Game Theory and Distributed Control

Jason R. Marden[*] Jeff S. Shamma[†]

July 9, 2012

Abstract

Game theory has been employed traditionally as a modeling tool for describing and influencing behavior in societal systems. Recently, game theory has emerged as a valuable tool for controlling or prescribing behavior in distributed engineered systems. The rationale for this new perspective stems from the parallels between the underlying decision making architectures in both societal systems and distributed engineered systems. In particular, both settings involve an interconnection of decision making elements whose collective behavior depends on a compilation of local decisions that are based on partial information about each other and the state of the world. Accordingly, there is extensive work in game theory that is relevant to the engineering agenda. Similarities notwithstanding, there remain important differences between the constraints and objectives in societal and engineered systems that require looking at game theoretic methods from a new perspective. This chapter provides an overview of selected recent developments of game theoretic methods in this role as a framework for distributed control in engineered systems.

1 Introduction

Distributed control involves the design of decision rules for systems of interconnected components to achieve a collective objective in a dynamic or uncertain environment. One example is teams of mobile autonomous systems, such as unmanned aerial vehicles (UAVs), for uses such as search and rescue, cargo delivery, scientific data collection, and homeland security operations. Other application examples can be found in sensor and data networks, communication networks,

[*] J.R. Marden is with the Department of Electrical, Computer and Energy Engineering, UCB 425, Boulder, Colorado 80309-0425, jason.marden@colorado.edu.

[†] J.S. Shamma is with the School of Electrical and Computer Engineering, Georgia Institute of Technology, 777 Atlantic Dr NW, Atlanta, GA 30332-0250, shamma@gatech.edu.

transportation systems, and energy [63, 74]. Technological advances in embedded sensing, communication, and computation all point towards an increasing potential and importance of such networks of autonomous systems.

In contrast to the traditional control paradigm, distributed control architectures do not have a central entity with access to all information or authority over all components. A lack of centralized information is possible even when components can communicate. Communications can be costly, e.g., because of energy conservation, or even inadmissible, e.g., for stealthy operations. Furthermore, the latency/time-delay required to distribute information in a large scale system may be impractical in a dynamically evolving setting. Accordingly, the collective components somehow must coordinate globally using distributed decisions based on only limited local information.

One approach to distributed control is to view the problem from the perspective of game theory. Since game theory concerns the study of interacting decision makers, the relevance of game theory to distributed control is easily recognized. Still, this perspective is a departure from the traditional study of game theory, where the focus has been the development of models and methods for applications in economic and social sciences. Following the discussion in [44, 77], we will refer to the to the traditional role of game theory as the "descriptive" agenda, and its application to distributed control as the "engineering" agenda.

The first step in deriving a game theoretic model is to identify the basic elements of the game, namely the players/agents and their admissible actions. In distributed control problems, there also is typically a global objective function that reflects the performance of the collective as a function of their joint actions. The following examples illustrate these elements in various applications:

- *Consensus/synchronization:* The agents are mobile platforms. The actions are agent orientations (e.g, positions or velocities). The global objective is for agents to align their orientations with each other [67].

- *Distributed routing:* Agents are mobile vehicles. The actions are paths from sources to destination. The global objective is to minimize network traffic congestion [69].

- *Sensor coverage:* The agents are mobile sensors. The actions are sensor paths. The global objective is to service randomly arriving spatially distributed targets in shortest time [56].

- *Wind energy harvesting:* The agents are wind turbines. The actions are the blade pitch angle and rotor speed. The global objective is to maximize the overall energy generated by the turbines [50].

- *Vehicle-target assignment:* The agents are heterogeneous mobile weapons with complementary capabilities. The actions are the selection of potential targets. The global objective is to maximize the overall expected damage [6].

- *Content distribution:* The agents are computer nodes. The actions are which files to store locally under limited storage capacity. The global objective is to service local file requests from users while minimizing peer-to-peer content requests [18].

- *Ad hoc networks:* The agents are mobile communication nodes. The actions are to form a network structure. Global objectives include establishing connectivity while optimizing performance specifications such as required power or communication hop lengths [65].

There is some flexibility in defining what constitutes a single player. For example in wind energy harvesting, a player could be a single turbine or a group of turbines. The determining factor is the extent to which the group can act as a single unit with shared information.

With the basic elements in place, the next step is to specify agent utility functions. Here the difference between the descriptive and engineering agenda becomes more apparent. Whereas in the descriptive agenda, utility functions are part of the modeling process, in the engineering agenda, utility functions constitute a *design choice*.

An important consideration in specifying the utility functions is the implication on the global objective. With utility functions in place, the game is fully specified. If one takes a solution concept such as Nash equilibrium to represent the outcome of the game, then these outcomes should be desirable as measured by the global objective.

With the game now fully specified, with players, actions, and utility functions, there is another important step that again highlights a distinction between the descriptive and engineering agenda. Namely, one must specify the dynamics through which agents will arrive at an outcome or select from a set of possible outcomes.

There is extensive work in the game theory literature that explores how a solution concept such as a Nash equilibrium might emerge, i.e., the "learning in games" program [21, 27, 87]. Quoting Arrow [5],

> "The attainment of equilibrium requires a disequilibrium process."

The work in learning in games seeks to understand how a plausible learning/adaptation disequilibrium process may (or may not) converge, and thereby reinforce the role of Nash equilibrium as a predictive outcome in the descriptive agenda. By contrast, in the engineering agenda, the role of such learning processes is to *guide* the agents towards a solution. Accordingly, the specification of the learning process also constitutes a *design choice*.

There is some coupling between the two design choices of utility functions and learning processes. In particular, it can be advantageous in designing utility functions to assure that the resulting game has an underlying structure (e.g., being a potential game or weakly acyclic game) so that one can exploit learning processes that converge for such games.

To recap, the engineering agenda requires both designing agent utility functions and learning processes. At this point, it is worthwhile highlighting various design considerations that play a more significant role in the engineering agenda:

- *Information:* One can impose various restrictions on the information available to each agent. Two natural widely considered scenarios are: i) agents can measure the actions of other agents or ii) agents can only measure their own action and perceived rewards. For example, in distributed routing, agents may observe the routes taken by other agents (which could be informationally intense), or, more reasonably, measure only their own experienced congestions.

- *Efficiency:* There can be several Nash equilibria, with some more desirable than others in terms of the global objective. This issue has been the subject of significant recent research in terms of the so called "price of anarchy" [69] in distributed routing problems.

- *Computation:* Each stage of a learning algorithm requires agent computations. Excessive computational demands per stage can render a learning algorithm impractical.

- *Dynamic constraints:* Learning algorithms can guide how agent actions evolve over time, and these actions may be restricted because of inherent agent limitations. For example, agents may have mobility limitations in that the current position restricts the possible near term positions. More generally, agent evolution may be subject to constraints in the form of physically constraining state dynamics (e.g., so-called Dubins vehicles [12]).

- *Time complexity:* A learning algorithm may exhibit several desirable features in terms of informational requirements, computational demands per stage, and efficiency, but require a excessive number of iterations to converge. One limitation is that some of the problems of distributed control, such as weapon/target assignment, have inherent computational complexity. Distributed implementation is a subclass of centralized implementation, and accordingly inherits computational complexity limitations.

Many factors contribute to the appeal of game theory for distributed control. First, in recognizing the relevance of game theory, one can benefit from the extensive existing work in game theory to build the engineering agenda. Second, the associated learning processes promise autonomous system operations in the sense of perpetual self-configuration in unknown or non-stationary environments and with robustness to disruptions or component failures. Finally, the separate design of utility functions and learning processes offers a modular approach to accommodate both different global objectives and underlying physical domain specific constraints.

The remainder of this chapter outlines selected results in the development of a engineering agenda in game theory for distributed control. Sections 2 and 3 present the design of utility functions and learning processes, respectively. Section 4 presents an expansion of these ideas in terms of a broader notion of game design. Finally, Section 5 provides some concluding remarks.

Preliminaries

A set of agents is denoted $N = \{1, 2, ..., n\}$. For each $i \in N$, $\mathcal{A}_i$ denotes the set of actions available to agent i. The set of joint actions is $\mathcal{A} = \mathcal{A}_1 \times ... \times \mathcal{A}_N$ with elements $a = (a_1, a_2, ..., a_n)$. The utility function of agent i is a mapping $U_i : \mathcal{A} \to \mathbb{R}$. We will often presume that there is also a global objective function $W : \mathcal{A} \to \mathbb{R}$. An action profile $a^* \in \mathcal{A}$ is a pure strategy Nash equilibrium (or just "equilibrium") if

$$U_i(a_i^*, a_{-i}^*) = \max_{a_i \in \mathcal{A}_i} U_i(a_i, a_{-i}^*)$$

for all $i \in N$.

2 Utility Design

In this section we will survey results pertaining to utility design for distributed engineered systems. Utility design for societal systems has been studied extensively in the game theoretic literature, e.g., cost sharing problems [60–62, 85] and mechanism design [22]. The underlying goal for utility design in societal systems is to augment players' utility functions in an admissible fashion to induce desirable outcomes.

Unlike mechanism design, the agents in the engineered agenda are programmable components. Accordingly, there is no concern that agents are not truthful in reporting information or obedient in executing instructions. Nonetheless, many of the contributions stemming from the cost sharing literature is immediately applicable to utility design in distributed engineered systems.

2.1 Cost/Welfare Sharing Games

To formally study the role of utility design in engineered systems we consider the class of welfare/cost sharing games [52]. This is a particularly relevant class of games in that may of the aforementioned applications of distributed control resemble resource allocation or sharing.

A welfare sharing game consists of a set of agents N, a finite set of resources $\mathcal{R}$, and for each agent $i \in N$, an action set $\mathcal{A}_i \subseteq 2^{\mathcal{R}}$. Note that the action set represents the set of allowable resource utilization profiles. For example, if $\mathcal{R} = \{1, 2, 3\}$, the action set

$$\mathcal{A}_i = \{\{1, 2\}, \{1, 3\}, \{2, 3\}\}$$

reflects that agent i always uses two out three resources. An example where structured actions sets emerge is in distributed routing, where resources are roads, but admissible actions are paths. Accordingly, the action set represents sets of resources induced by the underlying network structure.

We restrict our attention to the class of *separable* system level objective functions of the form

$$W(a) = \sum_{r \in \mathcal{R}} W_r \left(\{a\}_r\right)$$

where $W_r : 2^N \to R^+$ is the objective function for resource r, and $\{a\}_r$ is the set of agents using resource r, i.e.,

$$\{a\}_r = \{i \in N : r \in a_i\}.$$

The goal of such welfare sharing games is to derive *admissible* agent utility functions such that the resulting game possess desirable properties. In particular, we focus on the design of *local* agent utility functions of the form

$$U_i(a_i, a_{-i}) = \sum_{r \in a_i} f_r \left(i, \{a\}_r\right) \tag{1}$$

where $f_r : N \times 2^N \to R$ is the *welfare sharing protocol*, or just “protocol”, at resource r. The protocol represents a mechanism for agents to evaluate the “benefit” of being at a resource given the choices of the other agents. Utility functions are “local” in the sense that the benefit of using a resource only depends on the set of other agents using that resource and not on the usage profiles of other resources.

Finally, a welfare sharing game is now by the tuple $G = (N, \mathcal{R}, \{\mathcal{A}_i\}, \{W_r\}, \{f_r\})$.

One of the important design considerations associated with engineered systems is that the structure of the specific resource allocation problem, i.e., the resource set $\mathcal{R}$ or the structure of the action sets $\{\mathcal{A}_i\}_{i \in N}$, is not known to the system designer a priori. Accordingly, a challenge in welfare sharing problems is to design a set of *scalable* protocols $\{f_r\}$ that efficiently applies to all games in the set

$$\mathcal{G} = \{G = (N, \mathcal{R}, \{\mathcal{A}_i\}, \{W_r\}, \{f_r\}) : \mathcal{A}_i \subset 2^{\mathcal{R}}\}.$$

In other words, the set $\mathcal{G}$ is represents a family of welfare sharing games with different resource availability profiles. A protocol is “scalable” in the sense that the distribution of welfare does not depend on the specific structure of resource availability. Note that the above set of games can capture both variations in the agent set and resource set. For example, setting $\mathcal{A}_i = \emptyset$ is equivalent to removing agent i from the game. Similarly, letting the action sets satisfy $\mathcal{A}_i \subseteq 2^{\mathcal{R} \setminus \{r\}}$ for each agent i is equivalent to removing resource r from the specified resource allocation problem.

The evaluation of a protocol, $\{f_r\}$, takes into account the following considerations:

Potential game structure: Deriving an efficient dynamical process that converges to an equilibrium requires additional structure on the game environment. One such structure is that of *potential games* introduced in [58]. In a potential game there exists a potential function $\phi : \mathcal{A} \to R$ such that for any action profile $a \in \mathcal{A}$, agent $i \in N$, and action choice $a_i' \in \mathcal{A}_i$,

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

If a game is potential game, then an equilibrium is guaranteed to exist since any action profile $a^* \in \arg\max_{a \in \mathcal{A}} \phi(a)$ is an equilibrium[1]. Furthermore, there is a wide array of distributed learning algorithms which guarantee convergence to an equilibrium [71]. It is important to note that the global objective $W(\cdot)$ and $\phi(\cdot)$ can be different functions.

Efficiency of equilibria: Two well known worst case measures of efficiency of equilibria are the *price of anarchy (PoA)* and *price of stability (PoS)* [66]. The PoA provides an upper bound on the ratio between the performance of an optimal allocation versus an equilibrium. More specifically, for a game G, let $a^{\mathrm{opt}}(G) \in \mathcal{A}$ satisfy

$$a^{\mathrm{opt}}(G) \in \arg\max_{a \in \mathcal{A}} W(a; G);$$

let $\mathrm{NE}(G)$ denote the set of equilibria for G; and define

$$\text{PoA}(G) = \max_{a^{\mathrm{ne}} \in \mathrm{NE}(G)} \frac{W(a^{\mathrm{opt}}(G); G)}{W(a^{\mathrm{ne}}; G)}.$$

For example, a PoA of 2 ensures that for any game $G \in \mathcal{G}$ any equilibrium is at least 50% as efficient as the optimal allocation. The PoS, which represents a more optimistic worst case characterization, provides a lower bound on the ratio between the performance of the optimal allocation and the best equilibrium, i.e.,

$$\text{PoS}(G) = \min_{a^{\mathrm{ne}} \in \mathrm{NE}(G)} \frac{W(a^{\mathrm{opt}}(G); G)}{W(a^{\mathrm{ne}}; G)}.$$

2.2 Achieving Potential Game Structures

We begin by exploring the following question: is it possible to design scalable protocols and local utility functions to guarantee a potential game structure irrespective of the specific structure of the resource allocation problem? In this section we review two constructions which originated in the traditional economic cost sharing literature [85] that achieve this objective.

The first construction is the *marginal contribution protocol* [83]. For any resource $r \in \mathcal{R}$, player set $S \subseteq N$, and player $i \in N$,

$$f_r^{\mathrm{MC}}(i, S) := W_r(S) - W_r(S \setminus \{i\}). \tag{2}$$

[1] See [6] for examples of intuitive utility functions that do not result an equilibrium in vehicle-target assignment.

The marginal contribution protocol provides the following guarantees.

Theorem 2.1 (Wolpert and Tumer, [83]) *Let $\mathcal{G}$ be a class of welfare sharing games where the protocol for each resource $r \in \mathcal{R}$ is defined as the marginal contribution protocol in (2). Any game $G \in \mathcal{G}$ is a potential with potential function W.*

Irrespective of the underlying game, the marginal contribution protocols always ensures the existence of a potential games and consequently the existence of an equilibrium. Furthermore, since the resulting potential function is $\phi = W$, the PoS is guaranteed to be 1 when using the marginal contribution protocol. In general, the marginal contribution protocol need not provide any guarantee with respect to the PoA.

The second construction is known as the *weighted Shapley value* [26,29,76]. For any resource $r \in \mathcal{R}$, player set $S \subseteq N$, and player $i \in N$,

$$f_r^{\mathrm{WSV}}(i,S) := \sum_{T \subseteq S : i \in T} \frac{\omega_i}{\sum_{j \in T} \omega_j} \left(\sum_{R \subseteq T} (-1)^{|T|-|R|}\, W_r(R) \right). \tag{3}$$

where $\omega_i > 0$ is defined as the weight of player i. The Shapley value represents a special case of the weighted Shapley value when $w_i = 1$ for all agents $i \in N$. The weighted Shapley value protocol provides the following guarantees.

Theorem 2.2 (Marden and Wierman, 2008 [52]) *Let $\mathcal{G}$ be a class of welfare sharing games where the protocol for each resource $r \in \mathcal{R}$ is the weighted Shapley value protocol in (3). Any game $G \in \mathcal{G}$ is a (weighted) potential game*[2] *with potential function*

$$\phi^{\mathrm{WSV}}(a) := \sum_{r \in \mathcal{R}} \phi_r^{\mathrm{WSV}}\left(\{a\}_r\right),$$

where ϕ_r^{WSV} is the resource specific potential function defined (recursively) as follows:

$$\begin{aligned} \phi_r^{\mathrm{WSV}}(\emptyset) &= 0 \\ \phi_r^{\mathrm{WSV}}(S) &= \frac{1}{\sum_{i \in S} w_i} \left[W_r(S) + \sum_{i \in S} w_i \phi_r^{\mathrm{WSV}}(S \setminus \{i\}) \right], \;\; \forall S \subseteq N. \end{aligned}$$

[2] A weighted potential game is a generalization of a potential game with the following condition on the game structure. There exist a potential function $\phi : \mathcal{A} \to R$ and weights $w_i > 0$ for each agent $i \in N$ such that for any action profile $a \in \mathcal{A}$, agent $i \in N$, and action choice $a_i' \in \mathcal{A}_i$,

$$U_i(a_i', a_{-i}) - U_i(a_i, a_{-i}) = w_i \left(\phi(a_i', a_{-i}) - \phi(a_i, a_{-i}) \right).$$

The recursion presented in the above theorem directly follows from the potential function characterization of the weighted Shapley value derived in [29]. As with the marginal contribution protocol, the weighted Shapley value protocol always ensures the existence of a (weighted) potential game, and consequently the existence of an equilibrium, irrespective of the underlying structure of the resource allocation problem. However, unlike the marginal contribution protocol, the potential function is not $\phi = W$. Consequently, the PoS is not guaranteed to be 1 when using the marginal contribution protocol. In general, the weighted Shapley value protocol also does not provide any guarantees with respect to the PoA.

An important difference between the marginal contribution in (2) and the weighted Shapley value in (3) is that the weighted Shapley value protocol guarantees that the utility functions are *budget-balanced*, i.e., for any resource $r \in \mathcal{R}$ and agent set $S \subseteq N$,

$$\sum_{i \in S} f_r^{\mathrm{WSV}}(i, S) = W_r(S). \tag{4}$$

The marginal contribution utility, on the other hand, does not guarantee that utility functions are budget-balanced. Budget-balanced utility functions are important for the control (or influence) of societal systems where there is a cost or revenue that needs to be completely absorbed by the participating players, e.g., network formation [16] and content distribution [25]. Furthermore, budget-balanced (or budget-constrained) utility functions are also important for engineered systems by providing desirable efficiency guarantees [70, 80]; see forthcoming Theorems 2.3 and 2.5. However, the design of budget-balanced utility functions is computationally prohibitive in large systems since computing a weighted Shapley value requires a summation over an exponential number of terms.

2.3 Efficiency of Equilibria

The desirability of a potential game structure stems from the availability of various distributed learning/adaptation rules that lead to an equilibrium. Accordingly for the engineering agenda, an important consideration is the resulting PoA. This issue is related to a research thread within algorithmic game theory that focuses on *analyzing* the inefficiency of equilibria for classes of games where the agents' utility functions $\{U_i\}$ and system level objective W are specified (cf., Chapters 17–21 in [66]). While these results focus on analysis and not synthesis, they can be leveraged in utility design.

The following result, expressed in the context of resource sharing and protocols, requires the notion of submodular functions. An objective function W_r is *submodular* if for any agent set $S \subseteq T \subseteq N$ and any agent $i \in N$,

$$W_r(S \cup \{i\}) - W_r(S) \geq W_r(T \cup \{i\}) - W_r(T).$$

Submodularity reflects the diminishing marginal effect of assigning agents to resources. This property is relevant in a variety of engineering applications, including the aforementioned sensor coverage and vehicle-target assignment scenarios.

Theorem 2.3 (Vetta, 2002 [80]) *Let $\mathcal{G}$ be a class of welfare sharing games that satisfies the following conditions for each resource $r \in \mathcal{R}$:*

(i) The objective function W_r is submodular.

(ii) The protocol satisfies $f_r(i, S) \geq W_r(S) - W_r(S \setminus \{i\})$ for each set of agents $S \subseteq N$ and agent $i \in S$.

(iii) The protocol satisfies $\sum_{i \in S} f_r(i, S) \leq W_r(S)$ for each set of agents $S \subseteq N$.

Then for any game $G \in \mathcal{G}$, if an equilibrium exists, the PoA is 2.

Theorem 2.3 reveals two interesting properties. First, condition (ii) parallels the aforementioned marginal contribution protocol in (2). Second, condition (iii) relates to the the budget-balanced constraint associated with (weighted) Shapley value protocol in (4). Since both the marginal contribution protocol and Shapley value protocol guarantee the existence of an equilibrium, we can combine Theorems 2.1, 2.2, and 2.3 into the following corollary.

Corollary 2.1 *Let $\mathcal{G}$ be a class of welfare sharing games with submodular resource objective functions, W_r. Suppose one of the following two conditions is satisfied:*

(i) The protocol for each resource $r \in \mathcal{R}$ is the marginal contribution protocol in (2).

(ii) The protocol for each resource $r \in \mathcal{R}$ is the weighted Shapley value protocol in (3).

Then for any $G \in \mathcal{G}$, an equilibrium is guaranteed to exist and the PoA is 2.

Corollary 2.1 demonstrates that both the marginal contribution protocol and the Shapley value protocol guarantee desirable properties regarding the existence and efficiency of equilibria for a broad class of resource allocation problems with submodular objective functions. There are two shortcomings associated with this result. First, it does not reveal how the structure of the objective functions $\{W_r\}$ impacts the PoA guarantees beyond the factor of 2. For example, in the aforementioned vehicle-target assignment problem (cf., [6]) with submodular objective functions, both the marginal contribution and weighted Shapley value protocol will ensure that all resulting equilibria are at least 50% as efficient as the optimal assignment. It is unclear whether this factor of 2 is tight or the resulting equilibria will be more efficient than this general guarantee. Second, this corollary does not differentiate between the performance associated with the marginal contribution protocol and the weighted Shapley value protocol. For example, does the marginal contribution protocol outperform the weighted Shapley value protocol with respect to PoA guarantees? The following theorem begins to address these issues.

Theorem 2.4 (Marden and Roughgarden, 2010 [49]) *Let G be a welfare sharing game that satisfies the following conditions:*

(i) The objective function for each resource $r \in \mathcal{R}$ is submodular and anonymous.[3]

(ii) The protocol for each resource $r \in \mathcal{R}$ is the Shapley value protocol as in (3) with $\omega_i = 1$ for all agents $i \in N$.

(iii) The action set for each agent $i \in N$ is $\mathcal{A}_i = \mathcal{R}$.

Then an equilibrium is guaranteed to exist and the PoA is

$$1 + \max_{r \in \mathcal{R},\ m \leq n} \left\{ \max_{k \leq m} \left(\frac{W_r(k)}{W_r(m)} - \frac{k}{m} \right) \right\}. \tag{5}$$

Theorem 2.4 demonstrates that the structure of the welfare function plays a significant role in the underlying PoA guarantees. For example, suppose that the objective function for each resource is linear in the number of agents, e.g., $W_r(S) = |S|$ for all agent sets $S \subseteq N$. For this situation, the second term in (5) is 0 which means that the PoA is 1.

The final general efficiency result that we review in this section pertains to the efficiency of alternative classes of equilibria. In particular, we consider the class of *coarse correlated equilibria*, which represent a generalization of the class of Nash equilibria[4]. As with potential game structures, part of the interest in coarse correlated equilibria is the availability of simple adaptation rules that lead to time-averaged behavior consistent with coarse correlated equilibria [27, 87]. A joint distribution $z \in \Delta(\mathcal{A})$ is a coarse correlated equilibrium if for any player $i \in N$ and any action $a_i' \in \mathcal{A}_i$

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

where $\Delta(\mathcal{A})$ represent the simplex over the finite set $\mathcal{A}$ and z^a represents the component of the distribution z associated with the action profile a.

We extend the system level objective from allocations to a joint distribution $z \in \Delta(a)$ as

$$W(z) = \sum_{a \in \mathcal{A}} W(a) z^a.$$

Since the set of coarse correlated equilibria contains the set of Nash equilibria, the PoA associated with this more general set of equilibria can only degrade. However, the following theorem demonstrates that if the utility functions satisfy a "smoothness" then there is no such degradation. We will present this theorem with regards to utility functions as opposed to protocols for a more direct presentation.

[3] An objective function W_r is anonymous if $W_r(S) = W_r(T)$ for any agent sets $S, T \subseteq N$ such that $|S| = |T|$.

[4] Coarse correlated equilibria are also equivalent to the set of no-regret points [27, 87].

Theorem 2.5 (Roughgarden, 2009 [70]) *Consider any welfare sharing game G that satisfies the following conditions:*

(i) There exist parameters $\lambda > 0$ and $\mu > 0$ such that for any action profiles $a, a^* \in \mathcal{A}$

$$\sum_{i \in N} U_i(a_i^*, a_{-i}) \geq \lambda \cdot W(a^*) - \mu \cdot W(a). \tag{6}$$

(ii) For any action profile $a \in \mathcal{A}$, the agents' utility functions satisfy $\sum_{i \in N} U_i(a) \leq W(a)$.

Then the PoA of the set of coarse correlated equilibria is

$$\inf_{\lambda > 0, \mu > 0} \left\{ \frac{1+\mu}{\lambda} \right\}$$

where the infimum is over the set of admissible parameters that satisfy (6).

Many classes of games relevant to distributed engineered systems satisfy the "smoothness" condition set forth in (6). For example, the class of games considered in Theorem 2.3 satisfies the conditions of Theorem 2.5 with smoothness parameters $\lambda = 1$ and $\mu = 1$ [70]. Consequently, the PoA of 2 extends beyond just pure Nash equilibria to all coarse correlated equilibria.

3 Learning Design

The field of learning in games concerns the analysis of distributed learning algorithms and their convergence to various solution concepts or notions of equilibrium [21, 27, 87]. In the descriptive agenda, the motivation is that convergence of such algorithms provides some justification for a particular solution concept as a predictive model of behavior in a societal system.

This literature can be used as a starting point for the engineering agenda to offer solutions for how equilibria *should* emerge in distributed engineered systems. In this section we will survey results pertaining to learning design and highlight their applicability to distributed control of engineered systems.

3.1 Preliminaries: Repeated play of one-shot games

We will consider learning/adaptation algorithms in which agents repeatedly play over stages $t \in \{0, 1, 2, ...\}$. At each stage, an agent i chooses an action $a_i(t)$ according to the probability distribution $p_i(t) \in \Delta(\mathcal{A}_i)$. We refer to $p_i(t)$ as the *strategy* of agent i at time t. An agent's strategy at time t relies only on observations over stages $\{0, 1, 2, ..., t-1\}$.

Different learning algorithms are specified by the agents' information and the mechanism by which their strategies are updated as information is gathered. We categorize such learning algorithms into the following three classes of information structures.

- *Full Information:* For the class of full information learning algorithms, each agent knows the structural form of his own utility function and is capable of observing the actions of all other agents at every stage but does not know other agents' utility functions. Learning rules in which agents do not know the utility functions of other agents are also referred to as *uncoupled* [30, 32]. Full information learning algorithms can be written as

$$p_i(t) = F_i\big(a(0), ..., a(t-1); U_i\big). \tag{7}$$

for an appropriately defined functions $F_i(\cdot)$.

- *Oracle-Based Information:* For the class of oracle-based learning algorithms, each agent is capable of evaluating the payoff associated with alternative action choices—even though these choices were not selected. More specifically, the strategy adjustment mechanism of a given agent i can be written in the form

$$p_i(t) = F_i\left(\{U_i\left(a_i, a_{-i}\left(0\right)\right)\}_{a_i \in \mathcal{A}_i}, \ldots, \{U_i\left(a_i, a_{-i}\left(t-1\right)\right)\}_{a_i \in \mathcal{A}_i}\right). \tag{8}$$

- *Payoff-Based Information:* For the class of payoff-based learning algorithms, each agent has access to: (i) the action they played and (ii) the payoff they received. In this setting, the strategy adjustment mechanism of agent i takes the form

$$p_i(t) = F_i\left(\{a_i(0), U_i(a(0))\}, ..., \{a_i(t-1), U_i(a(t-1))\}\right). \tag{9}$$

Payoff-based learning rules are also referred to as *completely uncoupled* [3, 20].

The following sections review various algorithms from the literature on learning in games and highlight their relevance for the engineering agenda in terms of their limiting behavior, the resulting efficiency, and the requisite information structure.

3.2 Learning Nash Equilibria in Potential Games

We begin with algorithms for the special class of potential games. The relevance of these algorithms for the engineering agenda is enhanced by the possibility of constructing utility functions, as discussed in the previous section for resource allocation problems, to ensure a potential game structure.

3.2.1 Fictitious Play and Joint Strategy Fictitious Play

Fictitious play (cf., [21]) is representative of a full information learning algorithm. In Fictitious Play, each agent $i \in N$ tracks the empirical frequency of the actions of other players. Specifically, for any $t > 0$, let

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

where $I\{\cdot\}$ denotes the indicator function[5] The vector $q_i(t) \in \Delta(\mathcal{A}_i)$ reflects the percentage of time that agent i selected the action a_i over stages $\{0, 1, \ldots, t-1\}$. Define the empirical frequency vector for player i at time t as $q_i(t) = \{q_i^{a_i}(t)\}_{a_i \in \mathcal{A}_i}$. At each time t, each player seeks to maximize his expected utility under the presumption that all other players are playing independently accordingly the empirical frequency of their past actions. More specifically, the action of player i at time t is chosen according to

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

The following theorem establishes the convergence properties of Fictitious Play for potential games.

Theorem 3.1 (Monderer and Shapley, 1994 [57]) *Let G be a finite n-player potential game. Under Fictitious Play, the empirical distribution of the players' actions $\{q_1(t), q_2(t), \ldots, q_n(t)\}$ will converge to a (possibly mixed strategy) Nash equilibrium of the game G.*

One concern associated with utilizing Fictitious Play for prescribing behavior in distributed engineered systems is the informational and computational demands [23, 35, 48]. Here, each agent is required to track the empirical frequency of the past actions of all other agents, which is prohibitive in large scale systems. Furthermore, computing a best response is intractable in general since it requires computing an expectation over a joint action space whose cardinality grows exponentially in the number of agents and the cardinality of their action sets.

Inspired by the potential application of Fictitious Play for distributed control of engineered systems, several papers investigated maintaining the convergence properties associated with Fictitious Play while reducing the computational and informational demands on the agents [6, 23, 35, 39–41, 46, 48, 54]. One such learning algorithm is *Joint Strategy Fictitious Play with inertia* introduced in [48].

In Joint Strategy Fictitious Play with inertia (as with no-regret algorithms [27]), at each time $t > 0$ each agent $i \in N$ computes the average *hypothetical utility* for each action $a_i \in \mathcal{A}_i$, defined as

$$\begin{aligned} V_i^{a_i}(t) &= \frac{1}{t} \sum_{\tau=0}^{t-1} U_i(a_i, a_{-i}(\tau)), \qquad (10) \\ &= \left(\frac{t-1}{t}\right) V_i^{a_i}(t-1) + \frac{1}{t} U_i(a_i, a_{-i}(t-1)). \end{aligned}$$

The average hypothetical utility for action a_i at time t is the average utility that action a_i would have received up to time t provided that all other agents did not change their action. Note that this

[5] The indicator function $I\{\text{statement}\}$ equals 1 if the mathematical expression "statement" is true, and equals 0 otherwise.

computation only requires oracle-based information as opposed to the full information structure of Fictitious Play. Define the best response set of agent i at time t as

$$B_i(t) = \left\{ a_i \in \mathcal{A}_i : \operatorname*{arg\,max}_{a_i \in \mathcal{A}_i} V_i^{a_i}(t) \right\}.$$

The action of player i at stage t is chosen as follows:

- If $a_i(t-1) \in B_i(t)$ then $a_i(t) = a_i(t-1)$.
- If $a_i(t-1) \notin B_i(t)$ then

$$a_i(t) = \begin{cases} a_i(t-1) & \text{with probability } \epsilon \\ a_i \in B_i(t) & \text{with probability } \frac{1-\epsilon}{|B_i(t)|} \end{cases}$$

where $\epsilon > 0$ is the players' inertia. The following theorem establishes the convergence properties of Fictitious Play for *generic* potential games.[6]

Theorem 3.2 (Marden et al., 2009 [48]) *Let G be a finite n-player generic potential game. Under Joint Strategy Fictitious Play with Inertia, the joint action profile will converge almost surely to a pure Nash equilibrium of the game G.*

As previously mentioned, Joint Strategy Fictitious Play with inertia falls under the classification of oracle-based information. Accordingly, the informational and computational demands on the agents when using Joint Strategy Fictitious Play with inertia are reasonable in large scale systems—assuming the hypothetical utility can be measured. The availability of such measurements is application dependent. For example in distributed routing, the hypothetical utility could be estimated with some sort of "traffic report" at the end of each stage.

The name Joint Strategy Fictitious Play stems from the average hypothetical utility in (10) reflecting the expected utility for agent i under the presumption that all agents other than agent i select an action with a joint strategy[7] in accordance to the empirical frequency of their pasts joint decisions, i.e,

$$V_i^{a_i}(t) = \sum_{a_{-i} \in \mathcal{A}_i} U_i(a_i, a_{-i}) z_{-i}^{a_{-i}}(t)$$

where

$$z_{-i}^{a_{-i}}(t) = \frac{1}{t} \sum_{\tau=0}^{t-1} I\{a_{-i}(\tau) = a_{-i}\}.$$

[6] Here, "generic" me ands that for any agent $i \in N$, action profile $a \in \mathcal{A}$, and action $a_i' \in \mathcal{A}_i \setminus a_i$, $U_i(a_i, a_{-i}) \neq U_i(a_i', a_{-i})$. Weaker versions of genericity also ensure the characterization of the limiting behavior presented in Theorem 3.2, e.g., if all equilibria are strict.

[7] That is, unlike Fictitious Play, players are not presumed to play independently according to their individual empirical frequencies.

Joint strategy Fictitious Play also can be viewed as a "max-regret" variant of no-regret algorithms [27,46] with inertia where the the regret for action $a_i \in \mathcal{A}_i$ at time t is

$$R_i^{a_i}(t) \;=\; \frac{1}{t}\sum_{\tau=0}^{t-1}\left(U_i(a_i, a_{-i}(\tau)) - U_i(a_i(\tau), a_{-i}(\tau))\right). \tag{11}$$

Note that $\underset{a_i \in \mathcal{A}_i}{\arg\max}\; V_i^{a_i}(t) = \underset{a_i \in \mathcal{A}_i}{\arg\max}\; R_i^{a_i}(t)$, hence the algorithms are equivalent.

Finally, another distinction from Fictitious Play is that Joint Strategy Fictitious Play with inertia guarantees convergence to pure equilibria almost surely.

3.2.2 Simple Experimentation Dynamics

One concern with the implementation of learning algorithms, even in the case of full information, is the need to compute utility functions and the associated utility of different action choices (as in the computation of better or best replies). Such computations presume the availability of a closed-form expression of utility functions, which may impractical in many scenarios. A more realistic requirement is to have agents only *measure* a realized utility online, rather than compute utility values offline. Accordingly, several papers have focused on providing payoff-based dynamics with similar limiting behaviors as the preceding full information or oracle-based algorithms [7,20,24,54,68,88].

A representative example is the learning algorithm *Simple Experimentation Dynamics*, introduced in [54]. Each agent $i \in N$ maintains a pair of evolving local state variables $[\bar{a}_i, \bar{u}_i]$. These variables represent

- a *benchmark action*, $\bar{a}_i \in \mathcal{A}_i$, and
- a *benchmark utility*, $\bar{u}_i$, which is in the range of $U_i(\cdot)$.

Simple Experimentation Dynamics proceeds as follows:

1. **Initialization:** At stage $t = 0$, each player arbitrarily selects and plays any action, $a_i(0) \in \mathcal{A}_i$. This action will be set initially as the player's *baseline action* at stage 1, i.e., $\bar{a}_i(1) = a_i(0)$. Likewise, each player's *baseline utility* at stage 1 is initialized as $u_i(1) = U_i(a(0))$.

2. **Action Selection:** At subsequent stages, each player selects his baseline action with probability $(1 - \epsilon)$ or experiments with a new random action with probability ϵ. That is,

   - $a_i(t) = \bar{a}_i(t)$ with probability $(1 - \epsilon)$
   - $a_i(t)$ is chosen randomly (uniformly) over $\mathcal{A}_i$ with probability ϵ

   where $\epsilon > 0$ is the player's *exploration rate*. Whenever $a_i(t) \neq \bar{a}_i(t)$, we will say that player i "experimented".

3. **Baseline Action and Baseline Utility Update:** Each player compares the utility received, $U_i(a(t))$, with his baseline utility, $\bar{u}_i(t)$, and updates his baseline action and utility as follows:

   - If player i *experimented* (i.e., $a_i(t) \neq \bar{a}_i(t)$) and if $U_i(a(t)) > \bar{u}_i(t)$, then
     $$\bar{a}_i(t+1) = a_i(t),$$
     $$\bar{u}_i(t+1) = U_i(a(t)).$$
   - If player i *experimented* and if $U_i(a(t)) \leq \bar{u}_i(t)$, then
     $$\bar{a}_i(t+1) = \bar{a}_i(t),$$
     $$\bar{u}_i(t+1) = \bar{u}_i(t).$$
   - If player i *did not experiment* (i.e., $a_i(t) = \bar{a}_i(t)$), then
     $$\bar{a}_i(t+1) = \bar{a}_i(t),$$
     $$\bar{u}_i(t+1) = U_i(a(t)).$$

4. Return to Step 2 and repeat.

Theorem 3.3 (Marden et al., 2010 [54]) *Let G be a finite n-player potential game. Under Simple Experimentation Dynamics, given any probability $p < 1$, there exists an exploration rate $\epsilon > 0$ (sufficiently small), such that for all sufficiently large stages t, the joint action $a(t)$ is a Nash equilibrium of G with at least probability p.*

Theorem 3.3 demonstrates that one can attain convergence to equilibria even in the setting where agents have minimal knowledge regarding the underlying game. Note that for such payoff-based dynamics we attain probabilistic convergence as opposed to almost sure converges. The reasoning is that agents are unaware of whether or not they are at an equilibrium since they do not have access to oracle-based or full information. Consequently, the agents perpetually probe the system to reassess the baseline action and utility.

3.2.3 Equilibrium Selection: Log-linear Learning and Its Variants

The previous discussion establishes how distributed learning rules under various information structures can converge to a Nash equilibrium. However, these results are silent on the issue of equilibrium *selection*, i.e., determining which equilibria may be favored or excluded. Notions such as PoA and PoS give pessimistic and optimistic bounds, respectively, on the value of a global performance measure at an equilibrium as compared to its optimal value. Equilibrium selection offers a refinement of these bounds through the specific underlying dynamics.

The topic of equilibrium selection has been widely studied within the descriptive agenda. Two standard references are [37, 84], which discuss equilibrium selection between risk dominant or payoff dominant equilibrium in symmetric 2×2 games. As would be expected, the conclusions are sensitive to the underlying dynamics [9]. However, in the engineering agenda, one can exploit this dependence as an available degree of freedom (e.g., [15]).

This section will review equilibrium selection in potential games for a class of dynamics, namely log-linear learning and its variants, that converge to maximizer of the underlying potential function, ϕ. The relevance for the engineering agenda stems from results such as Theorem 2.1, which illustrate how utility design can ensure the resulting interaction framework is a potential game and that the optimal allocation corresponds to the optimizer of the potential function. Hence the optimistic PoS, which equals 1 for this setting, will be achieved through the choice of dynamics.

Log-linear learning, introduced in [11], is an asynchronous oracle-based learning algorithm. At each stage $t > 0$, a single agent $i \in N$ is randomly chosen and allowed to alter his current action. All other players must repeat their actions from the previous stage, i.e. $a_{-i}(t) = a_{-i}(t-1)$. At stage t, the selected player i employs the (Boltzmann distribution) strategy $p_i(t) \in \Delta(\mathcal{A}_i)$, given by

$$p_i^{a_i}(t) = \frac{e^{\frac{1}{\tau}U_i(a_i, a_{-i}(t-1))}}{\sum\limits_{\bar{a}_i \in \mathcal{A}_i} e^{\frac{1}{\tau}U_i(\bar{a}_i, a_{-i}(t-1))}}, \tag{12}$$

for a fixed "temperature", $\tau > 0$. As is well known for the Boltzmann distribution, for large τ, player i will select any action $a_i \in \mathcal{A}_i$ with approximately equal probability, whereas for diminishing τ, player i will select a best response to the action profile $a_{-i}(t-1)$, i.e.,

$$a_i(t) \in \underset{a_i \in \mathcal{A}_i}{\arg\max}\, U_i(a_i, a_{-i}(t-1))$$

with increasingly high probability.

The following theorem characterizes the limiting behavior associated with log-linear learning for the class of potential games.

Theorem 3.4 (Blume, 1993 [11]) *Let G be a finite n-player potential game. Log-linear learning induces an aperiodic and irreducible process of the joint action set $\mathcal{A}$. Furthermore, the unique stationary distribution $\mu(\tau) = \{\mu^a(\tau)\}_{a \in \mathcal{A}} \in \Delta(\mathcal{A})$ is given by*

$$\mu^a(\tau) = \frac{e^{\frac{1}{\tau}\phi(a)}}{\sum\limits_{\bar{a} \in \mathcal{A}} e^{\frac{1}{\tau}\phi(\bar{a})}}. \tag{13}$$

One can interpret the stationary distribution μ as follows. For sufficiently large times $t > 0$, $\mu^a(\tau)$ equals the probability that $a(t) = a$. As one decreases the temperature, $\tau \to 0$, all the weight of the stationary distribution $\mu(\tau)$ is on the joint actions that maximize the potential function. Again, the emphasis here is that log-linear learning, coupled with suitable utility design, converges probabilistically to the maximizer of the potential function, and hence underlying global objective.

A concern with log-linear learning as a tool for the engineering agenda is whether the specific assumptions on the both the game and learning algorithm are restrictive and thereby limit the applicability of log-linear learning for distributed control. In particular, log-linear learning imposes the following assumptions:

(i) The underlying process is *asynchronous* which implies that the agents can only update their strategies one at a time, thereby requiring some sort of coordination.

(ii) The updating agent can select any action in his action set. In distributed control applications, there may be evolving constraints on the available action sets (e.g., mobile robots with limited mobility or in an environment with obstacles).

(iii) The requisite information structure is oracle-based.

(iv) The agents' utility function constitute an exact potential game.

It turns out that these concerns can be alleviated through the use of similar learning rules with an alternative analysis. While Theorem 3.4 provides an explicit characterization of the resulting stationary distribution, an important consequence is that as $\tau \to 0$ the mass of the stationary distribution focuses on the joint actions that maximize the potential function. In the language of [86], potential functions maximizers are *stochastically stable*[8]

Recent work analysis how to relax the structure of log-linear learning while ensuring that the only stochastically stable states are the potential function maximizers. Reference in [1] demonstrates certain relaxations under which potential function maximizers need not be stochastically stable. Reference [51] demonstrates that it is possible to relax the structure carefully while maintaining the desired limiting behavior. In particular, [51] establishes a payoff-based learning algorithm, termed *payoff-based log linear learning*, which ensures that for potential games the only stochastically stable states are the potential function maximizers. We direct the readers to [51] for details.

3.2.4 Near Potential Games

An important consideration for the engineering agenda is to understand the "robustness" of learning algorithms, i.e., how do guaranteed properties degrade as underlying modeling assumptions are violated. For example, consider the weighted Shalpey value protocol defined in (3). The weighted Shapley value protocol requires a summation of an exponential number of terms, which can be computationally prohibitive in large-scale systems. While there are sampling approaches that can yield good approximations for the Shapley value [17], it is important to note that these sampled utilities will not constitute a potential game.

[8] An action profile $a \in \mathcal{A}$ is stochastically stable if $\lim_{\tau \to 0^+} \mu^a(\tau) > 0$.

Accordingly, several papers have focused on analyzed dynamics in *near potential games* [13, 14, 51]. We say that a game is $\delta > 0$ close to a potential game if there exists a potential function $\phi : \mathcal{A} \to R$ such that for any player $i \in N$, actions $a_i', a_i'' \in \mathcal{A}_i$, and joint action $a_{-i} \in \mathcal{A}_{-i}$, players' utility satisfies

$$|(U_i(a_i', a_{-i}) - U_i(a_i'', a_{-i})) - (\phi(a_i', a_{-i}) - \phi(a_i'', a_{-i}))| \leq \delta.$$

A game is a near potential game for such games where δ is sufficiently small. The work in [13, 14, 51] proves that the limiting behavior associated with associated with several classes of dynamics on near-potential games can be approximated by analyzing the dynamics on the closest potential game. Hence, the characterization of the limiting behavior for many of the learning algorithms for potential games immediately extend to near potential games.

3.3 Beyond Potential Games and Equilibria: Efficient Action Profiles

The discussion thus far has been limited to potential games and convergence to Nash equilibrium. Nonetheless, there is an extensive body of work that discusses convergence to broader classes of games (e.g., weakly-acyclic games) or alternative solution concepts (e.g., coarse and correlated equilibria). See [27, 87] for an extensive discussion. In this section, we depart from the preceding discussion on learning in games two ways. First, we do not impose a particular structure on the game[9]. Second, we focus on convergence to efficient joint actions, whether or not they may be an equilibrium of the underlying game. In doing so, we continue to exploit the prescriptive emphasis of the engineering agenda by treating the learning dynamics as a design element.

3.3.1 Learning Efficient Pure Nash Equilibria

We begin by reviewing the "mood-based" learning algorithms introduced in [68, 88]. For any finite n-player "interdependent" game where a pure Nash equilibrium exists, this algorithm guarantees (probabilistic) convergence to the pure Nash equilibrium that maximizes the sum of the agents' payoffs while adhering to a payoff-based information structure. Before stating the algorithm, we introduce the following definition of interdependence.

Definition 3.1 (Interdependence, [88]) *An n-person game G on the finite action space $\mathcal{A}$ is interdependent if, for every $a \in \mathcal{A}$ and every proper subset of agents $J \subset N$, there exists an agent $i \notin J$ and a choice of actions $a_J' \in \prod_{j \in J} \mathcal{A}_j$ such that $U_i(a_J', a_{-J}) \neq U_i(a_J, a_{-J})$.*

Roughly speaking, the interdependence condition states that it is not possible to divide the agents into two distinct subsets that do not mutually interact with one another.

[9] beyond the forthcoming technical connectivity assumption of "interdependence".

We will now present the version of the learning algorithm introduced in [68], which leads to *efficient* Nash equilibria. Without loss of generality we shall focus on the case where agent utility functions are strictly bounded between 0 and 1, i.e., for any agent $i \in N$ and action profile $a \in \mathcal{A}$ we have $1 > U_i(a) \geq 0$. As with the simple experimentation dynamics, each agent $i \in N$ maintains an evolving local state variables, now given by the triple $[\bar{a}_i, \bar{u}_i, m_i]$. These variables represent

- a *benchmark action* of agent i, $\bar{a}_i \in \mathcal{A}_i$.
- a *benchmark utility* of agent i, $\bar{u}_i$, which is in the range of $U_i(\cdot)$.
- a *mood* of agent i, $m_i \in \{C, D, H, W\}$. We will refer to the mood C as "content", D as "discontent", H as "hopeful", and W as "watchful".

The algorithm proceeds as follows:

1. **Initialization:** At stage $t = 0$, each player randomly selects and plays any action, $a_i(0)$. This action will be initially set as the player's *baseline action* at stage 1, i.e., $\bar{a}_i(1) = a_i(0)$. Likewise, the player's *baseline utility* at stage 1 is initialized as $u_i(1) = U_i(a(0))$. Finally, the player's *mood* at stage 1 is set as $m_i(1) = C$.

2. **Action Selection:** At each subsequent stage $t > 0$, each player selects his action according to the following rules. Let $x_i(t) = [\bar{a}_i, \bar{u}_i, m_i]$ be the state of agent i at time t. If the mood of agent i is content, i.e., $m_i = C$, the agent chooses an action $a_i(t)$ according to the following probability distribution

$$p_i^{a_i}(t) = \begin{cases} \frac{\epsilon}{|\mathcal{A}_i|-1} & \text{for } a_i \neq \bar{a}_i \\ 1-\epsilon & \text{for } a_i = \bar{a}_i \end{cases} \tag{14}$$

where $|\mathcal{A}_i|$ represents the cardinality of the set $\mathcal{A}_i$ and c is a constant that satisfies $c > n$. If the mood of agent i is discontent, i.e., $m_i = D$, the agent chooses an action a_i according to the following probability distribution

$$p_i^{a_i}(t) = \frac{1}{|\mathcal{A}_i|} \text{ for every } a_i \in \mathcal{A}_i \tag{15}$$

Note that the benchmark action and utility play no role in the agent dynamics when the agent is discontent. Lastly, if the agent is either hopeful or watchful, i.e., $m_i = H$ or $m_i = W$, the agent chooses an action $a_i(t)$ according to the following probability distribution

$$p_i^{a_i}(t) = \begin{cases} 0 & \text{for } a_i \neq \bar{a}_i \\ 1 & \text{for } a_i = \bar{a}_i \end{cases} \tag{16}$$

3. **Baseline Action, Baseline Utility, and Mood Update:** Once the agent selects an action $a_i(t) \in \mathcal{A}_i$ and receives the payoff $u_i(t) = U_i(a_i(t), a_{-i}(t))$, where $a_{-i}(t)$ is the action selected by all agents other than agent i at stage t, the state is updated according to the following rules. First, if the state of agent i at time t is $x_i(t) = [\bar{a}_i, \bar{u}_i, C]$ then the state $x_i(t+1)$ is derived from the following transition:

$$x_i(t) = [\bar{a}_i, \bar{u}_i, C] \longrightarrow x_i(t+1) = \begin{cases} [\bar{a}_i, \bar{u}_i, C] & \text{if } \ a_i(t) = \bar{a}_i, u_i(t) = \bar{u}_i, \\ [\bar{a}_i, u_i(t), H] & \text{if } \ a_i(t) = \bar{a}_i, u_i(t) > \bar{u}_i, \\ [\bar{a}_i, u_i(t), W] & \text{if } \ a_i(t) = \bar{a}_i, u_i(t) < \bar{u}_i, \\ [a_i(t), u_i(t), C] & \text{if } \ a_i(t) \neq \bar{a}_i, u_i(t) > \bar{u}_i, \\ [\bar{a}_i, \bar{u}_i, C] & \text{if } \ a_i(t) \neq \bar{a}_i, u_i(t) \leq \bar{u}_i. \end{cases}$$

Second, if the state of agent i at time t is $x_i(t) = [\bar{a}_i, \bar{u}_i, D]$ then the state $x_i(t+1)$ is derived from the following (probabilistic) transition:

$$x_i(t) = [\bar{a}_i, \bar{u}_i, D] \longrightarrow x_i(t+1) = \begin{cases} [a_i(t), u_i(t), C] & \text{with probability } \ \epsilon^{1-u_i(t)}, \\ [a_i(t), u_i(t), D] & \text{with probability } \ 1 - \epsilon^{1-u_i(t)}. \end{cases}$$

Third, if the state of agent i at time t is $x_i(t) = [\bar{a}_i, \bar{u}_i, H]$ then the state $x_i(t+1)$ is derived from the following transition:

$$x_i(t) = [\bar{a}_i, \bar{u}_i, H] \longrightarrow x_i(t+1) = \begin{cases} [a_i(t), u_i(t), C] & \text{if } \ u_i(t) \geq \bar{u}_i, \\ [a_i(t), u_i(t), W] & \text{if } \ u_i(t) < \bar{u}_i. \end{cases}$$

Lastly, if the state of agent i at time t is $x_i(t) = [\bar{a}_i, \bar{u}_i, W]$ then the state $x_i(t+1)$ is derived from the following transition:

$$x_i(t) = [\bar{a}_i, \bar{u}_i, W] \longrightarrow x_i(t+1) = \begin{cases} [a_i(t), u_i(t), H] & \text{if } \ u_i(t) > \bar{u}_i, \\ [a_i(t), u_i(t), D] & \text{if } \ u_i(t) \leq \bar{u}_i. \end{cases}$$

4. Return to Step 2 and repeat.

The above algorithm ensures convergence, in a stochastic stability sense, to the pure Nash equilibrium which maximizes the sum of the agents' payoffs. Before stating the theorem, we introduce the notation $NE(G)$ to represent the set of action profiles that are pure Nash equilibria of the game G.

Theorem 3.5 (Pradelski and Young, 2011 [68]) *Let G be a finite n-player interdependent game where a pure Nash equilibrium exists. Under the above algorithm, given any probability $p < 1$, there exists an exploration rate $\epsilon > 0$ (sufficiently small), such that for sufficiently large times t, $a(t) \in \arg\max_{a \in NE(G)} \sum_{i \in N} U_i(a)$ of G with at least probability p.*

3.3.2 Learning Pareto Efficient Action Profiles

One of the main issues regarding the asymptotic guarantees associated with the learning algorithm given in [68] is that the system performance associated with the best pure Nash equilibrium may be significantly worse than the optimal system performance, i.e., the system performance associated with optimal action profile. Accordingly, it would be desirable if the algorithm guarantees convergence to the action profile which maximizes the sum of the agents' utilities irrespective of whether this action profile constitutes a pure Nash equilibrium. We will now present a learning algorithm, termed *Distributed Learning for Pareto Optimality*, that builds on the developments in [68] and accomplishes such a task. As above, we shall focus on the case where agent utility functions are strictly bounded between 0 and 1. Consequently, for any action profile $a \in \mathcal{A}$ we have $n > \sum_{i \in N} U_i(a) \geq 0$. As with the dynamics presented in [68], each agent $i \in N$ maintains an evolving local state variable given by the triple $[\bar{a}_i, \bar{u}_i, m_i]$. These variables represent

- a *benchmark action* of agent i, $\bar{a}_i \in \mathcal{A}_i$.
- a *benchmark utility* of agent i, $\bar{u}_i$, which is in the range of $U_i(\cdot)$.
- a *mood* of agent i, $m_i \in \{C, D\}$. The moods "hopeful" and "watchful" are no longer used in this setting.

Distributed Learning for Pareto Optimality proceeds as follows:

1. **Initialization:** At stage $t = 0$, each player randomly selects and plays any action, $a_i(0)$. This action will be initially set as the player's *baseline action* at stage 1, i.e., $\bar{a}_i(1) = a_i(0)$. Likewise, the player's *baseline utility* at stage 1 is initialized as $u_i(1) = U_i(a(0))$. Finally, the player's *mood* at stage 1 is set as $m_i(1) = C$.

2. **Action Selection:** At each subsequent stage $t > 0$, each player selects his action according to the following rules. If the mood of agent i is content, i.e., $m_i(t) = C$, the agent chooses an action $a_i(t)$ according to the following probability distribution

$$p_i^{a_i}(t) = \begin{cases} \frac{\epsilon^c}{|\mathcal{A}_i| - 1} & \text{for } a_i \neq \bar{a}_i \\ 1 - \epsilon^c & \text{for } a_i = \bar{a}_i \end{cases} \tag{17}$$

   where $|\mathcal{A}_i|$ represents the cardinality of the set $\mathcal{A}_i$ and c is a constant that satisfies $c > n$. If the mood of agent i is discontent, i.e., $m_i(t) = D$, the agent chooses an action a_i according to the following probability distribution

$$p_i^{a_i}(t) = \frac{1}{|\mathcal{A}_i|} \quad \text{for every } \; a_i \in \mathcal{A}_i \tag{18}$$

   Note that the benchmark action and utility play no role in the agent dynamics when the agent is discontent.

3. **Baseline Action, Baseline Utility, and Mood Update:** Once the agent selects an action $a_i(t) \in \mathcal{A}_i$ and receives the payoff $U_i(a_i(t), a_{-i}(t))$, where $a_{-i}(t)$ is the action selected by all agents other than agent i at stage t, the state is updated according to the following rules. First, the baseline action and baseline utility at stage $t+1$ are set as

$$\begin{aligned} \bar{a}_i(t+1) &= a_i(t), \\ \bar{u}_i(t+1) &= U_i(a_i(t), a_{-i}(t)). \end{aligned}$$

The mood of agent i is updated as follows.

3a. If

$$\begin{bmatrix} \bar{a}_i(t) \\ \bar{u}_i(a(t)) \\ m_i(t) \end{bmatrix} = \begin{bmatrix} a_i(t) \\ U_i(a(t)) \\ C \end{bmatrix},$$

then $m_i(t+1) = C$.

3b. Otherwise,

$$m_i(t+1) = \begin{cases} C & \text{with probability } \epsilon^{1-U_i(a(t))} \\ D & \text{with probability } 1 - \epsilon^{1-U_i(a(t))} \end{cases}$$

4. Return to Step 2 and repeat.

Theorem 3.6 (Marden et al., 2011 [55]) *Let G be a finite n-player interdependent game. Under Distributed Learning for Pareto Optimality, given any probability $p < 1$, there exists an exploration rate $\epsilon > 0$ (sufficiently small), such that for sufficiently large stages t, $a(t) \in \arg\max_{a \in \mathcal{A}} \sum_{i \in N} U_i(a)$ of G with at least probability p.*

Distributed Learning for Pareto Optimality guarantees probabilistic convergence to the action profile that maximizes the sum of the agents' utility functions. As stated earlier, the maximizing action profile need *not* be a Nash equilibrium. Accordingly, in games such as the classical prisoner's dilemma game, this algorithm provides convergence to the action profile where each player cooperates even though this is a strictly dominated strategy. Likewise, for the aforementioned application of wind farm optimization, where each turbine's utility function represents the power generated by that turbine, Distributed Learning for Pareto Optimality guarantees convergence to the action profile that optimizes the total power production in the wind farm. As a consequence of the payoff-based information structure, this algorithm also demonstrates that optimizing system performance in wind farms does not require a characterization of the aerodynamic interaction between the turbines nor global information available to the turbines.

4 Exploiting the Engineering Agenda: State-Based Games

When viewed as dynamical systems, distributed learning algorithms are all described in terms of an underlying evolving state. In most cases, this state has an immediate interpretation in terms of the primitive elements of the game (e.g., empirical frequencies in Fictitious Play or immediately preceding actions in log-linear learning). In other cases, the state variable may be better interpreted as an auxiliary variable, not necessarily related to actions and payoffs. Rather, these variables are introduced into the dynamics to evoke desirable limiting behavior. One example is the "mood" in Distributed Learning for Pareto Optimality. Similarly, reference [73] illustrates how auxiliary states can overcome fundamental limitations in learning [31]. The introduction of such states again reflects the available degrees of freedom in the engineering agenda in that these variables need not have interpretations naturally relevant to the associated game.

In this section, we continue to explore and exploit this addition degree of freedom in the defining the game itself, and in particular, through a departure from utility design for normal form games. We begin this section by reviewing some of the limitations associated with the framework of strategic form games for distributed control. Next, we review the framework of *state based games*, introduced in [45], which represents a simplification of the framework of Markov games and is better suited to address the constraints and objective inherent to engineered systems. The key distinction between strategic form games and state based games is the introduction of an underlying state space into the game theoretic environment. Here, the state space presents the system designer with additional design freedom to address issues pertinent to distributed engineered systems. We conclude this section by illustrating how this additional state can be exploited in distributed control.

4.1 Limitations of Strategic Form Games

In this section we review two limitations of strategic form games for distributed engineered systems. The first limitation concerns the complexity associated with utility design. The second limitation concerns on the applicability of strategic form games for distributed optimization.

4.1.1 Limitations of Protocol Design

The marginal contribution (Theorem 2.1) and the weighted Shapley value (Theorem 2.2) represent two *universal* methodologies for utility design in distributed engineering system. By universal, we mean that these methodologies will ensure that the resulting game is a (weighted) potential game irrespective of the resource set $\mathcal{R}$, the structure of the objective functions $\{W_r\}_{r \in \mathcal{R}}$, or the structure of the agents' action sets $\{\mathcal{A}_i\}_{i \in N}$. Here, universality is of fundamental importance by allowing design methodologies to be applicable to a wide array of different applications.

Consider the weighted Shapley value protocol, defined in (3), which always ensures the existence of a pure Nash equilibrium through the use of budget-balanced utility functions. Budget-balanced utility functions play an important role for providing desirable efficiency guarantees as highlighted by the conditions set forth in Theorems 2.3 and 2.5. However, utilizing the weighted Shapley value protocol is computationally prohibitive in large systems as it requires computing a summation of an exponential number of terms. Are there alternative methodologies that guarantee the existence of a pure Nash equilibrium through the use of budget-balanced utility functions? The following theorem proves that the answer is no.

Theorem 4.1 (Marden and Wierman, 2011 [53]) *Let $\mathcal{G}$ be the set of welfare sharing games. A set of protocols $\{f_r(\cdot)\}$ is budget-balanced and guarantees the existence of any equilibrium in any game $G \in \mathcal{G}$ if and only if the protocol is a weighted Shapley value.*

Theorem 4.1 builds on work in [16] which derives a similar result for the analogous class of cost sharing games. This theorem proves that the only universal methodology that guarantees the existence of an equilibrium in any game through the use of budget-balanced protocols is the weighted Shapley value protocol. An alternative interpretation of Theorem 4.1 is the following: If a protocol is budget-balanced and does not represent a weighted Shapley value, then there exists a game $G \in \mathcal{G}$ such that an equilibrium does not exist. The value of this theorem is that it provides a *complete characterization* of the protocol design space that a system designer needs to consider when trying to design budget-balanced protocols that ensure the existence of a pure Nash equilibrium. In essence, this design space is completely parameterized by player weights $\{\omega_i\}_{i \in N}$. It is important to highlight that there are currently no characterizations for the complete class of protocols that ensure the existence of an equilibrium with relaxed budget-balanced constraints.

A second limitation associated with utility design for engineered systems focuses on the PoS when using budget-balanced protocols. Theorem 4.1 proves that the weighted Shapley value protocol is necessary for ensuring the existence of an equilibrium in such settings. The following theorem demonstrates that this requirement not only increases the computation complexity of such a task, but also comes with a degradation in the PoS.

Theorem 4.2 (Marden and Wierman, 2011 [53]) *Let $\mathcal{G}$ be the set of welfare sharing games with submodular objective functions and a fixed weighted Shapley value protocol. The PoS across the set of games $\mathcal{G}$ is* 2*.*

This theorem proves that in general it is impossible to guarantee that the optimal allocation is an equilibrium when using budget-balanced protocols. This is in contrast to non-budget balanced protocols, e.g., the marginal contribution protocol, which can achieve a PoS of 1 for such settings. Note that both the marginal contribution protocol and the weighted Shapley value protocol guarantee a PoA of 2 when restricting attention to welfare sharing games with submodular objective functions since both protocols satisfy the conditions of Theorem 2.3.

4.1.2 Distributed Optimization: Consensus

An extensively studied problem in distributed control is that of agreement and consensus [36,67,79]. In such consensus problems, there is a group of agents, N, and each agent $i \in N$ is endowed with an initial value $a_i(0) \in \mathbb{R}$. Agents update these values over stages, $t = 0, 1, 2, ...$ The goal is for each agent to compute the average of the initial endowments. Each agent i is only able to communicate with neighboring agents, specified by the subset $N_i \subseteq N$. Define the *interaction graph* as the graph formed by the nodes N and edges $E = \{(i, j) \in N \times N : j \in N_i\}$. The challenge in consensus problems is then to design update rules of the form

$$a_i(t) = F_i\left(\{\text{information about agent } j \text{ at time } t\}_{j\in N_i}\right) \tag{19}$$

so that $\lim_{t\to\infty} a_i(t) = a^* = \frac{1}{n}\sum_{i\in N} a_i(0)$, which represents the solution to the optimization problem

$$\begin{array}{cl} \max_{a\in\mathbb{R}^n} & -\frac{1}{2}\sum_{i\in N, j\in N_i} ||a_i - a_j||_2^2 \\ \text{s.t.} & \sum_{i\in N} a_i = \sum_{i\in N} a_i(0) \end{array} \tag{20}$$

provided that the interaction graph is connected. Furthermore, the control laws $\{F_i(\cdot)\}_{i\in N}$ should achieve the desired asymptotic guarantees for any initial value profile $a(0)$ and any connected interaction graph. This implies that the underlying control design must be invariant to these parameters in addition to the specific indices assigned to the agents.

One algorithm (among many variants) that achieves asymptotic consensus is distributed averaging [36,67,79], given by

$$a_i(t) = a_i(t-1) + \epsilon \sum_{j\in N_i} (a_j(t-1) - a_i(t-1)),$$

where $\epsilon > 0$ is a step-size. This algorithm imposes the constraint that for all times $t \geq 0$,

$$\sum_{i\in N} a_i(t) = \sum_{i\in N} a_i(0),$$

i.e., the average value is invariant. Hence, if the agents reach consensus on a common value, this value must represent the average of the initial values.

While the above description makes no explicit reference to games, there is considerable overlap with the present discussion of games and learning. Reference [47]) discusses how consensus and agreement problems can be viewed within the context of games and learning. However, the discussion is largely restricted to only asymptotic agreement, i.e., consensus but not necessarily to the original average.

Since most of the aforementioned learning rules converge to a Nash equilibrium, one could attempt to assign each agent $i \in N$ an admissible utility function such that (i) the resulting game is a potential game and (ii) all resulting equilibria solve the optimization problem in (20). To ensure scalability properties, we focus on meeting these objective using "spatially invariant" utility functions of the following form

$$U_i(a) \triangleq \mathcal{U}\left(\{a_j, a_j(0)\}_{j \in N_i}\right) \tag{21}$$

where the function $\mathcal{U}(\cdot)$ is invariant to specific indices assigned to agents. Note that the design of $\mathcal{U}(\cdot)$ leads to a well defined game irrespective of the agent set, N, initial value profile, $a(0)$, or the structure of the interaction graph $\{N_i\}_{i \in N}$. The following theorem demonstrates that it is *impossible* to design $\mathcal{U}(\cdot)$ such that for any game induced by an initial value profile and an undirected and connected interaction graph all resulting Nash equilibria solve the optimization problem in (20).

Theorem 4.3 (Na and Marden, 2011 [42]) *There does not exist a single $\mathcal{U}(\cdot)$ such that for any game induced by a connected and undirected interaction graph formed by the information sets $\{N_i\}_{i \in N}$, an initial value profile $a(0)$, and agents' utility functions of the form (21), the Nash equilibria of the induced game represent solutions to the optimization problem in (20).*

This theorem demonstrates that the framework of strategic form games is not rich enough to meet the design considerations pertinent to distributed engineered systems. While this limitation was illustrated here on the consensus problem, one might imagine that various other system level objectives could have similar limitations.

4.2 State Based Games

In this section we review the framework of *state based games*, introduced in [45], which represents an extension to the framework of strategic form games where an underlying state space to the game theoretic framework.[10] Here, the state is introduced as a coordinating device used to improve system level behavior. A (deterministic) state based game consists of the following elements:

(i) A set of agents N.

(ii) An underlying state space X.

(iii) A state-dependent action set $\mathcal{A}_i(x)$ for each agent $i \in N$ and state $x \in X$.

(iv) A state-dependent utility function $U_i : X \times \mathcal{A} \to R$ for each agent $i \in N$ where $\mathcal{A}_i = \cup_{x \in X} \mathcal{A}_i(x)$ and $\mathcal{A} = \prod_{i \in N} \mathcal{A}_i$.

(v) A deterministic state transition function $P : X \times \mathcal{A} \to X$.

[10] State based games represent a simplification of the class of Markov games [75] where the key difference lies in the discount factor associated with future payoffs. In Markov games, an agent's utility represents a discounted sum of future payoffs. Alternatively, in state based games, an agent's utility represents only the current payoff, i.e., the discount factor is 0. This difference greatly simplifies the analysis of such games.

Repeated play of a state based game produces a sequence of action profiles $a(0)$, $a(1)$, ..., and a sequence of states $x(0)$, $x(1)$, ..., where $a(t) \in \mathcal{A}$ is referred to as the action profile at time t and $x(t) \in X$ is referred to as the state at time t. The sequence of actions and states is generated according to the following process: at any time $t \geq 0$, each agent $i \in N$ *myopically* selects an $a_i(t) \in \mathcal{A}_i$, optimizing only the agent's potential payoff at time t. The state $x(t)$ and the action profile $a(t) := (a_1(t), \dots, a_n(t))$ together determine each agent's payoff $U_i(x(t), a(t))$ at time t. After all agents select their respective action, the ensuing state $x(t+1)$ is chosen according to the state transition function $x(t+1) = P(x(t), a(t))$ and the process is repeated.

The framework of state based games can be exploited to model phenomena that are of interest to distributed engineered systems. For example, a common assumption in game theory is that an agent can select any action in the agent's action set at any instance in time. In multiagent systems this assumption is not necessarily true. Rather, each agent has the ability to influence his action through different control strategies. Accordingly, we consider the situation where each agent $i \in N$ has a set of control strategies Π_i that the agent can use to influence the agent's action choice. Let $\Pi := \prod_i \Pi_i$ be the set of joint control strategies. We represent the action transition function of agent i by a deterministic (or stochastic) transition function $g_i : \mathcal{A}_i \times \Pi \rightarrow \mathcal{A}_i$. In a repeated state based game, we adopt the convention that $a_i(t+1) = g_i(a_i(t), \pi(t))$ for any agent $i \in N$ and time $t \geq 1$ where $\pi(t) = (\pi_1(t), ..., \pi_n(t))$ is the joint control decision at time t. This implies that an agent's ensuing action is potentially influenced by the control strategies of all agents. We will focus on the case where each agent has a null control strategy $\pi_i^0 \in \Pi_i$ such that for any agent $i \in N$ and action $a_i \in \mathcal{A}_i$ we have $a_i = g_i(a_i, \pi^0)$ where $\pi^0 = (\pi_1^0, ..., \pi_n^0)$.

Incorporating control-based decisions into the state based game framework requires the following: First, we embed the action profiles $\mathcal{A}$ and the original state space X into a new space $Y = \mathcal{A} \times X$. Each agent $i \in N$ now has a state invariant control strategy set Π_i as described above and a new state dependent utility function of the form $U_i : Y \times \Pi \rightarrow \mathbb{R}$. Lastly, the deterministic state transition function is now of the form $Q : Y \times \Pi \rightarrow Y$ which encompasses both the previous state transition function $P(\cdot)$ and the new action transition functions $\{g_i(\cdot)\}$. Repeated play of a state based game proceeds in the same fashion as before with the sole exception that action profiles $a(t)$ are replaced with control profile $\pi(t)$. We denote such a state based games G by the tuple $G = \{N, \{\Pi_i\}, \{U_i\}, Y, Q\}$.

We begin by introducing a class of games, termed *state based potential games* [42], which represent an extension of potential games to the framework of state based games.

Definition 4.1 (State Based Potential Game, [42]) *A state based game $G = \{N, \{\Pi_i\}, \{U_i\}, Y, Q\}$ is a state based potential game if there exists a potential function $\Phi : Y \times \Pi \rightarrow \mathbb{R}$ that satisfies the following two properties for every state $y \in Y$ and control profile $\pi \in \Pi$:*

1. *For any agent $i \in N$ with control $\pi_i' \in \Pi_i$*

$$U_i(y, \pi_i', \pi_{-i}) - U_i(y, \pi) = \Phi(y, \pi_i', \pi_{-i}) - \Phi(y, \pi)$$

2. *The potential function satisfies $\Phi(\tilde{y}, \pi^0) = \Phi(y, \pi)$ for the state $\tilde{y} = Q(y, \pi^0)$*

The first condition states that each agent's utility function is aligned with the potential function in the same fashion as in potential games [58]. The second condition relates to the evolution on the potential function along the state trajectory. We focus on the class of state based potential games since dynamics can be derived which converge to the following class of equilibria:

Definition 4.2 (Stationary State Nash Equilibrium, [42]) *A state action pair $[y^*, \pi^*]$ is a stationary state Nash equilibrium if*

1. *For any agent $i \in N$ we have $\pi_i^* \in \arg\max_{\pi_i \in \Pi_i} U_i(y^*, \pi_i, \pi_{-i}^*)$.*

2. *The state is a fixed point of the state transition function, i.e., $y^* = f(y^*, \pi^*)$.*

It can be shown that a stationary state Nash equilibrium is guaranteed to exist in any state based potential game [42]. Furthermore, there are several learning dynamics which will converge to such an equilibrium in state based potential games [42, 45].

4.3 Illustrations

4.3.1 Protocol Design

Section 4.1.1 highlight computational and efficiency limitations associated with designing protocols within the framework of strategic form games. Recently in [53], the authors shows that there exists a simple state-based protocol that overcomes both of these limitations. In particular, for welfare sharing games with submodular objective functions this state-based protocol is universal, budget-balanced, tractable, and ensures the existence of a stationary state Nash equilibrium. Furthermore, the PoS is 1 and PoA is 2 when using this state-based protocol. Hence, this protocol matches the performance of the marginal contribution protocol with respect to efficiency guarantees. We direct the readers to [53] for the specific details regarding this protocol.

4.3.2 Distributed Optimization

Consider the following generalization of the average consensus problem where there exists a set of agents N, an action set $\mathcal{A}_i = \mathbb{R}$ for each agent $i \in N$, a system level objective function $W : \mathcal{A} \to R$ which is concave and continuously differentiable, and a coupled constraint on the agents' action profile which is characterized by a set of m-linear inequalities represented in matrix form as $Za \leq C$. Here, the goal is to establish a set of local control laws of the form (19) such that the joint action profile converges to the solution of the following optimization problem

$$\begin{array}{ll} \max_{a \in \mathbb{R}^n, i \in N} & W(a) \\ s.t. & \sum_{i=1}^{n} Z_i^k a_i - C^k \leq 0, \quad k \in \{1, \ldots, m\} \end{array} \tag{22}$$

Here, the interaction graph encodes the desired locality in the control laws. Note that the objective for average consensus in (20) is a special case of the objective presented in (22).

Section 4.1.2 demonstrates that it is impossible to design scalable agent utility functions within the framework of strategic form games which ensured that all equilibria of the resulting game represented solutions to the optimization problem in (22). We will now review the methodologies developed in [42, 43] which accomplishes this task using the framework of state based games. Furthermore, the forthcoming design also ensures that the resulting game is a state based potential game; hence, there are available distributed learning algorithms for reaching the stationary state Nash equilibria of the resulting game [42,45]. The details of the design are as follows:

Agents: The agent set is $N = \{1, 2, ..., n\}$.

States: The starting point of the design is an underlying state space Y where each state $y \in Y$ is defined as a tuple $y = (a, e, c)$, where the components are as follows:

- The term $a = (a_1, \ldots, a_n) \in \mathbb{R}^n$ is the action profile.
- The term $e = (e_1, \ldots, e_n)$ is a profile of agent based estimation terms for the action profile a. Here, $e_i = (e_i^1, \ldots, e_i^n) \in \mathbb{R}^n$ is agent i's estimation for the joint action profile a. The term e_i^k captures agent i's estimate of agent k's action a_k.
- The term $c = (c_1, \ldots, c_n)$ is a profile of agent based estimation terms for the constraint violations. Here, $c_i = (c_i^1, \ldots, c_i^m) \in \mathbb{R}^m$ is agent i's estimation for the constraint violation $C - Za$. The term c_i^k captures agent i's estimate of the violation of the k-th constraint, i.e., $\sum_{j \in N} Z_{kj} a_j - C^k$.

Action Sets: Each agent $i \in N$ is assigned a set of control policies Π_i that permits the agents to change their value and change their estimation terms through communication with neighboring agents. Specifically, a control for agent i is defined as a tuple $\pi_i = (\hat{a}_i, \hat{e}_i, \hat{c}_i)$ whose components are as follows:

- The term $\hat{a}_i \in \mathbb{R}$ indicates a change in the agent's value.
- The term $\hat{e}_i := (\hat{e}_i^1, \cdots, \hat{e}_i^n)$ indicates a change in the agent's "action profile" estimation terms e_i. Here, $\hat{e}_i^k := \{\hat{e}_{i \to j}^k\}_{j \in N_i}$ where $\hat{e}_{i \to j}^k \in \mathbb{R}$ represents the estimation value that agent i "passes" to agent j regarding the action of agent k.
- The term $\hat{c}_i := (\hat{c}_i^1, \cdots, \hat{c}_i^m)$ indicates a change in the agent's "constraint violation" estimation terms c_i. Here, $\hat{c}_i^k := \{\hat{c}_{i \to j}^k\}_{j \in N_i}$ where $\hat{c}_{i \to j}^k \in \mathbb{R}$ represents the estimation value that agent i "passes" to agent j regarding the violation of the k-th constraint.

State Dynamics: We now describe how the state evolves as a function of the control choices $\pi(0)$, $\pi(1)$, ..., where $\pi(k)$ is the control profile at stage k. Define the initial state as $y(0) = [a(0), e(0), c(0)]$ where $a(0) = (a_1(0), ..., a_n(0))$ is the initial action profile, $e(0)$ is an initial estimation profile that satisfies

$$\sum_{i \in N} e_i^k(0) = n \cdot a_k(0), \tag{23}$$

for each agent $k \in N$, and $c(0)$ is an initial estimate of the constraint violations that satisfies

$$\sum_{i \in N} c_i^k(0) = \sum_{i \in N} Z_i^k \cdot a_i(0), \tag{24}$$

for each agent $k \in M$. Hence, the initial estimation terms are contingent on the initial action profile. We represent the state transition function $Q(\pi, y)$ by a set of local state transition functions $\{Q_i^a(y,\pi)\}_{i \in N}$, $\left\{Q_{i,j}^e(y,\pi)\right\}_{i,j \in N}$, and $\left\{Q_{i,k}^c(y,\pi)\right\}_{i \in N, k \in M}$. For any agent $i \in N$, state $y = (a, e, c)$, and control $\pi = (\hat{a}, \hat{e}, \hat{c})$ the state transition function pertaining to the action profile takes on the form

$$Q_i^a(y, \pi) \;\; = \;\; a_i + \hat{a}_i.$$

For any distinct agents $i, k \in N$, state $y = (a, e, c)$, and control $\pi = (\hat{a}, \hat{e}, \hat{c})$ the state transition function pertaining to the estimate of the action profile takes on the form

$$\begin{aligned} Q_{i,i}^e(\pi, y) &= e_i^i + n \cdot \hat{a}_i + \sum_{j \in N : i \in N_j} \hat{e}_{j \to i}^i - \sum_{j \in N_i} \hat{e}_{i \to j}^i \\ Q_{i,k \neq i}^e(y, \pi) &= e_i^k + \sum_{j \in N : i \in N_j} \hat{e}_{j \to i}^k - \sum_{j \in N_i} \hat{e}_{i \to j}^k. \end{aligned} \tag{25}$$

It is straightforward to show that for *any* sequence of control choices $\pi(0)$, $\pi(1)$, ..., the resulting state trajectory $y(t) = (a(t), e(t)) = Q(\pi(t-1), y(t-1))$ satisfies for all times $t \geq 1$ and agents $k \in N$

$$\sum_{i=1}^{n} e_i^k(t) = n \cdot a_k(t). \tag{26}$$

Lastly, for any agent $i \in N$, constraint $k \in M$, state $y = (a, e, c)$, and control $\pi = (\hat{a}, \hat{e}, \hat{c})$ the state transition function pertaining to the estimate of the constraint violations takes on the form

$$Q_{i,k}^c(y, \pi) \;\; = \;\; c_i^k + Z_i^k \hat{a}_i + \sum_{j \in N : i \in N_j} \hat{c}_{j \to i}^k - \sum_{j \in N_i} \hat{c}_{i \to j}^k$$

It is straightforward to show that for any sequence of action profiles $a(0), a(1), \ldots$, the resulting state trajectory $y(t) = (a(t), e(t), c(t)) = Q(y(t-1), \pi(t-1))$ satisfies

$$\sum_{i \in N} c_i^k(t) = \sum_{i \in N} z_i^k a_i(t) - C^k \tag{27}$$

for all $t \geq 1$ and constraints $k \in M$. Therefore

$$\sum_{i \in N} c_i^k(t) \leq 0 \Leftrightarrow \sum_{i \in N} z_i^k a_i(t) - C^k \leq 0 \tag{28}$$

for any constraint $k \in M$. Hence, the estimation terms encode information regarding whether the constraints are violated.

Agent Utility Functions: The last part of our design is the agents' utility functions. For any state $y \in Y$ and admissible control $\pi \in \Pi$ the utility function of agent i is defined as

$$U_i(y, \pi) = \sum_{j \in N_i} W(\tilde{e}_j^1, \tilde{e}_j^2, ..., \tilde{e}_j^n) - \sum_{j \in N_i} \sum_{k \in N} \left[\tilde{e}_i^k - \tilde{e}_j^k\right]^2 - \mu \sum_{j \in N_i} \sum_{k=1}^{m} \left[\max\left(0, \tilde{c}_j^k\right)\right]^2$$

where $\mu > 0$ and $(\tilde{v}, \tilde{e}, \tilde{c}) = Q(y, \pi)$ represents the ensuing state. Note that the agents' utility functions are both local and scalable.

Theorem 4.4 (Li and Marden, 2011 [42, 43]) *Consider the state based game depicted above. The designed game is a state based potential game with potential function*

$$\Phi(y, \pi) = \sum_{i \in N} W(\tilde{e}_i^1, \tilde{e}_i^2, ..., \tilde{e}_i^n) - \frac{1}{2} \sum_{i \in N} \sum_{j \in N_i} \sum_{k \in N} \left[\tilde{e}_i^k - \tilde{e}_j^k\right]^2 - \mu \sum_{j \in N} \sum_{k=1}^{m} \left[\max\left(0, \tilde{c}_j^k\right)\right]^2$$

where $(\tilde{a}, \tilde{e}, \tilde{c}) = Q(y, \pi)$ represents the ensuing state and $\mu > 0$. Furthermore, if the interaction graph is connected, undirected, and non-bipartite, then a state action pair $[y, \pi] = [(a, e, c), (\hat{a}, \hat{e}, \hat{c})]$ is a stationary state Nash equilibrium if and only if the following conditions are satisfied:

(i) The action profile a is an optimal point of the unconstrained optimization problem

$$\max_{a \in \mathcal{A}} W(a) - \frac{\mu}{n} \left[\sum_{k \in M} \max\left(0, \sum_{i \in N} Z_i^k a_i - C^k\right)\right]^2. \tag{29}$$

(ii) The estimation of the action profile e is consistent with a, i.e., for any $\forall i, k \in N$ we have $e_i^k = a_k$.

(iii) The estimation of the constraint violations c satisfies the following for any $\forall i \in N$ and $k \in M$

$$\max\left(0, c_i^k\right) = \frac{1}{n} \max\left(0, \sum_{i \in N} Z_i^k a_i - C^k\right).$$

(iv) The change in action profile satisfies $\hat{a}_i = 0$ for all agents $i \in N$.

(v) The net change in estimation for both the action profile and the constraint violation is 0, i.e.,

$$\sum_{j \in N : i \in N_j} \hat{e}_{j \to i}^k - \sum_{j \in N_i} \hat{e}_{i \to j}^k = 0 \qquad \forall i, k \in N,$$

$$\sum_{j \in N : i \in N_j} \hat{c}_{j \to i}^k - \sum_{j \in N_i} \hat{c}_{i \to j}^k = 0 \qquad \forall i \in N, k \in M.$$

This theorem characterizes the complete set of stationary state Nash equilibrium for the designed state-based game. There are several interesting properties regarding this characterization. First, the solutions to the unconstrained optimization problem incorporating penalty functions in (29) in general only represent solutions to the constrained optimization problem in (22) when the tradeoff parameter $\mu \to \infty$. However, in many settings, such as the consensus problem discussed in Section 4.1.2, any finite $\mu > 0$ will provide the equivalence between these two solution sets [42]. Second, the design methodology set forth in this section is universal and provides the desired equilibrium characteristics irrespective of the specific topological structure of the interaction graph or the agents' initial actions/values. Note that this was impossible when using the framework of strategic form games. Lastly, since the designed game represents a state based potential game, there exists learning dynamics which guarantee convergence to a stationary state Nash equilibrium [42].

5 Concluding Remarks

We conclude by mentioning some important topics not discussed in this chapter.

First, there has been work using game theoretic formulations for engineering applications over many decades. Representative topics include cybersecurity [2, (2010)], wireless networks [78, (2005)], robust control design [8, (1995)], team theory [34, (1972)], and pursuit-evasion [33, (1965)]. The material in this chapter focused on more recent trends that emphasize both game design and adaptation through learning in games.

Second is the issue of convergence rates. We reviewed how various learning rules under different information structures can converge asymptotically to Nash equilibria or other solution concepts. Practical implementation for engineering applications places demands on the requisite convergence rates. Furthermore, existing computational and communication complexity results constrain the limits of achievable performance in the general case [19,28]. Recent work has begun to address settings in which practical convergence is possible [4,38,59,72].

Finally, there is the obvious connection to distributed optimization. A theme throughout the paper is optimizing performance of a global objective function under various assumptions on available information and communication constraints. While the methods herein emphasis a game theoretic approach, there is extensive complementary work on modifying optimization algorithms (e.g., gradient decent, Newton's method, etc) to accommodate distributed architectures. Representative citations are [64,81,82] as well as the classic reference [10] .

References

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

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

[3] I. Arieli and Y. Babichenko. Average testing and the efficient boundary. Discussion paper, Department of Economics, University of Oxford and Hebrew University, 2011.

[4] I. Arieli and H.P. Young. Fast convergence in population games. University of Oxford Department of Economics Discussion Paper Series, 2011.

[5] K.J. Arrow. Rationality of self and others in an economic system. *The Journal of Buisiness*, **59**(4):S385–S399, 1986.

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

[7] Y Babichenko. Completely uncoupled dynamics and Nash equilibria. working paper, 2010.

[8] T. Basar and P. Bernhard. *$\mathcal{H}^\infty$-Optimal Control and Related Minimax Design Problems: A Dynamic Game Approach*. Birkhäuser, 1995.

[9] J. Bergin and B. L. Lipman. Evolution with state-dependent mutations. *Econometrica*, 64(4):943–956, 1996.

[10] D.P. Bertsekas and J.N. Tsitsiklis. *Parallel and Distributed Computation*. Prentice Hall, 1989.

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

[12] F. Bullo, E. Frazzoli, M. Pavone, K. Savla, and S. L. Smith. Dynamic vehicle routing for robotic systems. *Proceedings of the IEEE*, 99(9):1482–1504, 2011.

[13] U. O. Candogan, A. Ozdaglar, and P. A. Parrilo. Dynamics in near-potential games. Discussion paper, LIDS, MIT, 2011.

[14] U. O. Candogan, A. Ozdaglar, and P. A. Parrilo. Near-potential games: Geometry and dynamics. Discussion paper, LIDS, MIT, 2011.

[15] G.C. Chasparis and J.S. Shamma. Distributed dynamic reinforcement of efficient outcomes in multiagent coordination and network formation. *Dynamic Games and Applications*, **2**(1):18–50, 2012.

[16] H-L. Chen, T. Roughgarden, and G. Valiant. Designing networks with good equilibria. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 854–863, 2008.

[17] V. Conitzer and T. Sandholm. Computing shapley values, manipulating value division schemes, and checking core membership in multi-issue domains. In *Proceedings of AAAI*, 2004.

[18] G. Dan. Cache-to-cache: Could isps cooperate to decrease peer-to-peer content distribution costs? *IEEE Transactions on Parallel and Distributed Systems*, **22**(9):1469–1482, 2011.

[19] C. Daskalakis, P.W. Goldberg, and C.H. Papadimitriou. The complexity of computing a Nash equilibrium. In *STOC'06 Proceedings of the 38th Annual ACM Symposium on the Theory of Computing*, pages 71–78, 2006.

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

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

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

[23] A. Garcia, D. Reaume, and R. Smith. Fictitious play for finding system optimal routings in dynamic traffic networks. *Transportation Research B, Methods*, 34(2):147–156, January 2004.

[24] F. Germano and G. Lugosi. Global Nash convergence of Foster and Young's regret testing. *Games and Economic Behavior*, **60**:135–154, July 2007.

[25] M. Goemans, L. Li, V. S. Mirrokni, and M. Thottan. Market sharing games applied to content distribution in ad-hoc networks. In *Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, 2004.

[26] G. Haeringer. A new weight scheme for the shapley value. *Mathematical Social Sciences*, 52(1):88–98, July 2006.

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

[28] S. Hart and Y. Mansour. The communication complexity of uncoupled Nash equilibrium procedures. In *STOC'07 Proceedings of the 39th Annual ACM Symposium on the Theory of Computing*, pages 345–353, 2007.

[29] S. Hart and A. Mas-Colell. Potential, value, and consistency. *Econometrica*, 57(3):589–614, May 1989.

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

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

[32] S. Hart and A. Mas-Colell. Stochastic uncoupled dynamics and Nash equilibrium. *Games and Economic Behavior*, 57(2):286–303, 2006.

[33] Y.-C. Ho, A. Bryson, and S. Baron. Differential games and optimal pursuit-evasion strategies. *IEEE Transactions on Automatic Control*, **10**(4):385–389, 1965.

[34] Y.-C. Ho and K.-C. Chu. Team decision theory and information structures in optimal contra problems—Part I. *IEEE Transactions on Automatic Control*, **17**(1):15–22, 1972.

[35] 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.

[36] 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.

[37] M. Kandori, G. Mailath, and R. Rob. Learning, mutation, and long-run equilibria in games. *Econometrica*, 61:29–56, 1993.

[38] G.H. Kreindler and H.P. Young. Fast convergence in evolutionary equilibrium selection. University of Oxford Department of Economics Discussion Paper Series, 2011.

[39] D. Leslie and E. Collins. Convergent multiple-timescales reinforcement learning algorithms in normal form games. *Annals of Applied Probability*, 13:1231–1251, 2003.

[40] D. Leslie and E. Collins. Individual Q-learning in normal form games. *SIAM Journal on Control and Optimization*, **44**(2), 2005.

[41] D. Leslie and E. Collins. Generalised weakened fictitious play. *Games and Economic Behavior*, **56**(2):285–298, 2006.

[42] N. Li and J. R. Marden. Decoupling coupled constraints through utility design. Discussion paper, Department of ECEE, University of Colorado, Boulder, 2011.

[43] N. Li and J. R. Marden. Designing games for distributed optimization. Discussion paper, Department of ECEE, University of Colorado, Boulder, 2011.

[44] S. Mannor and J.S. Shamma. Multi-agent learning for engineers. *Artificial Intelligence*, pages 417–422, May 2007. special issue on Foundations of Multi-Agent Learning.

[45] J. R. Marden. State based potential games. Discussion paper, Department of ECEE, University of Colorado, Boulder, 2011.

[46] J. R. Marden, G. Arslan, and J. S. Shamma. Regret based dynamics: Convergence in weakly acyclic games. In *Proceedings of the 2007 International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, Honolulu, Hawaii, May 2007.

[47] J. R. Marden, G. Arslan, and J. S. Shamma. Connections between cooperative control and potential games. *IEEE Transactions on Systems, Man and Cybernetics. Part B: Cybernetics*, 39:1393–1407, December 2009.

[48] 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.

[49] J. R. Marden and T. Roughgarden. Generalized efficiency bounds for distributed resource allocation. In *Proceedings of the 48th IEEE Conference on Decision and Control*, December 2010.

[50] J. R. Marden, S.D. Ruben, and L.Y. Pao. Surveying game theoretic approaches for wind farm optimization. In *Proceedings of the AIAA Aerospace Sciences Meeting*, January 2012.

[51] J. R. Marden and J. S. Shamma. Revisiting log-linear learning: Asynchrony, completeness and a payoff-based implementation. *Games and Economic Behavior*, 2012. to appear.

[52] J. R. Marden and A. Wierman. Distributed welfare games. Discussion paper, Department of ECEE, University of Colorado, Boulder, 2008.

[53] J. R. Marden and A. Wierman. The limitations of utility design for multiagent systems. Discussion paper, Department of ECEE, University of Colorado, Boulder, 2011.

[54] J. R. Marden, H. P. Young, G. Arslan, and J. S. Shamma. Payoff based dynamics for multi-player weakly acyclic games. *SIAM Journal on Control and Optimization*, 48:373–396, February 2009.

[55] J. R. Marden, H. Peyton Young, and L. Y. Pao. Achieving pareto optimality through distributed learning. under submission, 2011.

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

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

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

[59] A. Montanari and A. Saberi. Convergence to equilibrium in local interaction games. In *FOCS'09 Proceedings of the 2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 303–312, 2009.

[60] H. Moulin. An efficient and almost budget balanced cost sharing method. *Games and Economic Behavior*, 70(1):107–131, 2010.

[61] H. Moulin and S. Shenker. Strategyproof sharing of submodular costs: budget balance versus efficiency. *Economic Theory*, 18(3):511–533, 2001.

[62] H. Moulin and R. Vohra. Characterization of additive cost sharing methods. *Economic Letters*, 80(3):399–407, 2003.

[63] R.M. Murray. Recent research in cooperative control of multivehicle systems. *Journal of Dynamic Systems, Measurement, and Control*, **129**(5):571–583, 2007.

[64] A. Nedic and A. Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, **54**(1):48–61, 2009.

[65] J. Neel, A.B. Mackenzie, R. Menon, L.A. Dasilva, J.E. Hicks, J.H. Reed, and R.P. Gilles. Using game theory to analyze wireless ad hoc networks. *IEEE Communications Surveys & Tutorials*, **7**(4):46–56, 2005.

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

[67] R. Olfati-Saber, J. A. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, **95**(1):215–233, January 2007.

[68] B. R. Pradelski and H. P. Young. Learning efficient Nash equilibria in distributed systems. Discussion paper, Department of Economics, University of Oxford, 2010.

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

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

[71] W.H. Sandholm. *Population Games and Evolutionary Dynamics*. MIT Press, 2012.

[72] D. Shah and J. Shin. Dynamics in congestion games. In *ACM SIGMETRICS*, pages 107–118, 2010.

[73] J. S. Shamma and G. Arslan. Dynamic fictitious play, dynamic gradient play, and distributed convergence to Nash equilibria. *IEEE Transactions on Automatic Control*, 50(3):312–327, March 2005.

[74] J.S. Shamma, editor. *Cooperative Control of Distributed Multi-Agent Systems*. Wiley-Interscience, 2008.

[75] L. S. Shapley. Stochastic games. *Proceedings of the National Academy of Sciences of the United States of America*, 39(10):1095–1100, 1953.

[76] L.S. Shapley. A value for n-person games. In H. W. Kuhn and A. W. Tucker, editors, *Contributions to the Theory of Games II (Annals of Mathematics Studies 28)*, pages 307–317. Princeton University Press, Princeton, NJ, 1953.

[77] 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. special issue on Foundations of Multi-Agent Learning.

[78] V. Srivastava, J. Neel, A.B. Mackenzie, R. Menon, L.A. Dasilva, J.E. Hicks, J.H. Reed, and R.P. Gilles. Using game theory to analyze wireless ad hoc networks. *IEEE Communications Surveys & Tutorials*, 7(4):46–56, 2005.

[79] 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.

[80] A. Vetta. Nash equilibria in competitive societies with applications to facility location, traffic routing, and auctions. In *FOCS*, pages 416–425, 2002.

[81] J. Wang and N. Elia. Control approach to distributed optimization. In *Proceedings of the 2010 48th Annual Allerton Conference on Communication, Control, and Computing*, pages 557–561, 2010.

[82] E. Wei, A. Ozdaglar, and A. Jadbabaie. A distributed Newton method for network utility maximization. arXiv:1005.2633, 2010.

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

[84] H. P. Young. The evolution of conventions. *Econometrica*, 61:57–84, 1993.

[85] H. P. Young. *Equity*. Princeton University Press, Princeton, NJ, 1994.

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

[87] H. P. Young. *Strategic Learning and its Limits*. Oxford University Press, 2005.

[88] H. P. Young. Learning by trial and error. *Games and Economic Behavior*, 65:626–643, 2009.