Performance Tradeoffs in Distributed Control Systems

by

Holly Borowski

B.S., U.S. Air Force Academy, 2004

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Aerospace Engineering Sciences

2016

This thesis entitled:
Performance Tradeoffs in Distributed Control Systems
written by Holly Borowski
has been approved for the Aerospace Engineering Sciences

Prof. Jason Marden

Prof. Eric Frew

add other readers

Date

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Borowski, Holly (Ph.D., Aerospace Engineering Sciences)

Performance Tradeoffs in Distributed Control Systems

Thesis directed by Prof. Jason Marden

Often the abstract will be long enough to require more than one page, in which case the macro “\OnePageChapter” should *not* be used.

But this one isn’t, so it should.

Dedication

To all of the fluffy kitties.

Acknowledgements

Here's where you acknowledge folks who helped. But keep it short, i.e., no more than one page, as required by the Grad School Specifications.

Contents

Chapter

Appendix

Todo list

Scan for phrases like ”this paper” and reword

Chapter 1

Introduction

Large scale systems consisting of many interacting subsystems are increasingly common, with applications such as computer networks, smart grids, and robotic sensor networks. Objectives for these systems are often complex; even centralized methods must make tradeoffs between performance and speed in order to optimize these non-convex, nonlinear systems. However, due to inherent limitations in computation, communication, or sensing, large multi-agent systems are often controlled in a distributed fashion. Here, individual agents must make decisions based on local, often incomplete information, potentially exacerbating tradeoffs between performance and speed. Furthermore, the distributed nature of such systems may introduce vulnerabilities to adversarial manipulation: by influencing small subsets of agents, a malicious agent may be able to degrade a system's overall performance.

The overarching goals of this dissertation are to (1) characterize tradeoffs between speed, performance, vulnerability, and information in distributed control systems, and (2) design algorithms with desirable guarantees in these aspects. In order to address these goals, this dissertation focuses specifically on the following questions:

try and fit these questions on the first page

- **When is fast convergence to near-optimal behavior possible?** (Chapter 2)

- **Can agents learn near-optimal correlated behavior despite severely limited information about one another's behavior?** (Chapter 3)

- **How does the structure of agent interaction impact the system's vulnerability to adversarial manipulation?** (Chapter 4)

1.1 Informal statement of contributions

This dissertations focuses on game theoretic methods for distributed control. Game theoretic learning algorithms have gained traction as a design tool for distributed control systems [42, 64, 24, 56, 20]. Here, a static game is repeated over time, and agents revise their strategies based on their objective functions and on observations of other agents' behavior.

The following informally summarizes this dissertation's primary contributions for game theoretic distributed control, with respect to the three research questions posed above.

Research question: When is fast convergence to near-optimal behavior possible?

Contribution: Fast convergence is possible with limited agent heterogeneity, provided agents revise their behavior according to a mild variant of log-linear learning.

Emergent collective behavior for game theoretic learning rules has been studied extensively in the literature, e.g., fictitious play [46, 19, 37], regret matching [27], and log-linear learning [1, 9, 55]. Although many of these learning rules have desirable asymptotic guarantees, their convergence times either remain uncharacterized or are prohibitively long [14, 31, 55, 26]. Characterizing convergence rates is key to determining whether a distributed algorithm is desirable for system control.

One well-studied method of optimizing behavior in a multi-agent system is to assign agent utilities to be their *marginal contribution* to the overall system objective [59], and have agents make decisions according to the *log-linear learning* rule [9]. In log-linear learning, agents primarily choose utility maximizing actions, but choose suboptimally with a small probability that decreases exponentially with respect to the associated payoff loss. In the long run, marginal contribution utilities with log-linear learning dynamics spends the majority of time at the global objective function maximizer. Unfortunately, worst-case convergence times for log-linear learning are known to be exponential in the number of agents, [55] often rendering this game theoretic control method impractical for large-scale distributed systems.

However, when system heterogeneity is limited, i.e., agents can be grouped into a small number of populations according to their action sets and impact on the objective function, a variant of log-linear learning achieves improved worst-case convergence times. In Chapter 2, I build on the work of [55] to derive this variant, which converges in near-linear time with respect to the number of agents.

Research question: Can agents learn near-optimal correlated behavior despite severely limited information about one another's behavior?

Contribution: Following the algorithm in Chapter 3, agents with limited knowledge of each other's behavior can learn a utility-maximizing correlated equilibrium.

Here, we focus on the scenario where the system objective is to maximize the sum of agents' payoffs. This type of objective can be useful when we wish to balance multiple local objectives.

Agents' average utilities can often be improved when they act according to a distribution over multiple joint actions, instead of staying fixed at a single joint action. In many cases, the desired collective behavior constitutes a *coarse correlated equilibrium.* A coarse correlated equilibrium is a probability distribution over the joint action space such that no agent can improve its payoff by deviating to a fixed action [6]. Previously, algorithms existed which converged to the *set* of coarse correlated equilibrium, e.g., [27], without selecting any particular equilibrium; these algorithms provided no performance guarantees. An algorithm which achieves a payoff maximizing coarse correlated equilibrium through deterministic, cyclic behavior is presented in [38]. However, predictable cyclic behavior may be undesirable in many settings, e.g., in the presence of an adversary.

In Chapter 3 I design and analyze an algorithm that converges to a payoff maximizing coarse correlated equilibrium when agents have no knowledge of others' behavior. Here, agents' utilities depend on collective behavior, but they have no way of evaluating the utility of alternative actions. My algorithm uses a common random signal as a coordinating entity to eventually drive agents toward the desired collective behavior. In the long run, day to day behavior is selected probabilistically according to the payoff maximizing coarse correlated equilibrium.

Research question: How does the structure of agent interaction impact a distributed system's vulnerability to adversarial manipulation?

Contribution: If every subset of agent interacts with sufficiently many other agents, the system is more resilient to adversarial manipulation.

Agents in a distributed system often interact and share information according to a network. The structure of this network not only has an impact on a distributed control algorithm's performance and speed, but also on its resilience to adversarial manipulation. A loosely connected network may be easier to influence, because an adversary may be able to more easily manipulate the information available to subsets of agents, thereby creating impacts that cascade throughout the system. On the other hand, a well-connected network may be more difficult to influence in this way. In Chapter 4, I investigate such vulnerabilities for *graphical coordination games* [57, 13] with agents revising their actions according to log-linear learning. In this work, I provided a condition based on network structure which guarantees resilience in a graphical coordination game.

1.2 Technical Preliminaries

OK, i took out the utility function + learning rule table, because I think it's distracting. The majority of my work isn't specific to resource allocation games, and doesn't do anything with utility design, so I don't want to discuss it too much. Think about this some more.

(1) the setting: a distributed control system. agents, actions, objective function

(2) Our approach: game theoretic control

  (a) Utility function that maps local information to a utility/payoff. Give a small preview of the types of utility function we use. Appropriate design varies depending on the application. My focus was on learning algorithms, so typically I assigned utilities that made analysis simple in order to gain intuition about the question I was trying to ask. Not sure whether to say this or how to convey it....

(b) Learning rule that dictates agents' decisions. Our focus is on probabilistic learning rules. Reference Markov chains section in the appendix. Discrete time, so this yields a sequence of joint actions $a(0), a(1), \ldots$. Small discussion that these are random variables, perhaps. Include an example (LLL) and state that the majority of the dissertation will focus on LLL, with the exception of the CCE work.

(3) Goal (in general): design utilities and learning rules with desired performance

(a) Desired long run performance. Given parameter $\varepsilon > 0$, should be able to tune learning algorithm parameters to achieve:

$$\lim_{t\to\infty} \mathbb{E}[W(a(t)] \geq \max_{a\in\mathcal{A}} W(a) - \varepsilon$$

(b) Fast convergence rates. Given $\delta > \varepsilon$, desired

$$\mathbb{E}[W(a(t))] \geq \max_{a\in\mathcal{A}} W(a) - \delta$$

for all $t \geq T$, for some "reasonable" convergence time T. Discussion here that T typically depends on the number of agents, action sets, etc., and can often be exponential in these parameters. Goal here is to design a learning rule that's linear. Reference an example and log-linear learning.

(c) Resilience to adversaries. Suppose an adversary is able to manipulate information available to small subsets of agents. Is it possible for limited info manipulation to cascade and create catastrophic performance failures? Goal is for it not to be.

1.3 Formal statement of contributions

Relate to the goals above. State each main theorem as a contribution here. First a reminder of the three main questions.

1.3.1 Fast convergence contributions

1.3.2 CCE contributions

1.3.3 Adversarial vulnerability contributions

Add large “future work” section. There’s a lot left to do!

Chapter 2

Fast Convergence in Semi-Anonymous Potential Games

When is fast convergence to a near optimal solution possible?

Emergent collective behavior for game theoretic learning rules has been studied extensively in the literature, e.g., fictitious play [46, 19, 37], regret matching [27], and log-linear learning [1, 9, 55]. Although many of these learning rules have desirable asymptotic guarantees, their convergence times either remain uncharacterized or are prohibitively long [14, 31, 55, 26]. Characterizing convergence rates is key to determining whether a distributed algorithm is desirable for system control.

In many multi-agent systems, the agent objective functions can be designed to align with the system-level objective function, yielding a **potential game** [47] whose potential function is precisely the system objective function. Here, the optimal collective behavior of a multi-agent system corresponds to the Nash equilibrium that optimizes the potential function. Hence, learning algorithms which converge to this efficient Nash equilibrium have proven useful for distributed control.

Log-linear learning is one algorithm that accomplishes this task [9]. Log-linear learning is a perturbed best reply process where agents predominantly select the optimal action given their beliefs about other agents' behavior; however, the agents occasionally make mistakes, selecting suboptimal actions with a probability that decays exponentially with respect to the associated payoff loss. As noise levels approach zero, the resulting process has a unique stationary distribution with full support on the efficient Nash equilibria. By designing agents' objective functions appropriately, log-linear learning can be used to define distributed control laws which converge to optimal steady-state behavior in the long run.

Unfortunately, worst-case convergence rates associated with log-linear learning are exponential in the game size [55]. This stems from inherent tension between desirable asymptotic behavior and convergence rates. The tension arises because small noise levels are necessary to ensure that the mass of the stationary distribution lies primarily on the efficient Nash equilibria; however, small noise levels also make it difficult to exit inefficient Nash equilibria, degrading convergence times.

Positive convergence rate results for log-linear learning and its variants are beginning to emerge for specific game structures [48, 33, 55, 4]. For example, in [48] the authors study the convergence rates of log-linear learning for a class of coordination games played over graphs. They demonstrate that underlying convergence rates are desirable provided that the interaction graph and its subgraphs are sufficiently sparse. Alternatively, in [55] the authors introduce a variant of log-linear learning and show that convergence times grow roughly linearly in the number of players for a special class of congestion games over parallel networks. They also show that convergence times remain linear in the number of players when players are permitted to exit and enter the game. Although these results are encouraging, the restriction to parallel networks is severe and hinders the applicability of such results to distributed engineering systems.

We focus on identifying whether the positive convergence rate results above extend beyond symmetric congestion games over parallel networks to games of a more general structure relevant to distributed engineering systems. Such guarantees are not automatic because there are many simplifying attributes associated with symmetric congestion games that do not extend in general (see Example 2). The main contributions of this chapter are as follows:

– We formally define a subclass of potential games, called **semi-anonymous potential games**, which are parameterized by populations of agents where each agent's objective function can be evaluated using only information regarding the agent's own decision and the aggregate behavior within each population. Agents within a given population have identical action sets, and their objective functions share the same structural form. The congestion games studied in [55] could be viewed as a semi-anonymous potential game with only one population.[1]

– We introduce a variant of log-learning learning that extends the algorithm in [55]. In Theorem 1, we prove that the convergence time of this algorithm grows roughly linearly in the number of agents for a fixed number of populations. This analysis explicitly highlights the potential impact of system-wide heterogeneity, i.e., agents with different action sets or objective functions, on the convergence rates. Furthermore, in Example 4 we demonstrate how a given resource allocation problem can be modeled as a semi-anonymous potential game.

– We study the convergence times associated with our modified log-linear learning algorithm when the agents continually enter and exit the game. In Theorem 2, we prove that the convergence time of this algorithm remains roughly linear in the number of agents provided that the agents exit and enter the game at a sufficiently slow rate.

The forthcoming analysis is similar in structure to the analysis presented in [55]. We highlight the explicit differences between the two proof approaches throughout, and directly reference lemmas within [55] when appropriate. The central challenge in adapting and extending the proof in [55] to the setting of semi-anonymous potential games is dealing with the growth of the underlying state space. Note that the state space in [55] is characterized by the aggregate behavior of a single population while the state space in our setting is characterized by the Cartesian product of the aggregate behavior associated with several populations. The challenge arises from the fact that the employed techniques for analyzing the mixing times of this process, i.e., Sobolev constants, rely heavily on the structure of this underlying state space.

2.1 Semi-Anonymous Potential Games

Consider a game with agents $N = \{1, 2, \ldots, n\}$. Each agent $i \in N$ has a finite action set denoted by $\mathcal{A}_i$ and a utility function $U_i : \mathcal{A} \to \mathbb{R}$, where $\mathcal{A} = \prod_{i \in N} \mathcal{A}_i$ denotes the set of joint actions. We express an action profile $a \in \mathcal{A}$ as (a_i, a_{-i}) where $a_{-i} = (a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n)$

[1] Semi-anonymous potential games can be viewed as a cross between a potential game and a finite population game [10].

denotes the actions of all agents other than agent i. We denote a game G by the tuple $G = (N, \{\mathcal{A}_i\}_{i \in N}, \{U_i\}_{i \in N})^2$.

Definition 1. A game G is a semi-anonymous potential game if there exists a partition $\mathcal{N} = (N_1, N_2, \dots, N_m)$ of N such that the following conditions are satisfied:

(i) For any population $N_\ell \in \mathcal{N}$ and agents $i, j \in N_\ell$ we have $\mathcal{A}_i = \mathcal{A}_j$. Accordingly, we say population N_ℓ has action set $\bar{\mathcal{A}}_\ell = \{\bar{a}_\ell^1, \bar{a}_\ell^2, \dots, \bar{a}_\ell^{s_\ell}\}^3$ where s_ℓ denotes the number of actions available to population N_ℓ. For simplicity, let $p(i) \in \{1, \dots, m\}$ denote the index of the population associated with agent i. Then, $\mathcal{A}_i = \bar{\mathcal{A}}_{p(i)}$ for all agents $i \in N$.

(ii) For any population $N_\ell \in \mathcal{N}$, let

$$X_\ell = \left\{ \left(\frac{v_\ell^1}{n}, \frac{v_\ell^2}{n}, \dots, \frac{v_\ell^{s_\ell}}{n} \right) \geq \mathbf{0} \, : \, \sum_{k=1}^{s_\ell} v_\ell^k = |N_\ell| \right\} \tag{2.1}$$

represent all possible aggregate action assignments for the agents within population N_ℓ. Here, the utility function of any agent $i \in N_\ell$ can be expressed as a lower-dimensional function of the form $\bar{U}_i : \bar{\mathcal{A}}_{p(i)} \times X \to \mathbb{R}$ where $X = X_1 \times \cdots \times X_m$. More specifically, the utility associated with agent i for an action profile $a = (a_i, a_{-i}) \in \mathcal{A}$ is of the form

$$U_i(a) = \bar{U}_i(a_i, a|_X)$$

where

$$a|_X = (a|_{X_1}, a|_{X_2}, \dots, a|_{X_m}) \in X, \tag{2.2}$$

$$a|_{X_j} = \frac{1}{n} \left\{ \left| \{ j \in N_\ell \, : \, a_j = \bar{a}_\ell^k \} \right| \right\}_{k=1,\dots,s_\ell} . \tag{2.3}$$

The operator $\cdot|_X$ captures each population's aggregate behavior in an action profile $\cdot$.

(iii) There exists a potential function $\phi : X \to \mathbb{R}$ such that for any $a \in \mathcal{A}$ and agent $i \in N$ with action $a_i' \in \mathcal{A}_i$,

$$U_i(a) - U_i(a_i', a_{-i}) = \phi(a|_X) - \phi((a_i', a_{-i})|_X). \tag{2.4}$$

[2] For brevity, we refer to G by $G = (N, \{\mathcal{A}_i\}, \{U_i\})$.

[3] We use the notation $\bar{\mathcal{A}}_\ell$ to represent the action set of the ℓth population, whereas $\mathcal{A}_i$ represents the action set of the ith agent.

If each agent $i \in N$ is alone in its respective partition, the definition of semi-anonymous potential games is equivalent to that of exact potential games in [47].

Example 1 (Congestion Games [8])**.** Consider a congestion game with players $N = \{1, \ldots, n\}$ and roads $R = \{r_1, r_2, \ldots, r_k\}$. Each road $r \in R$ is associated with a congestion function $C_r : \mathbb{Z}_+ \to \mathbb{R}$, where $C_r(k)$ is the congestion on road r with k total users. The action set of each player $i \in N$ represents the set of paths connecting player i's source and destination, and has the form $\mathcal{A}_i \subseteq 2^R$. The utility function of each player $i \in N$ is given by

$$U_i(a_i, a_{-i}) = -\sum_{r \in a_i} C_r(|a|_r),$$

where $|a|_r = |\{j \in N \,:\, r \in a_j\}|$ is the number of players in joint action a whose path contains road r. This game is a potential game with potential function $\phi : \mathcal{X} \to \mathbb{R}$

$$\phi(a|_X) = -\sum_{r \in R} \sum_{k=1}^{|a|_r} C_r(k). \tag{2.5}$$

When the players' action sets are symmetric, i.e., $\mathcal{A}_i = \mathcal{A}_j$ for all agents $i, j \in N$, then a congestion game is a semi-anonymous potential game with a single population. Such games, also referred to as anonymous potential games, are the focus of [55]. When the players' action sets are asymmetric, i.e., $\mathcal{A}_i \neq \mathcal{A}_j$ for at least one pair of agents $i, j \in N$, then a congestion game is a semi-anonymous potential game where populations consist of agents with identical path choices. The results in [55] are not proven to hold for such settings.

The following example highlights issues that arise when transitioning from a single population to multiple populations.

Example 2. Consider a resource allocation game with n players and three resources, $R = \{r_1, r_2, r_3\}$. Let n be even and divide players evenly into populations N_1 and N_2. Suppose that players in N_1 may select exactly one resource from $\{r_1, r_2\}$, and players in N_2 may select exactly one resource from $\{r_2, r_3\}$. The welfare garnered at each resource depends on how many players have selected that resource; the resource-specific welfare functions are

$$W_{r_1}(k) = 2k, \quad W_{r_2}(k) = \min\left\{3k, \frac{3}{2}n\right\}, \quad W_{r_3}(k) = k.$$

where $k \in \{0, 1, \ldots, n\}$ represents the number of agents selecting a given resource. The total system welfare is

$$W(a) = \sum_{r \in R} W_r(|a|_r)$$

for any $a \in \mathcal{A}$, where $|a|_r$ represents the number of agents selecting resource r under action profile a. Assign each agent's utility according to its marginal contribution to the system-level welfare: for agent i and action profile a

$$U_i(a) = W(a) - W(\emptyset, a_{-i}) \tag{2.6}$$

where $\emptyset$ indicates that player i did not select a resource. The marginal contribution utility in (2.6) ensures that the resulting game is a potential game with potential function W [59].

If the agents had symmetric action sets, i.e., if $\mathcal{A}_i = \{r_1, r_2, r_3\}$ for all $i \in N$, then this game has exactly one Nash equilibrium with $n/2$ players at resource r_1 and $n/2$ players at resource r_2. This Nash equilibrium corresponds to the optimal allocation.

In contrast, the two population scenario above has many Nash equilibria, two of which are: (i) an optimal Nash equilibrium in which all players from N_1 select resource r_1 and all players from N_2 select resource r_2, and (ii) a suboptimal Nash equilibrium in which all players from N_1 select resource r_2 and all players from N_2 select resource r_3. This large number of equilibria will significantly slow any equilibrium selection process, such as log-linear learning and its variants.

2.2 Main Results

Example 2 invites the question: can a small amount of heterogeneity break down the fast convergence results of [55]? In this section, we present a variant of log-linear learning [9] that extends the algorithm for single populations in [55]. In Theorem 1 we prove that for any semi-anonymous potential game our algorithm ensures (i) the potential associated with asymptotic behavior is close to the maximum and (ii) the convergence time grows roughly linearly in the number of agents for a fixed number of populations. In Theorem 2 we show that these guarantees still hold when agents are permitted to enter and exit the game. An algorithm which converges quickly to the potential function maximizer is useful for multi-agent systems because agent objective functions can often be designed so that the potential function is identical to the system objective function as in Example 2.

2.2.1 Modified Log-Linear Learning

The following modification of the log-linear learning algorithm extends the algorithm in [55]. Let $a(t) \in \mathcal{A}$ be the joint action at time $t \geq 0$. Each agent $i \in N$ updates its action upon ticks of a Poisson clock with rate $\alpha n / z_i(t)$, where

$$z_i(t) = |\{k \in N_{p(i)} \, : \, a_k(t) = a_i(t)\}|,$$

and $\alpha > 0$ is a design parameter which dictates the expected total update rate. A player's update rate is higher if he is not using a common action within his population. To continually modify his clock rate, each player must know the value of $z_i(t)$, i.e., the number of players within his population sharing his action choice, for all $t \in \mathbb{R}$. In many cases, agents also need this information to evaluate their utilities, e.g., when players' utilities are their marginal contribution to the total welfare, as in Example 2.

When player i's clock ticks, he chooses action $a_i \in \bar{\mathcal{A}}_{p(i)} = \mathcal{A}_i$ probabilistically according to

$$\begin{aligned} \text{Prob}[a_i(t^+) = a_i \;\big|\; a(t)] &= \frac{e^{\beta U_i(a_i, a_{-i}(t))}}{\sum_{a_i' \in \mathcal{A}_i} e^{\beta U_i(a_i', a_{-i}(t))}} \\ &= \frac{e^{\beta \phi(a(t)|\mathcal{X})}}{\sum_{a_i' \in \mathcal{A}_i} e^{\beta \phi((a_i', a_{-i}(t))|\mathcal{X})}}, \end{aligned} \tag{2.7}$$

for any $a_i \in \mathcal{A}_i$, where $a_i(t^+)$ indicates the agent's revised action and β is a design parameter that determines how likely an agent is to choose a high payoff action. As $\beta \to \infty$, payoff maximizing actions are chosen, and as $\beta \to 0$, agents choose from their action sets with uniform probability. The new joint action is of the form $a(t^+) = (a_i(t^+), a_{-i}(t)) \in \mathcal{A}$, where $t \in \mathbb{R}^+$ is the time immediately before agent i's update occurs. For a discrete time implementation of this algorithm and a comparison with the algorithm in [55], please see Appendix B.1.1.

The expected number of updates per second for the continuous time implementation of our modified log-linear learning algorithm is lower bounded by $m\alpha n$ and upper bounded by $(|\bar{\mathcal{A}}_1| + \cdots +$

$|\bar{\mathcal{A}}_m|)\alpha n$. To achieve an expected update rate at least as fast as the standard log-linear learning update rate, i.e., at least n per second, we set $\alpha \geq 1/m$. These dynamics define an ergodic, reversible Markov process for any $\alpha > 0$.

2.2.2 Semi-Anonymous Potential Games

Theorem 1 bounds the convergence time for modified log-linear learning in a semi-anonymous potential game and extends the results of [55] to semi-anonymous potential games. For notational simplicity, define $s := |\cup_{j=1}^{m} \overline{\mathcal{A}}_j|$.

Theorem 1. *Let $G = (N, \{\mathcal{A}_i\}, \{U_i\})$ be a semi-anonymous potential game with aggregate state space X and potential function $\phi : X \to [0,1]$. Suppose agents play according to the modified log-linear learning algorithm described above, and the following conditions are met:*

(i) The potential function is λ-Lipschitz, i.e., there exists $\lambda \geq 0$ such that

$$|\phi(x) - \phi(y)| \leq \lambda \|x - y\|_1, \quad \forall x, y \in X.$$

(ii) The number of players within each population is sufficiently large:

$$\sum_{i=1}^{m} |N_i|^2 \geq \sum_{i=1}^{m} |\bar{\mathcal{A}}_i| - m.$$

For any fixed $\varepsilon \in (0,1)$, if β is sufficiently large, i.e.,

$$\beta \geq \max\left\{ \frac{4m(s-1)}{\varepsilon} \log 2ms, \frac{4m(s-1)}{\varepsilon} \log \frac{8ms\lambda}{\varepsilon} \right\}, \tag{2.8}$$

then

$$\mathbb{E}[\phi(a(t)|_X)] \geq \max_{x \in X} \phi(x) - \varepsilon \tag{2.9}$$

for all

$$t \geq \frac{2^{2ms} c_1 e^{3\beta} m (m(s-1))!^2 n}{4\alpha} \left(\log\log(n+1)^{ms-m} + \log\beta + 2\log\frac{1}{\varepsilon} \right) \tag{2.10}$$

where c_1 is a constant that depends only on s.

We prove Theorem 1 in Appendix B.1.2. This theorem explicitly highlights the role of system heterogeneity, i.e., $m > 1$ distinct populations, on convergence times of the process. For the case when $m = 1$, Theorem 1 recovers the results of [55]. Observe that for a fixed number of populations, the convergence time grows as $n \log \log n$. Furthermore, note that a small amount of system heterogeneity does not have a catastrophic impact on worst-case convergence times as suggested by Example 2.

It is important to note that our bound is exponential in the number of populations and in the total number of actions. Therefore our results do not guarantee fast convergence with respect to these parameters. However, our convergence rate bounds may be conservative in this regard. Furthermore, as we will show in Section 2.3, a significantly smaller value of β may often be chosen in order to further speed convergence while still retaining the asymptotic properties guaranteed in (2.9).

2.2.3 Time Varying Semi-Anonymous Potential Games

In this section, we consider a trajectory of semi-anonymous potential games to model the scenario where agents enter and exit the system over time,

$$\mathcal{G} = \{G^t\}_{t\geq 0} = \{N^t, \{\mathcal{A}_i^t\}_{i\in N^t}, \{U_i^t\}_{i\in N^t}\}_{t\geq 0}$$

where, for all $t \in \mathbb{R}^+$, the game G^t is a semi-anonymous potential game, and the set of **active** players, N^t, is a finite subset of $\mathbb{N}$. We refer to each agent $i \in \mathbb{N} \setminus N^t$ as **inactive**; an inactive agent has action set $\mathcal{A}_i^t = \emptyset$ at time t. Define $\mathcal{X} := \cup_{t\in\mathbb{R}^+} X^t$, where X^t is the finite aggregate state space corresponding to game G^t. At time t, denote the partitioning of players per Definition 1 by $\mathcal{N}^t = \{N_1^t, N_2^t, \ldots, N_m^t\}$. We require that there is a fixed number of populations, m, for all time, and that the j-th population's action set is constant, i.e., $\forall j \in \{1, 2, \ldots, m\}$, $\forall t_1, t_2 \in \mathbb{R}^+$, $\bar{\mathcal{A}}_j^{t_1} = \bar{\mathcal{A}}_j^{t_2}$. We write the fixed action set for players in the j-th population as $\bar{\mathcal{A}}_j$.

Theorem 2. *Let $\mathcal{G}$ be a trajectory of semi-anonymous potential games with state space $\mathcal{X}$ and time-invariant potential function $\phi : \mathcal{X} \to [0, 1]$. Suppose agents play according to the modified* *log-linear learning algorithm and Conditions (i) and (ii) of Theorem 1 are satisfied. Fix $\varepsilon \in (0,1)$, assume the parameter β satisfies (2.8) and the following additional conditions are met:*

(iii) for all $t \in \mathbb{R}^+$, the number of players satisfies:

$$|N^t| \geq \max\left\{ \frac{4\alpha m e^{-3\beta}}{2^{2ms} c_1 m^2 (m(s-1))!^2}, 2\beta\lambda + 1 \right\}, \tag{2.11}$$

(iv) there exists $k > 0$ such that

$$|N_i^t| \geq |N^t| \,/\, k, \quad \forall i \in \{1, 2, \ldots, m\},\ \forall t \in \mathbb{R}^+, \tag{2.12}$$

(v) there exists a constant

$$\Lambda \geq 8 c_0 \varepsilon^{-2} e^{3\beta} (6\beta\lambda + e^{\beta} k(s-1)) \tag{2.13}$$

such that for any t_1, t_2 with $|t_1 - t_2| \leq \Lambda$,

$$\left|\left\{ i \in N^{t_1} \cup N^{t_2} \,:\, \mathcal{A}_i^{t_1} \neq \mathcal{A}_i^{t_1} \right\}\right| \leq 1, \tag{2.14}$$

and, if $i \in N^{t_1} \cap N^{t_2}$, then $i \in N_j^t$ for some $j \in \{1, \ldots, m\}$ and for all time $t \in [t_1, t_2]$, i.e., agents may not switch populations over this interval. Here, c_0 and c_1 do not depend on the number of players, and hence the constant Λ does not depend on n.

Then,

$$\mathbb{E}[\phi(a(t)|_{\mathcal{X}})] \geq \max_{x \in X(t)} \phi(x) - \varepsilon \tag{2.15}$$

for all

$$t \geq |N^0| e^{3\beta} c_0 \left(\frac{(ms-m)! \log(|N^0| + 2) + \beta}{\varepsilon^2} \right). \tag{2.16}$$

Theorem 2 states that, if player entry and exit rates are sufficiently slow as in Condition (v), then the convergence time of our algorithm is roughly linear in the number of players. However, the established bound grows quickly with the number of populations. Note that selection of parameter β impacts convergence time, as reflected in (2.16): larger β tends to slow convergence. However, the minimum β necessary to achieve an expected potential near the maximum, as in (2.15), is independent of the number of players, as given in (2.8). The proof of Theorem 2 follows a similar structure to the proof of Theorem 4 in [55] and is hence omitted for brevity. The significant technical differences arise due to differences in the size of the state space when $m > 1$. These differences give rise to Condition (iv) in our theorem.

2.3 Illustrative Examples

In this section, we consider resource allocation games with a similar structure to Example 2. In each case, agents' utility functions are defined by their marginal contribution to the system welfare, W, as in (2.6). Hence, each example is a potential game with potential function W.

Modified log-linear learning defines an ergodic, continuous time Markov chain; we denote its transition kernel by P and its stationary distribution by π. For relevant preliminaries on Markov chains, please refer to Appendix A.1, and for a precise definition of the transition kernel and stationary distribution associated with modified log-linear learning, please refer to Appendices B.1.1 and B.1.2.

Unless otherwise specified, we consider games with n players distributed evenly into populations N_1 and N_2. There are three resources, $R = \{r_1, r_2, r_3\}$. Players in population N_1 may choose a single resource from $\{r_1, r_2\}$ and players in population N_2 may choose a single resource from $\{r_2, r_3\}$. We represent a state by

$$x = \left(x_1^1, x_2^1, x_2^2, x_3^2\right), \tag{2.17}$$

where nx_1^1 and nx_2^1 are the numbers of players from N_1 choosing resources r_1 and r_2. Likewise, nx_2^2 and nx_3^2 are the numbers of players from N_2 choosing resources r_2 and r_3 respectively. Welfare functions for each resource depend only on the number of players choosing that resource, and are specified in each example. The system welfare for a given state is the sum of the welfare garnered at each resource, i.e.,

$$W(x) = W_{r_1}(nx_1^1) + W_{r_2}(n(x_2^1 + x_2^2)) + W_{r_3}(nx_3^2).$$

Player utilities are their marginal contribution to the total welfare, W, as in (2.6).

In Example 3, we directly the compute convergence times as in Theorem 1:

$$\min\{t \,:\, \mathbb{E}_{P^t(y,\cdot)}W(x) \geq \max_{x\in X} W(x) - \varepsilon,\, \forall y \in X\}, \tag{2.18}$$

for modified log-linear learning, the variant of [55], and standard log-linear learning. This direct analysis is possible due to the example's relatively small state space.

Example 3. Here, we compare convergence times of our log-linear learning variant, the variant of [55], and standard log-linear learning. The transition kernels for each process are described in detail in Appendix B.1.1.

Starting with the setup described above, we add a third population, N_3. Agents in population N_3 contribute nothing to the system welfare and may only choose resource r_2. Because the actions of agents in population N_3 are fixed, we represent states by aggregate actions of agents in populations N_1 and N_2 as in (2.17). The three resources have the following welfare functions for each $x = \left(x_1^1, x_2^1, x_2^2, x_3^2\right) \in X$:

$$\begin{aligned}
&W_{r_1}(x) = 2nx_1^1,\\
&W_{r_2}(x) = \min\left\{3(nx_1^1 + nx_1^2), \frac{3}{2}(nx_2^1 + nx_2^2)\right\},\\
&W_{r_3}(x) = nx_3^2.
\end{aligned}$$

Our goal in this example is to achieve an expected total welfare that is within 98% of the maximum welfare.

We fix the number of players in populations N_1 and N_2 at $n_1 = n_2 = 7$, and vary the number of players in population n_3 to examine the sensitivity of each algorithm's convergence rate to the size of N_3.

In our variant of log linear learning, increasing the size of population N_3 does not change the probability that a player from population N_1 or N_2 will update next. However, for standard log-linear learning and for the variant in [55], increasing the size of population N_3 significantly decreases the probability that players from N_1 or N_2 who are currently choosing resource r_2 will be selected for update.[4]

We select β in all cases so that, as $t \to \infty$, the expected welfare associated with the resulting stationary distribution is within 98% of its maximum. Then we examine the time it takes to come within $\varepsilon = 0.05$ of this expected welfare. We multiply convergence times by the number of players, n, to analyze the expected number of updates required to reach the desired welfare. These numbers represent the convergence times when the expected total number of updates per unit time is held constant as n increases. Table 3 depicts β values and expected numbers of updates.

For both log-linear learning and our modification, the required β to reach an expected welfare within 98% of the maximum welfare is independent of n_3 and can be computed using the expressions

$$\pi_x^{\text{LLL}} \propto e^{\beta W(x)} \binom{n_1}{nx_1^1, nx_2^1} \binom{n_2}{nx_2^2, nx_3^2}, \tag{2.19}$$

$$\text{and } \pi_x^{\text{MLLL}} \propto e^{\beta W(x)}. \tag{2.20}$$

These stationary distributions can be verified using reversibility arguments with the standard and modified log-linear learning probability transition kernels, defined in [55] and Appendix B.1.1 respectively. Unlike standard log-linear learning and our variant, the required β to reach an expected welfare of 98% of maximum for the log-linear learning variant of [55] does change with n_3. For each value of n, we use the probability transition matrix to determine the necessary values of β which yield an expected welfare of 98% of its maximum.

Our algorithm converges to the desired expected welfare in fewer updates than both alternate algorithms for all tested values of n_3, showing that convergence rates for log linear learning and the variant from [55] are both more sensitive to the number of players in population 3 than our algorithm.[5]

We are able to determine convergence times in Example 3 using each algorithm's probability

[4] Recall that in our log-linear learning variant and the one introduced in [55], an updating player chooses a new action according to (2.7); the algorithms differ only in agents' update rates. In our algorithm, an agent i in population N_j's update rate is $\alpha n \,/\, z_i^j(t)$, where $z_i^j(t)$ is the number of agents from population j playing the same action as agent i at time t. In the algorithm in [55], agent i's update rate is $\alpha n \,/\, \tilde{z}_i(t)$, where $\tilde{z}_i(t)$ is the **total** number of agents playing the same action as agent i.

[5] A high update rate for players in population N_3 was undesirable because they contribute no value. While this example may seem contrived, mild variations would exhibit similar behavior. For example, consider a scenario in which a relatively large population that contributes little to the total welfare may choose from multiple resources.

transition matrix, P, because the state space is relatively small. Here, we directly compute the distance of distribution $\mu(t) = \mu(0)P^t$ to the stationary distributions, π^{LLL} and π^{MLLL} for the selected values of β, where P and π. Examples 4 and 6, however, have significantly larger state spaces, making similar computations with the probability transition matrix unrealistic. Thus, instead of computing convergence times as in (2.18) we repeatedly simulate our algorithm from a worst case initial state and approximate convergence times based on average behavior. This method does not directly give the convergence time of Theorem 1, but the average performance over a sufficiently large number of simulations is expected to reflect expected behavior predicted by the probability transition matrix.

Example 4. In this example we consider a scenario similar the previous example, without the third population. That is, agents are evenly divided into two popultions, N_1 and N_2; we allow the total number of agents to vary. Agents in N_1 may choose either resource r_1 or r_2, and agents in N_2 may choose either resource r_2 or r_3. We consider welfare functions of the following form:

$$W_{r_1}(x) = \frac{e^{x_1^1} - 1}{e^2}, \quad W_{r_2}(x) = \frac{e^{2x_2^1 + 2x_3^2} - 1}{e^2}, \quad W_{r_3}(x) = \frac{e^{2.5x_4^2} - 1}{e^2}. \tag{2.21}$$

for $x = (x_1^1, x_2^1, x_2^2, x_3^2) \in X$. Here, the global welfare optimizing allocation is $a_i = r_2$ for all $i \in N$, i.e., $x^{\text{opt}} = (0, 1/2, 1/2, 0)$. Similar to Example 2, this example has many Nash equilibria, two of which are x^{opt} and $x^{\text{ne}} = (1/2, 0, 0, 1/2)$.

We simulated our algorithm with $\alpha = 1\,/\,4$ starting from the inefficient Nash equilibrium, x^{ne}. Here, β is chosen to yield an expected steady state welfare equal to 90% of the maximum. We examine the time it takes the average welfare to come within $\varepsilon = 0.05$ of this expected welfare.

Simulation results are shown in Figure 2.1 averaged over 2000 simulations with n ranging from 4 to 100. Average convergence times are bounded below by $2n \log \log n$ for all values of n, and are bounded above by $4n \log \log n$ when $n > 30$. These results support Theorem 1.

Example 5. In this example we investigate convergence times for modified log-linear learning when agents have larger action sets. We consider the situation where n agents are divided into two populations, N_1 and N_2. Agents in N_1 may choose from resources in $A_1 = \{r_1, r_2, \ldots, r_k\}$, and

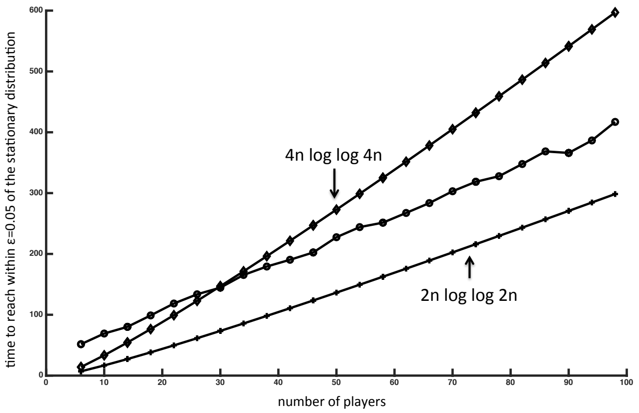

Figure 2.1: Example 4, number of players vs. average convergence times. Here, there are two equal-sized populations of agents, N_1 and N_2, and three resources r_1, r_2, and r_3. Agents in population N_1 may choose from resources r_1 and r_2, and agents in population N_2 may choose from resources r_2 and r_3. Welfare functions are given in (2.21).

agents in population N_2 may choose from resources in $A_2 = \{r_k, r_{k+1}, \ldots, r_{2k-1}\}$. That is, each agent may choose from k different resources, and the two populations share resource r_k. Suppose resource welfare functions are

$$W_{r_j}(x) = \begin{cases} x\,/\,4n & \text{if } j \neq k \\ x^2\,/\,n^2 & \text{if } j = k, \end{cases} \tag{2.22}$$

and suppose agents' utilities are given by their marginal contribution to the total welfare, as in (2.6). We allow k to vary between 5 and 15, and n to vary between 4 and 50.

The welfare maximizing configuration is for all agents to choose resource r_k; however, when all agents in populations N_1 and N_2 choose resources r_j and r_ℓ respectively, with $j, \ell \neq k$, this represents an inefficient Nash equilibrium. Along any path from this type of inefficient Nash equilibrium to the optimal configuration, when $n \geq 4$, at least $\lceil (n+4)/8 \rceil$ agents must make a utility-decreasing decision to move to resource r_k. Moreover, the additional resources are all alternative suboptimal choices each agent could make when revising its action; these alternate choices further slow convergence times. Figure 2.2 shows the average time it takes to reach a configuration whose welfare is 90% of the maximum, starting from an inefficient Nash equilibrium where all agents in N_1 choose resource r_1 and all agents in N_2 choose resource r_{2k-1}. Parameter β is selected so that the expected welfare is at least 90% of the maximum in the limit as $t \to \infty$. For each value of k, convergence times remain approximately linear in the number of agents, supporting Theorem 1.[6]

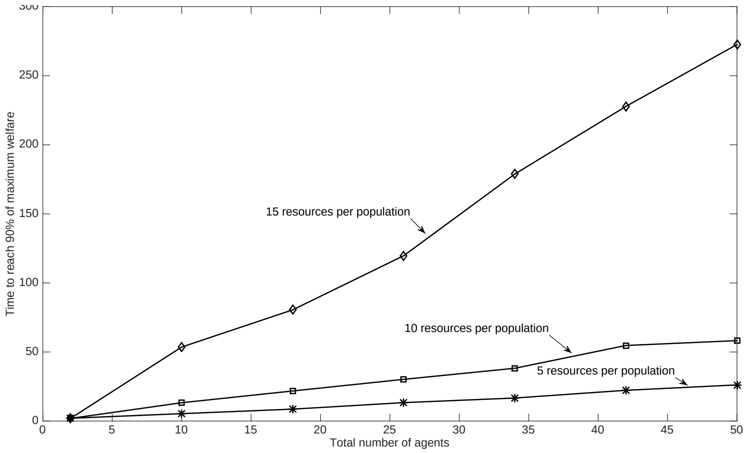

Figure 2.2: 5, number of agents vs. average time to reach 90% of the maximum welfare. Agents are separated into two populations, N_1 and N_2. Agents in N_1 choose from resources $r_1, r_2, \ldots, r_k$, and agents in N_2 choose from resources $r_k, r_{k+1} \ldots, r_{2k-1}$, where k varies from 5 to 15. Resource welfare functions are given by (2.22), agent utility functions are given by (2.6), and average convergence times are taken over 200 simulations.

In Example 6 we compare convergence times for standard and modified log-linear learning in a sensor-target assignment problem.

Example 6 (Sensor-Target Assignment)**.** In this example, we assign a collection of mobile sensors to four regions. Each region contains a single target, and the sensor assignment should maximize the probability of detecting the targets, weighted by their values. The targets in regions $R = \{r_1, r_2, r_3, r_4\}$ have values

$$v_1 = 1, \quad v_2 = 2, \quad v_3 = 3, \quad v_4 = 4 \tag{2.23}$$

respectively. Three types of sensors will be used to detect the targets: strong, moderate, and weak.

[6] In this example, convergence times appear super-linear in the size of populations' action sets. Note that the bound in (2.10) is exponential in the the sum of the sizes of each population's action set. Fast convergence with respect to parameter s warrants future investigation; in particular, convergence rates for our log-linear learning variant may be significantly faster than suggested in (2.10) under certain mild restrictions on resource welfare functions (e.g., submodularity) or for alternate log-linear learning variants (e.g., binary log-linear learning [5, 39]).

Detection probabilities of these three sensor types are:

$$p_s = 0.9, \quad p_m = 0.5, \quad p_w = 0.05. \tag{2.24}$$

The numbers of strong and weak sensors are $n_s = 1$ and $n_m = 5$. We vary the number of weak sensors, n_w.

The expected welfare for area r_i is the detection probability of the collection of sensors located at r_i weighted by the value of target i:

$$W_{r_i}(k_s, k_m, k_w) = v_i \left(1 - (1-p_s)^{k_s}(1-p_m)^{k_m}(1-p_w)^{k_w}\right),$$

where k_s, k_m and k_w represent the number of strong, moderate, and weak sensors located at region r_i. The total expected welfare for configuration a is

$$W(a) = \sum_{r \in R} W_r(|a|_r^s, |a|_r^m, |a|_r^w),$$

where $|a|_r^s$, $|a|_r^m$, and $|a|_r^w$ are the numbers of strong, moderate, and weak sensors choosing region r in a.

We assign agents' utilities according to their marginal contributions to the total welfare, W, as in (2.6). Our goal is to reach 98% of the maximum welfare. We set the initial state to be a worst-case Nash equilibrium.[7]

To approximate convergence times, we simulate each algorithm with the chosen β value[8] and compute a running average of the total welfare over 1000 simulations. In Figure 2.3 we show the average number of iterations necessary to reach 98% of the maximum welfare.

[7] The initial configuration is chosen by assigning weak agents to the highest value targets and then assigning strong agents to lower value targets. In particular, agents are assigned in order of weakest to strongest according to their largest possible marginal contribution. This constitutes an inefficient Nash equilibrium. As a similar example, consider a situation with two sensors with detection probabilities $p_1 = 0.5$ and $p_2 = 1$, and two targets with values $v_1 = 2$ and $v_2 = 1$. The assignment (sensor 1→ target 1, sensor 2→ target 2) is an inefficient Nash equilibrium, whereas the opposite assignment is optimal. The large state space makes it infeasible to directly compute a stationary distribution, and hence also infeasible to compute values of β that will yield precisely the desired expected welfare. Thus, we use simulations to estimate the β which yields an expected welfare of 98% of the maximum.

[8] To approximate the value of β which yields the desired steady-state welfare of 98% of maximum, we simulated the standard and modified versions of log-linear learning for 1×10^6 iterations for a range of β values. We then selected the β which yields an average welfare closest to the desired welfare during the final 5000 iterations. Note that we could instead set β according to (2.8) for the modified log-linear learning algorithm; however, in order to compare convergence times of modified and standard log-linear learning, we chose β to achieve approximately the same expected welfare for both algorithms.

For small values of n_w, standard log-linear learning converges more quickly than our modification, but modified log-linear learning converges faster than the standard version as n_w increases. The difference in convergence times is significant ($\approx$ 1000 iterations) for intermediate values of n_w. As the total number of weak sensors increases, (1) the probabilities of transitions along the paths to the efficient Nash equilibrium begin to increase for both algorithms, and (2) more sensor configurations are close to the maximum welfare. Hence, convergence times for both algorithms decrease as n_w increases.

This sensor-target assignment problem does not display worst-case convergence times with respect to the number of agents for either algorithm. However, it demonstrates a situation where our modification can have an advantage over standard log-linear learning. In log-linear learning, the probability that the strong sensor will update next decreases significantly as the number of agents grows. In modified log-linear learning this probability remains fixed. This property is desirable for this particular sensor-target assignment problem, since the single strong sensor contributes significantly to the total system welfare.

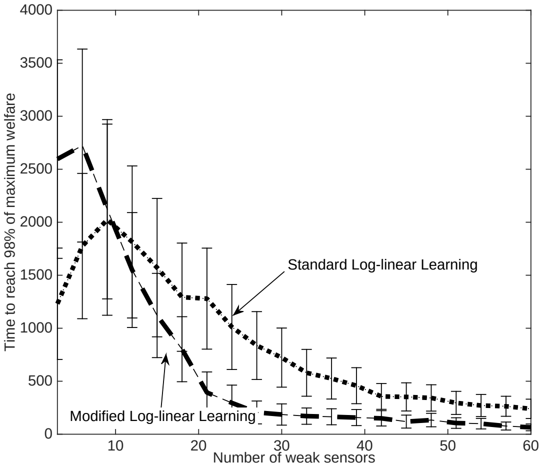

Figure 2.3: Example 6, number of weak sensors vs. average convergence times. Here, there are three types of sensors which may choose from four resources. Sensor detection probabilities and resource values are given in (2.24) and (2.23). We fix the number of strong and moderate sensors and vary the number of weak sensors. This figure shows the average time it takes for the average welfare to reach 98% of maximum. The average is taken over 1000 iterations, and convergence times correspond to a global update rate of 1 per second. Error bars show standard deviations of the convergence times.

Algorithm	n_3	β	Expected welfare	Expected # updates
Standard LLL	1	3.77	98%	9430
	5	3.77	98%	11947
	50	3.77	98%	40250
	500	3.77	98%	323277
LLL Variant from [55]	1	2.39	98%	1325
	5	2.44	98%	1589
	50	2.83	98%	3342
	500	3.72	98%	15550
Our LLL Variant	1	1.28	98%	743
	5	1.28	98%	743
	50	1.28	98%	743
	500	1.28	98%	743

Table 2.1: This table corresponds to Example 3. There are three populations of agents, N_1, N_2, and N_3, and three resources r_1, r_2, and r_3. Agents in population N_1 may choose from resources r_1 and r_2, and agents in population N_2 may choose from resources r_2 and r_3. Agents in population N_3 may only choose resource r_2. Welfare functions are given in (2.19); population N_3 contributes nothing to the overall system welfare. We examine the sensitivity of convergence times to the size of N_3, and keep the sizes of populations N_1 and N_2 fixed at 7. The third column of this table shows the values of β which yield an expected total welfare within 98% of the maximum. These values of β are constant for standard log-linear learning and for our variant, but grow with n for the algorithm in [55]. The final column shows the expected number of updates to achieve the desired near-maximum welfare. This value is constant for our algorithm, but increases with n for the other two. Global update rates are a design parameter dictated by parameter α; selecting a global update rate of n per second ($\alpha = 1/m$), convergence times would be a factor of n smaller than the number of updates shown.

In summary, we have extended the results of [55] to define dynamics for a class of semi-anonymous potential games whose player utility functions may be written as functions of aggregate behavior within each population. For games with a fixed number of actions and a fixed number of populations, the time it takes to come arbitrarily close to a potential function maximizer is linear in the number of players. This convergence time remains linear in the initial number of players even when players are permitted to enter and exit the game, provided they do so at a sufficiently slow rate.

Chapter 3

Learning Efficient Correlated Equlibria

Can agents in a distributed system learn a payoff maximizing correlated equilibrium, despite severely limited information about one another's behavior?

In this chapter, we focus on the scenario where the system objective is to maximize the sum of agents' payoffs. This type of objective can be useful when we wish to balance multiple local objectives.

Agents' average utilities can often be improved when they act according to a distribution over multiple joint actions, instead of staying fixed at a single joint action. In many cases, the desired collective behavior constitutes a *coarse correlated equilibrium.* A coarse correlated equilibrium is a probability distribution over the joint action space such that no agent can improve its payoff by deviating to a fixed action [6]. Previously, algorithms existed which converged to the *set* of coarse correlated equilibrium, e.g., [27], without selecting any particular equilibrium; these algorithms provided no performance guarantees. An algorithm which achieves a payoff maximizing coarse correlated equilibrium through deterministic, cyclic behavior is presented in [38]. However, predictable cyclic behavior may be undesirable in many settings, e.g., in the presence of an adversary.

For concreteness, consider a mild variant of the Shapley game with the following payoff matrix

	L	M	R
T	$1,-\varepsilon$	$-\varepsilon,1$	$0,0$
M	$0,0$	$1,-\varepsilon$	$-\varepsilon,1$
B	$-\varepsilon,1$	$0,0$	$1,-\varepsilon$

where $\varepsilon > 0$ is a small constant. In this game, there are two players (Row, Column); the row player has three actions (T,M,B), and the column player has three actions (L,M,R). The numbers in the table above are the players' payoffs for each of the nine joint actions. The unique Nash equilibrium for this game occurs when each player uses a probabilistic strategy that selects each of the three actions with probability 1/3. This yields an expected payoff of approximately 1/3 to each player. Alternatively, a joint distribution that places a mass of 1/6 on each of the six joint actions that yield non-zero payoffs to the players yields an expected payoff of approximately 1/2 to each player. Note that this distribution cannot be realized by independent strategies associated with the two players, but instead represents a specific correlated equilibrium.

As the above example demonstrates, distributed learning algorithms that converge to efficient correlated equilibria can be desirable from a system-wide perspective. In line with this theme, results presented in [35] rely on looking for cyclic behavior against a bounded memory opponent. Additionally, a recent result in [38] proposed a distributed algorithm that guarantees that the empirical frequency of the agents' collective behavior will converge to an efficient correlated equilibrium; however, convergence in empirical frequencies is attained through deterministic cyclic behavior of the agents. For example, in the above Shapley game, the algorithm posed in [38] guarantees that the collective behavior of the agents will follow the cycle $(T, L) \rightarrow (T, M) \rightarrow (M, M) \rightarrow (M, R) \rightarrow (B, R) \rightarrow (B, L) \rightarrow (T, L)$ with high probability. Following this deterministic cycle results in an empirical frequency of play that equates to the efficient correlated equilibrium highlighted above; however, at any time instance the players are not playing a joint strategy in accordance with this efficient correlated equilibrium.

Predictable, cyclic behavior may be desirable from a system-wide perspective for many applications, e.g., data ferrying [12]. However, such behavior could be exploited in many other situations, e.g., team versus team zero-sum games [29, 58]. By viewing each team as a single player, classical results for two-player zero-sum games suggest that a team's desired strategy is to play its security strategy, which can be characterized by a probability distribution over the team's joint action space. Distributed learning algorithms that can stabilize specific joint strategies, such as correlated equilibria, may be necessary for providing strong performance guarantees in such settings.

In this paper we present a distributed learning algorithm that ensures the agents collectively play a joint strategy corresponding to the efficient correlated equilibrium. With regards to the Shapley game, our algorithm guarantees that the agents collectively play the highlighted joint distribution with high probability. Attaining such guarantees on the underlying joint strategy is non-trivial as we aim to ensure desired correlated behavior through the design of learning rules where individual agents make independent decisions in response to local information. The key element of our algorithm that makes this correlation possible is the introduction of a common random signal to the agents, which is incorporated into their local decision-making rule. Another important feature of our algorithm is that it is completely uncoupled [18], i.e., agents make decisions based only on their received utility and their observation of the common random signal. In such settings, agents have no knowledge of the payoff or behavior of other agents, nor do they have any information regarding the structural form of their utility functions.

It is important to highlight the recent results which focus on efficient centralized algorithms for computing specific correlated equilibria [30, 51, 50]. Such algorithms often require a complete characterization of the game which is unavailable in many engineering multiagent systems. Hence, the applicability of such results to the design and control of multiagent systems may be limited.

3.1 Background

We consider the framework of finite strategic form games where there exists an agent set $N = \{1, 2, \ldots, n\}$, and each agent $i \in N$ is associated with a finite action set $\mathcal{A}_i$ and a utility function $U_i : \mathcal{A} \to [0, 1]$ where $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \times \cdots \times \mathcal{A}_n$ denotes the joint action space. We represent such a game by the tuple $G = (N, \{U_i\}_{i \in N}, \{\mathcal{A}_i\}_{i \in N})$.

In this paper we focus on the class of coarse correlated equilibria [6]. A coarse correlated equilibrium is characterized by a joint distribution $q = \{q^a\}_{a \in \mathcal{A}} \in \Delta(\mathcal{A})$, where $\Delta(\mathcal{A})$ represents the simplex over the finite set $\mathcal{A}$, such that for any agent $i \in N$ and action $a_i' \in \mathcal{A}_i$,

$$\sum_{a \in \mathcal{A}} U_i(a_i, a_{-i}) q^a \geq \sum_{a \in \mathcal{A}} U_i(a_i', a_{-i}) q^a, \tag{3.1}$$

where $a_{-i} = \{a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n\}$ denotes the collection of action of all players other than player i.[1] Informally, a coarse correlated equilibrium represents a joint distribution where each agent's expected utility for going along with the joint distribution is at least as good as his expected utility for deviating to any fixed action. We say a coarse correlated equilibrium q^* is **efficient** if it maximizes the sum of the expected payoffs of the agents, i.e.,

$$q^* \in \arg\max_{q \in \mathrm{CCE}} \sum_{i \in N} \sum_{a \in \mathcal{A}} U_i(a) q^a, \tag{3.2}$$

where $\mathrm{CCE} \subset \Delta(\mathcal{A})$ denotes the set of coarse correlated equilibria. It is well known that $\mathrm{CCE} \neq \emptyset$ for any game G.

This paper focuses on deriving a distributed learning algorithm that ensures the collective behavior of the agents converges to an efficient coarse correlated equilibrium. We adopt the framework of repeated one-shot games, where a static game G is repeated over time and agents use observations from previous plays of the game to formulate a decision. More specifically, a repeated one-shot game yields a sequence of action profiles $a(0)$, $a(1)$, $\ldots$, where at each time $t \in \{0, 1, 2, \ldots\}$ the decision of each agent i is chosen independently accordingly to the agent's strategy at time t, which we denote by $p_i(t) = \{p_i^{a_i}(t)\}_{a_i \in \mathcal{A}_i} \in \Delta(\mathcal{A}_i)$.

A learning rule dictates how each agent selects its strategy given available information from previous plays of the game. One type of learning rule, known as **completely uncoupled** or **payoff based** [18], takes on the form:

$$p_i(t) = F_i\left(\{a_i(\tau), U_i(a(\tau))\}_{\tau = 0, \ldots, t-1}\right) \tag{3.3}$$

Completely uncoupled learning rules represent one of the most informationally restrictive classes of learning rules since the only knowledge that each agent has about previous plays of the game is (i) the action the agent played and (ii) the utility the agent received.

[1] We will express an action profile $a \in \mathcal{A}$ as $a = (a_i, a_{-i})$.

We gauge the performance of a learning rule $\{F_i\}_{i \in N}$ by the resulting asymptotic guarantees. With that goal in mind, let $q(t) \in \Delta(\mathcal{A})$ represent the agents' collective strategy at time t, which is of the form

$$q^{(a_1,\ldots,a_n)}(t) = p_1^{a_1}(t) \times \cdots \times p_n^{a_n}(t) \tag{3.4}$$

where $\{p_i(t)\}_{i \in N}$ are the individual agent strategies at time t. The goal of this paper is to derive learning rules that guarantee the agents' collective strategy constitutes an efficient coarse correlated equilibrium the majority of the time, i.e., for all sufficiently large times t,

$$\Pr\left[q(t) \in \underset{q \in \text{CCE}}{\arg\max} \sum_{i \in N} \sum_{a \in \mathcal{A}} U_i(a) q^a\right] \approx 1. \tag{3.5}$$

Attaining this goal using learning rules of the form (3.3) is impossible because such rules do not allow for correlation between the players, i.e., the agents' collective strategies are restricted to being of form (3.4). Accordingly, we modify the learning rules in (3.3) by giving each agent access to a common random signal $z(t)$ at each period $t \in \{0, 1, \ldots\}$ that is i.i.d. and drawn uniformly from the interval $[0, 1]$. Now, the considered distributed learning rule takes the form

$$p_i(t) = F_i\left(\{a_i(\tau), U_i(a(\tau)), z(t))\}_{\tau=0,\ldots,t-1}\right). \tag{3.6}$$

As we show in the following section, this common signal can be used as a coordinating entity to reach collective strategies beyond the form given in (3.4).

3.2 A learning algorithm for attaining efficient correlated equilibria

In this section, we present a specific learning rule of the form (3.6) that guarantees the agents' collective strategy constitutes an efficient coarse correlated equilibrium the majority of the time. This algorithm achieves the desired convergence guarantees by exploiting the common random signal $z(t)$ through the use of **signal-based strategies**.

3.2.1 Preliminaries

Consider a situation where each agent $i \in N$ commits to a signal-based strategy of the form $s_i : [0, 1] \to \mathcal{A}_i$ which associates with each signal $z \in [0, 1]$ an action $s_i(z) \in \mathcal{A}_i$. With an abuse of notation, we consider a finite parameterization of such signal-based strategies, which we refer to as **strategies**, of the form $S_i = \cup_{\omega=1}^{\Omega}(\mathcal{A}_i)^{\omega}$ where $\Omega \geq 1$ is a design parameter identifying the granularization of the agent's possible strategies. A strategy $s_i = (a_i^1, \ldots, a_i^{\omega}) \in S_i$, $\omega \leq \Omega$, defines a mapping of the form

$$s_i(z) = \begin{cases} a_i^1 & \text{if} \quad z \in [0, 1/\omega) \\ a_i^2 & \text{if} \quad z \in [1/\omega, 2/\omega) \\ \vdots & \quad \vdots \\ a_i^{\omega} & \text{if} \quad z \in [(\omega-1)/\omega, 1]. \end{cases} \tag{3.7}$$

These strategies divide the unit interval into at most Ω regions of equal length and associate each region with a specific action in the agent's action set. If the agents commit to a strategy profile $s = (s_1, s_2, \ldots, s_n) \in S = \prod_{i \in N} S_i$, the resulting joint strategy $q(s) = \{q^a(s)\}_{a \in \mathcal{A}} \in \Delta(\mathcal{A})$ satisfies

$$q^a(s) = \int_0^1 \prod_{i \in N} I\{s_i(z) = a_i\} dz$$

where $I\{\cdot\}$ is the indicator function. Lastly, the set of joint distributions that can be realized by the strategies S is

$$q(S) = \{q \in \Delta(\mathcal{A}) : q(s) = q \text{ for some } s \in S\}.$$

3.2.2 Informal algorithm description

The forthcoming algorithm is reminiscent of the trial and error learning algorithm introduced in [62] and can be viewed at a high level through the following diagram.

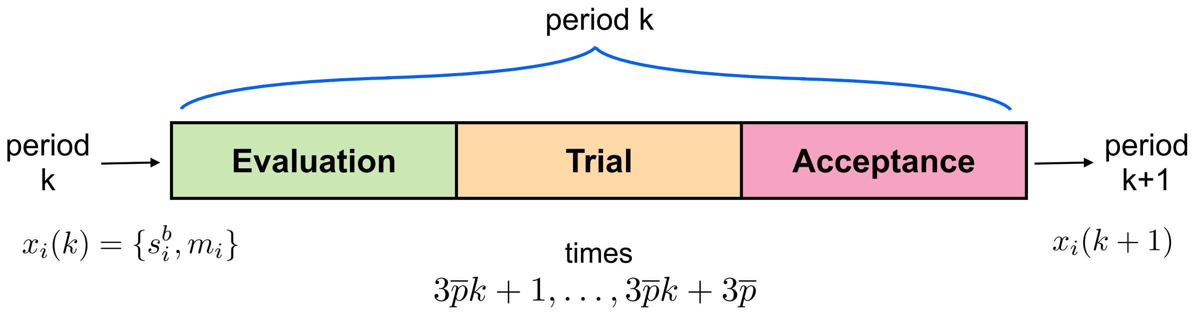
period k
period k
Evaluation
Trial
Acceptance
period k+1
$x_i(k) = \{s_i^b, m_i\}$
times
$3\overline{p}k + 1, \ldots, 3\overline{p}k + 3\overline{p}$
$x_i(k+1)$

Figure 3.1: Learning algorithm phases within each time period

The times $\{1, 2, \ldots\}$ will be broken up into periods of length $3\bar{p}$ where $\bar{p} > 1$ is an interval whose length will be defined formally below. At the beginning of each period k, each agent $i \in N$ has a local state variable of the form $x_i(k) = [s_i^b, m_i]$ where $s_i^b \in S_i$ is the agent's baseline strategy and m_i is the agent's mood. The agent's baseline strategy corresponds to the strategy the agent is accustomed to playing. The agent's mood m_i, which can either be Content or Discontent, dictates how likely each agent is to select its baseline strategy during a given period. Roughly speaking, a content agent is more likely to select its baseline strategy while a discontent agent is more likely to try an alternate strategy.

Each period $k > 0$, which consists of the time steps $\{3\bar{p}k+1, \ldots, 3\bar{p}(k+1)\}$, will be broken up into three distinct phases called **evaluation**, **trial**, and **acceptance**. The behavior of the agents in each of these phases is highlighted below:

– **Evaluation Phase:** The first phase is the **evaluation phase**. In this phase, each agent establishes a baseline utility, u_i^b, associated with its current baseline strategy, s_i^b. All agents commit to their baseline strategies during this entire phase.

– **Trial Phase:** The second phase is the **trial phase**. During this phase, each agent has the opportunity to experiment with an alternate trial strategy, s_i^t, in order to determine whether changing its baseline strategy could be advantageous. An agent's mood determines how likely it is to experiment. In particular, a content agent will use its baseline strategy s_i^b during the trial phase with high probability. On the other hand, a discontent player is likely to experiment with a trial strategy $s_i^t \neq s_i^b$. The exact probabilities associated with this selection process will be described in detail in the forthcoming section.

– **Acceptance Phase:** The third phase is the **acceptance phase**. Here, an agent who experimented during the trial phase decides whether to accept its trial strategy or revert to its baseline strategy. Agents who did not experiment during the trial phase commit to their baseline strategies and observe payoff changes which occur due to others' changes in strategy.

3.2.3 Formal algorithm description

We begin by defining a constant $c > n$, an experimentation rate $\varepsilon \in (0, 1)$, and the length of a phase to be $\bar{p} = \lceil 1/\delta^{nc+1} \rceil$ time steps, for some small $\delta \in (0, 1)$. A period consists of the evaluation, trial, and acceptance phases, and hence is $3\bar{p}$ time steps long. Let $x_i = x_i(k) = [s_i^b, m_i]$ represent that state of each agent $i \in N$ at the beginning of some period $k \in \{1, 2, \dots\}$. We will formally present the algorithm using the same general structure given in previous section.

Agent Dynamics: Here we describe how individual agents make decisions within a given period. Decisions of an agent $i \in N$ are influenced purely by its state at the beginning of the k-th period, $x_i(k)$, and by payoffs received during the k-th period. We specify agents' behavior during the k-th period for the three phases highlighted above.

– **Evaluation Phase:** The evaluation phase consists of the times $t \in \{3\bar{p}k + 1, \dots, 3\bar{p}k + \bar{p}\}$. Throughout this phase, each agent commits to its baseline strategy s_i^b. At the end of the phase, each agent computes its average baseline utility,

$$u_i^b = \frac{1}{\bar{p}} \sum_{\tau = 3\bar{p}k+1}^{3\bar{p}k+\bar{p}} U_i\big(s_1^b(z(\tau), \dots, s_n^b(z(\tau))\big), \tag{3.8}$$

where $z(\tau)$ denotes the common random signal observed at time τ. Here, u_i^b is viewed as an assessment of the performance associated with the baseline strategy s_i^b.

– **Trial Phase:** After the evaluation phase comes the trial phase which consists of the times

$t \in \{(3\bar{p}k+\bar{p})+1,\ldots,3\bar{p}k+2\bar{p}\}$. During the trial phase each player $i \in N$ may try a strategy other than its baseline, and must commit to this trial strategy, $s_i^t \in S_i$, over the entire phase. Agents' trial strategies are selected according to the following rule:

- **Content,** $m_i = C$**:** When agent i is content, its trial strategy, $s_i^t \in S_i$, is chosen according to the distribution

$$\Pr\left[s_i^t = s_i\right] = \begin{cases} 1-\varepsilon^c & \text{if } s_i = s_i^b \\ \varepsilon^c \,/\, |\mathcal{A}_i| & \text{for any } s_i = a_i \in \mathcal{A}_i \end{cases} \tag{3.9}$$

  A strategy $s_i^t = a_i$ means that agent i commits to playing action a_i for the entire trial phase of the k-th period, i.e., the strategy does not depend on the common random signal. Observe that a content player predominantly selects its baseline strategy during the trial phase.

- **Discontent,** $m_i = D$**:** When agent i is discontent, its trial strategy, s_i^t, is chosen randomly from the set S_i,

$$\Pr\left[s_i^t = s_i\right] = 1 \,/\, |S_i| \text{ for all } s_i \in S_i. \tag{3.10}$$

At the end of the trial phase, each agent computes its average utility:

$$u_i^t = \frac{1}{\bar{p}} \sum_{\tau=3\bar{p}k+\bar{p})+1}^{3\bar{p}k+2\bar{p}} U_i\big(s_1^t(z(\tau),\ldots,s_n^t(z(\tau))\big). \tag{3.11}$$

Here, u_i^t is viewed as an assessment of the performance associated with the baseline strategy s_i^t.

– **Acceptance Phase:** The last phase is the acceptance phase which consists of times $t \in \{(3\bar{p}k+2\bar{p})+1,\ldots,3\bar{p}k+3\bar{p}\}$. The primary purpose of the acceptance phase is to further evaluate changes in the payoffs between u_i^b and u_i^t. Each agent $i \in N$ commits to an acceptance strategy, denoted by $s_i^a \in S_i$, over the entire acceptance phase. Each agent's acceptance strategy is selected according to the following.

- **Content,** $m_i = C$**:** When agent i is content, its acceptance strategy is chosen as follows:

$$s_i^a = \begin{cases} s_i^t & \text{if } u_i^t > u_i^b + \delta, \\ s_i^b & \text{if } u_i^t \le u_i^b + \delta. \end{cases} \tag{3.12}$$

That is, players only repeat their trial strategy if their performance was high enough relative to the performance of the baseline strategy.

- **Discontent,** $m_i = D$**:** When agent i is discontent, the acceptance strategy is set as $s_i^a = s_i^t$.

Following the acceptance phase, each agent computes its average utility:

$$u_i^a = \frac{1}{\bar{p}} \sum_{\tau=(3\bar{p}k+2\bar{p})+1}^{3\bar{p}k+3\bar{p}} U_i\big(s_1^a(z(\tau), \ldots, s_n^a(z(\tau))\big). \tag{3.13}$$

Here, u_i^a is viewed as an assessment of the performance associated with the baseline strategy s_i^a.

State Dynamics: After the agent dynamics comes the state dynamics which specifies how the state of each agent evolves. The state of each agent $i \in N$ at the beginning of the $k+1$-st stage, i.e., $x_i(k+1)$, is influenced purely its state at the beginning of the k-th period, i.e., $x_i(k)$, the strategies s_i^b, s_i^t and s_i^a, and the payoffs received during the k-th period. The state dynamics are broken into the following cases:

– **Content and No Experimentation,** $m_i = C, s_i^t = s_i^b$**:** If agent i was content at the start of the k-th period and did not experiment in the trial phase, its state at the beginning of the $(k+1)$-st period is chosen as follows:

- If $u_i^a \geq u_i^b - \delta$,

$$x_i(k+1) = \begin{cases} \big[s_i^a = s_i^b,\, C\big] & \text{w.p. } 1-\varepsilon^{2c}, \\ \big[s_i^a = s_i^b,\, D\big] & \text{w.p. } \varepsilon^{2c}. \end{cases} \tag{3.14}$$

- If $u_i^a < u_i^b - \delta$,

$$x_i(k+1) = \Big[s_i^a = s_i^b,\, D\Big] \tag{3.15}$$

Accordingly, if the agent's average payoff during the acceptance phase is low enough, then it will become discontent.

– **Content and Experimentation,** $m_i = C, s_i^t \neq s_i^b$**:** If agent i was content at the start of the k-th period and experimented during the trial phase, its state at the beginning of the $(k+1)$-st period is chosen as

$$x_i(k+1) = [s_i^a, C]\,. \tag{3.16}$$

In this case the agent's average payoff during the acceptance phase does not impact its underlying state dynamics.

– **Discontent,** $m_i = D$**:** If agent i was discontent at the start of the k-th period, its state at the beginning of the $(k+1)$-th period is chosen as follows

$$x_i(k+1) = \begin{cases} [s_i^a, C] & \text{w.p. } \varepsilon^{1-u_i^a}, \\ [s_i^a, D] & \text{w.p. } 1-\varepsilon^{1-u_i^a}. \end{cases} \tag{3.17}$$

Here, the agents are more likely to become content with strategies the yield higher average payoffs.

3.3 Main Result

Throughout this paper we focus on games where there is some degree of coupling between the utility functions of the agents. The following definition of interdependence, taken from [62], captures this notion of coupling.

Definition 2. A game G with agents $N = \{1, 2, \ldots, n\}$ is said to be **interdependent** if, for every $a \in \mathcal{A}$ and every proper subset of agents $J \subset N$, there exists an agent $i \notin J$ and a choice of actions $a'_J \in \prod_{j \in J} \mathcal{A}_j$ such that $U_i(a'_J, a_{-J}) \neq U_i(a_J, a_{-J})$.

Roughly speaking, the definition of interdependence states that it is not possibly to partition the group of agents into two sets whose actions do not impact one another's payoffs.

The following theorem characterizes the limiting behavior associated with the proposed algorithm.

Theorem 3. *Let* $G = (N, \{U_i\}, \{\mathcal{A}_i\})$ *be a finite interdependent game. First, suppose* $q(S) \cap \mathrm{CCE} \neq \emptyset$*. Given any probability* $p < 1$*, if the exploration rate* ε *is sufficiently small, and if* $\delta = \varepsilon$*, then for* *all sufficiently large times t,*[2]

$$\Pr\left[q(s(t)) \in \underset{q \in q(S) \cap \mathrm{CCE}}{\arg\max} \sum_{i \in N} \sum_{a \in \mathcal{A}} U_i(a) q^a \right] > p.$$

Alternatively, suppose $q(S) \cap \mathrm{CCE} = \emptyset$. Given any probability $p < 1$, if the exploration rate ε is sufficiently small and $\delta = \varepsilon$, then for all sufficiently large times t,

$$\Pr\left[q(s(t)) \in \underset{q \in q(S)}{\arg\max} \sum_{i \in N} \sum_{a \in \mathcal{A}} U_i(a) q^a \right] > p.$$

We prove Theorem 3 in Appendix B.2.

A few remarks are on order regarding Theorem 3. First, observe that the proposed algorithm is of the form (3.6). Second, the condition $q(S) \cap \mathrm{CCE} \neq \emptyset$ implies the agents can realize specific joint distributions that are coarse correlated equilibria through the joint strategy set S. When this is the case, the above algorithm ensures the agents predominantly play a strategy $s \in S$ where the resulting joint distribution $q(s)$ corresponds to the efficient coarse correlated equilibrium. Alternately, the condition $q(S) \cap \mathrm{CCE} = \emptyset$ implies there are no agent strategies that can characterize a coarse correlated equilibrium. When that is the case, the above algorithm ensures the agents predominantly play strategies that have full support on the action profiles $a \in \mathcal{A}$ that maximize the sum of the agents payoffs, i.e., $\arg\max_{a \in \mathcal{A}} \sum_{i \in N} U_i(a)$.

3.4 Illustrative Example

Here, we present an example where agents update their strategies according to the algorithm above, and their actions converge to an efficient coarse correlated equilibrium.

Example 7. Consider a game with two players, (Row, Column), and the following payoff matrix:

	L	M	R
T	0, 0	0, 1	0.85, 0.75
M	1, 0	0, 0	0, 0
B	0.75, 0.85	0, 0	0, 0

[2] For the proof of Theorem 3, we require $\delta = \varepsilon$. However, in practice, fixing $\delta > \varepsilon$ in order to shorten the period length, $\bar{p}$, often yields similar results, as we demonstrate in Example 7.

The efficient coarse correlated equilibrium in this game places probability 0.5 on joint action (T,R), and probability 0.5 on joint action (B,L), i.e.,

$$q^{(T,R)} = q^{(B,L)} = 0.5, \tag{3.18}$$

and $q^a = 0$ for $a \notin \{(T,R),(B,L)\}$. The expected utility associated with this coarse correlated equilibrium is $U_i(q) = 0.8$.

For each value of ε in $\{0.15, 0.1, 0.015, 0.01\}$, we simulated our algorithm for 20 times over 10^5 iterations, fixing $\delta = 0.14$. The table below shows the percentage of the last 5×10^4 iterations spent in the efficient coarse correlated equilibrium as in (3.18).[3]

ε	**% time in efficient CCE**
0.15	9%
0.1	16%
0.015	84%
0.01	87%

Note that as ε decreases, more time is spent in the efficient coarse correlated equilibrium, as predicted by Theorem 3.

The majority of distributed learning literature has focused on identifying learning rules that converge to Nash equilibria. However, alternate forms of behavior, such as correlated equilibrium, can often lead to significant improvements in system-wide behavior. This chapter focuses on identifying learning rules that converge to joint distributions that do not necessarily constitute Nash equilibria. In particular, we have a provided a distributed learning rule, similar in spirit to the learning rule in [38], that ensuers agents play strategies that constitute efficient coarse correlated equilibria. A mild variant of the proposed algorithm could also ensure the agents play strategies that constitute correlated equilibria, as opposed to coarse correlated equilibria. Future work seeks

[3] We did not simulate our algorithm for smaller values of ε because convergence rates slow significantly as $\varepsilon \to 0$, reducing the algorithm's practicality. Next research steps include improving this algorithm's convergence rates.

to investigate the applicability of such algorithms in the context of team versus team zero-sum games.

Chapter 4

Understanding Adversarial Influence in Distributed Systems

How can network structure in a distributed system create vulnerabilities to adversarial influence?

Engineering and social systems often consist of many agents making decisions based on locally available information. In an engineering system, a distributed decision making strategy can be necessary when communication, computation, or sensing limitations preclude a centralized control strategy. For example, a group of unmanned aircraft performing surveillance in a hostile area may use a distributed control strategy to limit communication and thus remain undetected. Social systems are inherently distributed: individuals typically make decisions based on personal objectives and the behavior of friends and acquaintances. For example, the decision to adopt a recently released technology, such as a new smartphone, may depend both on the quality of the item itself and on friends' choices.

While there are many advantages of distributed decision making, it can create vulnerability to adversarial manipulation. Adversaries may attempt to influence individual agents by corrupting the information available to them, creating a chain of events which could degrade the system's performance. Work in the area of cyber-physical systems has focused on reducing the potential impact of adversarial interventions through detection mechanisms: detection of attacks in power networks [28], estimation and control with corrupt sensor data [15, 7], and monitoring [52]. In contrast to this research, our work focuses on characterizing the impact an adversary may have on distributed system dynamics when no mitigation or detection measures are in place.

We use graphical coordination games, introduced in [13,57], to study the impact of adversarial manipulation. The foundation of a graphical coordination game is a simple two agent coordination game, where each agent must choose between one of two alternatives, $\{x, y\}$, with payoffs depicted by the following payoff matrix which we denote by $u(\cdot)$:

	x	y
x	$1+\alpha,\ 1+\alpha$	$0,\ 0$
y	$0,\ 0$	$1,\ 1$

2×2 coordination game, g, with utilities $u(a_i, a_j)$, $a_i, a_j \in \{x, y\}$, and payoff gain $\alpha > 0$

where $\alpha > 0$ defines the relative quality of conventions (x, x) over (y, y). Both agents prefer to agree on a convention, i.e., (x, x) or (y, y), than disagree, i.e., (x, y) or (y, x), with a preference to agreeing on (x, x). The goal of deriving local agent dynamics which lead to the efficient Nash equilibrium (x, x) is challenging because of the existence of the inefficient Nash equilibrium (y, y). Deviating from (y, y) for an individual agent is accompanied by an immediate payoff loss of 1 to 0; hence, myopic agents may be reluctant to deviate, stabilizing the inefficient equilibrium (y, y).

This two player coordination game can be extended to an n-player **graphical coordination game** [32, 63, 48], where the interactions between the agents $N = \{1, 2, \ldots, n\}$ is described by an underlying graph $\mathcal{G} = (N, E)$ where $E \subseteq N \times N$ denotes the interdependence of agents' objectives. More formally, an agent's total payoff is the sum of payoffs it receives in the two player games played with its neighbors $\mathcal{N}_i = \{j \in N : (i, j) \in E\}$, i.e., for a joint decision $a = (a_1, \ldots, a_n) \in \{x, y\}^n$, the utility of agent i is

$$U_i(a_1, \ldots, a_n) = \sum_{j \in \mathcal{N}_i} u(a_i, a_j). \tag{4.1}$$

Joint actions $\vec{x} := (x, x, \ldots, x)$ and $\vec{y} := (y, y, \ldots, y)$, where either all players choose x or all players choose y, are Nash equilibria of the game; other equilibria may emerge depending on the structure of graph $\mathcal{G}$. In any case, $\vec{x}$ is the unique efficient equilibrium, since it maximizes agents' total payoffs. Graphical coordination games can model both task allocation in engineering systems as well as the evolution of social convention in marketing scenarios.

The goal in this setting is to prescribe a set of decision-making rules that ensures emergent behavior is aligned with the efficient Nash equilibrium $\vec{x}$ irrespective of the underlying graph $\mathcal{G}$ and the choice of α. Any such rule must be accompanied by a degree of noise (or mistakes) as agents must be enticed to deviate from inefficient Nash equilibrium. Log-linear learning [9,55] is one distributed decision making rule that selects the efficient equilibrium, $\vec{x}$, in the graphical coordination game described above. Although agents predominantly maximize their utilities under log-linear learning, selection of the efficient equilibrium is achieved by allowing agents to choose suboptimally with some small probability that decreases exponentially with respect to the associated payoff loss.

The equilibrium selection properties of log-linear learning extend beyond coordination games to the class of potential games [47], which often can be used to model engineering systems where the efficient Nash equilibrium is aligned with the optimal system behavior [40,42,60]. Hence, log-linear learning can be a natural choice for prescribing control laws in many distributed engineering systems [42,64,24,56,20], as well as for analyzing the emergence of conventions in social systems [61,55]. This prompts the question: can adversarial manipulation alter the emergent behavior of log-linear learning in the context of graphical coordination games (or more broadly in distributed engineering systems)?

We study this question in the context of the above graphical coordination games. Here, we model the adversary as additional nodes/edges in our graph, where the action selected by these adversaries (which we fix as the inferior convention y) impacts the utility of the neighboring agents and thereby influences the agents' decision-making rule as specified by log-linear learning. We focus on three different models of adversary behavior, referred to as **fixed, intelligent**; **mobile, random**; and **mobile, intelligent**.

- A fixed intelligent adversary aims to influence a fixed set $S \subseteq N$. To these agents the adversary appears to be a neighbor who always selects alternative y. We assume that S is selected based on the graph structure $\mathcal{G}$ and α.

- A mobile, random adversary connects to a random collection of agents $S(t) \subseteq N$ at each time, $t \in \mathbb{N}$ using no information on graph structure, $\mathcal{G}$, or payoff gain, α.

- A mobile, intelligent agent connects to a collection of agents, $S(t) \subseteq N$, at each time, $t \in \mathbb{N}$ using information on graph structure, $\mathcal{G}$, payoff gain α, and the current action profile, $a(t)$.

We will discuss each type of adversary's influence on an arbitrary graph, and then analyze the worst case influence on a set of agents interacting according to a line. We specify the values of payoff gain α for which an adversary can stabilize joint action $\vec{y}$, showing that a mobile, intelligent agent can typically stabilize joint action $\vec{y}$ for larger values of α than a mobile, random agent, and a mobile, random agent can typically stabilize $\vec{y}$ for larger values of α than a fixed, intelligent agent.

4.1 The model

Suppose agents in N interact according to the graphical coordination game above, with underlying graph $\mathcal{G} = (N, E)$, alternatives $\{x, y\}$ and payoff gain α. We denote the joint action space by $\mathcal{A} = \{x, y\}^n$, and we write

$$(a_i, a_{-i}) = (a_1, a_2, \ldots, a_i, \ldots, a_n) \in \mathcal{A}$$

when considering agent i's action separately from other agents' actions.

Now, suppose agents in N update their actions according to the **log-linear learning** algorithm at times $t = 0, 1, \ldots$, producing a sequence of joint actions $a(0), a(1), \ldots$. We assume agents begin with joint action, $a(0) \in \mathcal{A}$, and let $a(t) = (a_i, a_{-i}) \in \mathcal{A}$. At time $t \in \mathbb{N}$, an agent $i \in N$ is selected uniformly at random to update its action for time $t + 1$; all other agents' actions will remain fixed. Agent i chooses its next action probabilistically according to:[1]

$$\begin{aligned}\Pr[a_i(t+1) = x \mid a_{-i}(t) = a_{-i}]& \\ &= \frac{\exp\left(\beta \cdot U_i(x, a_{-i})\right)}{\exp\left(\beta \cdot U_i(x, a_{-i}) + \exp\left(\beta \cdot U_i(y, a_{-i})\right)\right.}.\end{aligned} \tag{4.2}$$

[1] Agent i's update probability is also conditioned on the fact that agent i was selected to revise its action, which occurs with probability $1\,/\,n$. For notational brevity we omit this throughout, and $\Pr[a_i(t+1) = A \mid a_{-i}(t) = a_{-i}]$, for example, is understood to mean $\Pr[a_i(t+1) = x \mid a_{-i}(t) = a_{-i}$, i selected for update].

Parameter $\beta > 0$ dictates an updating agent's degree of rationality. As $\beta \to \infty$, agent i is increasingly likely to select a utility maximizing action, and as $\beta \to 0$, agent i tends to choose its next action uniformly at random. The joint action at time $t+1$ is $a(t+1) = (a_i(t+1), a_{-i}(t))$.

Joint action, $a \in \mathcal{A}$ is **strictly stochastically stable** [17] under log-linear learning dynamics if, for any $\varepsilon > 0$, there exist $B < \infty$ and $T < \infty$ such that

$$\Pr[a(t) = a] > 1 - \varepsilon, \quad \text{for all } \beta > B, t > T \tag{4.3}$$

where $a(t)$ is the joint action at time $t \in \mathbb{N}$ under log-linear learning dynamics.

Joint action $\vec{x}$ is strictly stochastically stable under log-linear learning dynamics over graphical coordination game G [9]. We will investigate conditions when an adversary can destabilize $\vec{x}$ and stabilize an alternate equilibrium.

Consider the situation where agents in N interact according to the graphical game G, and an adversary seeks to convert as many agents in N to play action y as possible.[2] At each time, $t \in \mathbb{N}$ the adversary attempts to influence a set of agents $S(t) \subseteq N$ by posing as a friendly agent who always plays action y. Agents' utilities, $\tilde{U} : \mathcal{A} \times 2^N \to \mathbb{R}$, are now a function of adversarial and friendly behavior, defined by:

$$\tilde{U_i}((a_i, a_{-i}), S) = \begin{cases} U_i(a_i, a_{-i}) & \text{if } i \notin S \\ U_i(a_i, a_{-i}) & \text{if } a_i = x \\ U_i(a_i, a_{-i}) + 1 & \text{if } i \in S, a_i = y \end{cases} \tag{4.4}$$

where $(a_i, a_{-i}) \in \mathcal{A}$ represents friendly agents' joint action, and $S \subseteq N$ represents the set influenced by the adversary. If $i \in S(t)$, agent i receives an additional payoff of 1 for coordinating with the adversary at action y at time $t \in \mathbb{N}$; to agents in $S(t)$ the adversary appears to be a neighbor playing action y. By posing as a player in the game, the adversary has manipulated the utilities of agents belonging to S, providing an extra incentive to choose the inferior alternative, y.

[2] In this paper we consider a single adversary which may influence multiple agents. Our models can be extended to multiple adversaries whose objectives are either aligned or conflicting.

Suppose agents revise their actions according to log-linear learning as in (4.2), where the utility, U_i defined in (4.1) is replaced by $\tilde{U}_i$ in (4.4). An agent $i \in N$ which revises its action at time $t \in \mathbb{N}$ bases its new action choice on the utility $\tilde{U}_i(a(t), S(t))$ if $i \in S(t)$, increasing the probability that agent i updates its action to y. By posing as a player in the coordination game, an adversary manipulates agents' utility functions.thereby modifying their decision making rules.

4.2 Summary of results

In the following sections, we will precisely define three models of adversarial behavior: fixed, intelligent; mobile, random; and mobile, intelligent. Each type of adversary has a fixed capability, k, i.e., $|S(t)| = k$ for all $t \in \mathbb{N}$. Our analysis of these models will provide insight into an adversary's influence on a general graph, $\mathcal{G}$, and we derive exact bounds on α for adversarial influence on a line. Values of α for which each type of agent can stabilize $\vec{y}$ in the line are summarized below and in Figure 4.1.

- A fixed, intelligent adversary with capability k can stabilize joint action $\vec{y}$ when $\alpha < k/(n-1)$ (Theorem 7).

- A mobile, random adversary with capability $k \leq n-1$ can stabilize joint action $\vec{y}$ when $\alpha < 1$ (Theorem 8).

- A mobile, intelligent adversary with capability $k = 1$ can stabilize joint action $\vec{y}$ when $\alpha < 1$ (Theorem 9).

- A mobile, intelligent adversary with capability $k \geq 2$ can stabilize joint action $\vec{y}$ when $\alpha < n/(n-1)$ (Theorem 9).

Note that a mobile, random adversary's influence is the same for any capability k with $1 \leq k \leq n-1$. Similarly, a mobile, intelligent adversary does not increase its influence on agents in a line by increasing its capability above $k = 2$.

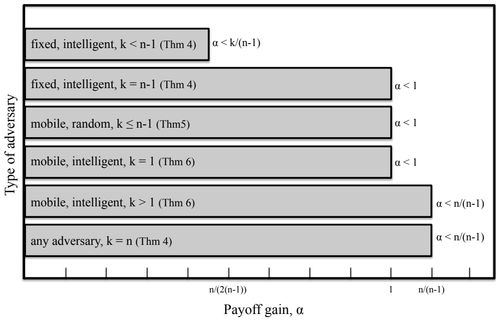

Figure 4.1: Values of α for which each type of adversary can stabilize joint action $\vec{y}$ in an n-agent line

4.3 Main results

Here, we present a detailed version of the results summarized above.

4.3.1 Universal resilience to an adversary

A graphical coordination game G is universally resilient to an adversary if $\vec{x}$ is strictly stochastically stable for all possible influenced sets $S(t)$, $t \in N$ and adversarial capability, $k \leq n$. The following theorem provides sufficient conditions that ensure G is universally resilient. For sets $S, T \subseteq N$, define

$$d(S,T) := |\{\{i,j\} \in E \,:\, i \in S, j \in T\}|.$$

Theorem 4. *Let $\mathcal{G} = (N, E)$, and suppose an adversary influences some set $S(t)$ with $|S(t)| = k$ at each $t \in \mathbb{N}$. If*

$$\alpha > \frac{|T| - d(T, N \setminus T)}{d(T, N)}, \quad \forall T \subseteq N \tag{4.5}$$

Then $\vec{x}$ is strictly stochastically stable. In particular, if $|S(t)| = N$ for all $t \in \mathbb{N}$, (4.5) *is also a necessary condition for strict stochastic stability of $\vec{x}$.*

The proof of Theorem 4 follows by using a straightforward adaptation of Proposition 2 in [63] to our adversarial model, included in Appendix B.3.2

When α satisfies (4.5), an adversary cannot influence the game for any $S(t)$. If $\vec{x}$ is strictly stochastically stable when the adversary influences set $S(t) = N$ for all $t \in \mathbb{N}$, then $\vec{x}$ will be strictly stochastically stable for any sequence of influenced sets, $S(t) \subseteq N$. In this case, game G is resilient in the presence of any adversary with capability $k \leq n$.[3]

When (4.5) is satisfied for some $T \subseteq N$, this means that agents in T have a sufficiently large proportion of neighbors in N. In this case, T can only be influenced by an adversary when the payoff gain, α, is small.

4.3.2 Fixed, intelligent adversarial influence

In the fixed, intelligent model of behavior, the adversary knows graph structure, $\mathcal{G}$, and the value of payoff gain, α. Using this information it influences some fixed subset,

$$S(t) = S \subseteq N, |S| = k, \quad \forall t \in \mathbb{N},$$

aiming to maximize the number of agents playing y in a stochastically stable state. Agents in N update their actions according to log-linear learning as in (4.2) with utilities

$$\tilde{U}_i(a(t), S(t)) = \tilde{U}_i(a(t), S), \quad \forall t \in \mathbb{N}.$$

We begin with two theorems which provide conditions for stochastic stability in an arbitrary graph $\mathcal{G}$ influenced by an adversary, and then we analyze stability conditions in detail for the line.

Theorem 5. *Suppose agents in N are influenced by a fixed, intelligent adversary with capability k. Joint action $\vec{x}$ is strictly stochastically stable for any influenced set S with $|S| = k$ if and only if*

$$\alpha > \frac{|T \cap S| - d(T, N \setminus T)}{d(T, N)}, \tag{4.6}$$

$\forall T \subseteq N$, $T \neq \emptyset$ and $\forall S \subseteq N$ with $|S| = k$.

Theorem 6 provides conditions which ensure an adversary can stabilize joint action $\vec{y}$.

[3] Our results can naturally be extended to a multi-agent scenario. The primary differences occur when multiple adversaries can influence a single friendly agent (or, equivalently, when an adversary's influence is weighted by some factor greater than 1). In this scenario, multiple adversaries can more easily overpower the influence of friendly agents on agent i. We will address this in future work.

Theorem 6. *A fixed, intelligent adversary with capability k can stabilize $\vec{y}$ by influencing set $S \subseteq N$ with $|S| = k$ if and only if*

$$\alpha < \frac{d(T, N \setminus T) + k - |T \cap S|}{d(N \setminus T, N \setminus T)} \tag{4.7}$$

for all $T \subseteq N$, $T \neq N$.

The proofs of Theorems 5 and 6 follow similarly to the proof of Theorem 4 and are omitted for brevity.

The line: We now analyze a fixed, intelligent adversary's influence on the line. Let $\mathcal{G} = (N, E)$ with $N = \{1, 2, \ldots, n\}$ and $E = \{\{i, j\} : j = i + 1\}$, i.e., $\mathcal{G}$ is a line with n nodes. Define

$$[t] := \{1, 2, \ldots, t\} \subseteq N, \text{ and } [i, j] := \{i, i + 1, \ldots, j\} \subseteq N.$$

Theorem 7 summarizes stability conditions for the line influenced by a fixed, intelligent adversary.

Theorem 7. *Suppose $\mathcal{G}$ is influenced by a fixed, intelligent adversary with capability k. Then:*

(1) Joint action $\vec{x}$ is strictly stochastically stable under any influenced set $S \subseteq N$ with $|S| = k$ if and only if

$$\alpha > \max\left\{\frac{k-1}{k}, \frac{k}{n-1}\right\}. \tag{4.8}$$

(2) If $\alpha < \frac{k}{n-1}$ and the adversary distributes influenced set S as evenly as possible along the line, so that

$$|S \cap [i, i + t]| \leq \left\lceil \frac{kt}{n} \right\rceil$$

for any set of nodes $[i, i + t] \subseteq N$, with $1 \leq i \leq n - t$, $t \leq n$ then $\vec{y}$ is strictly stochastically stable.

(3) Joint action $\vec{y}$ is strictly stochastically stable for all influenced sets S with $|S| = k$ if and only if

$$\alpha < \frac{1 + k - t}{n - t - 1}, \quad \forall t = 1, \ldots, k. \tag{4.9}$$

(4) If $\frac{k}{n-1} < \alpha < \frac{k-1}{k}$, the adversary can influence at most

$$t_{\max} = \max\left\{ t \,:\, \alpha < \frac{\min\{t,k\}-1}{t} \right\}$$

agents to play $\vec{y}$ in the stochastically stable state by distributing S as evenly as possible along $[t]$, so that

$$|S \cap [i, i+\ell]| \le \left\lceil \frac{k\ell}{t} \right\rceil \text{ and } S \cap [t+1, n] = \emptyset$$

for any set of nodes $[i, i+\ell] \subset N$ with $1 \le i \le t-\ell$, and $\ell < t$.

The proof of Theorem 7 is in Appendix B.3.2.

4.3.3 Mobile, random adversarial influence

Now, consider an adversary which influences a randomly chosen set $S(t) \subseteq N$ at each $t \in \mathbb{N}$. The adversary chooses each influenced set, $S(t)$, independently according to a uniform distribution over $\mathcal{S}_k := \{S \in 2^N \,:\, |S| = k\}$ An updating agent $i \in N$ revises according to (4.2), where $i \in S(t)$ with probability $k\,/\,n$.

The line: Suppose a mobile, random adversary attempts to influence a set of agents arranged in a line. Theorem 7 addresses the scenario where $k = n$, since in this case random and fixed agents are equivalent. Hence, Theorem 8 focuses on the case where $1 \le k \le n-1$.

Theorem 8. *Suppose $\mathcal{G} = (N, E)$ is a line, and agents in N update their actions according to log-linear learning in the presence of a random, mobile adversary with capability k, where $1 \le k \le n-1$. Then joint action $\vec{x}$ is strictly stochastically stable if and only if $\alpha > 1$, and joint action $\vec{y}$ is strictly stochastically stable if and only if $\alpha < 1$.*

Theorem 8 is proved in Appendix B.3.3.

Note that a mobile, random adversary with capability $k = 1$ stabilizes $\vec{y}$ for the same values of α as a mobile, random adversary with any capability $k \le n-1$. Recall that a fixed, intelligent adversary with capability k could only stabilize $\vec{y}$ when $\alpha < k\,/(n-1)$. In this sense, a mobile, random adversary with capability $k = 1$ has wider influence than a fixed, intelligent adversary with capability $k \le n-2$.

4.3.4 Mobile, intelligent adversarial influence

Now suppose the adversary chooses $S(t)$ at each $t \in \mathbb{N}$ based on joint action, $a(t)$. We assume a mobile, intelligent adversary with capability k chooses $S(t)$ according to a policy $\mu : \mathcal{A} \to \mathcal{S}_k$ that maximizes the number of agents playing y in a stochastically stable state, given graph structure, $\mathcal{G}$, and payoff gain α. Again, agents in N update their actions according to log-linear learning as in (4.2), with agent i's utility at time $t \in \mathbb{N}$ given by $\tilde{U}_i(a(t), \mu(a(t))$. We denote the set of optimal adversarial policies for a given capability k by

$$\mathcal{M}_k = \underset{\mu \in M_k}{\arg\max} \max_{a \text{ stable under } \mu} |\{i \in N \ : \ a_i = y\}| \tag{4.10}$$

where M_k represents the set of all mappings $\mu : \mathcal{A} \to \mathcal{S}_k$, and "$a$ stable under μ" denotes that joint action $a \in \mathcal{A}$ is strictly stochastically stable under μ. [4]

The line: Theorem 9 establishes conditions for strict stochastic stability of joint actions $\vec{x}$ and $\vec{y}$ in the line influenced by a mobile, intelligent adversary.

Theorem 9. *Suppose $\mathcal{G} = (N, E)$ is a line, and agents in N update their actions according to log-linear learning. Further suppose a mobile intelligent adversary influences set $S(t)$ at each $t \in \mathbb{N}$ according to an optimal policy for the line, $\mu^\star \in \mathcal{M}_k$.*

(1) If the adversary has capability $k = 1$ then $\vec{x}$ is strictly stochastically stable if and only if $\alpha > 1$, and $\vec{y}$ is strictly stochastically stable if and only if $\alpha < 1$.

In particular, when $k = 1$, the policy $\mu^\star : \mathcal{A} \to \mathcal{S}_1$ with:

$$\mu^\star(a) = \begin{cases} \{1\} & \text{if } a = \vec{x} \\ \{t+1\} & \text{if } a = (\vec{y}_{[t]}, \vec{x}_{[t+1,n]}), \\ & \qquad t \in \{1, 2, \ldots, n-1\} \\ \{1\} & \text{otherwise} \end{cases} \tag{4.11}$$

is optimal, i.e., $\mu^\star \in \mathcal{M}_1$

[4] Note that the optimal set of policies, $\mathcal{M}_k$, depends highly on the structure of graph $\mathcal{G}$, as does the stationary distribution π^μ. In order to maintain notational simplicity, we do not explicitly write this dependence.

(2) If $2 \leq k \leq n$, then $\vec{x}$ is strictly stochastically stable if and only if $\alpha > n\,/(n-1)$, and $\vec{y}$ is strictly stochastically stable if and only if $\alpha < n\,/(n-1)$.

If $2 \leq k \leq n-1$, any policy $\mu^\star : \mathcal{A} \to \mathcal{S}_k$ satisfying:

(a) $1 \in \mu^\star(\vec{x})$

(b) $1, n \in \mu^\star(\vec{y})$

(c) For any $a \in \mathcal{A}$, $a \neq \vec{x}, \vec{y}$, there exists $i \in \mu^\star(a)$ such that $a_i = x$ and either $a_{i-1} = y$ or $a_{i+1} = y$

is optimal.

The proof of Theorem 9 is included in Appendix B.3.4. Recall that a mobile, random agent with $k \geq 1$ and a fixed, intelligent agent with $k = n-1$ can stabilize $\vec{y}$ any time $\alpha < 1$; an adversary who can intelligently influence a different single agent in N each day can stabilize $\vec{y}$ under these same conditions. If the intelligent, mobile adversary has capability $k \geq 2$, it can stabilize $\vec{y}$ when $\alpha < n\,/(n-1)$, i.e., under the same conditions as an adversary with capability $k = n$.

We have shown that a mobile, intelligent adversary with capability $k \geq 2$ can stabilize joint action $\vec{y}$ in a line for any $\alpha < n\,/(n-1)$. Next, an intelligent, mobile adversary with capability $k = 1$ and a random, mobile adversary with capability $k \leq n-1$ can stabilize $\vec{y}$ when $\alpha < 1$. Finally, a fixed, intelligent adversary with capability k can stabilize $\vec{y}$ when $\alpha < k\,/(n-1)$. Recall that a fixed, intelligent adversary can also stabilize a joint action where some subset of agents play action y; this only occurs when $\alpha < (\min\{t, k\} - 1)\,/\,t < 1$ for some $t \leq n$.

In future work, we will address the scenario where multiple adversaries aim to influence agents in N.By heavily influencing a single agent, adversaries can cause this agent to choose action y with near certainty. Due to cascading effects, this can allow adversaries to stabilize joint action $\vec{y}$ for significantly larger values of payoff gain, α.

Chapter 5

Conclusion

This is where the conclusion will go.

fix appendix titles for adversarial work to include theorem names

Bibliography

[1] C. Alós-Ferrer and N. Netzer. The logit-response dynamics. Games and Economic Behavior, 68(2):413–427, 2010.

[2] T. Alpcan and T. Basar. Network Security: A Decision and Game-Theoretic Approach. Cambridge University Press, 1st edition, 2010.

[3] I. Arieli and Y. Babichenko. Average Testing and the Efficient Boundary. Journal of Economic Theory, 147:2376–2398, 2012.

[4] I. Arieli and H.P. Young. Fast convergence in population games. 2011.

[5] G. Arslan, J. R. Marden, and J. S. Shamma. Autonomous vehicle-target assignment: a game theoretical formulation. ASME Journal of Dynamic Systems, Measurement and Control, 129(5):584–596, 2007.

[6] R. Aumann. Correlated Equilibrium as an Expression of Bayesian Rationality. Econometrica, 55(1):1–18, 1987.

[7] C.Z. Bai and V. Gupta. On Kalman filtering in the presence of a compromised sensor: Fundamental performance bounds. Proceedings of the American Control Conference, 2014.

[8] M. Beckmann, C.B. McGuire, and C. B. Winsten. Studies in the Economics of Transportation. Yale University Press, New Haven, 1956.

[9] L. Blume. The statistical mechanics of strategic interaction. Games and Economic Behavior, 1993.

[10] L. Blume. Population Games. Econometrica, 68(5):1127–1150, 1996.

[11] O. Boussaton and J. Cohen. On the distributed learning of Nash equilibria with minimal information. 6th International Conference on Network Games, Control, and Optimization, 2012.

[12] A. Carfang, E. Frew, and D. Kingston. A Cascaded Approach to Optimize Aircraft Trajectories for Persistent Data Ferrying. In AIAA Gui, 2013.

[13] R. Cooper. Coordination Games. Cambridge University Press, Cambridge, UK, 1999.

[14] G. Ellison. Basins of attraction, long-run stochastic stability, and the speed of step-by-step evolution. The Review of Economic Studies, pages 17–45, 2000.

[15] H. Fawzi, P. Tabuada, and S. Diggavi. Secure estimation and control for cyber-physical systems under adversarial attacks. IEEE Transactions on Automatic Control, 59:1454–1467, 2014.

[16] D. Foster and R. Vohra. Calibrated Learning and Correlated Equilibrium. Games and Economic Behavior, 21:40–55, oct 1997.

[17] D. Foster and H. Young. Stochastic evolutionary game dynamics. Theoretical Population Biology, 38(2), 1990.

[18] D. Foster and H. Young. Regret testing: learning to play Nash equilibrium without knowing you have an opponent. Theoretical Economics, 1:341–367, 2006.

[19] D. P. Foster and H. P. Young. On the Nonconvergence of Fictitious Play in Coordination Games. Games and Economic Behavior, 25(1):79–96, oct 1998.

[20] M. Fox and J. Shamma. Communication, convergence, and stochastic stability in self-assembly. In 49th IEEE Conference on Decision and Control (CDC), dec 2010.

[21] A. Frieze and R. Kannan. Log-Sobolev inequalities and sampling from log-concave distributions. The Annals of Applied Probability, 1998.

[22] P. Frihauf, M. Krstic, and T. Bas. Nash Equilibrium Seeking in Noncooperative Games. IEEE Transactions on Automatic Control, 57(5):1192–1207, 2012.

[23] B. Gharesifard and J. Cortes. Distributed convergence to Nash equilibria by adversarial networks with directed topologies. In 51st IEEE Conference on Decision and Control, 2012.

[24] T. Goto, T. Hatanaka, and M. Fujita. Potential game theoretic attitude coordination on the circle: Synchronization and balanced circular formation. 2010 IEEE International Symposium on Intelligent Control, pages 2314–2319, sep 2010.

[25] Z. Han, D. Niyato, W. Saad, T. Baar, and A. Hjørungnes. Game Theory in Wireless and Communication Networks: Theory, Models, and Applications. Cambridge University Press, 1st edition, 2012.

[26] S. Hart and Y. Mansour. How long to equilibrium? The communication complexity of uncoupled equilibrium procedures. Games and Economic Behavior, 69:107–126, may 2010.

[27] S. Hart and A. Mas-Colell. A Simple Adaptive Procedure Leading to Correlated Equilibrium. Econometrica, 68(5):1127–1150, 2000.

[28] J.M. Hendrickx, K.H Johansson, R.M. Jungers, H. Sandberg, and K.C. Sou. Efficient Computations of a Security Index for False Data Attacks in Power Networks. IEEE Transactions on Automatic Control, 59(12):3194–3208, 2014.

[29] Y. Ho and F. Sun. Value of Information in Two-Team Zero-Sum Problems. Journal of Optimization Theory and Applications, 14(5), 1974.

[30] A. Jiang and K. Leyton-Brown. Polynomial-time computation of exact correlated equilibrium in compact games. In Proceedings of the Twelfth ACM Electronic Commerce Conference, feb 2011.

[31] M. Kandori, G. Mailath, and R. Rob. Learning, Mutation, and Long Run Equilibria in Games. Econometrica, 61(1):29–56, 1993.

[32] M. Kearns, M.L. Littman, and S. Singh. Graphical Models for Game Theory. In 17th Conference in Uncertainty in Artificial Intelligence, 2001.

[33] G. Kreindler and H. Young. Fast convergence in evolutionary equilibrium selection. 2011.

[34] S. Lasauce and H. Tembine. Game Theory and Learning for Wireless Networks. Elsevier, 1st edition, 2011.

[35] Y. Lim. Game Theoretic Distributed Coordination. PhD thesis, Georgia Tech, 2014.

[36] A. MacKenzie and L. DaSilva. Game Theory for Wireless Engineers. Morgan & Claypool Publishers, 1st edition, 2006.

[37] J. Marden. Joint strategy fictitious play with inertia for potential games. IEEE Transactions on Automatic Control, 54(2):208–220, 2009.

[38] J. Marden. Selecting Efficient Correlated Equilibria Through Distributed Learning. in submission, 2013.

[39] J. Marden, G. Arslan, and J. Shamma. Connections between cooperative control and potential games illustrated on the consensus problem. Proceedings of 2007 the European Control Conference, 2007.

[40] J. Marden, G. Arslan, and J. Shamma. Regret based dynamics: convergence in weakly acyclic games. In Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems, volume 5, 2007.

[41] J. Marden and J. Shamma. Revisiting log-linear learning: Asynchrony, completeness and payoff-based implementation. Games and Economic Behavior, 75(2):788–808, 2012.

[42] J. Marden and A. Wierman. Distributed welfare games. Operations Research, 61(1):155–168, 2013.

[43] J. Marden, H. Young, G. Arslan, and J. Shamma. Payoff Based Dynamics for Multi-Player Weakly Acyclic Games. SIAM Journal on Control and Optimization, 48(1):373–396, 2009.

[44] J. Marden, H. Young, and L. Pao. Achieving pareto optimality through distributed learning. dec 2011.

[45] I. Menache and A. Ozdaglar. Network Games: Theory, Models, and Dynamics. Morgan & Claypool Publishers, 1st edition, 2011.

[46] D. Monderer and L. Shapley. Fictitious Play Property for Games with Identical Interests. Journal of Economic Theory, (68):258–265, 1996.

[47] D. Monderer and L. Shapley. Potential games. Games and Economic Behavior, 14:124–143, 1996.

[48] A. Montanari and A. Saberi. The spread of innovations in social networks. Proceedings of the National Academy of Sciences, pages 20196–20201, 2010.

[49] R. Montenegro and P. Tetali. Mathematical Aspects of Mixing Times in Markov Chains. Foundations and Trends in Theoretical Computer Science, 1(3):237–354, 2006.

[50] LE Ortiz. Maximum entropy correlated equilibria. AISTATS, pages 347–354, 2007.

[51] C. Papadimitriou and T. Roughgarden. Computing correlated equilibria in multi-player games. Journal of the ACM, 55(3):1–29, jul 2008.

[52] F. Pasqualetti, F. Dörfler, and F. Bullo. Cyber-physical security via geometric control: Distributed monitoring and malicious attacks. Proceedings of the IEEE Conference on Decision and Control, (i):3418–3425, 2012.

[53] J. Poveda and N. Quijano. Distributed Extremum Seeking for Real-Time Resource Allocation. In American Control Conference, 2013.

[54] B. Pradelski and H. Young. Learning efficient Nash equilibria in distributed systems. 2012.

[55] D. Shah and J. Shin. Dynamics in Congestion Games. In ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, 2010.

[56] M. Staudigl. Stochastic stability in asymmetric binary choice coordination games. Games and Economic Behavior, 75(1):372–401, may 2012.

[57] E. Ullmann-Margalit. The Emergence of Norms. Oxford University Press, 1977.

[58] B. von Stengel and D. Koller. Team-Maxmin Equilibria. Games and Economic Behavior, 21(1-2):309 – 321, 1997.

[59] D. Wolpert and K. Tumer. An Introduction To Collective Intelligence. Technical report, Handbook of Agent technology. AAAI, 1999.

[60] D. Wolpert and K. Tumer. Optimal Payoff Functions for Members of Collectives. Advances in Complex Systems, 4:265–279, 2001.

[61] H. Young. The Evolution of Conventions. Econometrica, 61(1):57–84, 1993.

[62] H. Young. Learning by trial and error. Games and economic behavior, 65:626–643, 2009.

[63] H. P. Young. Colloquium Paper: The dynamics of social innovation. Proceedings of the National Academy of Sciences, 108:21285–21291, 2011.

[64] M. Zhu and S. Martinez. Distributed coverage games for mobile visual sensors (I): Reaching the set of Nash equilibria. In Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference, 2009.

Appendix A

Technical Preliminaries

A.1 Markov chain preliminaries

A continuous time Markov chain, $\{Z_t\}_{t\geq 0}$, over a finite state space Ω may be written in terms of a corresponding discrete time chain with transition matrix M [49], where the distribution $\mu(t)$ over Ω evolves as:

$$\mu(t) = \mu(0)e^{t(M-I)} = \mu(0)e^{-t}\sum_{k=0}^{\infty}\frac{t^k M^k}{k!}, \quad t \geq 0 \tag{A.1}$$

where we refer to M as the kernel of the process Z_t and $\mu(0) \in \Delta(\Omega)$ is the initial distribution. The following definitions and theorems are taken from [55, 49]. Let μ, ν be measures on the finite state space Ω. Total variation distance is defined as

$$\|\mu - \nu\|_{TV} := \frac{1}{2}\sum_{x\in\Omega}|\mu_x - \nu_x|. \tag{A.2}$$

and

$$D(\mu : \nu) := \sum_{x\in\Omega}\mu_x \log\frac{\mu_x}{\nu_x} \tag{A.3}$$

is defined to be the relative entropy between μ and ν. The total variation distance between two distributions can be bounded using the relative entropy:

$$\|\mu - \nu\|_{TV} \leq \sqrt{\frac{D(\mu : \nu)}{2}} \tag{A.4}$$

For a continuous time Markov chain with kernel M and stationary distribution π, the distribution $\mu(t)$ obeys

$$D(\mu(t) : \pi) \leq e^{-4t\rho(M)}D(\mu(0) : \pi), \quad t \geq 0 \tag{A.5}$$

where $\rho(M)$ is the Sobolev constant of M, defined by

$$\rho(M) := \inf_{\substack{f:\Omega\to\mathbb{R}:\\ \mathcal{L}(f)\neq 0}} \frac{\mathcal{E}(f,f)}{\mathcal{L}(f)} \tag{A.6}$$

with

$$\mathcal{E}(f,f) := \frac{1}{2}\sum_{x,y\in\Omega}(f(x)-f(y))^2 M(x,y)\pi_x \tag{A.7}$$

$$\mathcal{L}(f) := \mathbb{E}_\pi \log \frac{f^2}{\mathbb{E}_\pi f^2}. \tag{A.8}$$

Here $\mathbb{E}_\pi$ denotes the expectation with respect to stationary distribution π. For a Markov chain with initial distribution $\mu(0)$ and stationary distribution π, the total variation and relative entropy mixing times are

$$T_{TV}(\varepsilon) := \min_t\{\|\mu(t)-\pi\| \leq \varepsilon\} \tag{A.9}$$

$$T_D(\varepsilon) := \min_t\{D(\mu(t):\pi) \leq \varepsilon\} \tag{A.10}$$

respectively. From [49], Corollary 2.6 and Remark 2.11,

$$T_D(\varepsilon) \leq \frac{1}{4\rho(M)}\left(\log\log\frac{1}{\pi_{\min}} + \log\frac{1}{\varepsilon}\right),$$

where $\pi_{\min} := \min_{x\in\Omega}\pi_x$. Applying (A.4),

$$\begin{aligned} T_{TV}(\varepsilon) &\leq T_D(2\varepsilon^2) \\ &\leq \frac{1}{4\rho(M)}\left(\log\log\frac{1}{\pi_{\min}} + 2\log\frac{1}{\varepsilon}\right). \end{aligned} \tag{A.11}$$

Hence, a lower bound on the Sobolev constant yields an upper bound on the mixing time for the Markov chain.

A.2 Resistance Trees

below is the resistance trees background from the CCE paper. Need to review and combine with the other resistance trees background (consistent notation, etc)

Define P^0 as the transition matrix for some nominal Markov process, and let P^ε be a perturbed version of this nominal process where the size of the perturbation is $\varepsilon > 0$. Throughout this paper, we focus on the following class of Markov chains.

Definition 3. A family of Markov chains defined over a finite state space X, whose transition matrices are denoted by $\{P^{\varepsilon}\}_{\varepsilon>0}$, is called a **regular perturbed process** of a nominal process P^0 if the following conditions are satisfied for all $x, x' \in X$:

(1) There exists a constant $c > 0$ such that P^{ε} is aperiodic and irreducible for all $\varepsilon \in (0, c]$.

(2) $\lim_{\varepsilon \to 0} P^{\varepsilon}_{x \to x'} = P^{0}_{x \to x'}$.

(3) If $P^{\varepsilon}_{x \to x'} > 0$ for some $\varepsilon > 0$, then there exists a constant $r(x \to x') \geq 0$ such that

$$0 < \lim_{\varepsilon \to 0} \frac{P^{\varepsilon}_{x \to x'}}{\varepsilon^{r(x \to x')}} < \infty. \tag{A.12}$$

The constant $r(x \to x')$ is referred to as the **resistance** of the transition $x \to x'$.

For any $\varepsilon > 0$, let $\mu^{\varepsilon} = \{\mu^{\varepsilon}_x\}_{x \in X} \in \Delta(X)$ denote the unique stationary distribution associated with P^{ε}. The theory of resistance trees presented in [61] provides efficient mechanisms for computing the support of the limiting stationary distribution, i.e., $\lim_{\varepsilon \to 0^+} \mu^{\varepsilon}$, commonly referred to as the stochastically stable states.

Definition 4. A state $x \in X$ is **stochastically stable** [17] if $\lim_{\varepsilon \to 0^+} \mu^{\varepsilon}_x > 0$, where μ^{ε} is the stationary distribution corresponding to P^{ε}.

In this paper, we adopt the technique provided in [61] for identifying the stochastically stable states through a graph theoretic analysis over the recurrent classes of the unperturbed process P^0. To that end, let $Y_0, Y_1, \ldots, Y_m$ denote the recurrent classes of P^0. Define $\mathcal{P}_{ij}$ to be the set of all paths connecting Y_i to Y_j, i.e., a path $p \in \mathcal{P}_{ij}$ is of the form $p = \{(x_1, x_2), (x_2, x_3), \ldots, (x_{k-1}, x_k)\}$ where $x_1 \in Y_i$ and $x_k \in Y_j$. The resistance associated with transitioning from Y_i to Y_j is defined as

$$r(Y_i, Y_j) = \min_{p \in \mathcal{P}_{ij}} \sum_{(x,x') \in p} r(x, x'). \tag{A.13}$$

The recurrent classes $Y_0, Y_1, \ldots, Y_m$ satisfy the following properties: (i) there is a zero resistance path, i.e., a sequence of transitions each with zero resistance, from any state $x \in X$ to at least one state y in one of the recurrent classes; (ii) for any recurrent class Y_i and any states $y_i, y'_i \in Y_i$, there is a zero resistance path from y_i to y_i'; and (iii) for any state $y_i \in Y_i$ and $y_j \in Y_j$, $Y_i \neq Y_j$, any path from y_i to y_j has strictly positive resistance.

The first step in identifying the stochastically stable states is to identify the resistance between the various recurrent classes. The second step focuses on analyzing spanning trees of the weighted, directed graph $\mathcal{G}$ whose vertices are recurrent classes of the process P^0, and whose edge weights are given by the resistances between classes in (A.13). Denote $\mathcal{T}_i$ to be the set of all spanning trees of $\mathcal{G}$ rooted at recurrent class Y_i. Next, we compute the stochastic potential of each recurrent class which is defined as follows:

Definition 5. The **stochastic potential** of recurrent class Y_i is

$$\gamma(Y_i) = \min_{\mathcal{T} \in \mathcal{T}_i} \sum_{(Y,Y') \in \mathcal{T}} r(Y, Y')$$

The following theorem characterizes the recurrent classes that are stochastically stable.

Theorem 10 ([61]). *Let P^0 be the transition matrix for a stationary Markov process over the finite state space X with recurrent communication classes $Y_1, \dots, Y_m$. For each $\varepsilon > 0$, let P^ε be a regular perturbation of P^0 with a unique stationary distribution μ^ε. Then:*

(1) As $\varepsilon \to 0$, μ^ε converges to a stationary distribution μ^0 of P^0.

(2) A state $x \in X$ is stochastically stable if and only if x is contained in a recurrent class Y_j that minimizes $\gamma(Y_j)$.

Appendix B

Proofs

B.1 Fast Convergence in Semi-Anonymous Potential Games: Background and Proofs

We begin with a problem formulation and notation summary for semi-anonymous potential games, and then proceed with a proof of Theorem 1. Next, we provide a problem formulation and summary for time-varying semi-anonymous potential games, followed by a proof of Theorem 2.

B.1.1 Semi-Anonymous Potential Games

The following Markov chain, M, over state space X is the kernel of the continuous time modified log-linear learning process for stationary semi anonymous potential games. Define $n_j := |N_j|$ to be the size of population j, define $s_j := |\bar{\mathcal{A}}_j|$, and let $\sigma := \sum_{j=1}^{m} s_j$. Let $e_j^k \in \mathbb{R}^{s_j}$ be the kth standard basis vector of length s_j for $k \in \{1, \ldots, s_j\}$. Finally, let

$$x = (x_j, \boldsymbol{x}_{-}j) = (x_1, x_2, \ldots, x_m) \in X,$$

where $x_j = (x_j^1, x_j^2, \ldots, x_j^{s_j})$ represents the proportion of players choosing each action within population j's action set. The state transitions according to:

- Choose a population $N_j \in \{N_1, N_2, \ldots, N_m\}$ with probability s_j/σ.
- Choose an action $\bar{a}_j^k \in \{\bar{a}_j^1, \bar{a}_j^2, \ldots, \bar{a}_j^{s_j}\} = \bar{\mathcal{A}}_j$ with probability $1/s_j$.
- If $x_j^k > 0$, i.e., at least one player from population j is playing action $\bar{a}_j^k$, choose $p \in \{p' \in N_j : \bar{a}_{p'} = \bar{a}_j^k\}$ uniformly at random to update according to (2.7). That is, transition to

$\left(x_j + \frac{1}{n}(e_j^\ell - e_j^k), x_{-j}\right)$ with probability

$$\frac{e^{\beta\phi\left(x_j+\frac{1}{n}(e_j^\ell-e_j^k),x_{-j}\right)}}{\sum_{t=1}^{s_j} e^{\beta\phi(x_j+\frac{1}{n}(e_j^\ell-e_j^k),x_{-j})}}$$

for each $\ell \in \{1, 2, \ldots, s_j\}$. [1]

This defines transition probabilities in M for transitions from state $x = (x_j, x_{-j}) \in X$ to a state of the form $y = \left(x_j + \frac{1}{n}(e_j^\ell - e_j^k), x_{-j}\right) \in X$ in which a player from population N_j updates his action, so that

$$M(x,y) = \frac{e^{\beta\phi\left(x_j+\frac{1}{n}(e_j^\ell-e_j^k),x_{-j}\right)}}{\sigma\sum_{t=1}^{s_j} e^{\beta\phi(x_j+\frac{1}{n}(e_j^t-e_j^k),x_{-j})}} \tag{B.1}$$

For a transition of any other form, $M(x, y) = 0$. Applying (A.1) to the chain with kernel M and global clock rate $\alpha\sigma n$, modified log-linear learning evolves as

$$\mu(t) = \mu(0)e^{\alpha\sigma n t(M-I)}. \tag{B.2}$$

Notation summary for stationary semi-anonymous potential games: Let $G = \{N, \{A_i\}, \{U_i\}\}$ be a stationary semi-anonymous potential game. The following summarizes the notation corresponding to game G.

- X - aggregate state space corresponding to the game G
- $\phi : X \to \mathbb{R}$ - the potential function corresponding to game G
- M - probability transition kernel for the modified log-linear learning process
- α - design parameter for modified log-linear learning which may be used to adjust the global update rate

[1] Agents' update rates are the only difference between our algorithm, standard log-linear learning, and the log-linear learning variant of [55]. In standard log-linear learning, players have uniform, constant clock rates. In our variant and the variant of [55], agents' update rates vary with the state. For the algorithm in [55], agent i's update rate is $\alpha n \,/\, \tilde{z}_i(t)$, where $\tilde{z}_i(t)$ is the **total** number of players selecting the same action as agent i. The discrete time kernel of this process is as follows [55]: (1) Select an action $a_i \in \cup_{i\in N} A_i$ uniformly at random. (2) Select a player who is currently playing action a_i uniformly at random. This player updates its action according to (2.7). The two algorithms differ when at least two populations have overlapping action sets.

- $\mu(t) = \mu(0)e^{\alpha nt(M-I)}$ - distribution over state space X at time t when beginning with distribution $\mu(0)$ and following the modified log-linear learning process
- N_j - the jth population
- $n_j := |N_j|$ - the size of the jth population
- $\bar{\mathcal{A}}_j$ - action set for agents belonging to population N_j
- $\bar{a}_j^k$ - the kth action in population N_j's action set
- $s := | \cup_{j=1}^m \overline{\mathcal{A}}_j|$ - size of the union of all populations' action sets
- $s_j := |\bar{\mathcal{A}}_j|$ - size of population N_j's action set
- $e_j^k \in \mathbb{R}^{s_j}$ - kth standard basis vector of length s_j
- $\sigma := \sum_{j=1}^m s_j$ - sum of sizes of each population's action set
- π - stationary distribution corresponding to the modified log-linear learning process for game G.
- $(x_j, x_{-j}) = (x_1, x_2, \ldots, x_m) \in X$, a state in the aggregate state space, where $x_j = (x_j^1, x_j^2, \ldots, x_j^{s_j})$.

B.1.2 Proof of Theorem 1

We require two supporting lemmas to prove Theorem 1. The first establishes the stationary distribution for modified log-linear learning as a function of β and characterizes how large β must be so the expected value of the potential function is within $\varepsilon/2$ of maximum. The second upper bounds the mixing time to within $\varepsilon/2$ of the stationary distribution for the modified log-linear learning process.

Lemma 1. *For the stationary semi-anonymous potential game $G = (N, \mathcal{A}_i, U_i)$ with state space X and potential function $\phi : X \to [0, 1]$, the stationary distribution for modified log-linear learning is*

$$\pi_x \propto e^{\beta\phi(x)}, \quad x \in X \tag{B.3}$$

Moreover, if condition (i) of Theorem 1 is satisfied and β *is sufficiently large as in* (2.8)*, then*

$$\mathbb{E}_\pi[\phi(x)] \geq \max_{x\in X} \phi(x) - \varepsilon/2. \tag{B.4}$$

Proof: The form of the stationary distribution follows from standard reversibility arguments, using (B.1) and (B.3).

For the second part of the proof, define the following:

$$\begin{aligned} C_\beta &:= \sum_{x\in X} e^{\beta\phi(x)}, \\ x^\star &:= \arg\max_{x\in X} \phi(x) \\ B(x^\star,\delta) &:= \{x \in X \,:\, \|x - x^\star\|_1 \leq \delta\} \end{aligned}$$

where $\delta \in [0,1]$ is a constant which we will specify later. Because π is of exponential form with normalization factor C_β, the derivative of $\log C_\beta$ with respect to β is $\mathbb{E}_\pi[\phi(x)]$. Moreover, it follows from (B.3) that $\mathbb{E}_\pi[\phi(x)]$ is monotonically increasing in β, so we may proceed as follows:

$$\begin{aligned} \mathbb{E}_\pi[\phi(x)] &\geq \frac{1}{\beta}(\log C_\beta - \log C_0) \\ &= \phi(x^\star) + \frac{1}{\beta}\log\frac{\sum_{x\in X} e^{\beta(\phi(x)-\phi(x^\star))}}{|X|} \\ &\overset{(a)}{\geq} \phi(x^\star) + \frac{1}{\beta}\log\frac{\sum_{x\in B(x^\star,\delta)} e^{-\beta\delta\lambda}}{|X|} \\ &= \phi(x^\star) + \frac{1}{\beta}\log\frac{|B(x^\star,\delta)|e^{-\beta\delta\lambda}}{|X|} \\ &= \phi(x^\star) - \delta\lambda + \frac{1}{\beta}\log\left(\frac{|B(x^\star,\delta)|}{|X|}\right) \end{aligned}$$

where (a) is from the fact that ϕ is λ-Lipschitz and the definition of $B(x^\star,\delta)$. Using intermediate results in the proof of Lemma 6 of [55], $|B(x^\star,\delta)|$ and $|X|$ are bounded as:

$$|B(x^\star,\delta)| \geq \prod_{i=1}^{m}\left(\frac{\delta(n_i+1)}{2ms_i}\right)^{s_i-1}, \text{ and} \tag{B.5}$$

$$|X| \leq \prod_{i=1}^{m}(n_i+1)^{s_i-1}. \tag{B.6}$$

Now,

$$\mathbb{E}_\pi[\phi(x)] \geq \phi(x^\star) - \delta\lambda + \frac{1}{\beta}\log\left(\frac{|B(x^\star,\delta)|}{|X|}\right)$$

$$
\begin{aligned}
&\geq \phi(x^\star) - \delta\lambda + \frac{1}{\beta}\log\left(\frac{\prod_{i=1}^m \left(\frac{\delta(n_i+1)}{2ms_i}\right)^{s_i-1}}{\prod_{i=1}^m (n_i+1)^{s_i-1}}\right) \\
&= \phi(x^\star) - \delta\lambda + \frac{1}{\beta}\log\prod_{i=1}^m\left(\frac{\delta}{2ms_i}\right)^{s_i-1} \\
&\geq \phi(x^\star) - \delta\lambda + \frac{m(s-1)}{\beta}\log\left(\frac{\delta}{2ms}\right)
\end{aligned}
$$

Consider two cases: (i) $\lambda \leq \varepsilon/4$, and (ii) $\lambda > \varepsilon/4$. For case (i), choose $\delta = 1$ and let $\beta \geq \frac{4m(s-1)}{\varepsilon}\log 2ms$. Then,

$$
\begin{aligned}
\mathbb{E}_\pi[\phi(x)] &\geq \phi(x^\star) - \delta\lambda + \frac{m(s-1)}{\beta}\log\left(\frac{\delta}{2ms}\right) \\
&\geq \phi(x^\star) - \varepsilon/4 - \frac{m(s-1)}{\beta}\log 2ms \\
&\geq \phi(x^\star) - \varepsilon/4 - \frac{\varepsilon m(s-1)}{4m(s-1)\log 2ms}\log 2ms \\
&= \phi(x^\star) - \varepsilon/2
\end{aligned}
$$

For case (ii), note that $\lambda > \varepsilon/4 \implies \delta = \varepsilon/4\lambda < 1$ so we may choose $\delta = \varepsilon/4\lambda$. Let $\beta \geq \frac{4m(s-1)}{\varepsilon}\log\left(\frac{8\lambda ms}{\varepsilon}\right)$. Then

$$
\begin{aligned}
\mathbb{E}_\pi[\phi(x)] &\geq \phi(x^\star) - \delta\lambda + \frac{m(s-1)}{\beta}\log\left(\frac{\delta}{2ms}\right) \\
&= \phi(x^\star) - \varepsilon/4 + \frac{m(s-1)}{\beta}\log\left(\frac{\varepsilon}{8\lambda ms}\right) \\
&= \phi(x^\star) - \varepsilon/4 - \frac{m(s-1)}{\beta}\log\left(\frac{8\lambda ms}{\varepsilon}\right) \\
&\geq \phi(x^\star) - \varepsilon/4 - \frac{\varepsilon m(s-1)}{4m(s-1)\log\left(\frac{8\lambda ms}{\varepsilon}\right)}\log\left(\frac{8\lambda ms}{\varepsilon}\right) \\
&= \phi(x^\star) - \varepsilon/2
\end{aligned}
$$

as desired. □

Lemma 2. *For the Markov chain defined by modified log-linear learning with kernel M and stationary distribution π, if the number of players within each population satisfies condition (ii) of Theorem 1, and t is sufficiently large as in* (2.10)*, then*

$$
\|\mu(t) - \pi\|_{TV} \leq \varepsilon/2. \tag{B.7}
$$

Proof: We begin by establishing a lower bound on the Sobolev constant for the Markov chain, M. We claim that, for the Markov chain M defined in Appendix B.1.1, if $\phi : X \to [0,1]$ and $m + \sum_{i=1}^{m} n_i^2 \geq \sigma$, then

$$\rho(M) \geq \frac{e^{-3\beta}}{c_1 m(m(s-1))!^2 n^2} \tag{B.8}$$

for some constant c_1 which depends only on s. Then, from (A.11), a lower bound on the Sobolev constant yields an upper bound on the mixing time for the chain M.

Using the technique of [55], we compare the Sobolev constants for the chain M and a similar random walk on a convex set. The primary difference is that our proof accounts for dependencies on the number of populations, m, whereas theirs considers only the $m = 1$ case. As a result, our state space is necessarily larger. We accomplish this proof in four steps. In step 1, we define $M^\star$ to be the Markov chain M with $\beta = 0$, and establish the bound $\rho(M) \geq e^{-3\beta}\rho(M^\star)$. In step 2, we define a third Markov chain, $M^\dagger$, and establish the bound $\rho(M^\star) \geq \frac{1}{s}\rho(M^\dagger)$. Then, in step 3, we establish a lower bound on the Sobolev constant of $M^\dagger$. Finally, in step 4, we combine the results of the first three steps to establish (B.8). We now prove each step in detail.

Step 1, M to $M^\star$: Let $M^\star$ be the Markov chain M with $\beta = 0$, and let $\pi^\star$ be its stationary distribution. In $M^\star$ an updating agent chooses his next action uniformly at random. Per Equation (B.3) with $\beta = 0$, the stationary distribution $\pi^\star$ of $M^\star$ is the uniform distribution. Let $x, y \in X$. We bound $\pi_x/\pi_x^\star$ and $M(x,y)/M^\star(x,y)$ in order to use Corollary 3.15 in [49]:

$$\frac{\pi_x}{\pi_x^\star} = \frac{e^{\beta\phi(x)}}{\sum_{y\in X} e^{\beta\phi(y)}} \cdot \frac{\sum_{y\in X} e^0}{e^0} = \frac{|X| e^{\beta\phi(x)}}{\sum_{y\in X} e^{\beta\phi(y)}}$$

Since $\phi(x) \in [0,1]$ for all $x \in X$, this implies

$$e^{-\beta} \leq \frac{\pi_x}{\pi_x^\star} \leq e^{\beta} \tag{B.9}$$

Similarly, for $y = (x_j + \frac{1}{n}(e_j^k - e_j^\ell), x_{-j})$,

$$\frac{M(x,y)}{M^\star(x,y)} = \frac{s_j e^{\beta\phi(y)}}{\sum_{r=1}^{s_j} e^{\beta\phi(x_j + \frac{1}{n}(e_i^k - e_i^r), x_{-j})}}$$

Since $\phi(x) \in [0,1]$ for all $x \in X$, for any $x, y \in X$ of the above form,

$$e^{-\beta} \leq \frac{M(x,y)}{M^\star(x,y)} \leq e^{\beta}. \tag{B.10}$$

For a transition to any y not of the form above, $M(x, y) = M^\star(x, y) = 0$. Using this fact and Equations (B.9) and (B.10), we apply Corollary 3.15 in [49]:

$$\rho(M) \geq e^{-3\beta}\rho(M^\star). \tag{B.11}$$

Step 2, $M^\star$ to $M^\dagger$: Consider the Markov chain $M^\dagger$ on X, where transitions from state x to y occur as follows:

- Choose a population N_j with probability s_j/σ
- Choose $k \in \{1, \ldots, s_j - 1\}$ and choose $\kappa \in \{-1, 1\}$, each uniformly at random.
  - $*$ If $\kappa = -1$ and $x_j^k > 0$, then $y = (x_j + \frac{1}{n}(e_j^{s_j} - e_j^k), x_{-k})$.
  - $*$ If $\kappa = 1$ and $x_j^{s_j} > 0$, then $y = (x_j + \frac{1}{n}(e_j^k - e_j^{s_j}), x_{-j})$.

Since $M^\dagger(x, y) = M^\dagger(y, x)$ for any $x, y \in X$, $M^\dagger$ is reversible with the uniform distribution over X. Hence the stationary distribution is uniform, and $\pi^\dagger = \pi^\star$.

For a transition x to y in which an agent from population N_j changes his action, $M^\star(x, y) \geq \frac{1}{s_j}M^\dagger(x, y)$, implying

$$M^\star(x, y) \geq \frac{1}{s}M^\dagger(x, y), \quad \forall x, y \in X \tag{B.12}$$

since $s \geq s_j$, $\forall i \in \{1, \ldots, m\}$. Using (B.12) and the fact that $\pi^\star = \pi^\dagger$, we apply Corollary 3.15 from [49]:

$$\rho(M^\star) \geq \frac{1}{s}\rho(M^\dagger) \tag{B.13}$$

Step 3, $M^\dagger$ to a random walk: The following random walk on

$$C = \left\{ (z_1, \ldots, z_m) \in \mathbb{Z}_+^{\sigma - m} \,:\, z_j \in \mathbb{Z}_+^{s_j - 1},\ \sum_{k=1}^{s_j - 1} z_j^k \leq n_j,\ \forall j \right\}$$

is equivalent to $M^\dagger$. Transition from $x \to y$ in C as follows:

- Choose $j \in [\sigma - m]$ and $\kappa \in \{-1, 1\}$, each uniformly at random
- $y = \begin{cases} x + \kappa e_j & \text{if } x + \kappa e_j \in C \\ x & \text{otherwise} \end{cases}$.

The stationary distribution of this random walk is uniform. We lower bound the Sobolev constant, $\rho(M^{\dagger})$, which, using the above steps, lower bounds $\rho(M)$ and hence upper bounds the mixing time of our algorithm.

Let $g : C \to \mathbb{R}$ be an arbitrary function. To lower bound $\rho(M^{\dagger})$, we will lower bound $\mathcal{E}(g, g)$ and upper bound $\mathcal{L}(g)$. The ratio of these two bounds in turn lower bounds the ratio $\mathcal{E}(g, g)/\mathcal{L}(g)$; since g was chosen arbitrarily this also lower bounds the Sobolev constant. We will use a theorem due to [21] which applies to an extension of a function $g : C \to \mathbb{R}$ to a function defined over the convex hull of C; here we define this extension.

Let K be the convex hull of C. Given $g : C \to \mathbb{R}$, we follow the procedure of [21, 55] to extend g to a function $g_\varepsilon : K \to \mathbb{R}$. For $x \in C$, let $C(x)$ and $C(x, \varepsilon)$ be the $\sigma - m$ dimensional cubes of center x and sides 1 and $1 - 2\varepsilon$ respectively. For sufficiently small $\varepsilon > 0$ and $z \in C(x)$, define $g_\varepsilon : K \to \mathbb{R}$ by:

$$g_\varepsilon(z) := \begin{cases} g(x) & \text{if } z \in C(x, \varepsilon) \\ \frac{(1+\eta(z))g(x)+(1-\eta(z))g(y)}{2} & \text{otherwise} \end{cases}$$

where $y \in C$ is a point such that $D := C(x) \cap C(y)$ is the closest face of $C(x)$ to z (if more than one y satisfy this condition, one such point may be chosen arbitrarily), and $\eta := \frac{\operatorname{dist}(z,D)}{\varepsilon} \in [0, 1)$. The dist function represents standard Euclidean distance in $\mathbb{R}^{\sigma - m}$.

Define

$$I_\varepsilon := \int_K \left|\nabla g_\varepsilon(z)\right|^2 dz \tag{B.14}$$

$$J_\varepsilon := \int_K g_\varepsilon(z)^2 \log \frac{g_\varepsilon(z)^2 \operatorname{vol}(K)}{\int_K g_\varepsilon(y)^2 dy} dz. \tag{B.15}$$

Applying Theorem 2 of [21] for $K \in \mathbb{R}^{\sigma - m}$ with diameter $\sqrt{\sum_{i=1}^m n_i^2}$, if $m + \sum_{i=1}^m n_i^2 \geq \sigma$,

$$\frac{\varepsilon I_\varepsilon}{J_\varepsilon} \geq \frac{1}{A \sum_{i=1}^m n_i^2}. \tag{B.16}$$

We lower bound $\mathcal{E}(g, g)$ in terms of $\varepsilon I_\varepsilon$ and then upper bound $\mathcal{L}(g)$ in terms of J_ε to obtain a lower bound on their ratio with Equation (B.16). The desired lower bound on the Sobolev constant follows.

Using similar techniques to [55], we lower bound $\mathcal{E}(g,g)$ in terms of $\varepsilon I_\varepsilon$ as

$$I_\varepsilon \leq \frac{|C|(\sigma - m)}{\varepsilon}\mathcal{E}(g,g) + O(1).$$

Then, $\varepsilon I_\varepsilon \leq_{\varepsilon\to 0} |C|(\sigma - m)\mathcal{E}(g,g)$, and hence

$$\mathcal{E}(g,g) \underset{\varepsilon\to 0}{\geq} \frac{\varepsilon I_\varepsilon}{|C|(\sigma - m)}. \tag{B.17}$$

Again, using similar techniques as [55], we bound J_ε as

$$\frac{J_\varepsilon}{\mathrm{vol}(K)} \geq \frac{|C|}{2^{2(\sigma-m)}\,\mathrm{vol}(K)(\sigma - m)!^2}\mathcal{L}(f).$$

Then

$$\mathcal{L}(g) \underset{\varepsilon\to 0}{\leq} \frac{2^{2(\sigma-m)}(\sigma - m)!^2}{|C|}J_\varepsilon. \tag{B.18}$$

Step 4, Combining inequalities: Using inequalities (B.16), (B.17), and (B.18),

$$\frac{\mathcal{E}(f,f)}{\mathcal{L}(f)} \geq \frac{1}{2^{2(\sigma-m)}A(\sigma - m)(\sigma - m)!^2\sum_{i=1}^m n_i^2}, \tag{B.19}$$

$\forall f : C \to \mathbb{R}$. Therefore,

$$\begin{aligned}\rho(M^\dagger) &= \min_{f:C\to\mathbb{R}} \frac{\mathcal{E}(f,f)}{\mathcal{L}(f)} \\ &\geq \frac{1}{2^{2(\sigma-m)}A(\sigma - m)(\sigma - m)!^2\sum_{i=1}^m n_i^2}\end{aligned} \tag{B.20}$$

Combining equations (B.11), (B.13), and (B.20)

$$\rho(M) \geq \frac{e^{-3\beta}}{2^{2ms}c_1 m^2 (m(s-1))!^2 n^2}$$

where c_1 is a constant depending only on s.

From here, Lemma 2 follows by applying Equation (A.11) in a similar manner as the proof of Equation (23) in [55]. The main difference is that the size of the state space is bounded as $|X| \leq \prod_{i=1}^m (n_i + 1)^{s_i+1}$ due to the potential for multiple populations. □

Combining Lemmas 1 and 2 results in a bound on the time it takes for the expected potential to be within ε of the maximum, provided β is sufficiently large. The lemmas and method of proof for Theorem 1 follow the structure of the supporting lemmas and proof for Theorem 3 in [55]. The main differences have arisen due to the facts that i) our analysis considers the multi-population case, so the size of our state space cannot be reduced as significantly as in the single population case of [55], and ii) update rates in our algorithm depend on behavior within each agent's own population, instead of on global behavior.

Proof of Theorem 1:

From Lemma 1, if condition (i) of Theorem 1 is satisfied and β is sufficiently large as in (2.8), then $\mathbb{E}_\pi[\phi(x)] \geq \max_{x\in X}\phi(x) - \varepsilon/2$. From Lemma 2, if condition (ii) of Theorem 1 is satisfied, and t is sufficiently large as in (2.10), then $\|\mu(t) - \pi\|_{TV} \leq \varepsilon/2$. Then

$$\begin{aligned}\mathbb{E}[\phi(a(t)|_X)] &= \mathbb{E}_{\mu(t)}[\phi(x)]\\ &\geq \mathbb{E}_\pi[\phi(x)] - \|\mu(t)-\pi\|_{TV}\cdot\max_{x\in X}\phi(x)\\ &\overset{(a)}{\geq} \max_{x\in X}\phi(x) - \varepsilon\end{aligned}$$

where (a) follows from (B.4), (B.7), and the fact that $\phi(x) \in [0,1]$. □

B.1.3 Time-Varying Semi-Anonymous Potential Games

To analyze the dynamics associated with modified log-linear learning, we must analyze the behavior of the time-varying Markov chain, M_t, which corresponds to the time varying game $G^t = (N^t, \{A_i^t\}, \{U_i^t\}\}$ for any $t \in \mathbb{R}^+$. Let $n(t) := |N^t|$ and $n_j^t := |N_j^t|$. The stationary distribution corresponding to M_t will be denoted by $\pi(t)$. Here, the state space varies with time; denote the aggregate state space corresponding to G^t by X^t, and define $\mathcal{X} := \cup_{t\in\mathbb{R}^+} X^t$.

A few additional definitions and notation will be useful in proving Theorem 2. We begin by identifying the times at which changes in the state space X^t may occur, i.e., a player becomes active or inactive. As in [55], consider the sequence of times $t_0 < t_1 < \cdots$ where $N_i(t) = N_i(t')$ for all $t, t' \in [t_\ell, t_{\ell+1})$, $i \in \{1, 2, \ldots, m\}$ and

$$\Lambda \leq |t_{\ell+1} - t_\ell| \leq 2\Lambda \tag{B.21}$$

for all $\ell = 0, 1, 2, \ldots$. The times in this sequence represent times at which a player may either become active or inactive, with additional times (when no change occurs) inserted if necessary to satisfy the upper bound of (B.21). For each t_ℓ, there are three cases:

(i) A player joins population N_j at action $\bar{a}_j^i$.

(ii) A player exits population N_j from action $\bar{a}_j^i$.

(iii) No change.

For cases (i) and (ii), the state space X^t changes when a player enters or exits a population. To assess the way this changes the distance between distributions $\pi(t)$ and $\mu(t)$, we project distributions from the old to the new state space using the projection operator,

$$\cdot|_{X^t} : X^{t^-} \to X^{t^+},$$

which is identical to the operator in [55]. Here X^{t^-} is the state space immediately before the change, and X^{t^+} is the state space immediately after.

Let $e_j^i \in \mathbb{R}^{s_j}$ be the ith standard basis vector of length s_j for $i \in \{1, 2, \ldots, s_i\}$, let $n = n(t^-)$ be the number of players at time t before the change occurs, and let $x = (x_j, x_{-j}) = (x_1, x_2, \ldots, x_m) \in X^{t^-}$.

Case (i): A player joins population N_j at action $\bar{a}_j^i$.

$$x|_{X^t} = \left(\frac{n(t^-)x_j + e_j^i}{n(t^-) + 1}, \frac{n(t^-)x_{-j}}{n(t^-) + 1} \right) \tag{B.22}$$

Case (ii): A player exits population N_j from action $\bar{a}_j^i$.

$$x|_{X^t} = \left(\frac{n(t^-)x_j + e_j^i}{n(t^-) - 1}, \frac{n(t^-)x_{-j}}{n(t^-) - 1} \right) \tag{B.23}$$

Case (iii): No change.

$$x|_{X^t} = x. \tag{B.24}$$

We project a distribution $\mu(t^-) \in \Delta X^{t^-}$ to a distribution $\mu(t) \in \Delta X^t$ by assigning the mass of state $x \in X^{t^-}$ in $\mu(t^-)$ to state $x|_{X^t}$ in the projected distribution, i.e.,

$$\mu_{x|_{X^t}}(t) = \mu_x(t^-), \ \forall x \in X^{t^-}, \tag{B.25}$$

and $\mu_x(t) = 0$ if there is no state in X^{t^-} which projects to $x \in X^t$. Here, $\mu(t)$ is the distribution immediately after the change, $\mu(t^-)$ is the distribution immediately prior, and $\mu_x(t)$ denotes the mass on state x in the distribution $\mu(t)$. Using (B.25), we extend the projection operator to distributions as

$$\mu(t) = \mu(t^-)|_{X^t}. \tag{B.26}$$

For notational simplicity, define

$$\hat{\pi}(t_\ell) := \pi(t_\ell)|_{t_{\ell+1}} \tag{B.27}$$

as in [55]. Note that, in general, $\hat{\pi}(t_\ell) \neq \pi(t_{\ell+1})$, i.e., the projected stationary distribution is not the stationary distribution of the new Markov chain.

Notation summary for time-varying semi-anonymous potential games: Let $\mathcal{G} = \{G^t\}_{t\in\mathbb{R}^+}$, where G^t is a semi-anonymous potential game for all $t \in \mathbb{R}^+$. The following summarizes the notation corresponding to the time-varying game $\mathcal{G}$.

- X^t - aggregate state space corresponding to game G^t.
- $\mathcal{X} := \cup_{t\in\mathbb{R}^+} X^t$ - union of aggregate state spaces corresponding to games G^t for all $t \in \mathbb{R}^+$.
- $n(t) := |N^t|$ - number of active players at time t
- $n_j(t) := |N_j^t|$ - size of the jth population at time t
- $\pi(t)$ - stationary distribution corresponding to the Markov chain M_t for $t \in \mathbb{R}^+$.
- Λ - bound on the length of time between possible changes in the state space, satisfies $\Lambda \leq |t_{\ell+1} - t_\ell| \leq 2\Lambda$, for all times $t_\ell, t_{\ell+1}$, at which changes in the state space may occur.
- $\cdot|_{X^t} : X^{t^-} \to X^{t^+}$ - projection operator which projects distributions from state space X^{t^-} to X^{t^+}, where t^- denotes the time immediately prior to a possible change in the state space, and t^+ denotes the time immediately after.
- $\hat{\pi}(t_\ell) := \pi(t_\ell)|_{t_{\ell+1}}$ - the stationary distribution corresponding to M_{t_ℓ} projected to $X^{t_{\ell+1}}$

B.1.4 Proof of Theorem 2

To prove Theorem 2, we analyze the limiting behavior of the time-varying Markov chain, M_t, which governs the modified log-linear learning process. We use Lemma 1 from Appendix B.1.2 and Lemma 3 stated below. Lemma 3 is analogous to Lemma 2 for stationary semi-anonymous potential games, and establishes conditions under which the distribution $\mu(t)$ is within $\varepsilon\,/\,2$ of the stationary distribution.

Lemma 3. *For the trajectory of semi-anonymous potential games, $\mathcal{G} = \{G^t\}_{t \geq \mathbb{R}^+}$, if Conditions (i) - (iv) of Theorem 2 are met, then*

$$\|\mu(t) - \pi(t)\|_{TV} \leq \varepsilon\,/\,2 \tag{B.28}$$

for all t sufficiently large as in (2.16).

Proof:

To prove Lemma 3, we begin by bounding the change in entropy distance between $\mu(t)$ and $\pi(t)$ when a player becomes active or inactive. We claim that, if $n(t_\ell)$ satisfies (2.11), then

$$D\left(\mu(t_{\ell+1}) : \pi(t_{\ell+1})\right) \leq \left(1 - \frac{A_1}{n(t_\ell)}\right) D\left(\mu(t_\ell) : \pi(t_\ell)\right) + \frac{A_2}{n(t_\ell)}$$

where

$$A_1 := \frac{2e^{-3\beta}\Lambda}{c_0}, \quad A_2 := 6\beta\lambda + e^{\beta}c(s-1)$$

The majority of details needed to prove equation (**??**) follow closely to the proof of Lemma 7 in [55] and are omitted for brevity. However differences arise in establishing the fact that, for any $\ell > 0$, if $n_i(t_\ell) \geq n(t_\ell)\,/\,k$ for some $k > 0$, then

$$\sum_{x \in X^{t_{\ell+1}}} \mu_x(t_{\ell+1}) \log \frac{\hat{\pi}_x(t_\ell)}{\pi_x(t_{\ell+1})} \leq \frac{6\beta\lambda + e^{\beta}k(s_j - 1)}{n(t_\ell)}. \tag{B.29}$$

The primary difference between this portion of the proof and the proof of Lemma 8 in [55] is that we account for the fact that players may enter or exit any given population. There are three cases:

(i) No change in the number of players.

(ii) Player joins population N_j with action $\bar{a}^i_j$.

(iii) Player exits population N_j from action $\bar{a}^i_j$.

Case (i) is trivial, as $\hat{\pi}_x(t_\ell) = \pi_x(t_{\ell+1})$ for all $x \in \mathcal{X}$ when there is no change in the number of players, so the left hand side of (B.29) is zero. For Case (ii), let

$$n = n(t_{\ell+1}) = n(t_\ell) + 1 = n(t^-_{\ell+1}) + 1$$

and let

$$n_j = n_j(t_{\ell+1}) = n_j(t_\ell) + 1 = n_j(t^-_{\ell+1}) + 1.$$

We will use the following inequality:

$$\sum_{x \in X^{t_\ell+1}} \mu_x(t_{\ell+1}) \log \frac{\hat{\pi}_x(t_\ell)}{\pi_x(t_{\ell+1})} \leq \max_{x \in X^{t_\ell+1}: x^i_j > 0} \log \frac{\hat{\pi}_x(t_\ell)}{\pi_x(t_{\ell+1})}.$$

This is because, if the new player joins population j at action $\bar{a}^i_j$, then $\mu_x(t_{\ell+1}) = 0$ for any state x with $x^i_j = 0$. For $x \in X^{t_\ell+1}$,

$$\pi_x(t_{\ell+1}) = \frac{\exp(\beta\phi(x))}{C_1}$$
$$\hat{\pi}_x(t_\ell) = \begin{cases} \frac{\exp\left(\beta\phi\left(\frac{nx - e^i_j}{n-1}\right)\right)}{C_2} & \text{if } x^i_j > 0 \\ 0 & \text{otherwise} \end{cases}$$

where

$$C_1 = \sum_{x \in X^{t_\ell+1}} \exp(\beta\phi(x))$$
$$C_2 = \sum_{x \in X^{t_\ell+1}: x^i_j > 0} \exp\left(\beta\phi\left(\frac{nx - e^i_j}{n-1}\right)\right).$$

Then,

$$\sum_{x \in X^{t_\ell+1}} \mu_x(t_{\ell+1}) \log \frac{\hat{\pi}_x(t_\ell)}{\pi_x(t_{\ell+1})} \leq \max_{x \in X^{t_\ell+1}: x^i_j > 0} \log \frac{\hat{\pi}_x(t_\ell)}{\pi_x(t_{\ell+1})}$$

$$
\begin{aligned}
&= \log \frac{C_1}{C_2} + \max_{x \in X^{t_\ell + 1}: x_j^i > 0} \beta \left(\phi \left(\frac{nx - e_j^i}{n-1} \right) - \phi(x) \right) \\
&\leq \log \frac{C_1}{C_2} + \frac{2\beta\lambda}{n-1}. \qquad \text{(B.30)}
\end{aligned}
$$

The last inequality follows from the λ-Lipschitz property of ϕ. We can bound the C_1 / C_2 in a similar fashion as the proof of Lemma 8 in [55] to get

$$
\frac{C_1}{C_2} \leq 1 + \frac{4\beta\lambda + e^\beta k(s-1)}{n-1}. \qquad \text{(B.31)}
$$

The primary difference is that we must make use of the fact that there exists a constant $k > 0$ such that $n_j \geq n/k$ for all $j \in \{1, 2 \ldots, m\}$ to achieve this upper bound in terms of n.

Combining (B.30) and (B.31),

$$
\begin{aligned}
\sum_{x \in X^{t_\ell + 1}} \mu_x(t_{\ell+1}) \log \frac{\hat{\pi}_x(t_\ell)}{\pi_x(t_{\ell+1})} &\leq \log \frac{C_1}{C_2} + \frac{2\beta\lambda}{n-1} \\
&\leq \log \left(1 + \frac{4\beta\lambda + e^\beta k(s-1)}{n-1} \right) + \frac{2\beta\lambda}{n-1} \\
&\overset{(a)}{\leq} \frac{6\beta\lambda + e^\beta k(s-1)}{n-1} \\
&= \frac{6\beta\lambda + e^\beta k(s-1)}{n(t_\ell)} \qquad \text{(B.32)}
\end{aligned}
$$

(a) is from the fact that $\log(1 + x) \leq x$ for $x \geq 0$.

Case (iii) follows a similar argument as case (ii) to show that

$$
\sum_{x \in X^{t_\ell + 1}} \mu_x(t_{\ell+1}) \log \frac{\hat{\pi}_x(t_\ell)}{\pi_x(t_{\ell+1})} \leq \frac{6\beta\lambda}{n(t_{\ell+1})}.
$$

□

The remainder of the proof of Lemma 3 in a similar fashion as intermediate steps in the proof of Theorem 4 in [55]. The primary difference is that the size of the initial state space is instead bounded as $X^{t_0} \leq \prod_{i=1}^m (n_i(t_0) + 1)^{s_i - 1}$. □

Proof of Theorem 2:

Using Lemmas 1 and 3, the proof of Theorem 2 follows in exactly the same manner as the proof of Theorem 1. □

B.2 Learning Efficient Correlated Equilibria: Background and Proofs

Here, we provide the proof for Theorem 3. The formulation of the decision making process defined in Section 3.2 ensures that the evolution of the agents' states over the periods $\{0, 1, 2, \dots\}$ can be represented as a finite ergodic Markov chain over the state space

$$X = X_1 \times \cdots \times X_n \tag{B.33}$$

where $X_i = S_i \times \{C, D\}$ denotes the set of possible states of agent i. Let P^ε denote this Markov chain for some $\varepsilon > 0$, and $\delta = \varepsilon$. Proving Theorem 3 requires characterizing the stationary distribution of the family of Markov chains $\{P^\varepsilon\}_{\varepsilon>0}$ for all sufficiently small ε. We employ the theory of resistance trees for regular perturbed processes, introduced in [61], to accomplish this task. We begin by reviewing this theory and then proceed with the proof of Theorem 3.

We begin by restating the main results associated with Theorem 3 (setting $\delta = \varepsilon$) using the terminology defined in the previous section.

- If $q(S) \cap \text{CCE} \neq \emptyset$, then a state $x = \{x_i = [s_i, m_i]\}_{i \in N}$ is stochastically stable if and only if (i) $m_i = C$ for all $i \in N$ and (ii) the strategy profile $s = (s_1, \dots, s_n)$ constitutes an efficient coarse correlated equilibrium, i.e.,

$$q(s) \in \underset{q \in q(S) \cap \text{CCE}}{\arg\max} \sum_{i \in N} \sum_{a \in \mathcal{A}} U_i(a) q^a. \tag{B.34}$$

- If $q(S) \cap \text{CCE} = \emptyset$, then a state $x = \{x_i = [s_i, m_i]\}_{i \in N}$ is stochastically stable if and only if (i) $m_i = C$ for all $i \in N$ and (ii) the strategy profile $s = (s_1, \dots, s_n)$ constitutes an efficient action profile, i.e.,

$$q(s) \in \underset{q \in q(S)}{\arg\max} \sum_{i \in N} \sum_{a \in \mathcal{A}} U_i(a) q^a. \tag{B.35}$$

For convenience, and with an abuse of notation, define

$$U_i(s) := \sum_{a \in \mathcal{A}} U_i(a) q^a(s) \tag{B.36}$$

to be agent i's expected utility with respect to distribution $q(s)$, where $s \in S$.

The proof of Theorem 3 will consist of the following steps:

(i) Define the unperturbed process, P^0.

(ii) Determine the recurrent classes of process P^0.

(iii) Establish transition probabilities of process P^ε.

(iv) Determine the stochastically stable states of P^ε using Theorem 10.

Part 1: Defining the unperturbed process

The unperturbed process P^0 is effectively the process identified in Section 3.2 where $\varepsilon = 0$. Rather than dictate the entire process as done previously, here we highlight the main attributes of the unperturbed process that may not be obvious upon initial inspection.

- If agent i is content, i.e., $x_i = [s_i^b, C]$, the trial action is $s_i^t = s_i^b$ with probability 1. Otherwise, if agent i is discontent, the trial action is selected according to (3.10).

- The baseline utility u_i^b in (3.8) associated with joint baseline strategy s^b is now of the form

  $$u_i^b = U_i(s^b). \tag{B.37}$$

  This results from invoking the law of large numbers since $\bar{p} = \lceil 1/\varepsilon^{nc+1} \rceil$. The trial utility u_i^t and acceptance utility u_i^a are also of the same form.

- A content player will only become discontent if $u_i^a < u_i^b$ where associated payoffs are computed according to (B.37).

Part 2: Recurrent classes of the unperturbed process

The second part of the proof analyzes the recurrent classes of the unperturbed process P^0 defined above. The following lemma identifies the recurrent classes of P^0.

Lemma 4. *A state $x = (x_1, x_2, \ldots, x_n) \in X$ belongs to a recurrent class of the unperturbed process P^0 if and only if the state x fits into one of following two forms:*

- ***Form #1:*** *The state for each agent $i \in N$ is of the form $x_i = \left[s_i^b, C\right]$ where $s_i^b \in S_i$. Each state of this form comprises a distinct recurrent classes. We represent the set of states of this form by C^0.*

- ***Form #2:*** *The state for each agent $i \in N$ is of the form $x_i = \left[s_i^b, D\right]$ where $s_i^b \in S_i$. All states of this form comprise a single recurrent class, represented by D^0.*

Proof. We begin by showing that any state $x \in C^0$ is a recurrent class of the unperturbed process. According to P^0, if the system reaches state x, then it remains at x with certainty for all future time. Hence, each $x \in C^0$ is a recurrent class of P^0. Next, we show that D^0 constitutes a single recurrent class. Consider any two states $x, y \in D^0$. According to the unperturbed process, P^0, the probability of transitioning from x to y is strictly positive $\left(\geq \prod_{i \in N} 1/|S_i|\right)$; hence, the resistance of the transition $x \rightarrow y$ is 0. Further note that the probability of transitioning to any state not in D^0 is zero. Hence, D^0 forms a single recurrent class of P^0.

The last part of the proof involves proving that any state $x = \{[s_i^b, m_i]\}_{i \in N} \notin C^0 \cup D^0$ is not recurrent in P^0. Since $x \notin C^0 \cup D^0$, it consists of both content and discontent players. Denote the set of discontent players by $J = \{i \in N : m_i = D\} \neq \emptyset$. We will show that the discontent players J will play a sequence of strategies with positive probability that drives at least one content player to become discontent. Repeating this argument at most n times shows that any state x of the above form will eventually transition to the all discontent state, proving that x is not recurrent.

To that end, let $x(1) = x$ be the state at the beginning of the 1-st period. According to the unperturbed process P^0, each discontent player randomly selects a strategy $s_i \in S_i$ which becomes part of the player's state at the ensuing stage. Suppose each discontent agent selects a trial strategy $s_i = (a_i^1, \ldots, a_i^w) \in \mathcal{A}_i^w \subset S_i$ during the 1-st period, i.e., the discontent players select strategies of the finest granularization. Note that each agent selects a strategy with probability $\geq 1 \,/\, |S_i|$. Here, the trial payoff for each player $i \in N$ associated with the joint strategies $s = (\{s_i^b\}_{i \notin J}, \{s_i\}_{i \in J})$ is

$$\begin{aligned} u_i^t(s) &= \int_0^1 U_i(s(z))dz && \text{(B.38)} \\ &= \frac{1}{w}U_i(a) + \int_w^1 U_i(s'(z))dz, && \text{(B.39)} \end{aligned}$$

for some $a \in \mathcal{A}$ as $s_i(z) = s_i(z')$ for any $z, z' \in [0, 1/w]$ for any agent $i \in N$. If $u_i^t < u_i^b$ for any any agent $i \notin J$, agent i becomes discontent in the next stage and we are done.

For the remainder of the proof suppose $u_i^t(s) \geq u_i^b(s^b)$ for all agents $i \notin J$. This implies all agents $N \setminus J$ will be content at the beginning of the second stage. By interdependence, there exists a collective action $\tilde{a}_J \in \prod_{j \in J} \mathcal{A}_j$ and an agent $i \notin J$ such that $U_i(a) \neq U_i(\tilde{a}_J, a_{N \setminus J})$. Suppose each discontent agent selects a trial strategy $s_i' = (\tilde{a}_i^1, a_i^2, \ldots, a_i^w) \in \mathcal{A}_i^w \subset S_i$ during the second period, i.e., only the first component of the strategy changed. The trial payoff for each player $i \in N$ associated with the joint strategies $s' = (\{s_i^b\}_{i \notin J}, \{s_i'\}_{i \in J})$ is

$$\begin{aligned} u_i^t(s') &= \int_0^1 U_i(s'(z))dz \\ &= \frac{1}{w}U_i(\tilde{a}_J, a_{N \setminus J}) + \int_w^1 U_i(s'(z))dz \\ &\neq u_i^t(s) \end{aligned}$$

If $u_i^t(s') < u_i^t(s)$, agent i will become discontent at the ensuing stage and we are done. Otherwise, agent i will stay content at the ensuing stage. However, if each discontent agent selects a trial strategy $s_i'' = (a_i^1, a_i^2, \ldots, a_i^w) \in \mathcal{A}_i^w \subset S_i$ during the third period, we know $u_i^t(s'') < u_i^t(s')$, where $s'' = (\{s_i^b\}_{i \notin J}, \{s_i''\}_{i \in J})$. Hence, agent i will become discontent at the beginning of period 4. This argument can be repeated at most n times, completing the proof. □

Part 3: Transition probabilities of process P^ε

Here, we establish the transition probability $P^\varepsilon_{x \to x^+}$ for a pair of arbitrary states, $x, x^+ \in X$.

Let $x_i = [s_i, m_i]$, $x_i^+ = [s_i^+, m_i^+]$ for $i \in N$, $s = (s_1, s_2, \dots, s_n)$, and $s^+ = (s_1^+, s_2^+, \dots, s_n^+)$. Then,

$$P^{\varepsilon}_{x \to x^+} = \sum_{\tilde{s}s^t \in S} \sum_{\tilde{s}^a \in S} \Big(\Pr[x^+ \;\big|\; s^t = \tilde{s}^t,\, s^a = \tilde{s}^a] \\ \times \Pr[s^a = \tilde{s}^a \;\big|\; s^t = \tilde{s}^t] \Pr[s^t = \tilde{s}^t] \Big). \tag{B.40}$$

Note that the strategy selections and state transitions are conditioned on state x; for notational brevity we do not explicitly write this dependence. Here, s^t and s^a represent the joint trial and acceptance strategies during the period before the transition to x^+.. The double summation in (B.40) is over all possible trial actions, $\tilde{s}^t \in S$, and acceptance strategies, $\tilde{s}^a \in S$. However, recall from (3.14) - (3.17) that, when transitioning from x to x^+, not all strategies can serve as intermediate trial and acceptance strategies. In particular, transitioning to state x^+ requires that $s^a = s^+$; hence if $\tilde{s}^a \neq s^+$, then $\Pr[x^+ \;\big|\; s^t = \tilde{s}^t,\, s^a = \tilde{s}^a] = 0$, so we can rewrite (B.40) as:

$$P^{\varepsilon}_{x \to x^+} = \sum_{\tilde{s}^t \in S} \Big(\Pr[x^+ \;\big|\; s^t = \tilde{s}^t,\, s^a = s^+] \\ \times \Pr[s^a = s^+ \;\big|\; s^t = \tilde{s}^t] \Pr[s^t = \tilde{s}^t] \Big) \tag{B.41}$$

There are three cases for the transition probabilities in (B.41). Before proceeding, we make the following observations. The last term in (B.41), $\Pr[s^t = \tilde{s}^t]$, is defined in Section 3.2; we will not repeat the definition here. For the first two terms, agents' state transition and strategy selection probabilities are independent when conditioned state x and on the joint trial and acceptance strategy selections. Hence, we can write the first term as:

$$\Pr[x^+ \;\big|\; s^t = \tilde{s}^t, s^a = s^+] = \prod_{i \in N} \Pr[x_i^+ \;\big|\; s^t = \tilde{s}^t, s^a = s^+] \tag{B.42}$$

and the second term as:

$$\Pr[s^a = s^+ \;\big|\; s^t = \tilde{s}^t] = \prod_{i \in N} \Pr[s_i^a = s_i^+ \;\big|\; s^t = \tilde{s}^t]. \tag{B.43}$$

The following three cases specify individual agents' probability of choosing the acceptance strategy s_i^a in (B.43) and transitioning to state x_i^+ in (B.42).

Case (i) agent i is content in state x, i.e., $m_i = C$, and did not experiment, $s_i^t = s_i$:

For (B.43), since $s_i^a \in \{s_i^t, s_i\}$ we know that

$$\Pr[s_i^a = s_i^+ \mid s^t = \tilde{s}^t] = \begin{cases} 1 & \text{if } s_i^+ = s_i \\ 0 & \text{otherwise} \end{cases}.$$

In (B.42), for any trial strategy $s^t = \tilde{s}^t$, the probability of transitioning to a state x_i^+ depends on realized average payoffs u_i^b and u_i^a. In particular, if $x_i^+ = [s_i^+, C]$, then we must have that $u_i^a \geq u_i^b - \varepsilon$, so

$$\begin{aligned} &\Pr\Big[x_i^+ = [s_i^+, C] \mid s^a = s^+, s^t = \tilde{s}^t\Big] \\ &= \int_0^1 \Pr[u_i^b = \eta] \int_{\eta-\varepsilon}^1 \Pr[u_i^a = \nu \mid s^t = \tilde{s}^t, s^a = s^+] d\nu d\eta. \end{aligned}$$

Then, the probability that $x_i^+ = [s_i^+, D]$ is

$$1 - \Pr\Big[x_i^+ = [s_i^+, C] \mid s^a = s^+, s^t = \tilde{s}^t\Big].$$

Case (ii) agent i is content and experimented, $s_i^t \neq s_i$:

For (B.43), agent i's acceptance strategy depends on its average baseline and trial payoffs, u_i^b and u_i^t. Recall, if $u_i^t \geq u_i^b + \varepsilon$, then $s_i^a = s_i$, i.e., agent i's acceptance strategy is simply its baseline strategy from state x. Otherwise $s_i^a = s_i^t$. Utilities u_i^b and u_i^t depend on joint strategies s and s^t and on the common random signals sent during the corresponding phases. Therefore,

$$\begin{aligned} &\Pr[s_i^a = s_i^+ \mid s^t = \tilde{s}^t \neq s] \\ &\quad = \int_0^1 \int_0^1 \Pr[s_i^a = s_i^+ \mid u_i^b = \eta, u_i^t = \nu, s_i^t = s_i] \\ &\qquad\qquad \times \Pr[u_i^b = \eta] \Pr[u_i^t = \nu \mid s^t = \tilde{s}^t] d\eta d\nu \end{aligned}$$

In (B.42), since agent i remains content and sticks with its acceptance strategy from the previous period,

$$\Pr[x_i^+ \mid s^a = s^+, s^t = \tilde{s}^t] = \begin{cases} 1 & \text{if } s_i^+ = s_i^a \\ 0 & \text{otherwise} \end{cases}.$$

Case (iii) agent i is discontent:

For (B.43),

$$\Pr[s_i^a = s_i^+ \mid s^t = \tilde{s}^t] = \begin{cases} 1 & \text{if } s_i^+ = s_i^t \\ 0 & \text{otherwise} \end{cases}.$$

In (B.42), agent i's probability of becoming content depends only on its received payoff during the acceptance phase; it becomes content with probability $\varepsilon^{1-u_i^a}$ and remains discontent with probability $1-\varepsilon^{1-u_i^a}$. Hence, if $x_i^+ = [s_i^+, C]$,

$$\begin{aligned}\Pr\Big[x_i^+ = [s_i^+, C] \mid s^a = s^+, s^t = \tilde{s}^t \Big] \\ = \int_0^1 \varepsilon^{1-\eta} \Pr[u_i^a = \eta \mid s^a = s^+, s^t = \tilde{s}^t] d\eta.\end{aligned}$$

Then,

$$\begin{aligned}\Pr\Big[x_i^+ = [s_i^+, D] \mid s^a = s^+, s^t = \tilde{s}^t \Big] \\ = 1 - \Pr\Big[x_i^+ = [s_i^+, C] \mid s^a = s^+, s^t = \tilde{s}^t \Big]\end{aligned}$$

Now that we have established transition probabilities for process P^ε, we may state the following lemma.

Lemma 5. *The process P^ε is a regular perturbation of P^0.*

It is straightforward to see that P^ε satisfies the first two conditions of Definition 3 with respect to P^0. The fact that transition probabilities satisfy the third condition, Equation (A.12), follows from the fact that the dominant terms in $P^\varepsilon_{x\to y}$ are polynomial in ε. This is immediately clear in all but the incorporation of realized utilities into the transition probabilities, as in (B.41). However, for any joint strategy, s, and associated average payoff u_i, since

$$\mathbb{E}[u_i] = \mathbb{E}\left[\frac{1}{\bar{p}} \sum_{\tau=\ell}^{\ell+\bar{p}-1} U_i(s(z(\tau))) \right] = U_i(s).$$

for any time period of length $\bar{p}$ in which joint strategy s is played throughout the entire period. Moreover, $\mathrm{Var}\big[U_i(s(z(\tau)))\big] \leq 1$. Therefore, we may use Chebyschev's inequality and the fact that $\bar{p} = \lceil 1 \,/\, \varepsilon^{nc+2} \rceil$ to see that

$$\Pr\Big[\big|u_i - U_i(s)\big| \geq \varepsilon\Big] \leq \frac{\mathrm{Var}\big[U_i(s(z(\tau)))\big]}{\bar{p}\varepsilon^2} \leq \varepsilon^{nc}. \tag{B.44}$$

Note that this applies for all average utilities, u_i^b, u_i^t, and u_i^a in the aforementioned state transition probabilities.

Part 3: Determining the stochastically stable states

We begin by defining

$$C^\star := \{x = \{[s_i, m_i]\}_{i\in N}$$
$$: \; q(s) \in \text{CCE and } m_i = C, \, \forall i \in N\} \subseteq C^0$$

Here, we show that, if $C^\star$ is nonempty, then a state x is stochastically stable if and only if $q(s)$ satisfies (B.34). The fact that $q(s)$ must satisfy (B.35) when $C^\star = \emptyset$ follows in a similar manner. To accomplish this task, we (1) establish resistances between recurrent classes, and (2) compute stochastic potentials of each recurrent class.

B.2.1 Resistances between recurrent classes

We summarize resistances between recurrent classes in the following claim.

Claim 1. Resistances between recurrent classes satisfy:

For $x \in C^0$ with corresponding joint strategy s,

$$r(D^0 \to x) = \sum_{i\in N}(1 - U_i(s)). \tag{B.45}$$

For a transition of the form $x \to y$, where $x \in C^\star$ and $y \in (C^0 \cup D^0) \setminus \{x\}$,

$$r(x \to y) \geq 2c. \tag{B.46}$$

For a transition of the form $x \rightarrow y$ where $x \in C^0 \setminus C^\star$ and $y \in (C^0 \cup D^0) \setminus \{x\}$,

$$r(x \rightarrow y) \geq c. \tag{B.47}$$

For every $x \in C^0 \setminus C^\star$, there exists a path $x = x^0 \rightarrow x^1 \rightarrow \cdots \rightarrow x^m \in C^\star \cup D^0$ with resistance

$$r(x^j \rightarrow x^{j+1}) = c,\ \forall j \in \{0, 1, \ldots, m-1\}. \tag{B.48}$$

These resistances are computed in a similar manner to the proof establishing resistances in [38]; however, care must be taken due to the fact that there is a small probability that average received utilities fall outside of the window $U_i(s) \pm \varepsilon$ during a phase in which joint strategy s is played. We illustrate this by proving (B.45) in detail; the proofs are omitted for other types of transitions for brevity.

Proof. Let $x \in D^0$, $x^+ \in C^0$ with $x_i = [s_i, D]$ and $x_i^+ = [s_i^+, C]$ for each $i \in N$. Again, for notational brevity, we drop the dependence on state x in the following probabilities. Note that all agents must select $s^t = s_i^+$ in order to transition to state $x_i = [s_i^+, C]$; otherwise the transition probability is 0. we have

$$\begin{aligned}
P^\varepsilon_{x \rightarrow x^+} &\overset{(a)}{=} \Pr[x^+ \mid s^a = s^+, s^t = s^+] \cdot \Pr[s^a = s^+ \mid s^t = s^+] \cdot \Pr[s^t = s^+] \\
&\overset{(b)}{=} \Pr[x^+ \mid s^a = s^+, s^t = s^+] \cdot \Pr[s^t = s^+] \\
&\overset{(c)}{=} \Pr[x^+ \mid s^a = s^+, s^t = s^+] \cdot \prod_{i \in N} 1 \,/\, |S_i| \\
&= \prod_{i \in N} \frac{1}{|S_i|} \cdot \Pr[x_i^+ \mid s^a = s^+, s^t = s^+]
\end{aligned}$$

where: (a) follows from the fact that $s_i^a = s_i^t$ since $m_i = D$ in state x for all $i \in N$, (b) $\Pr[s^a = s^+ \mid s^t = s^+] = 1$ since all agents are discontent and hence commit to their trial strategies during the acceptance period, and (c) $\Pr[s^t = s^+] = \prod_{i \in N} 1 \,/\, |S_i|$ since each discontent agent selects its trial strategy uniformly at random from S_i.

We now show that

$$0 < \lim_{\varepsilon \rightarrow 0^+} \frac{P^\varepsilon_{x \rightarrow x^+}}{\varepsilon^{\sum_{i \in N} 1 - U_i(s^+)}} < \infty \tag{B.49}$$

satisfying (A.12). For notational simplicity, we define

$$U_i^+ := U_i(s^+) + \varepsilon, \quad U_i^- := U_i(s^+) - \varepsilon. \tag{B.50}$$

We first lower bound $P^\varepsilon_{x\to x^+}$:

$$\begin{aligned}
P^\varepsilon_{x\to x^+} &= \prod_{i\in N} \frac{1}{|S_i|} \Pr[x_i^+ \mid s^a = s^+, s^t = s^+] \\
&= \prod_{i\in N} \frac{1}{|S_i|} \int_0^1 \Pr[u_i^a = \eta \mid s^a = s^+, s^t = s^+] \varepsilon^{1-\eta} d\eta \\
&\geq \prod_{i\in N} \frac{1}{|S_i|} \int_{U_i^-}^{U_i^+} \Pr[u_i^a = \eta \mid s^a = s^+, s^t = s^+] \varepsilon^{1-\eta} d\eta \\
&\overset{(a)}{\geq} \prod_{i\in N} \frac{\varepsilon^{1-U_i^-}}{|S_i|} \int_{U_i^-}^{U_i^+} \Pr[u_i^a = \eta \mid s^a = s^+, s^t = s^+] d\eta \\
&\overset{(b)}{\geq} \prod_{i\in N} \frac{\varepsilon^{1-U_i^-}}{|S_i|} (1 - \varepsilon^{nc}) \\
&= \frac{\varepsilon^{\sum_{i\in N} 1-U_i^-} + O(\varepsilon^{nc})}{\prod_{i\in N} |S_i|}
\end{aligned} \tag{B.51}$$

where (a) is from the fact that $\varepsilon^{1-\eta}$ is continuous and increasing in η for $\varepsilon \in (0,1)$, and (b) follows from (B.44). Continuing in a similar fashion, it is straightforward to show

$$P^\varepsilon_{x\to x^+} \leq \varepsilon^{\sum_{i\in N}(1-U_i^+)} + O(\varepsilon^{nc}). \tag{B.52}$$

Given (B.51) and (B.52), and the fact that U_i^+ and U_i^- satisfy (B.50), we have that $P^\varepsilon_{x\to x^+}$ satisfies (A.12) with resistance $\sum_{i\in N}(1 - U_i(s^+))$ as desired. □

B.2.2 Stochastic potentials

The following lemma specifies stochastic potentials of each recurrent class. Using resistances from Claim 1, the stochastic potentials follow from the same arguments as in [38]. The proof is repeated below for completeness.

Lemma 6. *Let $x \in C^0 \setminus C^\star$ with corresponding joint strategy s, and let $x^\star \in C^\star$ with corresponding* *joint strategy $s^\star$. The stochastic potentials of each recurrent class are:*

$$\begin{aligned}\gamma(D^0) &= c|C^0 \setminus C^\star| + 2c|C^\star|,\\ \gamma(x) &= \left(|C^0 \setminus C^\star| - 1\right)c + 2c|C^\star| + \sum_{i\in N}(1 - U_i(s)),\\ \gamma(x^\star) &= |C^0 \setminus C^\star|c + 2c\,(|C^\star| - 1) + \sum_{i\in N}(1 - U_i(s^\star)),\end{aligned}$$

Proof: In order to establish the stochastic potentials for each recurrent class, we will lower and upper bound them.

Lower bounding the stochastic potentials: To lower bound the stochastic potentials of each recurrent class, we determine the lowest possible resistance that a tree rooted at each of these classes may have.

1) Lower bounding $\gamma(D^0)$:

$$\gamma(D^0) \geq c|C^0 \setminus C^\star| + 2c|C^\star|$$

In a tree rooted at D^0, each state in C^0 must have an exiting edge. In order to exit a state in $C^0 \setminus C^\star$, only a single agent must experiment, contributing resistance c. To exit a state in $C^\star$, at least two agents must experiment, contributing resistance $2c$.

2) Lower bounding $\gamma(x)$, $x \in C^0 \setminus C^\star$:

$$\gamma(x) \geq \left(|C^0 \setminus C^\star| - 1\right)c + 2c|C^\star| + \sum_{i\in N}(1 - U_i(s))$$

Here, each state in $C^0 \setminus \{x\}$ must have an exiting edge, which contributes resistance $\left(|C^0 \setminus C^\star| - 1\right)c+$ $2c|C^\star|$. The recurrent class D^0 must also have an exiting edge, contributing at least resistance $\sum_{i\in N}(1 - U_i(s))$.

3) Lower bounding $\gamma(x^\star)$, $x^\star \in C^\star$:

$$\gamma(x^\star) \geq |C^0 \setminus C^\star|c + 2c\,(|C^\star| - 1) + \sum_{i\in N}(1 - U_i(s^\star))$$

Again, each state in $C^0 \setminus \{x^\star\}$ must have an exiting edge, which contributes resistance $\left(|C^0 \setminus C^\star| - 1\right)c+$ $2c|C^\star|$. The recurrent class D^0 must also have an exiting edge, contributing resistance at least $\sum_{i\in N}(1 - U_i(s^\star))$.

Upper bounding the stochastic potentials: In order to upper bound the stochastic potentials, we construct trees rooted at each recurrent class which have precisely the resistances established above.

1) Upper bounding $\gamma(D^0)$:

$$\gamma(D^0) \leq c|C^0 \setminus C^\star| + 2c|C^\star|$$

Begin with an empty graph with vertices X. For each state $x \in C^0 \setminus C^\star$, add a path ending in $C^\star \cup D^0$ so that each edge has resistance c. This is possible due to Claim 1. Now eliminate redundant edges; this contributes resistance at most $c|C^0 \setminus C^\star|$ since each state in $C^0 \setminus C^\star$ has exactly one outgoing edge. Finally, add an edge $x^\star \to D^0$ for each $x^\star \in C^0$; this contributes resistance $2c|C^\star|$.

2) Upper bounding $\gamma(x)$, $x \in C^0 \setminus C^\star$:

$$\gamma(x) \leq \left(|C^0 \setminus C^\star| - 1\right) c + 2c|C^\star| + \sum_{i \in N}(1 - U_i(s)),$$

This follows by a similar argument to the previous upper bound, except here we add an edge $D^0 \to x$ which contributes resistance $\sum_{i \in N}(1 - U_i(s))$.

3) Upper bounding $\gamma(x^\star)$, $x^\star \in C^\star$:

$$\gamma(x^\star) \leq |C^0 \setminus C^\star| c + 2c\left(|C^\star| - 1\right) + \sum_{i \in N}(1 - U_i(s^\star)),$$

This follows from an identical argument to the previous bound. □

We now use Lemma 6 to complete the proof of Theorem 3. For the first part, suppose $C^\star$ is nonempty, and let

$$x^\star \in \arg\max_{x \in C^\star} \sum U_i(s),$$

where joint strategy s corresponds to state x. Then,

$$\begin{aligned}
\gamma(x^\star) &= |C^0 \setminus C^\star| c + 2c\left(|C^\star| - 1\right) + \sum_{i \in N}(1 - U_i(s^*)) \\
&< |C^0 \setminus C^\star| c + 2c|C^\star| \quad (\text{since } c \geq n) \\
&= \gamma(D).
\end{aligned}$$

For $x \in C^0$,

$$\begin{aligned}\gamma(x^\star) &= |C^0 \setminus C^\star|c + 2c\left(|C^\star| - 1\right) + \sum_{i \in N}(1 - U_i(s^\star)) \\ &< |C^0 \setminus C^\star - 1|c + 2c\left(|C^\star|\right) + \sum_{i \in N}(1 - U_i(s)) \\ &= \gamma(x).\end{aligned}$$

For $x \in C^\star$ with

$$x \notin \arg\max_{x \in C^\star} \sum U_i(s),$$

$$\begin{aligned}\gamma(x^\star) &= |C^0 \setminus C^\star|c + 2c\left(|C^\star| - 1\right) + \sum_{i \in N}(1 - U_i(s^\star)) \\ &< |C^0 \setminus C^\star|c + 2c\left(|C^\star| - 1\right) + \sum_{i \in N}(1 - U_i(s) \\ &= \gamma(x).\end{aligned}$$

Applying Theorem 10, $x^\star$ is stochastically stable. Since all other states have strictly larger stochastic potential, **only** states $x^\star \in C^\star$ with $x^\star \in \arg\max_{x \in C^\star} \sum U_i(s)$ are stochastically stable. From state $x^\star$, if each agent plays according to its baseline strategy, then the probability that joint action $a \in \mathcal{A}$ is played at any given time is $\Pr(a = a') = q^{a'(s^\star)}$. This implies that a CCE which maximizes the sum of agents' payoffs is played with high probability as $\varepsilon \to 0$, after sufficient time has passed.

The second part of the theorem follows similarly by considering the case when $C^\star = \emptyset$. □

B.3 Understanding Adversarial Influence in Distributed Systems: Background and Proofs

We begin by reviewing the underlying Markov process for log-linear learning in the adversarial models, and then we provide the proofs corresponding to the theorems in Chapter 4.

B.3.1 Log-linear learning in the adversarial influence models

Log-linear learning dynamics define a family of aperiodic, irreducible Markov processes, $\{\tilde{P}_\beta\}_{\beta > 0}$, over state space $\mathcal{A} \times \mathcal{S}_k$ with transition probabilities parameterized by β [9]. Under our adversarial model, transition probabilities are

$$P_\beta(((a_i,a_{-i}),S)\to(a_i',a_{-i}),S')=\frac{1}{n}\Pr[a_i(t+1)=a_i'\,|\,a_{-i}(t)=a_{-i},S(t)=S] \tag{B.53}$$

for any $i\in N$, $a_i\in\{\vec{x},\vec{y}\}$, $(a_i,a_{-i})\in\mathcal{A}$ and $S,S'\in\mathcal{S}_k$. Here S transitions to S' according to the specified adversarial model. If a and $a'\in\mathcal{A}$ differ by more than one agent's action, then $P_\beta(a\to a')=0$.

For each model of adversarial behavior, it is straightforward to reduce $\tilde{P}_\beta$ to a Markov chain, P_β over state space $\mathcal{A}$. Since P_β is aperiodic and irreducible for any $\beta>0$, it has a unique stationary distribution, π_β, with $\pi_\beta P_\beta=\pi_\beta$.

As $\beta\to\infty$, the stationary distribution, π_β, associated with log-linear learning converges to a unique distribution, $\pi:=\lim_{\beta\to\infty}\pi_\beta$. If $\pi(a)=1$, then joint action a is **strictly stochastically stable** [17].[2]

As $\beta\to\infty$, transition probabilities $P_\beta(a\to a')$ of log-linear learning converge to the transition probabilities, $P(a\to a')$, of a best response process. Distribution π is one of possibly multiple stationary distributions of a best response process over game G.

B.3.2 Stability in the presence of a fixed, intelligent adversary

When a fixed, intelligent adversary influences set S, the corresponding influenced graphical coordination game is a potential game [47] with potential function

$$\Phi^S(a_i,a_{-i})=\frac{1}{2}\sum_{i\in N}\left(U_i(a_i,a_{-i})+2\cdot\mathbb{1}_{i\in S,a_i=y}\right). \tag{B.54}$$

This implies that the stationary distribution associated with log-linear learning influenced by a fixed adversary is

$$\pi(a)=\frac{\exp(\beta\cdot\Phi^S(a))}{\sum_{a'\in\mathcal{A}}\exp(\beta\cdot\Phi^S(a'))}, \tag{B.55}$$

for $a\in\mathcal{A}$ [9]. Hence, $a\in\mathcal{A}$ is strictly stochastically stable if and only if $\Phi^S(a)>\Phi^S(a')$ for all $a'\in\mathcal{A}$, $a'\neq a$.

[2] Note that this definition of strict stochastic stability is equivalent to the definition in the introduction.

Proof of Theorem 4: This proof adapts Proposition 2 in [63] to our adversarial model. Let $\mathcal{G} = (N, E)$ and suppose $S(t) = N$ for all $t \in \mathbb{N}$. Define $(\vec{y}_T, \vec{x}_{N\setminus T})$ to be the joint action $(a_1, \ldots, a_n)$ with $T = \{i : a_i = y\}$. It is straightforward to show that

$$\alpha > \frac{|T| - d(T, N \setminus T)}{d(T, N)}, \quad \forall T \subseteq N$$

if and only if

$$\begin{aligned} \Phi^N(\vec{x}) &= (1+\alpha)d(N,N) \\ &> (1+\alpha)d(N \setminus T, N \setminus T) + d(T,T) + |T| \\ &= \Phi^N(\vec{y}_T, \vec{x}_{N\setminus T}) \end{aligned} \tag{B.56}$$

for all $T \subseteq N$, $R \neq \emptyset$, implying the desired result. □

Proof of Theorem 7 part (a): Let $\mathcal{G} = (N, E)$ be a line influenced by an adversary with capability k. Joint action $\vec{x}$ is strictly stochastically stable for all $S \subseteq N$ with $|S| = k$ if and only if

$$\Phi^S(\vec{x}) > \Phi^S(\vec{y}_T, \vec{x}_{N\setminus T}) \iff (1+\alpha)d(N,N) > (1+\alpha)d(N \setminus T, N\setminus \quad T) + d(T,T) + |S \cap T|. \tag{B.57}$$

for all $S \subseteq N$ with $|S| = k$ and all $T \subseteq N$, $T \neq \emptyset$.

Define $t := |T|$, let p denote the number of components in the graph $\mathcal{G}$ restricted to T, and let ℓ denote the number of components in the graph restricted to $N \setminus T$. Since $T \neq \emptyset$, we have $p \geq 1$ and $\ell \in \{p-1, p, p+1\}$.

The case where $T = N$ implies

$$\Phi^S(\vec{x}) = (1+\alpha)(n-1) > n - 1 + k = \Phi^S(\vec{y}),$$

which holds if and only if $\alpha > k\,/(n-1)$.

If $T \subset N$, the graph restricted to $N \setminus T$ has at least one component, i.e., $\ell \geq 1$. Then,

$$\begin{aligned} \Phi^S(\vec{y}_T, \vec{x}_{N\setminus T}) &= (1+\alpha)(n-t-\ell) + t - p + |S \cap T| \\ &\leq (1+\alpha)(n-t-1) + t - 1 + \min\{k, t\} \end{aligned}$$

where the inequality is an equality when $T = [t]$ and $S = [k]$. Then,

$$\begin{aligned}\Phi^S(\vec{y}_T, \vec{x}_{N\setminus T}) &\leq (1+\alpha)(n-t-1) + t - 1 + \min\{k, t\} \\ &< (1+\alpha)(n-1) \\ &= \Phi^S(\vec{x})\end{aligned}$$

for all $T \subset N$ if and only if $\alpha > (k-1)\,/\,k$, as desired. □

Proof of Theorem 7 part (b): Suppose $\alpha < k\,/(n-1)$. Then

$$\Phi^S(\vec{y}) = n - 1 + k > (1+\alpha)(n-1) = \Phi^S(\vec{x})$$

for any $S \subseteq N$ with $|S| = k$. Then, to show that $\vec{y}$ is stochastically stable for influenced set S satisfying

$$|S \cap [i, i+t]| \leq \left\lceil \frac{kt}{n} \right\rceil,$$

it remains to show that $\Phi^S(\vec{y}) > \Phi^S(\vec{y}_T, \vec{x}_{N\setminus T})$ for any $T \subset N$ with $T \neq \emptyset$ and $T \neq N$. Suppose the graph restricted to set T has p components, where $p \geq 1$. Label these components as $T_1, T_2, \ldots, T_p$ and define $t := |T|$ and $t_i := |T_i|$ Let ℓ represent the number of components in the graph restricted to $N \setminus T$. Since $\mathcal{G}$ is the line graph, we have $\ell \in \{p-1, p, p+1\}$, and since $T \neq N$, $\ell \geq 1$.

For any $T \subset N$ with $T \neq N, T \neq \emptyset$, and $0 < t < n$,

$$\begin{aligned}&\Phi^S(\vec{y}_T, \vec{x}_{N\setminus T}) \\ &\quad = (1+\alpha)(n-t-\ell) + \sum_{j=1}^{p} (t_j - 1 + |S \cap T_j|) \\ &\quad < n - 1 + k \qquad \text{(B.58)} \\ &\quad = \Phi^S(\vec{y})\end{aligned}$$

where (B.58) is straightforward to verify. □

The proofs of parts (c) and (d) follow in a similar manner to parts (a) and (b), by using the potential function Φ^S for stochastic stability analysis.

B.3.3 Stability in the presence of a mobile, random adversary

The following lemma applies to any graphical coordination game in the presence of a mobile, random adversary with capability $k \leq n-1$. It states that a mobile random adversary decreases the resistance of transitions when an agent in N changes its action from x to y, but does not change the resistance of transitions in the opposite direction.

Lemma 7. *Suppose agents in N update their actions according to log-linear learning in the presence of a mobile, random adversary with capability k, where $1 \leq k \leq n-1$. Then the resistance of a transition where agent $i \in N$ changes its action from x to y is:*

$$r((x,a_{-i}) \to (y,a_{-i})) = \max\left\{U_i(x,a_{-i}) - U_i(y,a_{-i}) - 1, 0\right\} \tag{B.59}$$

and the resistance of a transition where agent $i \in N$ changes its action from y to x is:

$$r((y,a_{-i}) \to (x,a_{-i})) = \max\left\{U_i(y,a_{-i}) - U_i(x,a_{-i}), 0\right\}. \tag{B.60}$$

Here $U_i : \mathcal{A} \to \mathbb{R}$, defined in (4.1), *is the utility function for agent i in the uninfluenced game, G.*

Proof: In the presence of a mobile, random agent,

$$\begin{aligned} P_\beta\left((x,a_{-i}) \to (y,a_{-i})\right) = \frac{1}{n}\Bigg(&\frac{k}{n} \cdot \frac{\exp(\beta(U_i(y,a_{-i})+1))}{\exp(\beta(U_i(y,a_{-i})+1)) + \exp(\beta U_i(x,a_{-i}))} \\ &+ \frac{n-k}{n} \cdot \frac{\exp(\beta U_i(y,a_{-i}))}{\exp(\beta U_i(y,a_{-i})) + \exp(\beta U_i(x,a_{-i}))}\Bigg) \end{aligned}$$

Define $P_\varepsilon\left((x,a_{-i}) \to (y,a_{-i})\right)$ by substituting $\varepsilon = e^{-\beta}$ into the above equation. It is straightforward to see that

$$0 < \lim_{\varepsilon \to 0^+} \frac{P_\varepsilon\left((x,a_{-i}) \to (y,a_{-i})\right)}{\varepsilon^{U_i(x,a_{-i}) - U_i(y,a_{-i}) - 1}} < \infty,$$

implying

$$r((x,a_{-i}) \to (y,a_{-i})) = \max\left\{U_i(x,a_{-i}) - U_i(y,a_{-i}) - 1, 0\right\}.$$

Equation (B.60) may be similarly verified. □

Proof of Theorem 8: First we show that, for any $\alpha > 0$, $\vec{x}$ and $\vec{y}$ are the only two recurrent classes of the unperturbed process, P, for the line. Then we show that, for the perturbed process,

$R(\vec{x},\vec{y}) < R(\vec{y},\vec{x}) \iff \alpha > 1$ and $R(\vec{y},\vec{x}) < R(\vec{x},\vec{y}) \iff \alpha < 1$. That is, when $\alpha > 1$ and β is large, the lowest resistance path from $\vec{x}$ to $\vec{y}$ occurs with higher probability than the lowest resistance path from $\vec{y}$ to $\vec{x}$ in P_β, and vice versa when $\alpha < 1$. Combining this with Theorem **??** proves Theorem 8.

Recurrent classes of P for the line: Note that, $P(\vec{x},a) = 0$ for all $a \in \mathcal{A}, a \neq \vec{x}$, and $P(\vec{y},a) = 0$ for all $a \in \mathcal{A}, a \neq \vec{y}$, implying $\vec{x}$ and $\vec{y}$ are recurrent. To show that no other state is recurrent, we will show that, for any $a \in \mathcal{A} \setminus \{\vec{x},\vec{y}\}$, there is a sequence of positive probability transitions in P leading from a to $\vec{x}$.

Let $a \in \mathcal{A}$ with $a \neq \vec{x},\vec{y}$. Without loss of generality, choose $i, i+1$ such that $a_i = y$ and $a_{i+1} = x$. Denote $(a_i, a_{-i}) = a$, and note that:

$$P((y,a_{-i}) \to (x,a_{-i})) = \frac{1}{n} \cdot \frac{n-k}{n} > 0 \tag{B.61}$$

for any $k \leq n-1$ and $\alpha > 0$. Since (B.61) holds for any $a \neq \vec{x},\vec{y}$, we can construct a sequence of at most $n-1$ positive probability transitions leading to joint action $\vec{x}$. Therefore a cannot be recurrent in P.

Resistance between recurrent classes $\vec{x}$ and $\vec{y}$: We will show that for all $1 \leq k \leq n-1$,

$$R(\vec{y},\vec{x}) = 1, \quad \forall \alpha > 0, \tag{B.62}$$

$$R(\vec{x},\vec{y}) \geq \alpha, \quad \forall \alpha > 0, \tag{B.63}$$

$$\text{and} \quad R(\vec{x},\vec{y}) = \alpha, \quad \forall \alpha \leq 1. \tag{B.64}$$

For (B.62), we have $r(\vec{y},(x,y,\ldots,y)) = 1$, and $r(\vec{y},a) \geq 1$ for any $a \neq \vec{y}$, implying that $R(\vec{y},\vec{x}) \geq 1$. Then, since

$$r\left((\vec{x}_{[t]},\vec{y}_{[t+1,n]}),(\vec{x}_{[t+1]},\vec{y}_{[t+2,n]})\right) = 0,$$

for any $1 \leq t \leq n-1$, and

$$r\left((\vec{x}_{[n-1]},\vec{y}_{[n,n]}),\vec{x}\right) = 0,$$

the path $\vec{y} \to (x,y,\ldots,y) \to (x,x,y,\ldots,y) \to \cdots \to \vec{y}$ has resistance 1. Since we know $R(\vec{y},\vec{x}) \geq 1$, this implies that $R(\vec{y},\vec{x}) = 1$.

Now, for (B.63), since $r(\vec{x}, a) \geq \alpha$ for any $a \neq \vec{x}$, this implies $R(\vec{x}, \vec{y}) \geq \alpha$. In particular $r(\vec{x} \to (y, x, \ldots, x)) = \alpha$. When $\alpha < 1$,

$$r\left((\vec{y}_{[t]}, \vec{x}_{[t+1,n]}), (\vec{y}_{[t+1]}, \vec{x}_{[t+2,n]})\right) = 0$$

for any $1 \leq t \leq n-1$, and

$$r\left((\vec{y}_{[n-1]}, \vec{x}_{[n,n]}), \vec{y}\right) = 0,$$

implying that the path $\vec{x} \to (y, x, \ldots, x) \to (y, y, \ldots, x) \to \cdots \to \vec{y}$ has resistance α when $\alpha \leq 1$. Hence $R(\vec{x}, \vec{y}) = \alpha$.

Combining (B.62) - (B.64) with Theorem **??** establishes Theorem 8. □

B.3.4 Stability in the presence of an intelligent, mobile agent

Define P_β^μ to be the Markov process associated with log-linear learning in the presence of a mobile, intelligent adversary using policy μ.

Proof of Theorem 9 part (a): Let $G = (N, E)$ be the line, influenced by a mobile, intelligent adversary with capability $k = 1$. For any policy $\mu : \mathcal{A} \to \mathcal{S} = N$, if $\alpha \neq 1$, only $\vec{x}$ and $\vec{y}$ are recurrent in the unperturbed process, P^μ. This can be shown via an argument similar to the one used in the proof of Theorem 8.

Define $\mu^\star$ as in (4.11). We will show that, (1) in $P_\beta^{\mu^\star}$, $\vec{x}$ is stochastically stable if and only if $\alpha > 1$, and $\vec{y}$ is stochastically stable if and only if $\alpha < 1$, and (2) $\mu^\star$ is optimal, i.e., if $\alpha = 1$, $\vec{x}$ is stochastically stable for any $\mu \in M_1$, and if $\alpha > 1$, $\vec{x}$ is strictly stochastically stable for any $\mu \in M_1$.

For policy $\mu \in M_1$, let $r^\mu(a, a')$ denote the single transition resistance from a to $a' \in \mathcal{A}$ in P_β^μ, and let $R^\mu(a, a')$, denote the resistance of the lowest resistance path from a to $a' \in \mathcal{A}$.

For any $\mu \in M_1$, we have $r^\mu(\vec{x}, a) \geq \alpha$, $\forall a \in \mathcal{A}, a \neq \vec{x}$, and $r^\mu(\vec{y}, a) \geq 1, \quad \forall a \in \mathcal{A}, a \neq \vec{y}$. Therefore

$$R^\mu(\vec{x} \to \vec{y}) \geq \alpha, \text{ and } R^\mu(\vec{y}, \vec{x}) \geq 1. \tag{B.65}$$

If $\alpha < 1$, the path $\vec{x} \rightarrow (y, x, \ldots, x) \rightarrow (y, y, x, \ldots, x) \rightarrow \cdots \rightarrow \vec{y}$ in $P_\beta^{\mu^\star}$ has total resistance α. Equation (B.65) implies that $R^{\mu^\star}(\vec{x}, \vec{y}) = \alpha < 1 \leq R^{\mu^\star}(\vec{y}, \vec{x})$, so by Theorem **??**, $\vec{y}$ is strictly stochastically stable in $P^{\mu^\star}$.

If $\alpha = 1$, it is straightforward to show that both $\vec{x}$ and $\vec{y}$ are stochastically stable in $P_\beta^{\mu^\star}$. Moreover, for any $\mu \in \mathcal{M}$, either the resistance of path

$$\vec{y} \rightarrow (x, y, \ldots, y) \rightarrow (x, x, y, \ldots y) \rightarrow \cdots \rightarrow \vec{x}$$

or the resistance of path

$$\vec{y} \rightarrow (y, \ldots, y, x) \rightarrow (y \ldots, y, x, x) \rightarrow \cdots \rightarrow \vec{x}$$

is 1, and hence it is impossible to find a policy with $R^\mu(\vec{x}, \vec{y}) < R^\mu(\vec{y}, \vec{x})$.

If $\alpha > 1$, similar arguments show that $R^\mu(\vec{y}, \vec{x}) = 1$ for any $\mu \in M_k$. Combining this with (B.65) implies that $\vec{x}$ is stochastically stable for any P_β^μ, $\mu \in \mathcal{M}$. □

Proof of (b): Again let $G = (N, E)$ be the line, and suppose the adversary has capability k with $2 \leq k \leq n-1$. We will show that, for a policy $\mu^\star$ which satisfies Conditions 1 - 3 of Theorem 9, $\vec{x}$ is strictly stochastically stable in $P^{\mu^\star}$ if and only if $\alpha > \frac{n}{n-1}$, and $\vec{y}$ is strictly stochastically stable if and only if $\alpha < \frac{n}{n-1}$. Since this is the same bound on α when we have an adversary with capability n, from Theorem 7 part (a), this also proves that policy $\mu^\star$ is optimal, i.e., no other policy can stabilize a state $a \in \mathcal{A}$ with $a_i = \vec{y}$ for some $i \in N$ when $\alpha > \frac{n}{n-1}$.

First note that only $\vec{y}$ is recurrent in $P^{\mu^\star}$ when $\alpha \leq 1$, and hence $\vec{y}$ is strictly stochastically stable in $P_\beta^{\mu^\star}$.

Now assume $\alpha > 1$. Again, it is straightforward to verify that only $\vec{x}$ and $\vec{y}$ are recurrent in $P^{\mu^\star}$. Note that $r(\vec{x} \rightarrow a) \geq \alpha, \forall a \neq \vec{x}$, and $r(\vec{y} \rightarrow a) = 2, \forall a \neq \vec{y}$. Moreover, the path $\vec{x} \rightarrow (y, x, \ldots, x) \rightarrow (y, y, x, \ldots, x) \rightarrow \cdots \rightarrow \vec{y}$ has total resistance $\alpha + (n-2)(\alpha - 1)$ in $P_\beta^{\mu^\star}$.

It is straightforward to verify that this is the least resistance path from $\vec{x}$ to $\vec{y}$ in $P_\beta^{\mu^\star}$, implying $R(\vec{x}, \vec{y}) = \alpha + (n-2)(\alpha-1)$. The path $\vec{y} \rightarrow (x, y, \ldots, y) \rightarrow (x, x, y, \ldots, y) \rightarrow \cdots \rightarrow \vec{x}$ has resistance 2; hence $R(\vec{y} \rightarrow \vec{x}) = 2$. □