we have

\[
| J^i_\pi - J^i_{\tilde{\pi}} | \leq \psi, \quad \forall i \in \mathcal{N}, x \in \mathcal{X} \text{ and } \pi \in \Pi,
\]

where \( \tilde{\pi} \) is the \( \rho \)-perturbed policy associated to \( \pi \).

**Proof:** This follows from Lemma [2].

### 5 Decoupling learning and adapting

We now outline our algorithmic approach to finding \( \epsilon \)-equilibrium in a symmetric stochastic game. The approach taken here builds on a technique presented in [12], which decouples learning and adaptation. This decoupled design is used to mitigate the challenges relating to learning in a non-stationary environment.

During a learning phase, each agent follows a fixed perturbed policy and estimates whether it is \( \epsilon \)-best-responding to the (unobserved) joint policy of the remaining agents. At the end of a learning phase, the agents synchronously update their policies, which will then be followed for the subsequent learning phase. At its core, this approach consists of four parts:
1. Time is partitioned into intervals called “exploration phases,” the \( k^{th} \) lasting \( T_k \in \mathbb{N} \) stage games, beginning with the stage game at time \( t_k := \sum_{l=0}^{k-1} T_l \) and ending after the stage game at \( t_k + T_k - 1 \);
2. Within an exploration phase, agent \( i \in \mathcal{N} \) follows a fixed policy and obtains feedback data on state-action-cost trajectories.
3. Within an exploration phase, agent \( i \) processes feedback data for policy evaluation, estimation of best-response sets, and estimation of state-action values.
4. Between the \( k^{th} \) and \( (k+1)^{th} \) exploration phases, agent \( i \) uses the learned information to update the baseline policy from \( \pi_i^k \) to \( \pi_i^{k+1} \in \Pi_i \). We focus here on \( \epsilon \)-satisficing update rules; that is, we focus on algorithms that prescribe no updating when player \( i \) is already \( \epsilon \)-best-responding to the remaining players.

Although the exploration phase technique is employed in our main algorithm here, this paper differs from [12] and [28] in multiple substantial ways. The most obvious difference is the class of games considered, with the present paper focusing on symmetric games and the earlier papers studying stochastic teams and their generalizations. The latter classes always admit an equilibrium in the set of stationary and deterministic policies, and so search can be restricted to this small finite set. Moreover, in the case of weakly ayclic games, inertial best-response dynamics can be used to drive play to such a deterministic equilibrium policy. Neither of these properties holds for general symmetric games. For example, consider the game of Rock-Paper-Scissors, presented in Figure 1. In this stateless game, the unique 0-equilibrium is to select each action with uniform probability, and one can show that for sufficiently small \( \epsilon \), all \( \epsilon \)-equilibria will be in a small neighbourhood of the unique 0-equilibrium [39]. Inertial best-response dynamics—which restricts agents to selecting deterministic policies—results in cyclical behaviour in this game, and cannot be used to find an \( \epsilon \)-equilibrium.

|       | Rock | Paper | Scissors |
|-------|------|-------|----------|
| Rock  | 0,0  | -1,1  | 1,-1     |
| Paper | -1,-1| 0,0   | -1,1     |
| Scissors | -1,1 | 1,-1  | 0,0      |

Rock-Paper-Scissors game

Figure 1: Player I (II) picks a row (column), and its reward (to be maximized) is the 1st (2nd) entry of the chosen cell.

As the preceding example shows, the adaptation mechanisms used in [12] and [28] may be ill-suited for use in some symmetric games. In order to include randomized policies in the search space, the main algorithm of this paper finely quantizes the policy spaces \( \Delta_i \) and allows for policy updates that move through the interior of this set rather than traversing boundary points. Moreover, to avoid making specific structural assumptions on the cost function, we include random search of the quantized set when a player is not \( \epsilon \)-best-responding in order to allow for a greater variety of revision paths to be explored during play. The subsequent subsections elaborate on this process.

5.1 Policy revision with oracle

We now specify a particular policy update rule to be used in the sequel, and we study the behaviour resulting from this rule under the unrealistic assumption that each player has access to an oracle for obtaining the information required for its policy update. This section focuses on the purely adaptive part of our algorithm, with the challenges of learning postponed to the next section. Our main algorithm is later analyzed as a noise perturbed version of this oracle process.

We propose an update rule that builds on the principle of satisficing, described above in Section 3. In particular, our rule instructs an agent to not change its policy when it is already \( \epsilon \)-best-responding. When not already \( \epsilon \)-best-responding, agent \( i \) is instructed to update its policy as follows: with small probability, \( \epsilon^i \in (0,1) \), select a policy in \( \Pi_i^t \) uniformly at random; with complement probability, \( 1 - \epsilon^i \), switch to another policy in \( \Pi_i^t \) that is determined by the Q-factors for the current environment. The mechanism actually used is up to the algorithm designer and can incorporate knowledge of the game if desired, provided this update only uses the state-action values and the current policy of the agent when deciding the next policy. For concreteness, we now give one potential subroutine for
stepping in the direction of a best-response, called UpdateRule$^i$. This subroutine is taken from [22] (c.f. Table 5), and we note that it is not the only alternative; more effective subroutines may exist, depending on the setting.

Algorithm 1: UpdateRule$^i(·|\pi)$ (with oracle access to Q-factors)

1. **Input** Joint policy $\pi = (\pi_{old}^i, \pi_{−i}^−) \in \Pi$
2. **Set Parameters**
3. for each $(x, u^i) \in X \times U^i$
   4. $\eta^i_{x,u^i} := \min \{ \pi_{old}^i(u^i|x), \eta^i / (|U^i| − 1) \}$
5. **Receive** $Q^i_{\pi_{−i}}$ (from oracle)
6. for each $(x, u^i) \in X \times U^i$
   7. $H^i_{x,u^i} := \begin{cases} -\eta^i_{x,u^i}, & \text{if } u^i \neq \text{argmax}_{a^i} Q^i_{\pi_{−i}}(x, a^i) \\ \sum_{a^i \neq u^i} \eta^i_{x,a^i}, & \text{else.} \end{cases}$
8. $\pi_{\text{mid}}^i(x|u) := \pi_{old}^i(u^i|x) + H^i_{x,u^i}$
9. **Output** $\pi_{new}^i = \text{proj}_{\Pi^i}(\pi_{\text{mid}}^i)$, (the policy in $\Pi^i$ closest to $\pi_{\text{mid}}^i$)

In UpdateRule$^i$, we have assumed oracle access to Q-factors for the environment determined by the game and the joint policy $\pi_{−i}^−$. Later, in Algorithm 3, we will introduce a subroutine called IndependentUpdateRule$^i$ that is effectively the same as UpdateRule$^i$, except that it uses learned Q-factors instead of the correct values. (We note that if the argmax in Line 7 is not a singleton, then any tie-breaking procedure may be used to select among the maximizers.)

Algorithm 2: Policy Revision Process for agent $i \in N$ (with oracle access)

1. **Set Parameters**
   2. $\epsilon^i \in (0, 1)$: experimentation probability when not $\epsilon$-best-responding
   3. $\eta^i > 0$ for UpdateRule$^i$ (Algorithm 1)
4. **Initialize**
   5. $\pi_0^i \in \Pi^i$: initial policy
5. for $k \geq 0$ ($k^{th}$ policy update)
   6. if $\pi_k^i$ is an $\epsilon$-best-response to $\pi_{−i}^−$ then
      7. $\pi_{k+1}^i = \pi_k^i$
   else
      8. $\pi_{k+1}^i \leftarrow \begin{cases} \gamma^i \sim \text{UpdateRule}^i(·|\pi_k), & \text{w.p. } 1 − \epsilon^i \\ \pi^i \sim \text{Unif}(\Pi^i), & \text{w.p. } \epsilon^i \end{cases}$
   9. Go to $k + 1$

We assume that players update their policies in an independent manner, conditional on the oracle information. The process resulting from all players’ following Algorithm 2 is a time homogeneous Markov chain on $\Pi$, and it is immediate that any $\epsilon$-equilibrium in $\Pi$ is an absorbing state.

**Lemma 5** Let $G$ be a symmetric game. If all players update their policies according to Algorithm 2 and Assumption 1 holds, then

$$\lim_{k \to \infty} \Pr(\pi_k \in \Delta_{eq}^\epsilon) = 1.$$ 

**Proof:** As noted above, any $\epsilon$-equilibrium of $G$ in the set $\Pi$ is an absorbing set for this Markov chain. Furthermore, the argument used in the proof of Theorem 1 can be used here to show that there exist $\epsilon$-revision paths within the restricted set $\Pi$ from any $\pi_0 \in \Pi$ to $\Pi \cap \Delta_{eq}^\epsilon$.

---

5For example, if the game was known to be a team, one could replace this routine with inertial best-responding.
For each $\gamma \in \Pi$, let $L_\gamma$ denote the shortest path of positive probability from $\gamma$ to an $\epsilon$-equilibrium in $\Pi$, and let $p_\gamma > 0$ be the probability of following this path when starting at $\gamma$. Define $L = \max\{L_\gamma : \gamma \in \Pi\}$ and $p := \min\{p_\gamma : \gamma \in \Pi\} > 0$, where the inequality holds because $\Pi$ is a finite set. Then,

$$\Pr \left( \bigcap_{j=1}^{J} \{ \pi_{k+jL} \text{ is not } \epsilon\text{-equilibrium} \mid \pi_k \} \leq (1 - p)^J \to 0, \text{ as } J \to \infty. \right)$$

Although Algorithm 2 cannot be used by online independent learners, it does offer some insights for designing the policy adaptation mechanism. In the next section, we combine the adaptation mechanism of Algorithm 2 with learning to replace the oracle and achieve guarantees on finding $\epsilon$-equilibrium even with independent learners in an online setting.

6 Synchronized two timescale learning algorithm

In this section, we present our main algorithm, Algorithm 3, which can be interpreted as a noisy, two-timescale implementation of Algorithm 2. Before giving the main algorithm, we present IndependentUpdateRule', which is a variant of UpdateRule' that can be implemented by independent learners who have estimated their Q-factors and do not know the joint policy of the remaining agents. IndependentUpdateRule' is presented in Algorithm 3.

Algorithm 3: IndependentUpdateRule'($\pi_{\text{old}}$, $Q^i$) (with estimated Q-factors)

1. **Input** Individual policy $\pi_{\text{old}}^i \in \Pi^i$; estimated Q-factors $Q^i \in \mathbb{R}^{X \times U}$
2. **Set Parameters** $\eta^i > 0$
3. **for** each $(x, u^i) \in X \times U^i$
4. \hspace{1em} $\eta_{x, u^i}^i := \min \{ \pi_{\text{old}}^i(u^i|x), \eta^i/(|U^i| - 1) \}$
5. **for** each $(x, u^i) \in X \times U^i$
6. \hspace{1em} $H_{x, u^i}^i := \left\{ \begin{array}{ll} -\eta_{x, u^i}^i, & \text{if } u^i \neq \arg\max_{a^i} Q^i(x, a^i) \\ \sum_{a_i \neq u^i} \eta_{x, a_i}^i, & \text{else.} \end{array} \right.$
7. \hspace{1em} $\pi_{\text{mid}}^i(u^i|x) := \pi_{\text{old}}^i(u^i|x) + H_{x, u^i}^i$
8. **Output** $\pi_{\text{new}}^i = \text{proj}_{\Pi^i}(\pi_{\text{mid}}^i)$, (the policy in $\Pi^i$ closest to $\pi_{\text{mid}}^i$)

6.1 Main Results

Our main algorithm is given below, as Algorithm 3. To state our main results, we recall that the game $G$ and constant $\epsilon > 0$ are fixed and that the restricted set of policies $\Pi$ is chosen to satisfy Assumption 1.

Assumption 2. For all $i \in N$, $\rho^i \in (0, \bar{\rho})$ and $\delta^i \in (0, \bar{\delta})$, where $\bar{\rho}$ and $\bar{\delta}$ are constants defined in the appendices, on page 18.

The constant $\bar{\delta}$ depends only on the game, $\epsilon$, and $\Pi$, while $\bar{\rho}$ depends on the game, $\epsilon$, and the collection $\{\delta^i\}_{i \in N}$. We also make the following standard assumption on the state transition kernel $P$, which is necessary to ensure that no proper subset of states is absorbing.

Assumption 3. For any $x, x' \in X$, there exists $H \in \mathbb{N}$, states $x = s_0, \ldots, s_H$, $s_{H+1} = x'$, and joint actions $a_0, \ldots, a_H$ such that

$$\prod_{j=0}^{H} P(s_{j+1}|s_j, a_j) > 0.$$
Algorithm 4: for agent $i$

1 Set Parameters
2 $Q^i \subset \mathbb{R}^{X \times U^i}$ and $J^i \subset \mathbb{R}^X$: compact sets
3 $\{T_k\}_{k \geq 0}$: a sequence in $\mathbb{N}$ of learning phase lengths
4 set $t_0 = 0$ and $t_{k+1} = t_k + T_k$ for all $k \geq 0$.
5 $\rho^i \in (0, 1)$: action experimentation probability
6 $\epsilon^i \in (0, 1)$: random policy updating probability
7 $\eta^i \in (0, 1)$: parameter for IndependentUpdateRule$^i$ (Algorithm$^3$)
8 $\delta^i \in (0, \infty)$: tolerance level for sub-optimality
9 $\{\alpha_n^i\}_{n \geq 0}$: step sizes such that $\alpha_n^i \in [0, 1]$, $\sum_n \alpha_n^i = \infty$, $\sum_n (\alpha_n^i)^2 < \infty$

10 Initialize baseline policy $\pi_0^i \in \Pi^i$, $Q_0^i \in Q^i$, $J_0^i \in J^i$ (all arbitrary)
11 Receive initial state $x_0 \in X$
12 for $k \geq 0$ (the $k^{th}$ exploration phase)
13 for $t = t_k, t_k + 1, \ldots, t_{k+1} - 1$ // Policy evaluation loop
14 Select $u_t^i = \begin{cases} \pi_k^i(x_t), & \text{w.p. } 1 - \rho^i \\ u_t^i \sim \text{Unif}(U^i), & \text{w.p. } \rho^i \end{cases}$
15 Receive cost $c^i(x_t, u_t^i, u_{t-1}^i)$, state $x_{t+1}$
16 Set $n_t^i = \text{the number of visits to } (x_t, u_t^i)$ in the $k^{th}$ learning phase up to $t$
17 Set $n_t^i = \text{the number of visits to } x_t$ in the $k^{th}$ learning phase up to $t$
18 $Q_{t+1}^i(x_t, u_t^i) = (1 - \alpha_{n_t^i}^i)Q_t^i(x_t, u_t^i) + \alpha_{n_t^i}^i \left[ c^i(x_t, u_t^i, u_{t-1}^i) + \beta^i \min_{v^i} Q_t^i(x_t, v^i) \right]$
19 $J_{t+1}^i(x_t) = (1 - \alpha_{n_t^i}^i)J_t^i(x_t) + \alpha_{n_t^i}^i \left[ c^i(x_t, u_t^i, u_{t-1}^i) + \beta^i J_t^i(x_t+1) \right]$
20 if $\forall x \in X, J_{k+1}^i(x) \leq \min_{a^i} Q_{k+1}^i(x, a^i) + \epsilon + \delta^i$ then $i$ is satisfied
21 else $i$ not satisfied
22 if $i$ is satisfied then // Baseline policy update
23 $\pi_{k+1}^i \leftarrow \pi_k^i$
24 else
25 $\pi_{k+1}^i \leftarrow \begin{cases} \hat{\pi}^i \sim \text{IndependentUpdateRule}^i(\pi_k^i, Q_{k+1}^i), & \text{w.p. } 1 - \epsilon^i \\ \pi^i \sim \text{Unif}(\Pi^i), & \text{w.p. } \epsilon^i \end{cases}$
26 Reset $Q_{t+1}^i$ to any $Q^i \in Q^i$ (e.g., project $Q_{t+1}^i$ on $Q^i$)
27 Reset $J_{t+1}^i$ to any $J^i \in J^i$

Theorem 2 Let $G$ be a symmetric game, $\epsilon > 0$, and $\Pi \subset \Delta$ be the quantized set of joint policies satisfying Assumption$^7$ Suppose all players follow Algorithm$^4$ and suppose Assumptions$^7$ and$^8$ hold. Then, for any $\psi > 0$, there exists $\bar{T} = \bar{T}(\psi, \epsilon, \Pi, \{\delta^i, \rho^i\}_{i \in \mathbb{N}}) \in \mathbb{N}$ such that if $T_k \geq \bar{T}$ for all $k$, then

$$\Pr(\pi_k \in \Delta_{eq}^\psi) \geq 1 - \psi, \text{ for all sufficiently large } k.$$ 

The proof of Theorem$^2$ is given in Appendix$^C$.

7 Simulations

In this section, we present a simulation study of Algorithm$^4$ applied to the familiar game of Rock-Paper-Scissors (RPS), whose stage game was given in$^1$ and is recalled in Figure$^2$.

We set $\beta = 0$, $\epsilon = 0.2$ and ran 100 independent trials of Algorithm$^4$ with each trial consisting of 1500 exploration phases of length 10,000, for a total of 15 million stage games per trial. The quantized set of policies $\Pi^i$ for each player $i \in \{A, B\}$ was chosen to be the set of probability vectors with with each entry having a single-digit after the decimal point, i.e.

$$\Pi^i = \{(p_a, p_b, p_c) \in \mathbb{R}^3 : p_a + p_b + p_c = 1, \text{ and } p_a, p_b, p_c \in \{0, 0.1, \cdots, 0.9, 1\}\}.$$
Figure 2: Player A (B) picks a row (column), and its reward (to be maximized) is the 1st (2nd) entry of the chosen cell.

When not randomly updating its policy, the agent employed a modified version of the independent update rule of Algorithm 3. The results, which are in line with our theoretical results, are summarized in Figure 3.

Figure 3: Frequency of $\{ \pi_t \in \Delta_{\epsilon_{eq}} \cap \Pi \}$, averaged over 100 trials.

8 Discussion

The algorithm presented here is—to our knowledge—the first independent learner that can provably drive play to approximate equilibrium in $N$ player symmetric games without assuming further structure, such as a zero-sum structure or admission of a potential function. One disadvantage of this generality is that, without relying on game-specific structure, the algorithm can be quite slow to converge to $\epsilon$-equilibrium when $\epsilon$ is small. This is most likely to happen when the choice of update function is not well-suited to the game at hand, in which case play is driven to $\epsilon$-equilibrium through random search alone. This reliance on random search is problematic when employing uniform quantizers, because the cardinality of the finite set $\Pi$ will explode as the radius of a bin is taken to 0. In that case, the mixing time of the underlying Markov chain will also be extremely large, and the algorithm will not be reliable in the short-run. It would be desirable to give an upper bound on the requisite bin radius $\xi(\epsilon, G)$ in terms of the cost function and transition kernel of the game. This
way, one could potentially determine quickly whether this algorithm can be used as is or whether it requires modification.

A second shortcoming of the approach presented here (and elsewhere, e.g. [12, 28, 9, 11, 23]) is that it requires agents to synchronously change policies at the end of exploration phases. In some settings this is justifiable, but in other settings it can be seen as a form of coordination between players, which might be questionable given the decentralized nature of the problem. Empirically, it appears as though perfect synchrony is not needed for driving play to equilibrium, but a rigorous proof has yet to be given. The difficulty here comes in large part from the complexity of conditional probabilities: one agent’s learning carries information about the state-action trajectory and so is correlated with the learning of another agent, so arguments that rely on conditioning and an appeal to the oracle process do not go through cleanly.

Another direction for future work is to switch focus away from the tabular setting, where each state-action is treated individually, and to consider action value learners that employ function approximation instead. With minor modifications, the main algorithmic ideas can be extended to be applicable to games with large state and action sets.

Finally, we note that the special structure required for the algorithm design—\(\epsilon\)-revision paths—is likely a structural property shared by other interesting classes of games. In a different class of games, if one can prove that a particular quantization \(\Pi\) of \(\Delta\) admits \(\epsilon\)-revision paths from any initial policy \(\pi_0 \in \Pi\) to an approximate equilibrium policy in \(\Pi\), then the same algorithm can be used for that class. However, guaranteeing that such paths exist in the absence of symmetry may be difficult.

9 Conclusions

In this paper, we considered independent learning algorithms for a class of stochastic games that is characterized by symmetry across players, both in costs and in state dynamics. To develop an algorithm that drives play to approximate equilibrium, we studied the properties of symmetric games in order to find exploitable structure. To this end, we defined \(\epsilon\)-revision paths and the \(\epsilon\)-revision paths property, and we showed that our class of symmetric games satisfies the \(\epsilon\)-revision paths property for any \(\epsilon \geq 0\).

After establishing the exploitable structure of the \(\epsilon\)-revision paths property, we presented a new algorithm for playing symmetric stochastic games, and we prove that this algorithm drives play to approximate equilibrium in the long run. The algorithm can be thought of as a two timescale procedure, wherein learning and adaptation occur at different timescales: players hold their policies constant during the fast timescale and use their observations to perform policy evaluation, and then players change their policies on the slower timescale to leverage this learned information. Our main algorithm, Algorithm 4, uses the same ‘exploration phase’ technique that was presented in [12], but here we employ a very different adaptation mechanism for changing policies and furthermore we explore an altogether different policy set. Our main algorithm combines “\(\epsilon\)-satisficing” with random search to explore a quantized subset of the agent’s policies.
A Proofs of the lemmas

Proof of Lemma 2

Fix $\epsilon > 0$ and $\gamma \in \Delta$. For some $T \in \mathbb{N}$, we note that the probability of a given trajectory $\{(s_k, a_k)\}_{k=0}^{T}$ is the following:

$$\Pr^\gamma \left( \{(x_k = s_k, u_k = a_k)\}_{k=0}^{T} | x_0 = x \right) = \prod_{k=0}^{T} \gamma(a_k | s_k) \cdot \prod_{k=0}^{T-1} P(s_{k+1} | s_k, a_k), \quad (6)$$

where $\gamma(a_k | s_k) = \prod_{j \in \mathcal{N}} \gamma^j(a_k^j | s_k)$ for each $0 \leq k \leq T$.

Let $\tilde{\gamma} \in \Delta$ be any policy. Taking $T$ large enough that

$$\beta^{T+1} \frac{1}{1 - \beta} \max \left\{ \{c^j(x, u) : j \in \mathcal{N}, x \in \mathcal{X}, u \in \mathcal{U}\} \right\} < \epsilon$$

holds, we have that

$$|J_x^\gamma(\gamma) - J_x^\gamma(\tilde{\gamma})| \leq \left| E^\gamma \left( \sum_{k=0}^{T} (\beta^i)^k c^i(x_k, u_k) | x_0 = x \right) - E^{\tilde{\gamma}} \left( \sum_{k=0}^{T} (\beta^i)^k c^i(x_k, u_k) | x_0 = x \right) \right| + \epsilon/2,$$

Define a function $F_T : \mathcal{X} \times \mathcal{U}^{T+1} \to \mathbb{R}$ by $F_T(s_0, a_0, \ldots, s_T, a_T) = \sum_{k=0}^{T} (\beta^i)^k c^i(s_k, a_k)$. We then have

$$E^\gamma \left( \sum_{k=0}^{T} (\beta^i)^k c^i(x_k, u_k) | x_0 = x \right) = \sum_{\omega \in (\mathcal{X} \times \mathcal{U})^{T+1}} F_T(\omega) \Pr^\gamma \left( \{(x_k, u_k)_{k=0}^{T} = \omega | x_0 = 0 \right).$$

For ease of notation, we write $\Pr^\gamma(\omega | x_0 = x) := \Pr^\gamma \left( \{(x_k, u_k)_{k=0}^{T} = \omega | x_0 = x \right).$ Then,

$$\left| E^\gamma \left( \sum_{k=0}^{T} (\beta^i)^k c^i(x_k, u_k) | x_0 = x \right) - E^{\tilde{\gamma}} \left( \sum_{k=0}^{T} (\beta^i)^k c^i(x_k, u_k) | x_0 = x \right) \right| \leq \sum_{\omega \in (\mathcal{X} \times \mathcal{U})^{T+1}} \left| F_T(\omega) \right| \cdot \left| \Pr^\gamma(\omega | x_0 = x) - \Pr^{\tilde{\gamma}}(\omega | x_0 = x) \right|. \quad (7)$$

From the continuity of the right-hand side of (6), the result follows: taking $\left| \gamma^i(a^i | s) - \tilde{\gamma}^i(a^i | s) \right|$ to be sufficiently small for all $(s, a^i) \in \mathcal{X} \times \mathcal{U}$ for every $i \in \mathcal{N}$, we can ensure that

$$\left| \Pr^\gamma(\omega | x_0 = x) - \Pr^{\tilde{\gamma}}(\omega | x_0 = x) \right| \leq \frac{\epsilon}{2} \left| \mathcal{X} \times \mathcal{U} \right|^{T+1} \max_{\omega'} \left| F_T(\omega') \right|,$$

which bounds the righthand side of (7) by $\epsilon/2$, completing the proof. \ □

Note that the preceding proof did not require symmetry, and indeed Lemma 2 holds for any stochastic game (1).
B Supporting Lemmata and results from the literature

Two results for single-agent MDPs

Consider a single-agent MDP, using the notation from (1) but omitting the player indices. Suppose the agent uses a behaviour policy $\gamma \in \Delta$ to interact with its environment and collect observations in the form of a trajectory $\{(x_t, u_t, c(x_t, u_t))\}_{t=0}^\infty$. Let $Q_0 \in \mathbb{R}^X \times U$ be a given initial condition and suppose the agent updates its Q-factors according to the following rule:

$$Q_{t+1}(x, u) = Q_t(x, u) + \theta_t(x, u) \left( c(x_t, u_t) + \beta \min_{a \in U} Q_t(x_{t+1}, a) - Q_t(x, u) \right), \quad (8)$$

for all $(x, u) \in X \times U$, $t \geq 0$, where $\theta_t(x, u) \in [0, 1]$ is a random step-size parameter.

In this paper, for simplicity, we assume a particular form for the step-size parameters $\theta_t(x, u)$ that ensure they are determined by the sample path up to time $t$: that is, $\theta_t(x, u)$ is measurable with respect to the $\sigma$-algebra generated by the variables $\{(x_k, u_k)\}_{k=0}^t$.

**Assumption 4** Let $\{\alpha_t\}_{t \geq 0}$ be a sequence in $[0, 1]$ such that

$$\sum_{t \geq 0} \alpha_t = \infty \quad \sum_{t \geq 0} \alpha_t^2 < \infty.$$  

For any $t \geq 0$, $(x, u) \in X \times U$, define the random variables

$$n_t(x, u) = |\{k : (x_k, u_k) = (x, u), 0 \leq k \leq t\}|.$$  

Assume that the step-size variables $\theta_t$ in (8) are given by

$$\theta_t(x, u) = \begin{cases} \alpha_{n_t(x,u)} & \text{if } (x_t, u_t) = (x, u), \\ 0 & \text{otherwise.} \end{cases}$$

We now state a useful result on the uniformity of convergence of $Q_t \to Q^*$ in its initial condition $Q_0$.

**Lemma 6** Let $Q \subset \mathbb{R}^X \times U$ be a compact set and let $\gamma \in \Delta$ be the policy used by the agent. Let $(Q_t)_{t \geq 0}$ be the sequence obtained via Q-learning and suppose Assumption 4 holds. If every $(x, u) \in X \times U$ is visited infinitely often $\Pr^\gamma$-almost surely, then for any $\eta > 0$, there exists $T = T(\eta, Q) \in \mathbb{N}$ such that

$$\Pr^\gamma \left( \sup_{t \geq T} \|Q_t - Q^*\|_\infty \leq \eta \right) \geq 1 - \eta, \quad \forall Q_0 \in Q.$$  

For a proof of Lemma 6 see [12] Lemma 1.

An analogous result holds when estimating $J_\gamma(x)$ using a stochastic approximation algorithm similar to Q-learning. Suppose the agent follows a policy $\gamma$ and observes a trajectory of states, actions, and costs $\{(x_t, u_t, c(x_t, u_t))\}_{t \geq 0}$, and updates its value function estimates according to the rule

$$J_{t+1}(x) = \begin{cases} (1 - \alpha_{m_t(x)} J_t(x_t)) + \alpha_{m_t(x)} \left( c(x_t, u_t) + \beta J_t(x_{t+1}) \right) & \text{if } x_t = x \\ J_t(x) & \text{if } x_t \neq x, \end{cases} \quad (9)$$

where $J_0 \in \mathbb{R}^X$ is given and $m_t(x) = |\{k : x_k = x, 0 \leq k \leq t\}|$, i.e. $m_t(x)$ is the number of visits to $x \in X$ by time $t$.

**Lemma 7** Let $\gamma \in \Delta$ be a stationary policy and let $J \subset \mathbb{R}^X$ be a compact set. Suppose that any state $x \in X$ is visited infinitely often $\Pr^\gamma$-almost surely. Let $(J_t)_{t \geq 0}$ be the sequence obtained via (9), and suppose Assumption 4 holds. For any $\eta > 0$, there exists $\bar{T} = \bar{T}(\eta, J) \in \mathbb{N}$ such that

$$\Pr^\gamma \left( \sup_{t \geq \bar{T}} \|J_t - J^*\|_\infty \leq \eta \right) \geq 1 - \eta, \quad \forall J_0 \in J,$$  

where $J^*(x) = J_\gamma(x)$ for each $x \in X$.

The proof of Lemma 7 parallels that of [12] Lemma 1, and is omitted.
Results for learning in stochastic games

As discussed in §2.4, if players \(-i\) follow a stationary policy \(\gamma^{-i} \in \Delta^{-i}\), then player \(i\) faces an environment equivalent to an MDP, and we denoted \(i\)'s optimal Q-factors for this environment by \(Q^*_{\gamma^{-i}}\). We now review several existing results pertaining to the regularity of Q-learning in stochastic games.

Assumption 5 For each agent \(i \in \mathcal{N}\), \(\{\alpha^i_t\}_{t \geq 0}\) is a sequence of step-size parameters in \([0, 1]\) satisfying \(\sum_{t \geq 0} \alpha^i_t = \infty\), and \(\sum_{t \geq 0} (\alpha^i_t)^2 < \infty\). Assume that each agent \(i\) updates its Q-factors according to the rule

\[
Q^i_{t+1}(x, u^i) = (1 - \theta^i_t(x, u^i)) Q^i_t(x, u^i) + \theta^i_t(x, u^i) \left( \epsilon^i(x, u^i_t, u^{-i}_t) + \beta^i \min_{a^i \in \mathcal{U}^i} Q^i_t(x_{t+1}, a^i) \right),
\]

with

\[
\theta^i_t(x, u^i) = \begin{cases} \alpha^i_t(x, u^i), & \text{if } (x_t, u^i_t) = (x, u^i), \\ 0, & \text{otherwise}, \end{cases}
\]

where for each \((x, u^i) \in \mathcal{X} \times \mathcal{U}^i\), \(n^i_t(x, u^i) := \sum_{k=0}^t \mathbb{1}_{\{k : (x_k, u^i_k) = (x, u^i), \text{ for } 0 \leq k \leq t\}}\) counts the number of visits to \((x, u^i)\) up to time \(t\).

To state our next lemma, we recall Assumption 3

Assumption 3 For any \(x, x' \in \mathcal{X}\), there exists \(H \in \mathbb{N}\), states \(x = s_0, \ldots, s_H, s_{H+1} = x'\), and joint actions \(a_0, \ldots, a_H\) such that

\[
\prod_{j=0}^H P(s_{j+1}|s_j, a_j) > 0.
\]

Assumption 3 is a standard condition on the transition probabilities that requires that any state can be reached from any other. We note that under Assumption 3, if players follow any positively perturbed policy (i.e. perturbed with \(\rho^i > 0\) for each agent \(i\)) then any \((s, a) \in \mathcal{X} \times \mathcal{U}\) will be visited infinitely often with probability one.

With the preceding assumption, we now consider what happens to agent \(i\)'s Q-factors when \(i\) follows some perturbed policy \(\tilde{\pi}^i\) for some \(\pi^i \in \Pi^i\), its quantized set of policies, while the remaining players follow a policy \(\tilde{\pi}^{-i} \in \Pi^{-i}\).

Lemma 8 Let \(\pi \in \Pi\) be a given stationary joint policy and let \(\tilde{\pi}\) be the associated \(\rho\)-perturbed policy, with \(\rho^i > 0\) for each \(i\). For each \(i \in \mathcal{N}\), let \(Q^i \subset \mathbb{R}^{\mathcal{X} \times \mathcal{U}^i}\) be a compact set. Suppose Assumptions 3 and 5 hold. For any \(\eta > 0\), there exists \(T = T(\eta) \in \mathbb{N}\) such that if \(t \geq T\) and \(Q^i_0 \in Q^i\) for each \(i \in \mathcal{N}\), then

\[
\Pr_{\tilde{\pi}^i} \left( \|Q^i_t - Q^i_{\pi^i}\|_\infty \leq \eta, \forall i \in \mathcal{N} \right) \geq 1 - \eta.
\]

Proof: By Assumption 3, all states are visited infinitely often \(\Pr_{\tilde{\pi}}\)-almost surely. We then invoke Lemma 8 on each of the finitely many players in \(\mathcal{N}\) and the proof follows. (C.f. Lemma 2 in [12]
Appendix B).)

We now get a result analogous to Lemma 8 but pertaining to estimates of \(J^i_x\).

Assumption 6 For each agent \(i \in \mathcal{N}\), \(\{\alpha^i_t\}_{t \geq 0}\) is a sequence of step-size parameters in \([0, 1]\) satisfying \(\sum_{t \geq 0} \alpha^i_t = \infty\), and \(\sum_{t \geq 0} (\alpha^i_t)^2 < \infty\). Assume that each agent \(i\) updates its so-called J-factors according to the rule

\[
J^i_{t+1}(x) = \left(1 - \hat{\theta}^i_t(x)\right) J^i_t(x) + \hat{\theta}^i_t(x) \left( \epsilon^i(x, u^i_t, u^{-i}_t) + \beta^i J^i_t(x_{t+1}) \right),
\]

with

\[
\hat{\theta}^i_t(x) = \begin{cases} \alpha^i_{m^i_t}(x), & \text{if } x_t = x, \\ 0, & \text{otherwise}, \end{cases}
\]

where for each \(x \in \mathcal{X}\), \(m^i_t(x) := \sum_{k=0}^t \mathbb{1}_{\{k : x_k = x, \text{ for } 0 \leq k \leq t\}}\) counts the number of visits to \(x\) up to time \(t\).
Lemma 9 Let $\pi \in \Pi$ be a given stationary joint policy and let $\hat{\pi}$ be the associated $\rho$-perturbed policy, with $\rho^i > 0$ for each $i$. For each $i \in \mathcal{N}$, let $\mathcal{J}^i \subset \mathbb{R}^x$ be a compact set. Suppose Assumptions 3 and 6 hold. For any $\psi > 0$, there exists $\bar{T} = \bar{T}(\psi) \in \mathbb{N}$ such that if $t \geq \bar{T}$, then

$$\Pr[\hat{\pi} \left( |J^i_t - J^i_{\pi}(\hat{\pi})| \leq \psi, \forall i \in \mathcal{N}, x \in \mathcal{X} \right) \geq 1 - \psi,$$

for any $J^i_0 \in \mathcal{J}^i$.

Proof: Since $\rho^i > 0$ for each $i \in \mathcal{N}$, Assumption 3 guarantees that each $x \in \mathcal{X}$ is visited $\Pr[\hat{\pi}]$-almost surely. The result then follows by invoking Lemma 7 for each player $i \in \mathcal{N}$ and taking the maximum time over all (finitely many) players. Invoking Lemma 7 is possible as $i$ faces an MDP when $-i$ follows baseline policy $\pi^{-i}$ (that is to say, $-i$ actually follows policy $\pi^{-i}$).

Parameter Restrictions

With the same (fixed) $\epsilon > 0$ used in Assumption 1 define $\bar{\delta} = \bar{\delta}(\epsilon, \Pi)$ to be the minimum positive distance between $\epsilon$ and the amount by which some player’s policy is suboptimal. That is, $\bar{\delta} := \min \{S \setminus \{0\}, \}$ where

$$S := \left\{ \epsilon - \left( J^i_{\pi}(\pi^i), \pi^{-i} \right) - \min \{Q^i_{\pi^{-i}}(x, a^i) \} : i \in \mathcal{N}, \pi^{-i} \in \Pi^{-i}, x \in \mathcal{X} \right\}.$$

The set $S$ is finite, since $\mathcal{N}, \mathcal{X}$, and $\Pi$ are finite. We note that $\bar{\delta}$ depends on both $\epsilon > 0$ and the finite quantization $\Pi$.

$\bar{\delta}$ will serve as an upper bound for additive terms that each agent will use when determining whether they are $\epsilon$-best-responding, i.e. $\delta^i \in (0, \bar{\delta})$. We fix $\{\delta^i\}_{i \in \mathcal{N}}$ and introduce an upper bound $\bar{\rho}$ such that $\rho^i \leq \bar{\rho}$ for all $i \in \mathcal{N}$ implies

$$\|Q^i_{\pi^{-i}} - Q^i_{\pi^{-i}}\| < \frac{1}{2} \min_{j \in \mathcal{N}} \{\delta^j, \bar{\delta} - \delta^j\}, \forall i \in \mathcal{N}, \pi^{-i} \in \Pi^{-i}, \text{ and } (10)$$

$$\|J^i_{\pi}(\pi^{-i}) - J^i_{\pi}(\pi^{-i})\| < \frac{1}{2} \min_{j \in \mathcal{N}} \{\delta^j, \bar{\delta} - \delta^j\}, \forall i \in \mathcal{N}, \pi^{-i} \in \Pi^{-i}. \text{ (11)}$$

Such $\bar{\rho}$ exists by Lemmas 3 and 4.

Lemma 10 Let $\pi \in \Pi$ be a joint policy, let $\pi$ be its $\rho$-perturbed version, and let $\Pr[\pi]$ denote the associated probability measure. Suppose $\delta^i \in (0, \bar{\delta})$ and $\rho^i \in (0, \bar{\rho})$ for each $i \in \mathcal{N}$. For any $\psi > 0$, there exists $T = T(\psi)$ such that, if $t \geq T$, we have

$$\Pr[\pi \left( \|Q^i_t - Q^i_{\pi^{-i}}\|_\infty < \frac{1}{2} \min_{j \in \mathcal{N}} \{\delta^j, \bar{\delta} - \delta^j\}, \forall i \in \mathcal{N} \right) \geq 1 - \psi,$$

for any $Q^i_0 \in Q^i$.

Proof 1 This follows from Lemma 8 and (10). See also Lemma 4 in [12, Appendix B].

Lemma 11 Let $\pi \in \Pi$ be a joint policy, let $\pi$ be its $\rho$-perturbed version, and let $\Pr[\pi]$ denote the associated probability measure. Suppose $\delta^i \in (0, \bar{\delta})$ and $\rho^i \in (0, \bar{\rho})$ for each $i \in \mathcal{N}$. For any $\psi > 0$, there exists $\bar{T} = \bar{T}(\psi)$ such that, if $t \geq \bar{T}$, we have

$$\Pr[\pi \left( |J^i_t(\pi) - J^i_t(x)| < \frac{1}{2} \min_{j \in \mathcal{N}} \{\delta^j, \bar{\delta} - \delta^j\}, \forall i \in \mathcal{N}, x \in \mathcal{X} \right) \geq 1 - \psi,$$

for any $J^i_0 \in \mathcal{J}^i$.

Proof 2 This follows from Lemma 9 and (11).

Strictly speaking, the quantity $\bar{\rho}$ used in both of Lemmas 10 and 11 depends on the choice of $\{\delta^i\}_{i \in \mathcal{N}}$. We suppress this dependence for ease of notation.
C Proofs of main results

Lemma 12 Let $0 < p < u < 1$ and $y_0 \in [0, 1]$. For each $k \geq 0$, define $y_{k+1} = uy_k + p(1 - y_k)$. Then,

$$\lim_{k \to \infty} y_k = \frac{p}{1 - u + p}$$

Proof: We find an explicit form for $y_k = (u - p)^ky_0 + p \sum_{j=0}^{k-1}(u - p)^j$, and the result follows by taking limits.

Proof of Theorem 2

We begin by defining a number of events of interest. For any $k \geq 0$, the event $E^j_k$ is the event that $J$-factor learning in the $k$th exploration phase is accurate to our desired precision, i.e.

$$E^j_k := \left\{ \left| J_{k+1}(x) - J^*_x(\pi_k) \right| < \frac{1}{2} \min_{j \in \mathbb{N}} \{ \delta, \tilde{\delta} - \delta^j \}, \forall x \in \mathcal{X}, i \in \mathcal{N} \right\}.$$

The event $E^Q_k$ is defined analogously for Q-factors:

$$E^Q_k := \left\{ \left\| Q_{k+1}^i - Q^{\pi_k}_{\pi_k} \right\|_\infty < \frac{1}{2} \min_{j \in \mathbb{N}} \{ \delta, \tilde{\delta} - \delta^j \}, \forall i \in \mathcal{N} \right\}.$$

Then we define $E_k = E^j_k \cap E^Q_k$, and for $l \geq 1$ we define $E_{k:k+l} := E_k \cap E_{k+1} \cap \cdots \cap E_{k+l}$. Conditional on $E_k$, it holds that $\pi_k \in \text{BR}_k(\pi_k)$ if and only if $J_{k+1}(x) < \min_{\pi_k} Q_{k+1}^i(x, a^i) + \epsilon + \delta^j$ for every state $x \in \mathcal{X}$. This is true by the final remark of §23 and the construction of $\delta$. Thus, once $E_k$ is assumed, players correctly determine whether or not they are $\epsilon$-best-responding to the other players’ policy.

Recall the object $L$ from the proof of Lemma 5 for any $\pi_0 \in \Pi \setminus \Delta^\epsilon_{\pi_0}$, there exists an $\epsilon$-revision path into $\Pi \cap \Delta^\epsilon_{\pi_0}$, and therefore there exists a shortest such path, whose length is denoted $L_{\pi_0}$. We then take $L := \max \{ L_{\pi_0} : \pi_0 \in \Pi \}$.

In the proof of Lemma 5, we also introduced a quantity $p$, to lower bound the probability of transitioning from any initial policy into $\Pi \cap \Delta^\epsilon_{\pi_0}$ in $L$ steps. Here, when the Q-factors are noise perturbed, this lower bound may no longer hold, as the dynamics pushing play to equilibrium may place lower probability on each necessary transition. Nevertheless, another lower bound exists for the case where Q-factors are estimated:

$$\Pr \left( \pi_{k+L} \in \Pi \cap \Delta^\epsilon_{\pi_0} \mid E_k, E_{k+1}, \cdots, E_{k+L}, \pi_k \notin \Pi \cup \Delta^\epsilon_{\pi_0} \right) \geq \prod_{i \in \mathcal{N}} \left( \epsilon^i / \| \Pi^i \| \right)^L =: p_{\text{min}} > 0.$$

The preceding bound is explained as follows. Given any $\pi_0 \in \Pi \setminus \Delta^\epsilon_{\pi_0}$, take a shortest $\epsilon$-revision path into $\Pi \cap \Delta^\epsilon_{\pi_0}$. Such a path has length at most $L$, and at every step along this path (say the $(k + l)^{th}$ for $0 \leq l \leq L - 1$) each player either stays put when $\epsilon$-best-responding (which has conditional probability 1 when the event $E_{k+l}$ occurs) or the player might switch (or stay put) when not $\epsilon$-best-responding. The conditional probability of selecting the correct successor policy to follow this path, conditional on $E_{k+l}$ occurring, is then $\epsilon^i / \| \Pi^i \|$ for player $i$.

For ease of notation, let $B_k := \{ \pi \in \Pi \cap \Delta^\epsilon_{\pi_0} \}$, for $k \geq 0$. By the previous remarks, we have the following inequality, for any $k \geq 0$:

$$\Pr(B_{k+L}) = \Pr(B_{k+L} \mid B_k) \Pr(B_k) + \Pr(B_{k+L} \mid B_k^c) \Pr(B_k^c) \geq \Pr(B_{k+L} \mid B_k \cap E_{k:k+L}) \Pr(E_{k:k+L} \mid B_k) \Pr(B_k) + \Pr(B_{k+L} \mid B_k^c \cap E_{k:k+L}) \Pr(E_{k:k+L} \mid B_k^c) \Pr(B_k^c) \geq 1 \cdot \Pr(E_{k:k+L} \mid B_k) \Pr(B_k) + p_{\text{min}} \Pr(E_{k:k+L} \mid B_k^c) (1 - \Pr(B_k)).$$

We claim that $\Pr(E_{k:k+L} \mid B_k)$ and $\Pr(E_{k:k+L} \mid B_k^c)$ can be made arbitrarily high by taking the exploration phase lengths $T_k, \ldots, T_{k+L}$ to be suitably large. Indeed, the Q-factors and J-factors for
the $k^{th}$ exploration phase and onward are conditionally independent of the past up to time $t_k$, given the initial conditions, $Q_{t_k}^i, J_{t_k}^i, x_{t_k}$. Since Lemma 6 and Lemma 7 hold uniformly in their initial conditions (see [12, Appendix A] for more details), this claim holds by a union bound argument, making $\Pr(E_{k+l}|B_k)$ and $\Pr(E_{k+l}|B_k^c)$ suitably large for each $0 \leq l \leq L$. (The final remark is enabled by Lemmas 10 and 11.)

We now consider what happens when $\Pr(E_{k:k+L}|B_k) \geq u$ and $\Pr(E_{k:k+L}|B_k^c) \geq u$ for some $u \in (0, 1)$. For fixed $k \in \{0, 1, \ldots, L - 1\}$, define the following quantities:

$$h_k^{(m)} := \Pr(B_{k+mL}), \quad \text{for } m \geq 0$$

$$y_k^{(0)} := \Pr(B_k),$$

$$y_k^{(m+1)} := uy_k^{(m)} + up_{\text{min}}(1 - y_k^{(m)}), \quad \text{for } m \geq 0$$

We have that $h_{m+1}^{(k)} \geq uh_m^{(k)} + up_{\text{min}}(1 - h_m^{(k)})$ by the law of total probability, conditioning on whether $B_{k+mL}$ occurred. It is easy to verify, then, that $h_m^{(k)} \geq y_m^{(k)}$ for all $m \geq 0$. (Since $h_0^{(k)} = y_0^{(k)}$ and $h_{m+1}^{(k)} \geq (u - up_{\text{min}})h_m^{(k)} + up_{\text{min}} \geq (u - up_{\text{min}})y_m^{(k)} + up_{\text{min}} = y_{m+1}^{(k)}$.)

But then, by Lemma 12 with $u = u$ and $p = up_{\text{min}}$, we have that $\lim_{m \to \infty} y_m^{(k)} = \frac{up_{\text{min}}}{1 - u^* + up_{\text{min}}}$. Thus, we select $u^*$ sufficiently large that

$$\frac{u^*p_{\text{min}}}{1 - u^* + u^*p_{\text{min}}} > 1 - \frac{\psi}{2}.$$ 

Finally, by taking exploration phase lengths $\{T_k\}_{k \geq 0}$ large enough that $\Pr(E_{k:k+L}|B_k) \geq u^*$ and $\Pr(E_{k:k+L}|B_k^c) \geq u^*$ hold for all $k \geq 0$, we get our result. \qed
References

[1] P. Hernandez-Leal, M. Kaisers, T. Baarslag, and E. M. de Cote, “A survey of learning in multiagent environments: Dealing with non-stationarity,” arXiv preprint arXiv:1707.09183, 2017.

[2] M. Tan, “Multi-agent reinforcement learning: Independent vs. cooperative agents,” in Proceedings of the Tenth International Conference on Machine Learning, pp. 330–337, 1993.

[3] C. Claus and C. Boutilier, “The dynamics of reinforcement learning in cooperative multiagent systems,” in Proceedings of the Tenth Innovative Applications of Artificial Intelligence Conference, Madison, Wisconsin, pp. 746–752, 1998.

[4] K. Zhang, Z. Yang, and T. Başar, “Multi-agent reinforcement learning: A selective overview of theories and algorithms,” Handbook of Reinforcement Learning and Control, pp. 321–384, 2021.

[5] L. Matignon, G. J. Laurent, and N. Le Fort-Piat, “Independent reinforcement learners in cooperative markov games: a survey regarding coordination problems,” Knowledge Engineering Review, vol. 27, no. 1, pp. 1–31, 2012.

[6] L. Matignon, G. J. Laurent, and N. Le Fort-Piat, “Coordination of independent learners in cooperative Markov games,” HAL preprint hal-00370889, 2009.

[7] C. Daskalakis, D. J. Foster, and N. Golowich, “Independent policy gradient methods for competitive reinforcement learning,” arXiv preprint arXiv:2101.04233, 2021.

[8] E. Wei and S. Luke, “Lenient learning in independent-learner stochastic cooperative games,” The Journal of Machine Learning Research, vol. 17, no. 1, pp. 2914–2955, 2016.

[9] J. R. Marden, H. P. Young, G. Arslan, and J. S. Shamma, “Payoff-based dynamics for multiplayer weakly acyclic games,” SIAM Journal on Control and Optimization, vol. 48, no. 1, pp. 373–396, 2009.

[10] J. R. Marden and J. S. Shamma, “Revisiting log-linear learning: Asynchrony, completeness and payoff-based implementation,” Games and Economic Behavior, vol. 75, no. 2, pp. 788–808, 2012.

[11] D. P. Foster and H. P. Young, “Regret testing: Learning to play Nash equilibrium without knowing you have an opponent,” Theoretical Economics, vol. 1, pp. 341–367, 2006.

[12] G. Arslan and S. Yüksel, “Decentralized Q-learning for stochastic teams and games,” IEEE Transactions on Automatic Control, vol. 62, no. 4, pp. 1545–1558, 2017.

[13] G. Brown, “Iterative solutions of games by fictitious play,” in Activity Analysis of Production and Allocation (T. Koopmans, ed.), pp. 374–376, New York: Wiley, 1951.

[14] J. Robinson, “An iterative method of solving a game,” Annals of Mathematics, vol. 54, pp. 296–301, 1951.

[15] B. Swenson, C. Eksin, S. Kar, and A. Ribeiro, “Distributed inertial best-response dynamics,” IEEE Transactions on Automatic Control, vol. 63, no. 12, pp. 4294–4300, 2018.

[16] C. Eksin and A. Ribeiro, “Distributed fictitious play for multiagent systems in uncertain environments,” IEEE Transactions on Automatic Control, vol. 63, no. 4, pp. 1177–1184, 2017.

[17] S. Sen, M. Sekaran, and J. Hale, “Learning to coordinate without sharing information,” in Proceedings of the 12th National Conference on Artificial Intelligence, pp. 426–431, 1994.

[18] M. L. Littman, “Markov games as a framework for multi-agent reinforcement learning,” in Machine Learning Proceedings 1994, pp. 157–163, Elsevier, 1994.

[19] J. Hu and M. P. Wellman, “Nash q-learning for general-sum stochastic games,” Journal of Machine Learning Research, vol. 4, no. Nov, pp. 1039–1069, 2003.

[20] M. L. Littman, “Friend-or-foe Q-learning in general-sum games,” in ICML, vol. 1, pp. 322–328, 2001.

[21] M. Lauer and M. Riedmiller, “An algorithm for distributed reinforcement learning in cooperative multi-agent systems,” in In Proceedings of the Seventeenth International Conference on Machine Learning, Citeseer, 2000.
[22] M. Bowling and M. Veloso, “Multiagent learning using a variable learning rate,” *Artificial Intelligence*, vol. 136, pp. 215–250, 2002.

[23] F. Germano and G. Lugosi, “Global Nash convergence of Foster and Young’s regret testing,” *Games and Economic Behavior*, vol. 60, no. 1, pp. 135–154, 2007.

[24] G. C. Chasparis, A. Arapostathis, and J. S. Shamma, “Aspiration learning in coordination games,” *SIAM J. Control and Optimization*, vol. 51, no. 1, pp. 465–490, 2013.

[25] J. R. Marden, H. P. Young, and L. Y. Pao, “Achieving Pareto optimality through distributed learning,” *SIAM Journal on Control and Optimization*, vol. 52, no. 5, pp. 2753–2770, 2014.

[26] L. Matignon, G. J. Laurent, and N. Le Fort-Piat, “Hysteretic Q-learning: an algorithm for decentralized reinforcement learning in cooperative multi-agent teams,” in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 64–69, IEEE, 2007.

[27] M. O. Sayin, K. Zhang, D. S. Leslie, T. Basar, and A. Ozdaglar, “Decentralized Q-learning in zero-sum Markov games,” *arXiv preprint arXiv:2106.02748*, 2021.

[28] B. Yongacoglu, G. Arslan, and S. Yüksel, “Decentralized learning for optimality in stochastic dynamic teams and games with local control and global state information,” *IEEE Transactions on Automatic Control*, to appear.

[29] A. M. Fink, “Equilibrium in a stochastic n-person game,” *Journal of Science of the Hiroshima University, Series AI (Mathematics)*, vol. 28, no. 1, pp. 89–93, 1964.

[30] C. Watkins, *Learning from Delayed Rewards*. PhD thesis, Cambridge University, 1989.

[31] C. Watkins and P. Dayan, “Q-Learning,” *Machine Learning*, vol. 8, no. 3, pp. 279–292, 1992.

[32] J. N. Tsitsiklis, “Asynchronous stochastic approximation and Q-learning,” *Machine Learning*, vol. 16, no. 3, pp. 185–202, 1994.

[33] H. A. Simon, “Rational choice and the structure of the environment,” *Psychological review*, vol. 63, no. 2, p. 129, 1956.

[34] R. Radner, “Satisficing,” in *Optimization Techniques IFIP Technical Conference*, pp. 252–263, Springer, 1975.

[35] R. Cassidy, C. Field, and M. Kirby, “Solution of a satisficing model for random payoff games,” *Management Science*, vol. 19, no. 3, pp. 266–271, 1972.

[36] A. Charnes and W. W. Cooper, “Deterministic equivalents for optimizing and satisficing under chance constraints,” *Operations Research*, vol. 11, no. 1, pp. 18–39, 1963.

[37] O. Candogan, A. Ozdaglar, and P. A. Parrilo, “Near-potential games: Geometry and dynamics,” *ACM Transactions on Economics and Computation (TEAC)*, vol. 1, no. 2, pp. 1–32, 2013.

[38] R. Zhang, Z. Ren, and N. Li, “Gradient play in multi-agent Markov stochastic games: Stationary points and convergence,” *arXiv preprint arXiv:2106.00198*, 2021.

[39] P. Awasthi, M.-F. Balcan, A. Blum, O. Sheffet, and S. Vempala, “On Nash-equilibria of approximation-stable games,” in *International Symposium on Algorithmic Game Theory*, pp. 78–89, Springer, 2010.