Fast Convergence in Semi-Anonymous Potential Games

Holly Borowski and Jason R. Marden

***Abstract*—Log-linear learning has been extensively studied in both the game theoretic and distributed control literature. A central appeal of log-linear learning for distributed control of multiagent systems is that this algorithm often guarantees that the agents' collective behavior will converge in probability to the optimal configuration. However, the worst case convergence time can be prohibitively long, e.g., exponential in the number of players. Building off the work in [20], in this paper we formalize a modified log-linear learning algorithm whose worst case convergence time is roughly linear in the number of players. We prove this characterization for a class of potential games where the agents' utility functions can be expressed as a function of aggregate behavior within a finite collection of populations. Lastly, we show that the convergence time remains linear in the number of players even when the players are permitted to enter and exit the game over time.**

I. Introduction

Game theoretic learning algorithms have gained traction as a powerful design tool for distributed control systems [9], [10], [16], [21], [24]. Here, a static game is repeated over time and agents are permitted to revise their strategies in response to their objective functions and information about the behavior of the other agents. Emergent collective behavior for such revision strategies has been studied extensively in the literature, e.g., fictitious play [8], [15], [17], regret matching [12], and log-linear learning [1], [5], [20]. While many results prove that these learning rules possess desirable asymptotic guarantees, convergence times associated with these algorithms remain uncharacterized or have been shown to be prohibitively long [7], [11], [13], [20]. Characterizing convergence rates is key to determining whether a game theoretic algorithm is desirable for system control.

The class of games known as *potential games* [18] has received significant research attention with regards to distributed control. A game is a potential game if each agent's local objective function is "aligned" with some system level objective function. Although potential games may not naturally emerge in many social systems, there are methods for designing local objective functions in engineering systems so that (i) the resulting game is a potential game and (ii) the optimal behavior corresponds to a Nash equilibrium of the game [3], [16], [22]. Consequently, a surge of research interest has focused on deriving distributed learning algorithms that converge to this efficient Nash equilibrium in potential games.

This research was supported by AFOSR grants #FA9550-09-1-0538 and #FA9550-12-1-0359, by ONR grant #N00014-09-1-0751, and by the NASA Aeronautics Scholarship Program.

H. Borowski is a graduate research assistant with the Department of Aerospace Engineering, University of Colorado, Boulder., holly.borowski@colorado.edu.

J. R. Marden is with the Department of Electrical, Computer, and Energy Engineering, University of Colorado, Boulder, CO 80309, jason.marden@colorado.edu. Corresponding author.

Log-linear learning is one algorithm that accomplishes this task [5]. Log-linear learning can be viewed as a perturbed best reply process where the agents predominantly select the optimal action given their beliefs regarding the behavior of other agents; however, the agents will occasionally make mistakes and select suboptimal actions with a probability that decays exponentially with respect to the potential payoff loss. As noise levels tend to zero, the resulting process has a unique stationary distribution with full support on the efficient Nash equilibria. Thus, by designing the agents' local objective functions appropriately, log-linear learning can be used to derive distributed control laws with highly desirable asymptotic guarantees.

Unfortunately, convergence rates associated with log-linear learning in its nominal form have been shown to be exponential in the game size [20]. This stems from inherent tension between desirable asymptotic behavior and convergence rates. The tension arises from the fact that small noise levels are necessary for ensuring that the bulk of the mass of the stationary distribution lies on the efficient Nash equilibria; however, small noise levels also make it difficult to leave inefficient Nash equilibria which ultimately degrades the underlying convergence rates.

Positive results regarding convergence rates of log-linear learning and its variants are beginning to emerge for specific problem instantiations [2], [14], [19], [20]. For example, in [19] the authors study the convergence rates of log-linear learning for a class of coordination games played over graphs and demonstrate that the underlying convergence rates are desirable provided that the interaction graph is sufficiently sparse. Alternatively, in [20] 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. Furthermore, the authors show that the convergence times remain linear in the number of players even under the situations where 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.

In this paper, we focus on identifying whether the positive results regarding convergence rates highlighted 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 as there are many simplifying attributes associated with symmetric congestion games over parallel networks that do not extend to more general network structures, e.g., uniqueness of equilibria (see Example 2). The main contributions of this paper are as follows:

– First, we formally define a subclass of potential games, termed *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. Here, agents within a given population have identical action sets and the same structural form of their objective functions. For comparison, the framework studied in [20] could be viewed as a semi-anonymous potential game with only one population.[1]

– Second, we introduce a variant of log-learning learning that is similar in spirit to the algorithm introduced in [20]. In Theorem 1, we prove that the convergence time of this algorithm grows linearly in the number of agents in the context of semi-anonymous potential games for a fixed number of populations. In particular, this analysis explicitly highlights the potential impact of system-wide heterogeneity, i.e., agents with different action sets or objective functions, on the underlying convergence rates. Furthermore, in Example 3 we demonstrate how a given resource allocation problem can be modeled as a semi-anonymous potential game.

– Lastly, 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 linear in the number of agents provided that the agents exit and enter the game at a sufficiently slow rate.

II. Semi-Anonymous Potential Games

Consider a game with agents $N = \{1, 2, \ldots, n\}$ where 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 will frequently express an action profile $a \in \mathcal{A}$ as (a_i, a_{-i}) where $a_{-i} = (a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n)$ denotes the actions of all agents other than agent i. Similarly, we let $\mathcal{A}_{-i} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_{i-1} \times \mathcal{A}_{i+1} \times \cdots \times \mathcal{A}_n$ denote the action sets of all players excluding i. We will 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 $\mathcal{N} = (N_1, N_2, \ldots, N_m)$ of N such that the following conditions are satisfied:

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

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

$$X_j = \left\{ \left(\frac{v_j^1}{n}, \frac{v_j^2}{n}, \ldots, \frac{v_j^{s_j}}{n} \right) \geq \mathbf{0} \,:\, \sum_{k=1}^{s_j} v_j^k = |N_j| \right\} \quad (1)$$

represent all possible aggregate action assignments for the agents within population N_j. The utility function of any agent $i \in N_j$ can be expressed as a lower-dimensional function of the form $\bar{U}_j : \bar{\mathcal{A}}_j \times X \to \mathbb{R}$ where $X = X_1 \times \cdots \times X_m$. More specifically, the utility associated with agent i for an action profile $a \in \mathcal{A}$ is of the form

$$U_i(a) = \bar{U}_j(a|_X)$$

where

$$a|_X = (a|_{X_1}, a|_{X_2}, \ldots, a|_{X_m}) \in X, \quad (2)$$

$$a|_{X_j} = \frac{1}{n} \left\{ \left| \{ p \in N_j \,:\, a_p = \tilde{a}_j^k \} \right| \right\}_{k=1,\ldots,s_j} \quad (3)$$

the operator $a|_X$ captures each population's aggregate behavior in the action profile a.

(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 $a_i' \in \mathcal{A}_i$,

$$U_i(a) - U_i(a_i', a_{-i}) = \phi(a|_X) - \phi((a_i', a_{-i})|_X). \quad (4)$$

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 [18].

Example 1 (Congestion Games [4])**.** Consider a congestion game with players $N = \{1, \ldots, n\}$, roads $R = \{r_1, r_2, \ldots, r_k\}$, and 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 $i \in N$ is of the form $\mathcal{A}_i \subseteq 2^R$, e.g., all paths that connect the player's source and destination. The utility function of each player $i \in N$ is of the form

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

where $|a|_r = |\{j \in N \,:\, r \in a_j\}|$. It is well known that this game is a potential game with potential function $\phi : \mathcal{A} \to \mathbb{R}$

$$\phi(a) = -\sum_{r\in R} \sum_{k=1}^{|a|_r} C_r(k). \quad (5)$$

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 can be viewed as a semi-anonymous potential games with a single population. Such games, also referred to as anonymous potential games, are the central focus of [20]. When the players' action sets are asymmetric, i.e., $\mathcal{A}_i \neq \mathcal{A}_j$ for all agents $i, j \in N$, then a congestion game can be viewed as a semi-anonymous potential where populations correspond to agents with identical path choices. It is important to highlight that the results in [20] are not proven to hold for such settings.

The following example sheds some light into the 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 $R = \{r_1, r_2, r_3\}$. Suppose n is even and players are divided evenly into populations N_1 and N_2, where 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 only on how many players have selected that resource, i.e., for

[1]The framework of semi-anonymous potential games can be viewed as the cross between a potential game and a finite population game [6].

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

any $k \in \{0, 1, \ldots, n\}$, the resource-specific welfare functions are

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

and the total system welfare is $\sum_{r \in R} W_r(|a|_r)$. Lastly, define each agent's utility function as their marginal contribution to the system-level welfare, i.e., for any agent i and action profile a

$$U_i(a) = W(a) - W(\emptyset, a_{-i}) \tag{6}$$

where $\emptyset$ indicates that player i did not select a resource. Note that the marginal contribution utility in (6) always ensures that the resulting game is a potential game with potential function W [23]. If the agents had symmetric action sets, i.e., $\mathcal{A}_i = \{r_1, r_2, r_3\}$ for all agents $i \in N$, then this game has exactly one Nash equilibrium with $n/2$ players at resource r_1 and $n/2$ players at resource r_2. Furthermore, this Nash equilibrium corresponds to the optimal allocation. One the other hand, for the two population scenario depicted above, there are two Nash equilibria: (i) an efficient 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) an inefficient Nash equilibrium in which all players from N_1 select resource r_2 and all players from N_2 select resource r_3. Employing distributed algorithms that effectively deal with multiplicities of equilibria invariably comes at the expense of the underlying convergence rates. Hence, transitioning for single population to multi-population scenarios can induce significant challenges associated with the underlying algorithm design.

III. Main Results

In this section, we present the main results of this paper. We begin by posing a variant of the well-studied algorithm *log-linear learning* [5] that is similar in spirit to the algorithm presented for single populations in [20]. Next, in Theorem 1 we show that for any semi-anonymous potential game our algorithm ensures that (i) the asymptotic behavior is close to the potential function maximizer and (ii) the mixing time grows roughly linearly in the number of agents. Lastly, in Theorem 2 we show that these guarantees continue to hold even in situations where agents are permitted to enter and exit the game. The significance of attaining fast convergence to the potential function maximizer in semi-anonymous potential games stems from the fact that a system-designer can often design agent objective functions that ensure that the potential function maximizers correspond to optimal system behavior as shown in Example 2.

A. Modified Log-Linear Learning

The following modification of the log-linear learning algorithm is an extension of the algorithm in [20]. Let $a(t) \in \mathcal{A}$ be the action profile at time $t \geq 0$. Each agent $i \in N$ is associated with a Poisson clock of rate $\alpha n / z_i(t)$, where

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

where $\alpha > 0$ is a design parameter. Hence, a player's update rate is higher if he is not using a common action within his population. If player i's clock ticks, he chooses action $a_i \in \bar{\mathcal{A}}_{p(i)}$ probabilistically according to

$$\begin{aligned} \text{Prob}[a_i(t^+) = a_i] &= \frac{e^{\beta U_i^t(a_i, a_{-i}(t))}}{\sum_{a_i' \in \mathcal{A}_i^t} e^{\beta U_i^t(a_i, a_{-i}(t))}} \\ &= \frac{e^{\beta \phi(a(t)|_X)}}{\sum_{a_i' \in \mathcal{A}_i^t} e^{\beta \phi((a_i', a_{-i}(t))|_X)}}, \end{aligned} \tag{7}$$

where $a_i(t^+)$ indicates the agent's revised action and β 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.

The agents' update rates are the only difference between this algorithm and the standard log-linear learning continuous time implementation or the variant posed in [20]. In the standard implementation, agents have a fixed rate 1 Poisson clock, yielding n expected updates per second over periods when n is fixed. The expected number of updates per second for modified log-linear learning is lower bounded by $m\alpha n$ and upper bounded $(|\bar{\mathcal{A}}_1| + \cdots + |\bar{\mathcal{A}}_m|)\alpha n$. To achieve an expected update rate at least as fast as the standard log-linear learning update rate, set $\alpha \geq 1/m$. For any $\alpha > 0$, these dynamics define an ergodic, reversible Markov process.

B. Semi-Anonymous Potential Games

The following theorem extends the results of [20] to semi-anonymous potential games.

Theorem 1. *Let $G = (N, \{\mathcal{A}_i\}, \{U_i\})$ be a stationary 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, 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:

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

Then for any fixed $\varepsilon \in (0, 1)$, if the the parameter β is sufficiently large, i.e.,

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

then

$$\mathbb{E}[\phi(a(t)|_X)] \geq \max_{x \in X} \phi(x) - \varepsilon \tag{9}$$

for all

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

where c_1 is a constant that depends only on s.

This theorem explicitly highlights the role of system-wide heterogeneity, i.e., $m > 1$ distinct populations, on the underlying mixing times of the process. For the special case when $m = 1$, this theorem recovers the results of [20]. Observe that for a fixed number of populations, the mixing time grows as $n \log\log n$.

C. Time Varying Semi-Anonymous Potential Games

In this section, we expand the framework of semi-anonymous potential games to model the scenario where agents enter and exit the game over time. To that end, consider a trajectory of semi-anonymous potential games,

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

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* and the agent is associated with the action set $\mathcal{A}_i^t = \emptyset$ at time t. We define $X = \cup_{t\in\mathbb{R}^+} X^t$, where X^t is the finite aggregate state space corresponding to game G^t. At time t, we denote the partitioning of players per Definition 1 by $\mathcal{N}^t = (N_1^t, N_2^t, \ldots, N_m^t)$. Further, 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., $\forall j \in \{1, 2, \ldots, m\}$, $\forall t_1, t_2 \in R^+$, $\bar{\mathcal{A}}_j^{t_1} = \bar{\mathcal{A}}_j^{t_2}$. 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) - (ii) of Theorem 1 are satisfied. Then for any fixed $\varepsilon \in (0, 1)$, if the the parameter β satisfies (8) and the following additional conditions are met:*
(iii) For all $t \in \mathbb{R}^+$, the number of players satisfies:

$$|\mathcal{N}^t| \geq \max\left\{ \frac{4\alpha m e^{-3\beta}}{2^{2ms} c_1 m^2 (m(s-1))!^2}, 2\beta\lambda + 1 \right\}, \quad (10)$$

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

$$|N_i^t| \geq |\mathcal{N}^t| / k, \quad \forall i \in \{1, 2, \ldots, m\},\ \forall t \in \mathbb{R}^+. \quad (11)$$

(v) There exists a constant

$$\Lambda \geq 8 c_0 \varepsilon^{-2} e^{3\beta} (6\beta\lambda + e^{\beta} k(s-1)) \quad (12)$$

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

$$\left|\{ i \in \mathcal{N}^{t_1} \cup \mathcal{N}^{t_2} \,:\, \mathcal{A}_i^{t_1} \neq \mathcal{A}_i^{t_1} \}\right| \leq 1, \quad (13)$$

where c_0 and c_1 do not depend on the number of players. Accordingly, the constant Λ does not depend on n. Furthermore, at most one agent may become active or inactive in a time interval of length Λ, and agents may not switch populations over this interval, i.e., if $i \in \mathcal{N}^{t_1} \cap \mathcal{N}^{t_2}$, then $i \in N_j^t$ for some $j \in \{1, \ldots, m\}$ and for all time $t \in [t_1, t_2]$.
Then, we have

$$\mathbb{E}[\phi(a(t)|_X)] \geq \max_{x\in X(t)} \phi(x) - \varepsilon \quad (14)$$

for all

$$t \geq |\mathcal{N}^0| e^{3\beta} c_0 \left(\frac{(ms-m)! \log(n(0)+2) + \beta}{\varepsilon^2} - 2 \right). \quad (15)$$

Theorem 2 proves that, if player entry and exit rates are sufficiently slow, i.e., 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 (15): larger β tends to slow convergence. However, the minimum β necessary to achieve an expected potential near the maximum, as in (14), is independent of the number of players, as given in (8).

IV. Illustrative Examples

In the following examples we consider resource allocation games with a similar structure to Example 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 $x = (x_1, x_2, x_3, x_4)$, where nx_1 and nx_2 are the numbers of players from N_1 choosing resources r_1 and r_2 respectively. Likewise, nx_3 and nx_4 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 total system welfare for a given state is the sum of the welfare garnered at each resource, i.e.,

$$W(x) = W_{r_1}(nx_1) + W_{r_2}(n(x_2 + x_3)) + W_{r_3}(nx_4).$$

Player utilities are their marginal contribution to the total welfare, W, as in (6).

The first example supports the result of Theorem 1 that convergence time for modified log-linear learning grows as $\Theta(n \log\log n)$ via simulation of the algorithm.

Example 3. For $x = (x_1, x_2, x_3, x_4) \in \bar{\mathcal{X}}$, welfare functions for resources in R are

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

The global welfare optimizing allocation is $a_i = r_2$ for all $i \in N$, i.e., $x^{\text{opt}} = (0, 1/2, 1/2, 0)$.

We simulated our algorithm with rate $\alpha = 1/\sigma = 1/4$ starting from an inefficient Nash equilibrium $x^{\text{ne}} = (1/2, 0, 0, 1/2)$. Using Theorem 1, we examine the time it takes for the expected welfare with respect to the empirical frequency of joint actions to come within 90% of its maximum. Here, β is set appropriately for each value of n to ensure the

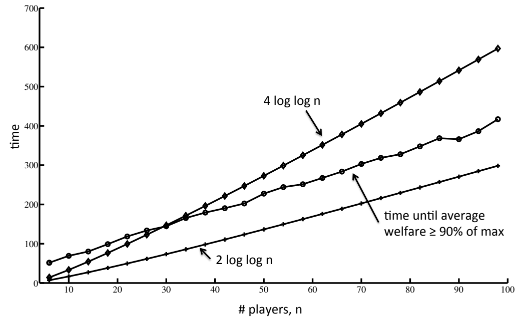

Fig. 1: Number of players vs. average time to reach 90% of maximum welfare

expected welfare with respect to the stationary distribution is 90% of maximum. The empirical frequency for a state $x \in \bar{\mathcal{X}}$ is

$$\nu_x(t) = \frac{1}{t} \sum_{\tau=1}^{t} \mathbb{I}\{x(\tau) = x\}.$$

The indicator function $\mathbb{I}$ returns 1 if the state $x(\tau)$ is x and 0 otherwise; $\nu_x(t)$ is the percentage of time from $\tau = 1$ to $\tau = t$ that the system has been in state x.

$$\mathbb{E}_{\nu(t)} W(x) \geq .9 \max_{x \in \bar{\mathcal{X}}} W(x)$$

per Theorem 1. Simulation results are shown in Figure 1 for an average over 2000 simulations with n ranging from 4 to 100. Average times until the expected welfare comes within 90% of its maximum are bounded below by $2n \log \log n$ for all n and bounded above by $4n \log \log n$ when $n > 30$. These results support Theorem 1.

Example 4. In this example we compare convergence times of our log-linear learning variant, the variant of [20], and standard log-linear learning. We use the probability transition kernels of each algorithm to compute the expected welfare over time.

Starting with the basic setup of the previous example, 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, states may be represented in the same way as in the previous example, simply reflecting the aggregate actions of players in populations N_1 and N_2. The three resources have the following submodular welfare functions for each $x = (x_1, x_2, x_3, x_4) \in \bar{\mathcal{X}}$:

$$\begin{aligned} W_{r_1}(x) &= 2nx_1, \\ W_{r_2}(x) &= \min\left\{3(nx_2 + nx_3), \frac{3}{2}(nx_1 + nx_2)\right\}, \\ W_{r_3}(x) &= nx_4. \end{aligned}$$

The welfare maximizing state is $x^{\text{opt}} = (1/2, 0, 1/2, 0)$, and the state $x^{\text{ne}} = (0, 1/2, 0, 1/2)$ represents an inefficient Nash equilibrium, which we set as the initial configuration. Suppose we wish to achieve an expected total welfare within 98% of the maximum.

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. We use the notation $x_5 = n_3/n$.

Recall that in both our log linear learning variant and in the variant introduced in [20], an updating player chooses its new action according to (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 selecting the same action as agent i at time t. The discrete time kernel of this process can be described using the following update rule:

- Select a population $N_j \in \{N_1, N_2, N_3\}$ uniformly at random.
- Select a resource $r \in \bar{\mathcal{A}}_j$ uniformly at random.
- Select a player uniformly at random from population N_j who is currently choosing resource r. This player updates its action according to (7).

Hence, in this example, 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.

In the algorithm of [20], 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, as applied to our multiple population setting, can be described using the following update rule [20]:

- Select a resource $r \in \{r_1, r_2, r_3\}$ uniformly at random.
- Select a player uniformly at random who is currently choosing resource r. This player updates its action according to (7).

In general, the two algorithms differ when at least two populations have overlapping action sets. Here, 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. This sensitivity to the size of N_3 is even more significant for standard log-linear learning, since all players are equally likely to be selected for update in the discrete time version.

We begin by selecting β so that, in the limit, the expected welfare is within 98% of its maximum. Then we examine the number of updates necessary to come within $\varepsilon = 0.05$ of this expected welfare. Often the global update rate is set to be n per second; in this case, convergence times are a factor of n smaller than the number of updates to convergence.

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

$$\pi_x^{\text{LLL}} \propto e^{\beta W(x)} \binom{n_1}{nx_1, nx_2} \binom{n_2}{nx_3, nx_4} \qquad (16)$$

$$\pi_x^{\text{MLLL}} \propto e^{\beta W(x)} \qquad (17)$$

These stationary distributions can be verified using reversibility arguments with the standard and modified log-linear learning probability transition kernels. 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 [20] does change with n_3. We estimate stationary distribution by raising the probability transition matrix for their log-linear learning variant to a sufficiently high power, then use this to determine the β necessary to reach the desired expected welfare.

Table 4 shows the required β values for each algorithm to yield the desired expected welfare with respect to its stationary distribution. The table also shows the number of updates until the expected total welfare has come within $\varepsilon = 0.05$ of this goal value. The expected value of the welfare after a given number of updates, k, is determined by simply raising the probability transition matrix to the kth power.

Our algorithm converges to the desired expected welfare in fewer updates than both alternate algorithms for all tested values of n_3; in fact, the number of updates required does not vary with n_3. The ratio between the convergence times of each of the two alternate algorithms and our algorithm approaches 0 as the number of players grow, showing that convergence rates for log linear learning and the variant from [20] are both more sensitive to the number of players in population 3 than our algorithm.

In this example, a high update rate for players in population N_3 was undesirable because they contribute no value. This example does not directly represent a practical scenario: the system could simply be modeled without the agents in population 3 to avoid convergence rate impacts illustrated here. However, mild variations to this example are expected to display similar behavior. For example, consider a scenario in which a relatively large population may choose from multiple resources, but contributes relatively little welfare at each.

V. Conclusion

We have extended the results of [20] to define dynamics for a class of semi-anonymous limited population 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.

Algorithm	n_3	β	Expected welfare	# updates to converge
Standard Log Linear Learning	1	3.77	98%	9430
	5	3.77	98%	11947
	50	3.77	98%	40250
	500	3.77	98%	323277
Log Linear Learning Variant from [20]	1	2.39	98%	1325
	5	2.44	98%	1589
	50	2.83	98%	3342
	500	3.72	98%	15550
Our Log Linear Learning Variant	1	1.28	98%	743
	5	1.28	98%	743
	50	1.28	98%	743
	500	1.28	98%	743

TABLE I: This table shows the required values of β for each algorithm to guarantee an expected total welfare within 98% of the maximum with respect to the stationary distribution, and how this value varies with the number of players for each algorithm. Note that for standard log-linear learning and for our variant, the β required to reach the desired expected welfare is constant, whereas is grows with n for the log-linear learning variant of [20]. The final column also shows the number of updates necessary to ensure the emergent behavior which guarantees this near-maximum welfare has been achieved. Note that this value does not increase with n for our algorithm, but does increase with n for the other two. Update rates are a design parameter in the log-linear learning algorithm; by selecting a global update rate of n per second, the convergence times would be a factor of n smaller than the number of updates shown.

References

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

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

[3] 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.

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

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

[6] L. E. Blume. Population games. *The Economy as a Complex Evolving System II*, pages 425–460, 1996.

[7] 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.

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

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

[10] 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, September 2010.

[11] 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.

[12] S. Hart and A. MasColell. A Simple Adaptive Procedure Leading to Correlated Equilibrium. *Econometrica*, 68(5):1127–1150, 2000.

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

[14] G. E. Kreindler and H. P. Young. Fast convergence in evolutionary equilibrium selection. 2011.

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

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

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

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

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

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

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

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

[23] DH Wolpert and K. Tumer. An Introduction To Collective Intelligence. Technical report, 1999.

[24] 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.