Game Theoretic Learning in Distributed Control *

Jason R. Marden† Jeff S. Shamma‡

November 1, 2016
May 11, 2017 (revised)

Abstract

In distributed architecture control problems, there is a collection of interconnected decision making components that seek to realize desirable collective behaviors through local interactions and by processing local information. Applications range from autonomous vehicles to energy to transportation. One approach to control of such distributed architectures is to view the components as players in a game. In this approach, two design considerations are the components' incentives and the rules that dictate how components react to the decisions of other components. In game theoretic language, the incentives are defined through utility functions and the reaction rules are online learning dynamics. This chapter presents an overview of this approach, covering basic concepts in game theory, special game classes, measures of distributed efficiency, utility design, and online learning rules, all with the interpretation of using game theory as a prescriptive paradigm for distributed control design.

Keywords: Learning in games; Evolutionary games; Multiagent systems; Distributed decision systems;

1 Introduction

There is growing interest in distributed architecture or networked control systems, with emergent applications ranging from smart grid to autonomous vehicle networks to mobile sensor platforms.

*This work was supported by ONR Grant #N00014-15-1-2762 and NSF Grant #ECCS-1351866 and by funding from King Abdullah University of Science and Technology (KAUST).

†J.R. Marden is with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, 5161 Harold Frank Hall, Santa Barbara, CA 93106, jrmarden@ece.ucsb.edu.

‡J.S. Shamma is with the King Abdullah University of Science and Technology (KAUST), Computer, Electrical and Mathematical Science and Engineering Division (CEMSE), Thuwal 23955–6900, Saudi Arabia, jeff.shamma@kaust.edu.sa.

As opposed to a traditional control system architecture, there is no single decision making entity with full information and full authority that acts as an overall system controller. Rather, decisions are made by a collective of interacting entities with local information and limited communication capabilities. The challenge is to derive distributed controllers to induce desirable collective behaviors.

One approach to distributed architecture systems is to view the decision making components as individual players in a game and to formulate the distributed control problem in terms of game theory. The basic elements of what constitutes a game are (i) a set of players or agents; (ii) for each player, a set of choices; and (iii) for each player, preferences over the *collective* choices of agents, typically expressed in the form of a utility function. In traditional game theory (e.g., [14]), these elements are a *model* of a collection of decision makers, typically in a societal context (e.g., competing firms, voters, bidders in an auction, etc.). In the context of distributed control, these elements are *design considerations* in that one has the degree of freedom on how to decompose a distributed control problem and how to design the preferences/utility functions to properly incentivize agents. Stated differently, game theory in this context is being used as a *prescriptive* paradigm, rather than a *descriptive* paradigm [28, 41].

Formulating a distributed control problem in game theoretic terms implicitly suggests that the outcome—or more appropriately, the solution concept—of the resulting game is a desirable collective configuration. The most well known solution concept is Nash equilibrium, in which each player's choice is optimal with respect to the choices of other agents. Other solution concepts, which are generalizations of Nash equilibrium, are correlated and coarse correlated equilibrium [49]. Typically, a solution concept does not uniquely specify the outcome of a game (e.g., a game can have multiple Nash equilibria), and so there is the issue that some outcomes are better than others.

A remaining concern is how a solution concept emerges at all. Given the complete description of a game, an outside party can proceed to compute (modulo computational complexity considerations [11]) a proposed solution concept realization. In actuality, the data of a game (e.g., specific utility functions) is distributed among the players and not necessarily shared or communicated. Rather, over time players might make observations of the choices of the other players and eventually the collective play converges to some limiting structure. This latter scenario is the topic of game theoretic learning, for which there are multiple survey articles and monographs (e.g., [13, 16, 40, 49]). Under the descriptive paradigm, the learning in games discussion provides a plausibility argument of how players may arrive at a specified solution concept realization. Under the prescriptive paradigm, the learning in games discussion suggests an *online algorithm* that can lead agents to a desirable solution concept realization.

This article will provide an overview of approaching distributed control from the perspective of game theory. The presentation will touch on each of the aforementioned aspects of problem

formulation, game design, and game theoretic learning.

2 Game Theoretic Distributed Resource Utilization

2.1 Setup

Various problems of interest take the form of allocating a collection of assets to utilize a set of resources to a desired effect. In sensor coverage problems (e.g., [10]), the "assets" are mobile sensors, and the "resources" are the regions to be covered by the sensors. For any given allocation, there is an overall score reflecting the quality of the coverage. In traffic routing problems (e.g, [37]), the "assets" are vehicles (or packets in a communication setting), and the "resources" are roads (or channels). The objective is to route traffic from origins to destinations in order to minimize a global cost such as congestion.

It will be instructive to intrepret the forthcoming discussion on distributed control in the framework of such distributed resource utilization problems. As previously mentioned, the framework captures a variety of applications of interest. Furthermore, focusing on this specific setting will enhance the clarity of exposition.

More formally, the problem is to allocate a collection of assets $N = \{1, 2, ..., n\}$ over a collection of resources $\mathcal{R} = \{1, 2, \ldots, m\}$ in order to optimize a given system level objective. The set $\mathcal{A}_i \subseteq 2^{\mathcal{R}}$ is the allowable resource selections by asset i. In terms of the previous examples, an allowable resource selection is an area covered by a sensor or a set of roads used by vehicle. The system level objective is a mapping $W : \mathcal{A} \to \mathbb{R}$ where $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_n$ denotes the set of joint resource selections. We denote a collective configuration by the tuple $a = (a_1, a_2, ..., a_n)$ where $a_i \in \mathcal{A}_i$ is the choice, or *action*, of agent i.

Moving towards a game theoretic model, we will identify the set of assets as the set of agents or players. Likewise, we will identify $\mathcal{A}_i$ as the choice set of agent i. We defer for now specifying a utility function for agent i.

Looking forward to the application of game theoretic learning, we will consider agents selecting actions iteratively over an infinite time horizon $t \in \{1, 2, ...\}$. Depending on the update rules of the agents, the outcome is a sequence of joint actions $a(1), a(2), a(3), ...$ The action of agent i at time t is chosen according to some update policy, $\pi_i(\cdot)$, i.e.,

$$a_i(t) = \pi_i\left(\text{information available to agent } i \text{ at time } t\right). \tag{1}$$

The update policy $\pi_i(\cdot)$ specifies how agent i processes available information to formulate a decision. We will be more explicit about the argument of the $\pi_i(\cdot)$'s in the forthcoming discussion. For now, the information available to an agent can include both knowledge regarding previous action choices of other agents and certain system-level information that is propagated throughout the

system.

The main goal is to design both the agents' utility functions and the agents' local policies $\{\pi_i\}_{i\in N}$ to ensure that the emergent collective behavior optimizes the global objective W in terms of the asymptotic properties of $W(a(t))$ as $t \to \infty$.

2.2 Prescriptive paradigm

Once the players and their choices have been set, the remaining elements in the prescriptive paradigm that are yet to be designed are (i) the agent utility functions and (ii) the update policies, $\{\pi_i\}_{i\in N}$. One can view this specification in terms of the following two-step design procedure:

Step #1: Game Design. The first step of the design involves defining the underlying interaction structure in a game theoretic environment. In particular, this choice involves defining a utility function for each agent $i \in N$ of the form $U_i : \mathcal{A} \to \mathbb{R}$. The utility of agent i for an action profile $a = (a_1, a_2, ..., a_n)$ is expressed as $U_i(a)$ or alternatively $U_i(a_i, a_{-i})$ where a_{-i} denotes the collection of actions other than player i in the joint action a, i.e., $a_{-i} = (a_1, ..., a_{i-1}, a_{i+1}, ..., a_n)$. A key feature of this design choice is the coupling of the agents' utility functions where the utility, or payoff, of one agent is affected by the actions of other agents.

Step #2: Learning Design. The second step involves defining the decision making rules for the agents. That is, how does each agent process available information to formulate a decision. A typical assumption in the framework of learning in games is that each agent uses historical information from previous actions of itself and other players. Accordingly, at each time t the decision of each agent $i \in N$ is made independently through a learning rule of the form

$$a_i(t) = \pi_i \left(\{a(\tau)\}_{\tau=1,...,t-1} ; U_i(\cdot) \right). \tag{2}$$

The are two important considerations in the above formulation. First, we stated for simplicity that agents can observe the actions of all other agents. In games with a graphical structure [22], one only requires historical information from a *subset* of other players. Other reductions are also possible, such as aggregate information of other players of even just measurements of one's own utility [13,16,40,49][1]. Second, implicit in the above construction is that the learning rule is defined independently of the utility function, and an agent's utility function then enters as a parameter of a specified learning rule.

[1] Alternative agent control policies where the policy of agent i also depends on previous actions of agent i or auxiliary "side information" could also be replicated by introducing an underlying state in the game-theoretic environment. The framework of state based games, introduced in [23], represents one such framework that could accomplish this goal.

This second consideration offers a distinction between conventional distributed control and game theoretic distributed control in the role of the utility function for the individual agents $\{U_i\}_{i\in N}$. An agent may be using a specific learning rule, but the realized behavior depends on the specified utility function. In a more conventional approach, there need not be such a decomposition. Rather, one might directly specify the agents' control policies $\{\pi_i\}_{i\in N}$ and perform an analysis regarding the emergent properties of the given design, e.g., as is done in models of flocking or bio-inspired controls [34]. An advantage of the decomposition is that one can analyze learning rules for classes of games and separately examine whether or not specified utility functions conform to such an assumed game class.

The following example demonstrates how a given distributed control policy $\{\pi\}_{i\in N}$ can be reinterpreted as a game-theoretic control approach with appropriately defined agent utility functions $\{U_i\}_{i\in N}$.

Example 2.1 (Consensus) *Consider the well-studied consensus/rendezvous problem [5,21,36,42] where the goal is to drive the agents to agreement on a state $x^* \in \mathbb{R}$ when each agent has limited information regarding the state of other agents in the systems. Specifically, we will say that the set of admissible states (or actions) of each agent $i \in N$ is $\mathcal{A}_i = \mathbb{R}$ and agent i at stage t can observe the previous state choices at stage $t-1$ of a set of neighboring agents denoted by $\mathcal{N}_i(t) \subseteq N \setminus \{i\}$. Consider the following localized averaging dynamics where the decision of an agent $i \in N$ at time t is of the form*

$$a_i(t) = \frac{1}{\mathcal{N}_i(t)} \sum_{j\in\mathcal{N}_i(t)} a_j(t-1). \tag{3}$$

Given an initial state profile $a(0)$, the dynamics in (3) *produces a sequence of state profiles $a(1)$, $a(2)$, ... Whether or not the state profiles converge to consensus under the above dynamics (or variants thereof) has been extensively studied in the existing literature [4, 36, 44].*

Now we will present a game-theoretic design that leads to the same collective behavior. More formally, consider a game-theoretic model where each agent $i \in N$ is assigned an action set $\mathcal{A}_i = \mathbb{R}$ and a utility function of the form

$$U_i(a_i, a_{-i}) = -\frac{1}{2\,|\mathcal{N}_i(t)|} \sum_{j\in\mathcal{N}_i(t)} (a_i - a_j)^2, \tag{4}$$

where $|\mathcal{N}_i(t)|$ denotes the cardinality of the set $\mathcal{N}_i(t)$. Now, suppose each agent follows the well known best-response learning rule of the form

$$a_i(t) \in B_i(a_{-i}(t)) = \arg\max_{a_i\in\mathcal{A}_i} U_i(a_i, a_{-i}(t-1)),$$

where $B_i(a_{-i}(t))$ is referred to as the best response set of agent i to the action profile $a_{-i}(t)$. Given an initial state profile $a(0)$, it is straightforward to show that the ensuing action or state profiles $a(1)$, $a(2)$, ..., will be equivalent for both design choices.

The above example illustrates the separation between the learning rule and the utility function. The learning rule is best response dynamics. When the utility function is the above quadratic form, then the combination leads to the usual distributed averaging algorithm. If the utility function is changed (e.g., weighted, non-quadratic, etc.), then the realization of best response learning is altered, as well as the structure of the game defined by the collection of the utility functions, but the learning rule remains best response dynamics.

An important property of best response dynamics and other learning rules of interest is that the actions of agent i can depend explicitly on the utility function of agent i but not (explicitly) on the utility functions of other agents. This property of learning rules in the learning in games literature is called being *uncoupled*. [3, 17, 19, 49]. Of course, the action stream of agent i, i.e., $a_i(0), a_i(1), ...$, does depend on the *actions* of other agents, but not the utility functions behind those actions.

It turns out that there are many instances in which control policies not derived from a game theoretic perspective can be reinterpreted as the realization of an uncoupled learning rule from a game theoretic perspective. These include control policies that have been widely studied in the cooperative control literature with application domains such as consensus and flocking [35, 43], sensor coverage [29, 32], routing information over networks [37], among many others.

While the design of such control policies can be approached in either a traditional perspective or a game-theoretic perspective, there are potential advantages associated with viewing control design from a game theoretic perspective. In particular, a game theoretic perspective allows for a modularized design architecture, i.e., the separation of game design and learning design, that can be exploited in a plug-and-play fashion to provide control algorithms with automatic performance guarantees:

Game Design Methodologies. There are several established methodologies for the design of agent objective functions, e.g., Shapley value and marginal contribution [26]. The methodologies, which will be briefly reviewed in Section 3.6, are systematic procedures for deriving the agent objective functions $\{U_i\}_{i\in N}$ from a given system-level objective function G. These methodologies often provide structural guarantees on the resulting game, e.g., existence of a pure Nash equilibrium or a potential game structure, that can be exploited in distributed learning.

Learning Design Methodologies. The field of learning in games has sought out to establish decision-making rules that lead to Nash equilibrium or other solution concepts in strategic form games. In general, it has been shown (see [19]) that there are no "natural" dynamics that converge to Nash equilibria for all games, where natural refers to dynamics that do not rely on some form of centralized coordination, e.g., exhaustive search of the joint action profiles. For example, there are no rules of the form (2) that provide convergence to a Nash equilibrium in any game. However,

the same limitations do not hold when we transition from "all games" to "all games of a given structure". In particular, there are several positive results in context of learning in games for special classes of games (e.g., potential games and variants thereof). These results, which will be discussed in Section 4, identify learning dynamics that yield desirable performance guarantees when applied to the realm of potential games.

Performance Guarantees. Merging a game design methodology with an appropriate learning design methodology can often result in agent control policies with automatic performance guarantees. For example, employing a game design where agent utility functions constitute a potential game coupled with a learning algorithm that ensures convergence to a pure Nash equilibrium in potential games, provides agent control policies that converge to the Nash equilibrium of the derived game. Furthermore, additional structure on the agents' utility functions can often be exploited to provide efficiency bounds on the Nash equilibria, c.f., price of anarchy [33], as well approximations for the underlying convergence rates [8, 31, 39].

Human-Agent Collaborative Systems. Game theory constitutes a design choice for control policies in distributed systems comprised purely of engineering components. However, when a networked system consists of both engineering and human-decision making entities, e.g., the smart grid, game theory transitions from a design choice to a necessity. The involvement of human decision-making entities in a system requires that the system-operator utilizes game theory for the purpose of modeling and influencing the human decision-making entities to optimize system-performance.

3 Solution Concepts, Game Structures, and Efficiency

Recall that an important metric in the game theoretic approach to distributed control is the asymptotic properties of a system level objective function, i.e., $W(a(t))$ as $t \to \infty$. These asymptotic properties depend on both aspects of the prescriptive paradigm, i.e., the utility functions and learning rule. The specification of utility functions in itself defines an underlying game that is repeatedly played over stages. In this section, we review properties related to this underlying game in terms of solution concepts, game structures, and measures of efficiency.

In this section we will temporarily distance ourselves from the design objectives set forth in this manuscript with the purpose of identifying properties of games that are relevant to our mission. To that end, we will consider a finite strategic form game G with agent set $N = \{1, 2, \ldots, n\}$ where each agent $i \in N$ has an action set $\mathcal{A}_i$ and a utility function $U_i : \mathcal{A} \to \mathbb{R}$. Further, there exists a system-level objective $W : \mathcal{A} \to \mathbb{R}$ that a system designer is interested in maximizing. We

will often denote such a game by the tuple $G = \{N, \{\mathcal{A}_i\}, \{U_i\}, W\}$ where we use the shorthand notation $\{\cdot\}$ instead of $\{\cdot\}_{i \in N}$ to denote the agents' action sets or utility functions.

3.1 Solution concepts

The most widely known solution concept in game theory is a pure Nash equilibrium, defined as follows.

Definition 3.1 *An action profile $a^{\text{ne}} \in \mathcal{A}$ is a pure Nash equilibrium if for any agent $i \in N$*

$$U_i(a_i^{\text{ne}}, a_{-i}^{\text{ne}}) \geq U_i(a_i, a_{-i}^{\text{ne}}), \forall a_i \in \mathcal{A}_i. \tag{5}$$

A pure Nash equilibrium represents an action profile where no agent has a unilateral incentive to alter its action provided that the behavior of the remaining agents is unchanged. A pure Nash equilibrium need not exist for any game G.

The definition of Nash equilibrium also extends to scenarios where the agents can probabilistically choose their actions. Define a strategy of agent i as $p_i \in \Delta(\mathcal{A}_i)$ where $\Delta(\mathcal{A}_i)$ denotes the simplex over the finite action set $\mathcal{A}_i$. We will express a strategy p_i by the tuple $\{p_i^{a_i}\}_{a_i \in \mathcal{A}_i}$ where $p_i^{a_i} \geq 0$ for any $a_i \in \mathcal{A}_i$ and $\sum_{a_i \in \mathcal{A}_i} p_i^{a_i} = 1$. We will evaluate the utility of an agent $i \in N$ for a strategy profile $p = (p_1, \ldots, p_n)$ as

$$U_i(p_i, p_{-i}) = \sum_{a \in \mathcal{A}} U_i(a) \times p_1^{a_1} \times \cdots \times p_n^{a_n}. \tag{6}$$

which has the usual interpretation of the expected utility under independent randomized actions.

We can now state the definition of Nash equilibrium when extended to mixed (or probabilistic) strategies.

Definition 3.2 *A strategy profile $p^{\text{ne}} \in \Delta(\mathcal{A}_1) \times \cdots \times \Delta(\mathcal{A}_n)$ is a mixed Nash equilibrium if for any agent $i \in N$*

$$U_i(p_i^{\text{ne}}, p_{-i}^{\text{ne}}) \geq U_i(p_i, p_{-i}^{\text{ne}}),\ \forall p_i \in \Delta(\mathcal{A}_i). \tag{7}$$

Unlike pure Nash equilibria, a mixed Nash equilibrium is guaranteed to exist in any[2] game G.

A common critique regarding the viability of pure or mixed Nash equilibria as a characterization of achievable behavior in multiagent systems is that the complexity associated with computing such equilibria is often prohibitive [11]. We now introduce a weaker solution concept, which is defined relative to a joint distribution $z \in \Delta(\mathcal{A})$, that does not suffer from such issues.

Definition 3.3 *A joint distribution $z = \{z^a\}_{a \in \mathcal{A}} \in \Delta(\mathcal{A})$ is a coarse correlated equilibrium if for any agent $i \in N$*

$$\sum_{a \in \mathcal{A}} U_i(a_i, a_{-i}) z^{(a_i, a_{-i})} \geq \sum_{a \in \mathcal{A}} U_i(a_i', a_{-i}) z^{(a_i, a_{-i})},\ \forall a_i' \in \mathcal{A}_i. \tag{8}$$

[2]Recall that we are assuming a finite set of players, each with a finite set of actions.

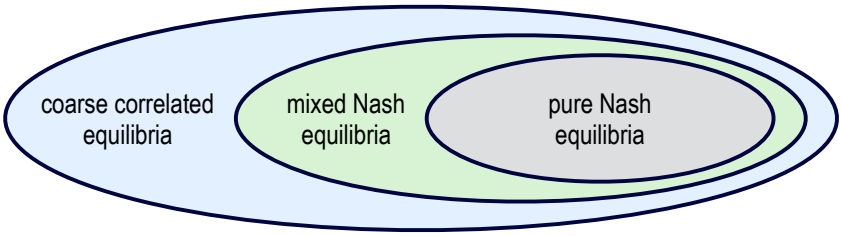

Figure 1: The relationship between the three solution concepts: pure Nash equilibrium, mixed Nash equilibrium, and coarse correlated equilibrium.

A coarse correlated equilibrium is a joint distribution z such that each agent's expected utility according to that distribution is at least as high as the agent's expected utility for committing to any fixed action $a_i' \in \mathcal{A}$ while all the other agents play according to their marginal distribution of z. It is straightforward to verify that any mixed Nash equilibrium is a coarse correlated equilibrium; hence, the set of coarse correlated equilibria is non-empty for any game, G. Furthermore, as we will see in Section 4.4, there are simple learning algorithms that ensure that the empirical frequency of play will approach the set of coarse correlated equilibria in a reasonable period of time. We will discuss techniques for characterizing the efficiency of this type of collective behavior in Section 3.3.[3]

Figure 1 highlights the relationship between the three solution concepts discussed above.

3.2 Measures of efficiency

It is important to highlight that the above equilibrium definitions have no dependence on the system level objective function. The goal here is to understand how the efficiency associated with such equilibria compares to the optimal behavior with respect to a system level objective function. Here, we investigate two common worst-case measures, termed *price of anarchy* and *price of stability* [33], for characterizing the inefficiency associated with equilibria in games.

The first measure that we consider is the price of anarchy, which is defined as the worst-case ratio between the performance of the worst equilibrium and the optimal system behavior. We use the terminology worst equilibrium as the price of anarchy could be defined by restricting attention to any of the aforementioned equilibrium sets. Focusing on pure Nash equilibria for simplicity, the price of anarchy associated with a game G is defined as

$$\mathrm{PoA}(G) = \min_{a^{\mathrm{ne}} \in \mathrm{PNE}(G)} \left\{ \frac{W(a^{\mathrm{ne}})}{W(a^{\mathrm{opt}})} \right\} \leq 1, \tag{9}$$

[3] Another common equilibrium set, termed correlated equilibrium, is similar to coarse correlated equilibrium where the difference lies in the consideration of conditional deviations as opposed to the unconditional deviations considered in (8). A formal definition of correlated equilibrium can be found in [49].

where $a^{\text{opt}} \in \arg\max_{a \in \mathcal{A}} W(a)$ and $\text{PNE}(G)$ denotes the set of pure Nash equilibria in the game G. Note that the price of anarchy given in (9) provides a lower bound on the performance associated with any pure Nash equilibrium in the game G.

The second measure that we consider is the price of stability, which is defined as the best-case ratio between the performance of the *best* equilibrium and the optimal system behavior. Focusing on pure Nash equilibria for simplicity, the price of stability associated with a game G is defined as

$$\text{PoS}(G) = \max_{a^{\text{ne}} \in \text{PNE}(G)} \left\{ \frac{W(a^{\text{ne}})}{W(a^{\text{opt}})} \right\} \leq 1. \tag{10}$$

By definition, $\text{PoS}(G) \geq \text{PoA}(G)$. The price of stability is a more optimistic measure of the efficiency loss associated with pure Nash equilibrium. When analyzing dynamics that converge to specific types of equilibrium, e.g., the best Nash equilibrium, the price of stability may be a more reasonable characterization of the efficiency associated with the limiting behavior.

The above definition of price of anarchy and price of stability also extend to situations where there is uncertainty regarding the structure of the specific game. To that end, let $\mathcal{G}$ denote a family of possible games. The price of anarchy and price of stability associated with the family of games is then defined as the worst-case performance of all games within that family, i.e.,

$$\text{PoA}(\mathcal{G}) = \min_{G \in \mathcal{G}} \{\text{PoA}(G)\}, \tag{11}$$

$$\text{PoS}(\mathcal{G}) = \min_{G \in \mathcal{G}} \{\text{PoS}(G)\}. \tag{12}$$

Clearly, $1 \geq \text{PoS}(\mathcal{G}) \geq \text{PoA}(\mathcal{G})$. For clarity, a $\text{PoA}(\mathcal{G}) = 0.5$ implies that regardless of the underlying game $G \in \mathcal{G}$, any pure Nash equilibrium is at least 50% efficient when compared to the performance of the optimal allocation for that game.

The definitions of price of anarchy and price of stability given in (9) and (10) can be extended to broader classes of equilibria, i.e., mixed Nash equilbria or coarse correlated equilibria, in the logical manner. To perform the above analysis for broader equilibrium sets, we extend the definition of the welfare function to a distribution $z \in \Delta(\mathcal{A})$ as $W(z) = \sum_{a \in \mathcal{A}} W(a) z^a$. Note that for a given family of games $\mathcal{G}$, the price of anarchy associated with pure Nash equilibria would be better (closer to 1) than the price of anarchy associated with coarse correlated equilibrium. Since coarse correlated equilibria contain Nash equilibria, one would naturally expect that the efficiency associated with equilibria could be far worse than the efficiency associated with Nash equilibria. Surprisingly, it often turns out that this is not the case as we will see below.

3.3 Smoothness

Charactering the inefficiency of equilibria often is challenging and often involves a non-trivial domain specific analysis. One attempt at providing a universal approach to characterizing efficiency loss in distributed systems, termed smoothness [38], is given in the following theorem.

Theorem 3.1 *Consider any game G where the agents' utility functions satisfy $\sum_{i\in N} U_i(a) \leq W(a)$ for any $a \in \mathcal{A}$. If there exists parameters $\lambda > 0$ and $\mu > -1$ such that for any two action profiles $a, a^* \in \mathcal{A}$*

$$\sum_i U_i(a_i^*, a_{-i}) \geq \lambda \cdot W(a^*) - \mu \cdot W(a), \tag{13}$$

then the efficiency associated with any coarse correlated equilibrium $z^{\text{cce}} \in \Delta(\mathcal{A})$ of G must satisfy

$$\frac{W(z^{\text{cce}})}{W(a^{\text{opt}})} \geq \frac{\lambda}{1+\mu}. \tag{14}$$

We will refer to a game G as $(\lambda, \mu)-$smooth if the game satisfies (13)*.*

Theorem 3.1 demonstrates that the problem of evaluating the price of anarchy in a given game can effectively be recast as a problem of solving for the appropriate coefficients (λ, μ) that satisfsy (13) and maximize $\frac{\lambda}{1+\mu}$. This analysis naturally extends to guarantees over a family of games $\mathcal{G}$,

$$\text{PoA}(\mathcal{G}) \geq \inf_{\lambda>0,\mu>-1} \left\{ \frac{\lambda}{1+\mu} : G \text{ is } (\lambda,\mu) \text{ smooth for all } G \in \mathcal{G} \right\}, \tag{15}$$

where the above expression is referred to as the *robust price of anarchy* [38]. In line with the forthcoming discussion (cf., Section 4.4), implementing a learning rule that leads to the set of coarse correlated equilibrium provides performance guarantees that conform to this robust price of anarchy.

One example of an entire class of games with known price of anarchy bounds is congestion games with affine congestion functions [37] (see also Example 3.1). Another class is valid utility games, introduced in [45], which is very relevant to distributed resource utilization problems. A critical property of valid utility games is a system-level objective that is *submodular*. Submodularity corresponds to a notion of decreasing marginal returns that is a common feature of many objective function in engineering systems. A set-based function $f : 2^N \to \mathbb{R}$ is submodular if for any $S \subseteq T \subseteq N \setminus \{i\}$, we have

$$f(S \cup \{i\}) - f(S) \geq f(T \cup \{i\}) - f(T). \tag{16}$$

In each of these settings, [38] has derived the appropriate smoothness parameters hence providing the price of anarchy guarantees. Accordingly, the resulting price of anarchy holds for coarse correlated equilibrium as well as Nash equilibrium.

Theorem 3.2 ([38, 45]) *Consider any game $G = (N, \{\mathcal{A}_i\}, \{U_i\}, W)$ that satisfies the following three properties:*

(i) The objective function W is submodular;

(ii) For any agent $i \in N$ and any action profile $a \in \mathcal{A}$,

$$U_i(a) \geq W(a) - W(a_i = \emptyset, a_{-i}),$$

where $a_i = \emptyset$ is when player i is removed from the game;

(iii) For any action profile $a \in \mathcal{A}$, the sum of the agents' utilities satisfies

$$\sum_{i \in N} U_i(a) \leq W(a).$$

We will refer to such a game as a valid utility game. Any valid utility game G is smooth with parameters $\lambda = 1$ and $\mu = 1$; hence, the robust price of anarchy is $1/2$ for the class of valid utility games. Accordingly, the efficiency guarantees associated with any coarse correlated equilibrium $z^{\text{cce}} \in \Delta(\mathcal{A})$ in a valid utility game satisfies

$$W(z^{\text{cce}}) \geq \left(\frac{1}{2}\right) W(a^{\text{opt}}).$$

One example of a valid utility game is the vehicle target assignment problem which will be presented in Example 3.3. Here, the system-level objective function is submodular and Condition (ii) in Theorem 3.2 is satisfied by the given design. Further, it is straightforward to verify that Condition (iii) is also satisfied. Accordingly, all coarse correlated equilibria in the designed game for the vehicle target assignment problem are at least 50% efficient. Consequently, the application of learning rules that lead to coarse correlated equilibria (cf., Section 4.4) will lead to a collective behavior in line with these efficiency guarantees.

3.4 Game structures

The two components associated with a game-theoretic design are the agent utility functions, which define an underlying game, and the learning rule. Both components impact various performance objectives associated with the distributed control design. The specification of the agent utility functions directly impacts the price of anarchy, which can be viewed as the efficiency associated with the asymptotic collective behavior. On the other hand, the specification of the learning algorithm dictates the transient behavior in its attempt to drive the collective behavior to the solution concept of interest.

At first glance it appears that the objectives associated with these two components are unrelated to one another. For example, one could employ a design where (i) the agents' utility functions are chosen to optimize the price of anarchy of pure Nash equilibria and (ii) a learning algorithm is employed that drives the collective behavior to a pure Nash equilibrium. Unfortunately, such decoupling is not necessarily possible due to limitations associated with (ii). As previously discussed, there are no "natural dynamics" of the form

$$a_i(t) = \Pi_i\left(a(0), a(1), \ldots, a(t-1); U_i\right) \tag{17}$$

that lead to a (pure or mixed) Nash equilibrium in every game [19], where "natural" refers to uncoupled dynamics (i.e., agents are uninformed of the utility functions of other agents) and rules out behaviors such as exhaustive search or centralized coordination.

Given such impossibility results, it is imperative that the game design component addresses objectives beyond just price of anarchy. In particular, it is of paramount importance that the resulting game has properties that can be exploited in distributed learning. In this section we will review such game structures. Each of these game structures provides a degree of alignment between the agents' utility functions $\{U_i\}$ and a system-level potential function $\phi : \mathcal{A} \to \mathbb{R}$.

The first class of games we introduce, termed potential games [30], exhibits perfect alignment between the agents' utility functions and the potential function ϕ.

Definition 3.4 (Potential Game) *A game G is an (exact) potential game if there exists a potential function $\phi : \mathcal{A} \to \mathbb{R}$ such that for any action profile $a \in \mathcal{A}$, agent $i \in N$, and action choice $a_i' \in \mathcal{A}_i$, we have*

$$U_i(a_i', a_{-i}) - U_i(a_i, a_{-i}) = \phi(a_i', a_{-i}) - \phi(a_i, a_{-i}). \tag{18}$$

Note that any maximizing action profile $a \in \arg\max_{a \in \mathcal{A}} \phi(a)$ is a pure Nash equilibrium; hence, a pure Nash equilibrium is guaranteed to exist in any potential game. Further, as we will see in the forthcoming Section 4, the structure inherent to potential games can be exploited to bypass the impossibility result highlighted above. In other words, there are natural dynamics that lead to a Nash equilibrium in any potential game. We will survey some of these dynamics in Section 4.

There are several variants of potential games that seek to relax the equality given in (18) while preserving the exploitability of the game structure for distributed learning. One of the properties that is commonly exploited in distributed learning is the monotonicity of the potential function along a *better reply path*, which is defined as follows:

Definition 3.5 (Better Reply Path) *A better reply path is a sequence of joint actions $a^1, a^2, \ldots, a^m$ such that for each $k \in \{1, \ldots, m-1\}$ (i) $a^{k+1} = (a_i, a_{-i}^k)$ for some agent $i \in N$ with action $a_i \in \mathcal{A}_i$, $a_i \neq a_i^k$, and (ii) $U_i(a^{k+1}) > U_i(a^k)$.*

Informally, a better reply path is a sequence of joint actions where each subsequent joint action is the result of an advantageous unilateral deviation. In a potential game, the potential function is monotonically increasing along a better reply path. Since the joint action set $\mathcal{A}$ is finite, any better reply will lead to a pure Nash equilibrium in a finite number of iterations. This property is known as the *finite improvement property* [30].[4]

We now introduce the class of weakly acyclic games which relaxes the finite improvement property condition.

[4]Commonly studied variants of exact potential games, e.g., ordinal or weighted potential games, also possess the finite improvement property.

Definition 3.6 (Weakly Acyclic Game) *A game G is weakly acyclic under better replies if for any joint action $a \in \mathcal{A}$ there exists a better reply path from a to a pure Nash equilibrium of G.*

As with potential games, a pure Nash equilibrium is guaranteed to exist in any weakly acyclic game. One advantage of considering broader game classes as a mediating layer for game theoretic control designs is the expansion of available game design methodologies for designing agent utility functions within that class.

3.5 Illustrative examples

At first glance it may appear that the framework of potential games (or weakly acyclic games) is overly restrictive as a framework for the design of networked control systems. Here, we provide three examples of potential games, which illustrates the breadth of the problem domains that can be modeled and analyzed within this framework.

The first example focuses on distributed routing and highlights how a reasonable model of user behavior, i.e., users seeking to minimize their experienced congestion, constitutes a potential game.

Example 3.1 (Distributed Routing) *A routing problem consists of a collection of self-interested agents that need to utilize a common network to satisfy their individual demands. The network is characterized by a collection of edges $E = \{e_1, \dots, e_m\}$ where each edge $e \in E$ is associated with an anonymous congestion function $c_e : \{1, 2, \dots\} \to \mathbb{R}$ that defines the congestion associated with that edge as a function of the number of agents using that edge. That is, $c_e(k)$ is the congestion on edge e when there are $k \geq 1$ agents using that edge. Each agent $i \in N$ is associated with an action set $\mathcal{A}_i \subseteq 2^E$, which satisfies the agent's underlying demands, as well as a local cost function $J_i : \mathcal{A} \to \mathbb{R}$ of the form*

$$J_i(a_i, a_{-i}) = \sum_{e \in a_i} c_e(|a|_e),$$

where $|a|_e = |\{i \in N : e \in a_i\}|$ denotes the number of agents using edge e in the allocation a.[5] *In general, a system designer would like to allocate the agents over the network to minimize the aggregate congestion given by*

$$C(a) = \sum_{e \in E} |a|_e \cdot c_e\left(|a|_e\right).$$

It is well-known that any routing game of the above form, which is commonly referred to as an anonymous congestion game, is a potential game with a potential function $\phi : \mathcal{A} \to \mathbb{R}$ of the form

$$\phi(a) = \sum_{e \in E} \sum_{k=1}^{|a|_e} c_e\left(k\right).$$

[5]Here, we use cost functions $J_i(\cdot)$ instead of utility functions $U_i(\cdot)$ in situation where the agents are minimizers instead of maximizers.

This implies that a pure Nash equilibrium is guaranteed to exist in any anonymous congestion, namely any action profile that minimizes $\phi(a)$. Furthermore, it is often the case that this is unique pure Nash equilibrium with regards to aggregate behavior, i.e., $a^{\mathrm{ne}} \in \arg\min_{a\in\mathcal{A}} \phi(a)$. The fact that the potential function and the system cost are not equivalent, i.e., $\phi(\cdot) \neq C(\cdot)$, can lead to inefficiencies of the resulting Nash equilibria.

The second example focuses on coordination games over graphs. A coordination game is typically posed between two agents where each agent's utility function favors agreement on an action choice over disagreement. However, the agents may have different preferences over which action is agreed upon. Graphical coordination games, or coordination games over graphs, extend such two agent scenarios to n agent scenarios where the underlying graph depicts the population that each agent is seeking to coordinate with.

Example 3.2 (Graphical Coordination Games) *Graphical coordination games characterize a class of strategic interactions where the agents' utility functions are derived from local interactions with neighboring agents. In a graphical coordination game, each agent $i \in N$ is associated with a common action set $\mathcal{A}_i = \bar{\mathcal{A}}$, a neighbor set $\mathcal{N}_i \subseteq N$, and a utility function of the form*

$$U_i(a) = \sum_{j\in\mathcal{N}_i} \mathcal{U}(a_i, a_j) \tag{19}$$

where $\mathcal{U} : \bar{\mathcal{A}} \times \bar{\mathcal{A}} \to \mathbb{R}$ captures the (symmetric) utility associated with a pairwise interaction. As an example, $\mathcal{U}(a_i, a_j)$ designates the payoff for agent i selecting action a_i that results from the interaction with agent j selecting action a_j. Throughout, we adopt the convention that the payoff $\mathcal{U}(a_i, a_j)$ is associated with the player i whose action a_i is the first in the tuple (a_i, a_j).

In the case where the common action set has two actions, i.e., $\bar{\mathcal{A}} = \{x, y\}$, and the interaction graph is undirected, i.e., $j \in \mathcal{N}_i \Leftrightarrow i \in \mathcal{N}_j$, it is straightforward to show that this utility structure gives rise to a potential game with a potential function of the form

$$\phi(a) = \frac{1}{2} \sum_{(i,j)\in E} \phi_{\mathrm{pw}}(a_i, a_j) \tag{20}$$

where $\phi_{\mathrm{pw}} : \bar{\mathcal{A}} \times \bar{\mathcal{A}} \to \mathbb{R}$ is a local potential function. One choice for this local potential function is the following:

$$\begin{aligned}
\phi_{\mathrm{pw}}(x, x) &= 0, \\
\phi_{\mathrm{pw}}(y, x) &= \mathcal{U}(y, x) - \mathcal{U}(x, x), \\
\phi_{\mathrm{pw}}(x, y) &= \mathcal{U}(y, x) - \mathcal{U}(x, x), \\
\phi_{\mathrm{pw}}(y, y) &= (\mathcal{U}(y, y) - \mathcal{U}(x, y)) - (\mathcal{U}(y, x) - \mathcal{U}(x, x)) .
\end{aligned}$$

Observe that any potential function $\phi'_{\mathrm{pw}} = \phi_{\mathrm{pw}} + \alpha$ where $\alpha \in \mathbb{R}$ also leads to a potential function for the given graphical coordination game.

The first two examples show how potential games could naturally emerge in two different types of strategic scenarios. The last example we present focuses on a engineering inspired resource allocation problem, termed the vehicle target assignment problem [32], where the vehicles' utility functions are engineered so that the resulting game is a potential game.

Example 3.3 (Vehicle Target Assignment Problem) *In the well-studied vehicle target assignment problem, there is a finite set of targets $\mathcal{T}$, and each target $t \in \mathcal{T}$ has a relative value of importance $v_t \geq 0$. Further, there are a set of vehicles $N = \{1, 2, \ldots, n\}$ where each vehicle $i \in N$ has an invariant success/destroy probability satisfying $0 \leq p_i \leq 1$ and a set of possible assignment $\mathcal{A}_i \subseteq 2^{\mathcal{T}}$. The goal of vehicle target assignment problem is to find an allocation of vehicles to targets $a \in \mathcal{A}$ to optimize a global objective $W : \mathcal{A} \to \mathbb{R}$ of the form*

$$W(a) = \sum_{t \in \mathcal{T}(a)} v_t \cdot \left(1 - \prod_{j : t \in a_j} (1 - p_j) \right)$$

where $\mathcal{T}(a) \subseteq \mathcal{T}$ denotes the collection of targets that are assigned to at least one agent, i.e., $\mathcal{T}(a) = \cup_{i \in N} a_i$.

Note that in this engineering based application there is no appropriate model of utility functions of the engineered vehicles. Rather, vehicle utility functions are designed with the goal of engineering desirable system-wide behavior. Consider one such design where the utility functions of the vehicles are set as the marginal contribution of the vehicles to the system level objective, i.e., for each vehicle $i \in N$ and allocation $a \in \mathcal{A}$ we have

$$\begin{aligned} U_i(a) &= \sum_{t \in a_i} v_t \cdot \left(1 - \prod_{j : t \in a_j} (1 - p_j) \right) - v_t \cdot \left(1 - \prod_{j \neq i : t \in a_j} (1 - p_j) \right), \\ &= \sum_{t \in a_i} v_t \cdot \left(p_i \prod_{j \neq i : t \in a_j} (1 - p_j) \right). \end{aligned}$$

Given this design of utility functions, it is straightforward to verify that the resulting game is a potential game with potential function $\phi(a) = W(a)$. This immediately implies that any optimal allocation, $a^{\text{opt}} \in \arg\max_{a \in \mathcal{A}} W(a)$, is a pure Nash equilibrium. However, other inefficient Nash equilibria may also exist due to the lack of uniqueness of Nash equilibrium for such scenarios.

3.6 A brief review of game design methodologies

The examples in the previous section illustrate various settings that happen to fall under the special category of potential games. Given that utility function specification is a design degree of freedom in the prescriptive paradigm, it is possible to exploit this degree of freedom to design utility functions to induce desirable structural properties.

There are several objectives that a system designer needs to consider when designing the game that defines the interaction framework of the agents in a multiagent system [26]. These goals could include (i) ensuring the existence of a pure Nash equilibrium, (ii) ensuring that the agents' utility functions fit into the realm of potential games, or (iii) ensuring that the agents' utility functions optimize the price of anarchy / price of stability over an admissible class of agent utility functions, e.g., local utility functions. While recent research has identified the full space of methodologies that guarantee (i) and (ii) [15], the existing research has yet to provide mechanisms for optimizing the price of anarchy.

The following theorem provides one methodology for the design of agent utility functions with guarantees on the resulting game structure [26, 47].

Theorem 3.3 *Consider the class of resource utilization problems defined in Section 2.1 with agent set N, action sets $\{\mathcal{A}_i\}$, and a global objective $W : \mathcal{A} \to \mathbb{R}$. Define the marginal contribution utility function for each agent $i \in N$ and allocation $a \in \mathcal{A}$ as*

$$U_i(a) = \phi(a) - \phi(a_i^b, a_{-i}), \tag{21}$$

where $\phi : \mathcal{A} \to \mathbb{R}$ is any system-level function and $a_i^b \in \mathcal{A}$ is the fixed baseline action for agent i. Then the resulting game $G = \{N, \{\mathcal{A}_i\}, \{U_i\}, W\}$ is an exact potential game where the potential function is ϕ.

A few notes are in order regarding Theorem 3.3. First, the assignment of the agents' utility functions is a byproduct of the chosen system-level design function ϕ and the transformation of ϕ into the agents' utility functions, which is given by (21) and the choice of the baseline action a_i^b for each agent $i \in N$. Observe that the utility design presented in Example 3.3 is precisely the design detailed in Theorem 3.3 where $\phi = W$ and $a_i^b = \emptyset$ for each agent $i \in N$. While a system designer could clearly set $\phi = W$, judging whether this design choice is effective centers on a detailed analysis regarding the properties of the resulting game, e.g., price of anarchy. In fact, recent research has demonstrated that setting $\phi = W$ does not optimize the price of anarchy for a large class of objective functions W. Furthermore, there are also alternative mechanisms for transforming the system-level function ϕ to agent utility functions $\{U_i\}$, as opposed to (21), that provide similar guarantees on the structure of the resulting game, e.g., Shapley and weighted Shapley values [15]. It remains an open question as to what combination, i.e., the transformation and system-level design function that the transformation operates on, gives rise to the optimal utility design.

4 Distributed Learning Rules

We now turn our attention towards distributed learning rules. We can categorize the learning algorithms into the following four areas:

Model-Based Learning. In model-based learning, each agent observes the past behavior of the other agents and uses this information to develop a model for the action choice of the other agents at the ensuing period. Equipped with this model, each agent can then optimally select its actions based on its expected utility at the ensuing time-step. As the play evolves, so do the models of other agents.

Robust Learning. A learning algorithm of the form (2) defines a systematic rule for how individual agents process available information to formulate a decision. Many of the learning algorithms in the existing literature provide guarantees on the asymptotic collective behavior provided that the agents following these rules precisely. Here, we explore the robustness of such learning algorithms, i.e., the asymptotic guarantees on the collective behavior preserved when agents follow variations of the prescribed learning rules stemming from delays in information or asynchronous clock rates.

Equilibrium Selection. The price of anarchy and price of stability are two measures characterizing the inefficiency associated with Nash equilibria. The differences between these two measures follow from the fact that Nash equilibria are often not unique. This lack of uniqueness of Nash equilibria prompts the question of whether deriving distributed learning that favor certain types of Nash equilibria is attainable. Focusing of the framework of potential games, we will review one such algorithm that guarantees the collective behavior will lead to the specific Nash equilibria that optimize the potential function. Note that when utility functions are engineered, as in Example 3.3, a system designer can often ensure that the resulting game is potential game where the action profiles that optimize the potential function coincide with the action profiles that optimize the system-level objective. (We reviewed one such methodology in Section 3.6.)

Universal Learning. All of the above learning algorithms provide asymptotic guarantees when attention is restricted to specific game structures, e.g., potential games or weakly acyclic games. Here, we focus on the derivation of learning algorithms that provide desirable asymptotic guarantees irrespective of the underlying game structure. Recognizing the previously discussed impossibility of natural and universal dynamics leading to Nash equilibria [19], we shift our emphasis from convergence to Nash equilibria to convergence to the set of coarse correlated equilibrium. We introduce one such algorithm, termed *regret matching*, that guarantees convergence to the set of coarse correlated equilibrium irrespective of the underlying game structure. Lastly, we discuss the implications of such learning algorithms on the efficiency of the resulting collective behavior.

We will primarily gauge the quality of a learning algorithm by characterizing the collective behavior as time $t \to \infty$. When merging a particular distributed learning algorithm with an underlying game, the efficiency analysis techniques presented in Section 3.2 can then be employed

to characterize the quality of the emergent collective behavior with regards to a given system-level objective.

4.1 Model-based learning

The central challenge in distributed learning is dealing with the fact that each agent's environment is inherently non-stationary in that the environment from the perspective of any agent consists of the behaviors of other agents, which are evolving. A common approach in distributed learning is to have agents make decisions in a myopic fashion, thereby neglecting the ramifications of an agent's current decision on the future behavior of the other agents. In this section we review two learning algorithms of this form that we categorize as model-based learning algorithms. In model-based learning, each agent observes the past behavior of the other agents and utilizes this information to develop a behavioral model of the other agents. Equipped with this behavioral model, each agent then performs a myopic best response seeking to optimize its expected utility. It is important to stress here that the goal is not to accurately model the behavior of the other agents in ensuing period. Rather, the goal is to derive systematic agent responses that will guide the collective behavior to a desired equilibrium.

4.1.1 Fictitious play

One of the most well-studied algorithms of this form is fictitious play [30]. Here, each agent uses the empirical frequency of past play as a model for the behavior of the other agents at the ensuing time-step. To that end, define the empirical frequency of play for each player $i \in N$ at time $t \in \{1, 2, \dots\}$ as $q_i(t) = \{q_i^{a_i}\}_{a_i \in \mathcal{A}_i} \in \Delta(\mathcal{A}_i)$ where

$$q_i^{a_i}(t) = \frac{1}{t} \sum_{\tau=0}^{t-1} I\{a_i(\tau) = a_i\}, \tag{22}$$

and $I\{\cdot\}$ is the usual indicator function. At time t, each agent seeks to myopically maximize its expected utility given the belief that each agent $j \neq i$ will select its action independently according to a strategy $q_j(t)$. This update rule takes on the form

$$a_i(t) \in \arg\max_{a_i \in \mathcal{A}_i} \sum_{a_{-i} \in \mathcal{A}_{-i}} U_i(a_i, a_{-i}) \prod_{j \neq i} q_j^{a_j}(t). \tag{23}$$

The following theorem provided in [30] characterizes the long run behavior associated with fictitious play in potential games.

Theorem 4.1 *Consider any exact potential game G. If all players follow the fictitious play learning rule, then the players' empirical frequencies of play $q_1(t), \dots, q_n(t)$ will converge to a Nash equilibrium of the game G.*

The fictitious play learning rule provides a mechanism to guide individual agent behavior in distributed control systems when the agents (i) can observe the previous action choices of the other agents in the system and (ii) have access to the structural form of their utility function. Further, fictitious play provides provable guarantees on the emergent collective behavior provided that the system can be modeled by an exact potential game. For example, consider the distributed routing problem given in Example 3.1 which can be modeled as a potential game irrespective of the number of agents, the number of edges, the topology of the network, or the edge-specific latency functions. Regardless of the structure of the routing problem, the fictitious play algorithm can be employed to drive the collective system behavior to a Nash equilibrium.

While the asymptotic guarantees associate with fictitious play in distributed routing problems is appealing, the implementation of fictitious play in such settings is problematic. First, each agent must be able to observe the specific behavior of all other agents in the network each period. Second, the choice of each agent at any time given in (23) requires (i) knowledge of the structural form of the agent's utility function and (ii) computing an expectation of its utility function, which involves evaluating a weighted summation over $|\mathcal{A}_{-i}|$ terms. In large-scale systems, such as distributed routing, each of these requirements could be prohibitive. Accordingly, research has attempted to alter the fictitious play algorithm to minimize such requirements while preserving the desirable asymptotic guarantees.

4.1.2 Variants of fictitious play

One of the first attempts to relax the implementation requirements associated with fictitious play centered on the computation of a best response given in (23). In [20], the authors proposed a sample-based approach for computing this best response, where each agent randomly drew samples of the other agents' behavior using their empirical frequencies of play and evaluated the average performance of each possible routing decision against the drawn samples. The choice with the best average performance was then substituted for the choice that maximized the agent's expected utility in (23), and the process was repeated. While simulations demonstrated reasonable performance even for limited samples, unfortunately preserving the theoretical asymptotic guarantees associated with fictitious play required that the number of samples drawn each period grew prohibitively large.

A second variant of fictitious play focused on the underlying asymptotic guarantees given in Theorem 4.1, which state that the empirical frequency of play converges to a Nash equilibrium. It is important to highlight this does not imply that the day-to-day behavior of the agents converges to a Nash equilibrium, e.g., the agents' day-to-day behavior could oscillate yielding a frequency of play consistent with a Nash equilibrium. Furthermore, the cumulative payoff may be less than the payoff associated with the limiting empirical frequencies. With this issue in mind, [12] introduced a variant of fictitious play that assures a specific payoff consistency property against arbitrary

environments, i.e., not just when other agents employ fictitious play.

4.1.3 Joint strategy ficitious play with inertia

The focus in model-based learning is not whether such models accurately reflect the behavior of the other agents. Rather, the focus is on whether systematic responses to potentially inaccurate models can guide the collective behavior to a desired equilibrium. The behavioral models used in fictitious play, i.e., assuming each agent will play a strategy independently according the the agent's empirical frequency of play, provided nice asymptotic guarantees but was prohibitive from an implementations perspective. Here, we consider a variant of fictitious play, termed joint strategy fictitious play (JSFP), which provides similar asymptotic guarantees while alleviating many of the computational and observational challenges associated with fictitious play [24]. The main difference between fictitious play and joint strategy fictitious play resides in the behavioral model of the other agents. In joint strategy fictitious play, each agent presumes that the other players will select an action collectively in accordance with their empirical frequency of their past joint play. In two player games, fictitious play and joint strategy fictitious play are equivalent. However, the learning algorithms yield fundamentally different behavior beyond two player games.

We begin by defining the average hypothetical utility of agent $i \in N$ for each action $a_i \in \mathcal{A}$ as

$$\bar{U}_i^{a_i}(t) = \frac{1}{t}\sum_{\tau=0}^{t-1} U_i(a_i, a_{-i}(\tau)) = \frac{t-1}{t}\bar{U}_i^{a_i}(t-1) + \frac{1}{t}U_i(a_i, a_{-i}(t-1)). \tag{24}$$

Note that this average hypothetical utility is computed under the belief that the action choices of the other agents remain unchanged. Now, consider the decision making rule where each agent $i \in N$ independently selects its action probabilistically according to the rule

$$a_i(t) = \begin{cases} \arg\max_{a_i \in \mathcal{A}_i} \bar{U}_i^{a_i}(t) & \text{with probability} \quad (1-\epsilon), \\ a_i(t-1) & \text{with probability} \quad \epsilon, \end{cases} \tag{25}$$

where $\epsilon > 0$ is referred to as the agent's inertia or probabilistic reluctance to change actions. Hence, with high probability, i.e., probability $(1-\epsilon)$, each agent selects the action that maximizes the agent's hypothetic utility.

The following theorem from [24] characterizes the long run behavior of joint strategy fictitious play in potential games.

Theorem 4.2 *Consider any exact potential game G. If all players following the learning algorithm joint strategy fictitious play defined above, then the joint action profile will converge almost surely to a pure Nash equilibrium of the game G.*

Hence, JFSP with inertia provides similar asymptotic guarantees to Fictitious Play while minimizing the computational and observational burden on the agents. The name "joint strategy fictitious play" is derived from the fact that maximizing the average hypothetical utility in (24) is

equivalent to maximizing an expected utility under the belief that all agents will play collectively according to the empirical frequency of their past joint play.

4.2 Robust distributed learning

Both Fictitious Play and Joint Strategy Fictitious Play are intricate decision-making rules that provide guarantees regarding the emergent collective behavior. A natural question that emerges when considering the practicality of such rules for control of networked systems is the robustness of these guarantees to common implementation issues including asynchronous clocks, noisy payoffs, delays in information, among others. This section highlights that the framework of potential games, or more generally weakly acyclic games, is inherently robust to such issues.

We review the result in [49] that deals with this exact issue. In particular, [49] demonstrates the robustness of weakly acyclic games by identifying a broad family of learning rules, termed *finite memory better response processes*, with the property that any rule within this family will provably guide the collective behavior to a pure Nash equilibrium in any weakly acyclic game.

A finite memory better reply process with inertia is any learning algorithm of the following form: at each time t, each agent selects its action independently according to the rule

$$a_i(t) = \begin{cases} B_i^m(h^m(t)) & \text{with probability} \quad (1-\epsilon), \\ a_i(t-1) & \text{with probability} \quad \epsilon, \end{cases} \tag{26}$$

where $m \geq 1$ is the size of the agent's memory, $\epsilon > 0$ is the agent's inertia, $h^m(t) = \{a(t-1), a(t-2), \ldots, a(t-m)\}$ denotes the previous m action profiles, and $B_i^m : \mathcal{A}^m \to \Delta(\mathcal{A}_i)$ is the finite memory better reply process.[6] A finite memory better reply process $B_i^m(\cdot)$ can be any process that satisfies the following properties:

- If the history is saturated, i.e., $h^m(t) = \{\bar{a}, \bar{a}, \ldots, \bar{a}, \bar{a}\}$ for some action profile $\bar{a} \in \mathcal{A}$, then the strategy $p_i = B_i^m(h^m(t))$ must satisfy
  - If $\bar{a}_i \in \arg\max_{a_i \in \mathcal{A}_i} U_i(a_i, \bar{a}_{-i})$, then $p_i^{\bar{a}_i} = 1$ and $p_i^{a_i} = 0$ for all $a_i \neq \bar{a}_i$.
  - Otherwise, if $\bar{a}_i \notin \arg\max_{a_i \in \mathcal{A}_i} U_i(a_i, \bar{a}_{-i})$, then $p_i^{a_i} > 0$ if and only if $U_i(a_i, \bar{a}_{-i}) \geq U_i(\bar{a}_i, \bar{a}_{-i})$.
- If the history is not saturated, then the strategy $p_i = B_i^m(h^m(t))$ can be any probability distribution in $\Delta(\mathcal{A}_i)$.[7]

[6] We write $a_i(t) = B_i^m(h^m(t))$ with the understanding that this implies that the action profile $a_i(t)$ is chosen randomly according to the probability distribution specified by $B_i^m(h^m(t))$.

[7] The actually definition of a finite-better reply process considered in [49] puts a further condition on the structure of B_i^m under the case where the memory is not saturated, i.e., the strategy assigns positive probability to any action with strictly positive regret. However, an identical proof holds for any B_i^m that satisfies the weaker conditions set forth in this chapter.

In summary, the only constraint imposed on a finite memory better reply process is that a better reply to saturated memory $\{a, \ldots, a\}$ is consistent with a better reply to the single action profile a.

The following theorem from [49] (Theorem 6.2) demonstrates the inherent robustness of weakly acyclic games.

Theorem 4.3 *Consider any weakly acyclic game G. If all agents follow a finite memory better reply process defined above, then the joint action profile will converge almost surely to a pure Nash equilibrium of the game G.*

One can view this result from two perspectives. The first perspective is that the system-designer has extreme flexibility in designing learning rules for weakly acyclic games that guarantee the agents' collective behavior will converge to a pure Nash equilibrium. The second perspective is that perturbations of a nominal learning rule, e.g., agents updating asynchronously or responding to delayed or inaccurate histories, will also satisfy the conditions above and ultimately lead behavior to a Nash equilibrium as well. These perspectives provide the basis for our claim of robust distributed learning.

4.3 Equilibrium selection in potential games

The preceding discussion focused largely on algorithms that ensured the emergent collective behavior constitutes a (pure) Nash equilibrium. In the case where there are multiple Nash equilibria, these algorithms provide no guarantees on which equilibrium is likely to emerge. Accordingly, characterizing the efficiency associated with the emergent collective behavior is equivalent to characterizing the efficiency associated with the worst performing Nash equilibrium, i.e., the price of anarchy.

In this section we explore the notion of equilibrium selection in distributed learning. That is, are there classes of distributed learning algorithms that converge to specific classes of equilibria? One motivation for pursuing such developments is the marginal cost utility, given in Theorem 3.3, which ensures that the optimal allocation is a Nash equilibrium, i.e., the price of stability is 1. Accordingly, the focus of this section will be on learning dynamics that converge to the most efficient action profile in potential games, i.e., the action profile that maximizes the potential function.

4.3.1 Log linear learning

We begin this subsection by describing a simple asynchronous best reply process, where each agent chooses a best reply when given the opportunity to revise its strategy. Let $a(t)$ represent the action profile at time t. The action profile at time $t+1$ is chosen as follows:

(i) An agent $i \in N$ is randomly picked to update its action according to a uniform distribution.

(ii) Agent i selects an action that is a best response to the action profile played by the other agents in the previous period, i.e.,

$$a_i(t+1) \in \arg\max_{a_i \in \mathcal{A}_i} U_i(a_i, a_{-i}(t)). \tag{27}$$

(iii) All other agents $j \neq i$ play their previous actions, i.e., $a_{-i}(t+1) = a_{-i}(t)$.

(iv) The process is then repeated.

It is straightforward to see that the above process will converge almost surely to a pure Nash equilibrium in any potential game by observing that $\phi(a(t+1)) \geq \phi(a(t))$ for all times t. Accordingly, the efficiency guarantees associated with the application of this algorithm to a potential game are in line with the price of anarchy of the game.

Here, a slight modification, or perturbation, is introduced of the above best reply dynamics that ensures that the resulting behavior leads to the pure Nash equilibrium that optimizes the potential function, i.e., $a^{\text{opt}} \in \arg\max_{a \in \mathcal{A}} \phi(a)$. The algorithm, known as log-linear learning or the logit response dynamics [1,6,7,25,48], follows the best reply process highlighted above where step (ii) is replaced by a noisy best response. More formally, step (ii) is now of the form:

(ii) Agent i selects an action $a_i(t+1)$ according to a probability distribution $p_i(t) = \{p_i^{a_i}(t)\}_{a_i \in \mathcal{A}_i} \in \Delta(\mathcal{A}_i)$ that is of the form

$$p_i^{a_i}(t) = \frac{e^{(1/T) \cdot U_i(a_i, a_{-i}(t))}}{\sum_{\tilde{a}_i \in \mathcal{A}_i} e^{(1/T) \cdot U_i(\tilde{a}_i, a_{-i}(t))}}, \tag{28}$$

where the parameter $T > 0$ is referred to as the temperature.

A few remarks are in order regarding the update protocol specified in (28). First, when $T \to \infty$, the agent's strategy is effectively a uniform distribution over the agent's action set. Second, when $T \to 0^+$, the agent's strategy is effectively the best response strategy given in (27). Lastly, we present this algorihtm (and the forthcoming Binary Log-Linear Learning) with regards to a fixed temperature parameter that is common to all agents. However, there are variations of this algorithm which allow for annealing of this temperature parameter that preserve the resulting asymptotic guarantees, e.g., [50].

The following theorem establishes the asymptotic guarantees associated with the learning algorithm log linear learning in potential games [6, 7, 48].

Theorem 4.4 *Consider any potential game G with potential function ϕ. If all players follow the learning algorithm log-linear learning with temperature $T > 0$, then the resulting process has a unique stationary distribution $\pi = \{\pi^a\}_{a \in \mathcal{A}} \in \Delta(\mathcal{A})$ of the form*

$$\pi^a = \frac{e^{(1/T) \cdot \phi(a)}}{\sum_{\tilde{a} \in \mathcal{A}} e^{(1/T) \cdot \phi(\tilde{a})}}. \tag{29}$$

The stationary distribution of the process given in (29) follows the same intuition as presented for the update protocol in (28). That is, when $T \to \infty$ the stationary distribution is effectively a uniform distribution over the joint action set $\mathcal{A}$. However, when $T \to 0^+$, all of the weight of the stationary distribution is concentrated on the action profiles that maximize the potential function ϕ. The above stationary distribution provides an accurate assessment of the resulting assymptotic behavior due to the fact that the log-linear learning process is both irreducible and aperiodic, hence (29) is the unique stationary distribution.

Merging log-linear learning with the marginal contribution utility design given in Theorem 3.3 leads to the following corollary.

Corollary 4.1 *Consider the class of resource allocation problems defined in Section 2.1 with agent set N, action sets $\{\mathcal{A}_i\}$, and a global objective $W : \mathcal{A} \to \mathbb{R}$. Consider the following game theoretic control design:*

(i) Assign each agent a utility function that captures the agent's marginal contribution to the global objective, i.e.,

$$U_i(a) = W(a) - W(a_i^b, a_{-i}), \tag{30}$$

where $a_i^b \in \mathcal{A}_i$ is any fixed baseline action for agent i.

(ii) Each agent follows the log-linear learning rule with temperature parameter $T > 0$.

Then the resulting process has a unique stationary distribution $\pi(T) = \{\pi^a(T)\}_{a\in\mathcal{A}} \in \Delta(\mathcal{A})$ of the form

$$\pi^a(T) = \frac{e^{(1/T)\cdot W(a)}}{\sum_{\tilde{a}\in\mathcal{A}} e^{(1/T)\cdot W(\tilde{a})}}. \tag{31}$$

Observe that this design rule ensures that the resulting asymptotic behavior will be concentrated around the allocations that maximize the global objective W. This fact has made this design methodology an attractive option for several domains including wind farms, sensor networks, coordination of unmanned vehicles, among others.

4.3.2 Binary log-linear learning

The framework of log-linear learning imposes a fairly rigid structure on the update process of the agents. This structure mandates that (i) only one agent updates the action choice at any iteration, (ii) agents are able to select any action in their action set, and (iii) agents are able to assess their utility for any alternative action choice given the observed behavior of the other agents. In general, [1] demonstrates that relaxing these structures arbitrarily can significantly alter the resulting asymptotic guarantees associated with log-linear learning. However, in each of the scenarios variations of log-linear learning can preserve the asymptotic guarantees while making the structure more amenable to engineering systems [25].

Here, we present a variation of log-linear learning that preserves the asymptotic guarantees associated with log-linear learning while accommodating restrictions in the agents' action sets. By restrictions in action sets, we mean that the set of actions available to a given agent is dependent on the agent's current action choice, and we express this dependence by the function $R_i : \mathcal{A}_i \to 2^{\mathcal{A}_i}$ where $a_i \in R_i(a_i)$ for all a_i. That is, if the choice of agent i at time t is $a_i(t)$, then the ensuing choice of the agent $a_i(t+1)$ must be contained in the set $R_i(a_i(t))$. Throughout this section, we consider restricted action sets that satisfy two properties:

(i) *Reversibility*: Let a_i, a_i' be any two action choices in $\mathcal{A}_i$. If $a_i' \in R_i(a_i)$ then $a_i \in R_i(a_i')$.

(ii) *Completeness*: Let a_i, a_i' be any two action choices in $\mathcal{A}_i$. There exists a sequence of actions $a_i = a_i^0, a_i^1, \ldots, a_i^m = a_i'$ with the property that $a_i^{k+1} \in R_i(a_i^k)$ for all $k \in \{0, \ldots, m-1\}$.

One motivation for considering restricted action sets of the above form is when the individual agents have mobility limitations, e.g., mobile sensor networks.

Note that the log-linear learning update rule given in (28) has full support on the agent's action set $\mathcal{A}_i$ thereby disqualifying this algorithm for use in the case where there are restrictions in action sets. Here, we seek to address the question of how to alter the algorithm so as to preserve the asymptotic guarantees, i.e., convergence in the stationary distribution to the action profile that maximizes the potential function. One natural variation would be to replace (28) with a strategy of the form: for any $a_i \in R_i(a_i(t))$

$$p_i^{a_i}(t) = \frac{e^{(1/T)\cdot U_i(a_i, a_{-i}(t))}}{\sum_{\tilde{a}_i \in R_i(a_i(t))} e^{(1/T)\cdot U_i(\tilde{a}_i, a_{-i}(t))}}, \tag{32}$$

and $p_i^{a_i}(t) = 0$ for any $a_i \notin R_i(a_i(t))$. However, such modifications can have drastic consequences on the resulting asymptotic guarantees. In fact, such a rule is not even able to guarantee that the potential function maximizer is the support of the limiting distribution as $T \to 0^+$ [25].

Here, we introduce a variation of log-linear learning, termed binary log-linear learning with restricted action sets [25], that preserves these asymptotic guarantees. Binary log-linear learning follows the same setup as log-linear learning where step (ii) is now of the form:

(ii) Agent i selects a trial action $a_i^{\mathrm{t}} \in R_i(a_i(t))$ according to any distribution with full support on the set $R_i(a_i(t))$. Conditioned on the selection of this trial action, the agent selects the action $a_i(t+1)$ according to a probability distribution $p_i(t) = \{p_i^{a_i}(t)\}_{a_i \in \mathcal{A}_i} \in \Delta(\mathcal{A}_i)$ of the form

$$p_i^{a_i}(t) = \begin{cases} a_i(t-1) & \text{with probability } \frac{e^{(1/T)\cdot U_i(a(t))}}{e^{(1/T)\cdot U_i(a(t))} + e^{(1/T)\cdot U_i(a_i^{\mathrm{t}}, a_{-i}(t))}}, \\ a_i^{\mathrm{t}} & \text{with probability } \frac{e^{(1/T)\cdot U_i(a_i^{\mathrm{t}}, a_{-i}(t))}}{e^{(1/T)\cdot U_i(a(t))} + e^{(1/T)\cdot U_i(a_i^{\mathrm{t}}, a_{-i}(t))}}, \end{cases} \tag{33}$$

where $p_i^{a_i}(t) = 0$ for any $a_i \notin \{a_i(t-1), a_i^{\mathrm{t}}\}$.

Much like log-linear learning, for any temperature $T > 0$ binary log-linear learning can be modeled by an irreducible and aperiodic Markov chain over the state space $\mathcal{A}$; hence, there is a unique stationary distribution which we denote by $\pi(T) = \{\pi^a(T)\}_{a\in\mathcal{A}}$. While log-linear learning provides the explicit form of the stationary distribution $\pi(T)$, the value of log-linear learning centers on the fact that the support of the limiting distribution is precisely the set of potential function maximizers, i.e.,

$$\lim_{T\to 0^+} \pi^a(T) > 0 \;\Leftrightarrow\; a \in \arg\max_{a\in\mathcal{A}} \phi(a)$$

The action profiles contained in the support of the limiting distribution are termed the *stochastically stable states*. Accordingly, log-linear learning ensures that an action profile is stochastically stable if and only if it is a potential function maximizer.

The following theorem from [25] characterizes the long run behavior of binary log-linear learning.

Theorem 4.5 *Consider any potential game G with potential function ϕ. If all players follow the learning algorithm binary log-linear learning with restricted action set and temperature $T > 0$, then an action profile is stochastically stable if and only if it is a potential function maximizer.*

This theorem demonstrates that a system-designer can effectively deal with restrictions in action sets by appropriately modifying the learning rule. However, a consequence of this is that we are no longer able to provide a precise characterization of the stationary distribution as a function of the temperature parameter T. Unlike log-linear learning, binary log-linear learning applied to such a game does not satisfy reversibility unless there are additional constraints imposed on the agents' restricted action sets, i.e., $|R_i(a_i)| = |R_i(a_i')|$ for all $i \in N$ and $a_i, a_i' \in \mathcal{A}_i$. Hence, in this theorem we forgo a precise analysis of the stationary distribution in favor of a coarse analysis of the stationary distribution that demonstrates roughly the same asymptotic guarantees.

4.3.3 Beyond asymptotic guarantees

In potential games, both log-linear learning and binary log-linear learning ensure that the resulting collective behavior can be characterized by the action profiles that maximize the potential function when the temperature $T \to 0^+$. Here, we focus on the question of characterizing the convergence rates of this process. That is, how long does it take for the collective behavior to reach these desired equilibrium points.

Several negative results have emerged regarding the convergence rates of such algorithms [11, 17, 39]. In particular, [17, 39] demonstrates that in general the amount of time that it may take to reach such an equilibrium could be exponential in both the number of agents and the cardinality of their action sets. Accordingly, research has shifted to identifying whether there are classes of games and variants of the above dynamics that exhibit more desirable guarantees on the convergence rates.

The following briefly highlights three domains where such positive results exist.

Symmetric Parallel Congestion Games. Consider the class of congestion games introduced in Example 3.1. A symmetric parallel congestion game is a congestion game where each agent $i \in N$ has an action set $\mathcal{A}_i = \mathcal{R}$; that is, any agent can choose any single edge from the the set of available roads $\mathcal{R}$. In [39], the authors demonstrate that the mixing times associated with log-linear learning could grow exponentially with regards to the number of players n even in such limited scenarios. However, the authors introduce a variant of log-linear learning, which effectively replaces Step (i) of the algorithm (pick an updating player uniformly) with a new procedure which biases the selection rate of certain agents based on the current action profile a. This modification of log-linear learning provides similar asymptotic guarantees with far superior transient guarantees. In particular, this variant of log-linear learning provides a mixing time that is nearly linear in the number of agents for this class of congestion games.

Semi-Anonymous Potential Games. In symmetric parallel congestion games all of the agents are anonymous (or identical) with regards to their impact on the potential function and their available action choices. More formally, we will call two agents $i, j \in N$ anonymous in a potential game if (i) $\mathcal{A}_i = \mathcal{A}_j$ and (ii) $\phi(a) = \phi(a')$ for any action profiles a, a' where $a'_i = a_j$, $a'_j = a_i$, and $a'_k = a_k$ for all $k \neq i, j$. Accordingly, let $C_1, \ldots C_m$ represent a minimal partition of N such that each set of agents C_k, $k \in \{1, \ldots, m\}$, is anonymous with respect to one another, i.e., any agents $i, j \in C_k$ are anonymous with respect to each other. The authors in [8] derive a variant of log-linear learning algorithm, similar to the algorithm for symmetric parallel congestion games in [39] highlighted above, that provides mixing times that are nearly linear in the number of agents n, but exponential in the number of indistinguishable groups of agents, m.

Graphical Coordination Games. Consider the family of graphical coordination games introduced in Example 3.2. In [31], the authors study the mixing times associated with log-linear learning in a special class of graphical coordination games where the underlying pairwise utility function constitutes a 2×2 symmetric utility function. In particular, the authors demonstrate that the structure of the network, in particular the min-cut of graph, is intimately related to the underlying speed of convergence. A consequence of this characterization is that the mixing times associated with log-linear learning is effectively linear in the number of agents when the underlying graph is sparse.

4.4 Universal learning

The preceding sections presented algorithms that guarantee convergence to Nash equilibria (or potential function maximizers) for specific game structures, e.g., potential games or weakly acyclic games. Here, we focus on the question of whether there are universal algorithms that provide con-

vergence to an equilibrium irrespective of the underlying game structure. With regards to Nash equilibrium, it turns out that such an objective is impossible as demonstrated by [19] which establishes that no natural dynamics converge to a Nash equilibrium in all games. Here, the phrase natural seeks to disqualify dynamics that can be thought of as an exhaustive search or utilizing a central coordinator. Nonetheless, by relaxing our equilibrium requirements focus from Nash equilibria to coarse correlated equilibria, such universal algorithms do exists. In the following, we survey the most well-known algorithm that achieves this objective and discuss its implications on the efficiency of this broader class of equilibria.

In this section we present an algorithm proposed in [18], referred to as *regret matching*, that guarantees convergence to the set of coarse correlated equilibrium. The informational demands and computations associated with the decision-making rule regret matching is very similar to those presented for the algorithm Joint Strategy Fictitious Play with inertia highlighted above. The main driver for each agent's strategy selection is the regret associated with each of its actions. For any time $t \in \{1, 2 \dots\}$, the regret of agent $i \in N$ for action $a_i \in \mathcal{A}_i$ is defined as

$$R_i^{a_i}(t) = \bar{U}_i^{a_i}(t) - \bar{U}_i(t), \tag{34}$$

where $\bar{U}_i(t) = \frac{1}{t}\sum_{\tau=0}^{t-1} U_i(a(\tau))$ is the average utility received by agent i up to time t and $\bar{U}_i^{a_i}(t) = \frac{1}{t}\sum_{\tau=0}^{t-1} U_i(a_i, a_{-i}(\tau))$ is the average utility that would have been received by agent i up to time t if the agent committed to action a_i all time steps and the behavior of the other agents were unchanged. Observe that $R_i^{a_i}(t) > 0$ implies that agent i could have received a higher average utility if the agent had committed to the action a_i for all previous time-steps and the action choices of the other agents was unchanged.

The regret matching algorithm proceeds as follows: at each time $t \in \{1, 2, \dots\}$, each agent $i \in N$ independently selects its action according to the strategy $p_i(t) \in \Delta(\mathcal{A}_i)$ of the form

$$p_i^{a_i}(t) = \frac{[R_i^{a_i}(t)]_+}{\sum_{\tilde{a}_i \in \mathcal{A}_i} \left[R_i^{\tilde{a}_i}(t)\right]_+} \tag{35}$$

where $[\cdot]_+$ denotes the projection to the positive orthant, i.e., $[x]_+ = \max\{x, 0\}$.

The following theorem characterizes the long run behavior of regret matching in any game.

Theorem 4.6 *Consider any finite game G. If all players follow the learning algorithm regret matching defined above, then the positive regret for any agent $i \in N$ and action $a_i \in \mathcal{A}_i$ asymptotically vanishes, i.e.,*

$$\lim_{t\to\infty} [R_i^{a_i}(t)]_+ = 0. \tag{36}$$

Alternatively, the empirical frequency of play converges to the set of coarse correlated equilibria.

The connection between the condition (36) and the definition of coarse correlated equilibria stems from the fact that an agent's regret and average utility can also be computed using the empirical frequency of play $z(t) = \{z^a(t)\}_{a\in\mathcal{A}}$ where

$$z^a(t) = \frac{1}{t}\sum_{\tau=0}^{t-1} I\{a(\tau) = a\}. \tag{37}$$

In particular, at any time $t \in \{1, 2, \dots\}$ we have that

$$U_i(z(t)) = \sum_{a\in\mathcal{A}} U_i(a) z^a(t) = \bar{U}_i(t). \tag{38}$$

Further, defining the marginal distribution of the empirical frequency of play of all agents $j \neq i$ as $z_{-i}^{a_{-i}}(t) = \sum_{a_i\in\mathcal{A}_i} z^{(a_i,a_{-i})}(t)$, we have

$$U_i(a_i, z_{-i}(t)) = \sum_{a_{-i}\in\mathcal{A}_{-i}} U_i(a_i, a_{-i}) z_{-i}^{a_{-i}}(t) = \bar{U}_i^{a_i}(t). \tag{39}$$

Accordingly, if a sequence of play $a(0)$, $a(1)$, $\dots$, $a(t-1)$, satisfies (36), then we know that the empirical frequency of play $z(t)$ satisfies

$$\lim_{t\to\infty} \{U_i(z(t)) - U_i(a_i, z_{-i}(t))\} \leq 0, \;\; \forall i \in N, \; a_i \in \mathcal{A}_i. \tag{40}$$

Hence, the limiting empirical frequency of play $z(t)$ is contained in the set of coarse correlated equilibria. Note that the convergence highlighted above does not state that the empirical frequency of play will converge to any specific correlated equilibrium; rather, it merely states that the empirical frequency of play will approach the set of coarse correlated equilibria.

Lastly, we presented a version of regret matching that provides convergence to the set of coarse correlated equilibria. Variants of the presented regret matching could also ensure convergence to the set of correlated equilibrium, which is a more rigid solution concept than presented in Definition 3.3. We direct the readers to [18, 49] for the details associated with this variation.

4.4.1 Equilibrium selection of correlated equilibrium

The set of correlated equilibra is much larger than the set of Nash equilibria and can potentially be exploited to provide systems with better performance guarantees. One example of such a system is the Shapley game, which is a two-player game with utility functions of the form

		Agent 2		
		A	B	C
Agent 1	A	$0, 0$	$0, 1$	$1, 0$
	B	$1, 0$	$0, 0$	$0, 1$
	C	$0, 1$	$1, 0$	$0, 0$

Payoff Matrix

There are no pure Nash equilibria in this game and the unique (mixed) Nash equilibrium is when each agent i employs a strategy $p_i = (1/3, 1/3, 1/3)$, which yields an expected payoff of $1/3$ to each agent. However, there is also a coarse correlated equilibrium where the distribution z has a value $1/6$ on each of the six joint actions where some agent receives non-zero payoff; z has a value 0 for the other three joint actions. This coarse correlated equilibrium yields an expected utility of $1/2$ to each agent and is clearly more desirable. One could easily imagine other scenarios, e.g., team versus team games, where specific coarse correlated equilibrium could provide significant performance improvements over any Nash equilibrium.

The problem with regret matching for exploiting this potential opportunity is that behavior is not guaranteed to converge to any specific coarse correlated equilibrium. Accordingly, the efficiency guarantees associated with coarse correlated equilibria cannot be better than the efficiency bounds associated with pure Nash equilibria and can often be quite worse. With this issue in mind, recent work in [9, 27] has sought to develop learning algorithms that converge to the efficient coarse correlated equilibrium, where efficiency is measured by the sum of the agents' expected utilities. Here, the algorithm introduced in [27] ensures that the empirical frequency of play will converge to the most efficient coarse correlated equilibrium while [9] provides an algorithm that guarantees that the day-to-day behavior of the agents will converge to the most efficient correlated equilibrium. Both of these algorithms view convergence in a stochastic stability sense.

The motivation for these developments centers on the fact that joint randomization, which can potentially be characterized by correlated equilibria, can be key to providing desirable system-level behavior. One example of such a system is a peer-to-peer file sharing system where users engage in interactions with other users to transfer files of interest and satisfy demands [46]. Here, [46] demonstrates that the optimal system performance is actually characterized by the most efficient correlated equilibrium as defined above. Another example of such a system is the problem of access control for wireless communications, where there are a collection of mobile terminals that compete over access to a common channel [2]. Optimizing system-throughput requires a level of correlation between the transmission strategies of the mobiles so as to minimize the chance of simultaneous transmissions and failures. The authors in [2] study the efficiency of correlated equilibria in this context. Identifying the role of correlated equilibrium equilibrium (and learning strategies for attaining specific correlated equilibrium) warrants further research attention.

5 Final Remarks

The goal of this chapter has been to highlight a potential role of game theoretic learning in the design of networked control systems. We reviewed several classes of learning algorithms accentuating their performance guarantees and reliance on game structures.

It is important to re-emphasize that game theoretic learning represents just a single dimension

of a game theoretic control design. The other dimension centers on the assignment of objective functions to the individual agents. The structure of these agent objective functions not only dictate convergence guarantees associated with various game theoretic learning algorithms, but can also be exploited to characterize the efficiency of the resulting behavior. To that end, consider the assignment of agent objective functions that yields a potential game and has a given price of anarchy. Marrying this design with a learning algorithm that guarantees convergence to a pure Nash equilibrium in potential games yields a game theoretic control design that ensures that the collective behavior will converge to a specific allocation (in particular a Nash equilbrium associated with the designed agent objective functions) and the efficiency of this allocation will be in line with the given price of anarchy.

Taking full advantage of this game theoretic approach requires assigning agent objective functions that yield a potential game and optimize the price of anarchy over all such objective functions. Unfortunately, the existing literature provides no mechanism for accomplishing this goal as utility design for distributed engineering systems is currently not well understood. A reason for this gap is that agent objective function are traditionally modeled to reflect agent preferences in a given social system, e.g., a reasonable objective for drivers on a transportation network is minimizing experienced congestion. Hence, efficiency measures in games, such as the price of anarchy, are traditionally viewed from an analysis perspective with virtually no design component. Reversing this trend and deriving systematic methodologies for utility design in multiagent systems represents a significant opportunity for game theoretic control moving forward.

References

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

[2] E. Altman, N. Bonneau, and M. Debbah. Correlated equilibrium in access control for wireless communications. In *5th International Conference on Networking*, 2006.

[3] Y Babichenko. Completely uncoupled dynamics and Nash equilibria. *Games and Economic Behavior*, 76:1–14, 2012.

[4] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J.N. Tsitsiklis. Convergence in multiagent coordination, consensus, and flocking. In *IEEE Conf. on Decision and Control*, 2005.

[5] V.D. Blondel, J.M. Hendrickx, A. Olshevsky, and J.N. Tsitsiklis. Convergence in multiagent coordination, consensus, and flocking. In *Proceedings of the Joint 44th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC'05)*, Seville, Spain, December 2005.

[6] L. Blume. The statistical mechanics of strategic interaction. *Games and Economic Behavior*, 5:387–424, 1993.

[7] L. Blume. Population games. In B. Arthur, S. Durlauf, and D. Lane, editors, *The Economy as an Evolving Complex System II*, pages 425–460. Addison-Wesley, Reading, MA, 1997.

[8] H. Borowski, J. R. Marden, and E. W. Frew. Fast convergence in semi-anonymous potential games. In *Proceedings of the IEEE Conference on Decision and Control*, 2013.

[9] H. P. Borowski, J. R. Marden, and J. S. Shamma. Learning efficient correlated equilibria. In *Proceedings of the IEEE Conference on Decision and Control*, 2014.

[10] J. Cortes, S. Martinez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '02)*, pages 1327–1332, Washington, DC, May 2002.

[11] C. Daskalakis, P.W. Goldberg, and C.H. Papadimitriou. The complexity of computing a nash equilibrium. *SIAM Journal on Computing*, 39(1):195–259, 2009.

[12] D. Fudenberg and D.K. Levine. Consistency and cautious fictitious play. *Games and Economic Behavior*, 19:1065–1089, July–September 1995.

[13] D. Fudenberg and D.K. Levine. *The Theory of Learning in Games*. MIT Press, Cambridge, MA, 1998.

[14] D. Fudenberg and J. Tirole. *Game Theory*. MIT Press, Cambridge, MA, 1991.

[15] R. Gopalakrishnan, J. R. Marden, and A. Wierman. Potential games are necessary to ensure pure Nash equilibria in cost sharing games. *Mathematics of Operations Research*, 39(4):1252–1296, 2014.

[16] S. Hart. Adaptive heuristics. *Econometrica*, **73**(5):1401–1430, 2005.

[17] S. Hart and Y. Mansour. How long to equilibrium? The communication complexity of uncoupled equilibrium procedures. *Games and Economic Behavior*, 69(1):107–126, 2010.

[18] S. Hart and A. Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68(5):1127–1150, 2000.

[19] S. Hart and A. Mas-Colell. Uncoupled dynamics do not lead to Nash equilibrium. *American Economic Review*, 93(5):1830–1836, 2003.

[20] T. J. Lambert III, M. A. Epelman, and R. L. Smith. A fictitious play approach to large-scale optimization. *Operations Research*, **53**(3):477–489, 2005.

[21] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Trans. on Automatic Control*, 48(6):988–1001, June 2003.

[22] M.J. Kearns, M.L. Littman, and S.P. Singh. Graphical models for game theory. In *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, pages 253–260, 2001.

[23] J. R. Marden. State based potential games. *Automatica*, 48:3075–3088, 2012.

[24] J. R. Marden, G. Arslan, and J. S. Shamma. Joint strategy fictitious play with inertia for potential games. *IEEE Transactions on Automatic Control*, 54:208–220, February 2009.

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

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

[27] J.R. Marden. Selecting efficient correlated equilibria through distributed learning. In *American Control Conference*, 2015.

[28] J.R. Marden and J.S. Shamma. Game theory and distributed control. In H.P. Young and S. Zamir, editors, *Handbook of Game Theory with Economic Applications*, volume 4, pages 861–899. Elsevier Science, 2015.

[29] S. Martinez, J. Cortes, and F. Bullo. Motion coordination with distributed information. *Control Systems Magazine*, 27(4):75–88, 2007.

[30] D. Monderer and L.S. Shapley. Fictitious play property for games with identical interests. *Journal of Economic Theory*, **68**:258–265, 1996.

[31] A. Montanari and A. Saberi. Convergence to equilibrium in local interaction games. In *50th Annual IEEE Symposium on Foundations of Computer Science*, 2009.

[32] R. A. Murphey. Target-based weapon target assignment problems. In P. M. Pardalos and L. S. Pitsoulis, editors, *Nonlinear Assignment Problems: Algorithms and Applications*. Kluwer Academic, Alexandra, Virginia, 1999.

[33] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, New York, NY, USA, 2007.

[34] R. Olfati-Saber. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Transactions on Automatic Control*, **51**:401–420, 2006.

[35] R. Olfati-Saber, J. A. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. In *Proceedings of the IEEE*, volume 95, pages 215–233, January 2007.

[36] R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–1533, 2003.

[37] T. Roughgarden. *Selfish Routing and the Price of Anarchy*. MIT Press, Cambridge, MA, USA, 2005.

[38] T. Roughgarden. Intrinsic robustness of the price of anarchy. In *Proceedings of STOC*, 2009.

[39] D. Shah and J. Shin. Dynamics in congestion games. In *ACM SIGMETRICS*, 2010.

[40] J.S. Shamma. Learning in games. In J. Baillieul and T. Samad, editors, *Encyclopedia of Systems and Control*. Springer-Verlag, London, 2014.

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

[42] B. Touri and A. Nedic. On ergodicity, infinite flow, and consensus in random models. *IEEE Transactions on Automatic Control*, 56(7):1593–1605, 2011.

[43] J. N. Tsitsiklis. Decentralized detection by a large number of sensors. Technical report, MIT, LIDS, 1987.

[44] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control*, **35**(9):803–812, 1986.

[45] A. Vetta. Nash equilibria in competitive societies with applications to facility location, traffic routing, and auctions. In *Foundations on Computer Science*, pages 416–425, 2002.

[46] Beibei Wang, Zhu Han, and K.J.R. Liu. Peer-to-peer file sharing game using correlated equilibrium. In *Information Sciences and Systems, 2009. CISS 2009. 43rd Annual Conference on*, pages 729–734, March 2009.

[47] D. Wolpert and K. Tumor. An overview of collective intelligence. In J. M. Bradshaw, editor, *Handbook of Agent Technology*. AAAI Press/MIT Press, 1999.

[48] H. P. Young. *Individual Strategy and Social Structure*. Princeton University Press, Princeton, NJ, 1998.

[49] H. P. Young. *Strategic Learning and its Limits*. Oxford University Press, 2004.

[50] M. Zhu and S. Martínez. Distributed coverage games for energy-aware mobile sensor networks. *SIAM Journal on Control and Optimization*, 51(1):1–27, 2013.

Index