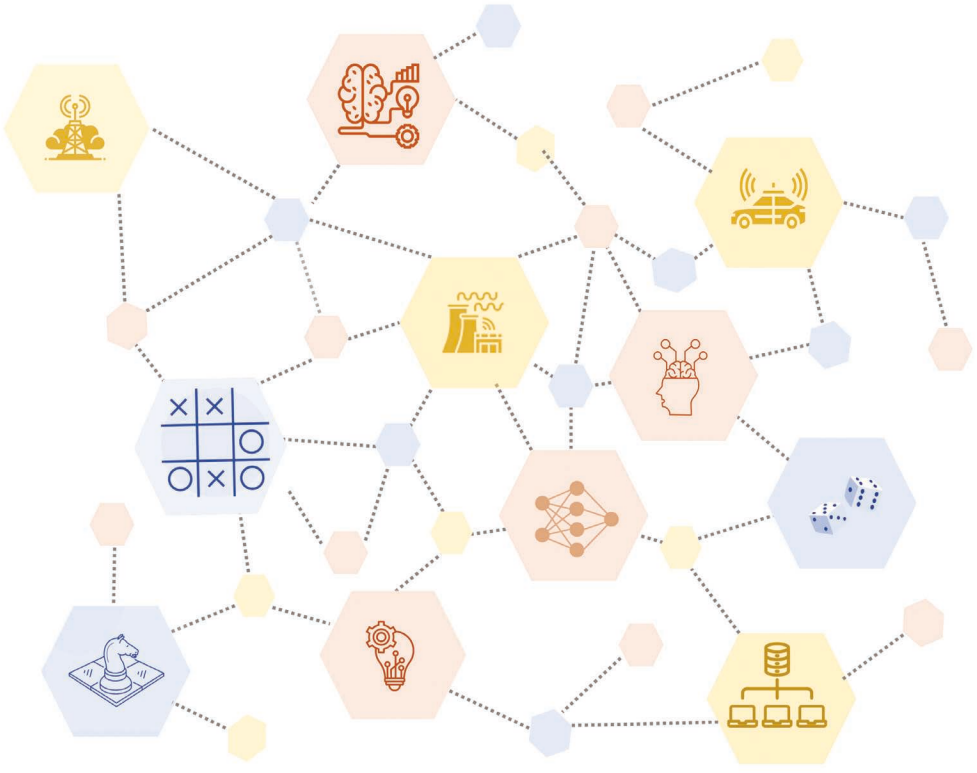
The Confluence of Networks, Games, and Learning a Game-Theoretic Framework for Multiagent Decision Making Over Networks

TAO LI, GUANZE PENG, QUANYAN ZHU, and TAMER BAŞAR



Multiagent decision making over networks has recently attracted an exponentially growing number of researchers from the systems and control community. The area has gained increasing momentum in engineering, social sciences, economics, urban science, and artificial intelligence as it serves as a prevalent framework for studying large and complex systems and has been widely applied to many problems, such as social networks analysis [1], [2], smart grid management [3], [4], wireless and communication networks [5]–[7], cybersecurity [8]–[10], critical infrastructures [11]–[13], and cyberphysical systems [14]–[16].

Due to the proliferation of advanced technologies and services in modern network applications, solving decision-making problems in multiagent networks calls for novel models and approaches that capture the following characteristics of emerging network systems and the design of autonomous controls:

» the heterogeneous nature of the underlying network, where multiple entities (represented by the set of

Digital Object Identifier 10.1109/MCS.2022.3171478
Date of current version: 18 July 2022

nodes) aim to pursue their own goals with independent decision-making capabilities
- the need for distributed or decentralized operation of the system, when the underlying network is of a complex topological structure and is too large to be managed in a centralized approach
- the need for creating network intelligence that is responsive to changes in the network and the environment as the system often operates in a dynamic or an adversarial environment.

Game theory provides a natural set of tools and frameworks for addressing these challenges and bridging networks to decision making. It entails the development of mathematical models that qualitatively and quantitively depict how the interactions of self-interested agents with different information and rationalities can attain a global objective or lead to emerging behaviors at a system level. Moreover, game-theoretic models capture the impact of the underlying network topology on the process of distributed decision making, where agents plan their moves independently according to their goals and local information available to them, such as their observations of their neighbors.

In addition to game-theoretic models over networks, learning theory is indispensable when designing decentralized management mechanisms for network systems to equip networks with distributed intelligence. Through the combination of game-theoretic models and associated learning schemes, such network intelligence allows heterogeneous agents to interact strategically with each other and learn to respond to uncertainties, anomalies, and disruptions, leading to desired collective behavior patterns over the network or an optimal system-level performance. The key feature of such network intelligence is that even though each agent's decision-making process is influenced by the others' decisions, the agents reach an equilibrium state (that is, a Nash equilibrium as we elucidate later, in an online and decentralized manner). To equip networks with distributed intelligence, networked agents should adapt themselves to the dynamic environment with limited and local observations over a large network that may be unknown to them. Computationally, decentralized learning scales efficiently to large and complex networks and requires no global information regarding the entire network (which is more practical compared with centralized control laws).

This article discusses the confluence of networks, games, and learning, which establishes a theoretical underpinning for understanding multiagent decision making over networks. We aim to provide a systematic treatment of game-theoretic learning methods and their applications in network problems, which meet the three requirements specified in "Summary." As shown in Figure 1, emerging network applications call for novel approaches. Thanks to their decentralized nature, game-theoretic models and associated learning methods provide an elegant approach for tackling network problems arising from various fields. Specifically, the objectives are threefold:

1) to provide a high-level introduction to game-theoretic models that apply to multiagent decision-making problems
2) to present the key analytical tool based on stochastic approximation and Lyapunov theory for studying learning processes in games and pinpoint extensively studied learning dynamics

Summary

Modern network systems with heterogeneous entities call for distributed and intelligent operations that are responsive to uncertainties, anomalies, and disruptions within a dynamic or an adversarial environment. The combination of game-theoretic models and learning-based approaches equips the system with decentralized intelligence, allowing heterogeneous agents to strategically interact with each other and learn to adjust their behaviors accordingly. This article presents an overview of the confluence of networks, games, and learning, providing a game-theoretic framework for multiagent decision making over networks. Its focus is on widely applied game-theoretic models and equilibrium concepts as well as associated learning schemes in games. According to their distinct natures in exploration, learning schemes are categorized into two main classes: *exploitative reinforcement learning and exploratory reinforcement learning*. A comparison of the resulting dynamics of learning algorithms from the two classes is presented, highlighting the connections and differences in their exploration processes as well as equilibrium-convergence properties. To demonstrate the broad applicability of this game-theoretic framework, this article discusses, in detail, some representative research on next-generation wireless networks, smart grids, and distributed machine learning (ML), while pointing the reader to other emerging networks applications. In addition to existing research on game-theoretic learning over networks, this article also highlights several new angles and research on learning in games that are, in part, driven by and closely related to recent advances in ML and artificial intelligence, including the study of equilibrium convergence in generic multiplayer games, acceleration techniques for speeding up learning processes, and extending learning algorithms to more complicated dynamic games. The overall objective is to provide the reader with a clear picture of the strengths and challenges of adopting game-theoretic learning methods within the context of network systems, and further identify fruitful future research directions in both theoretical and applied studies.

3) to introduce various multiagent systems and network applications that can be addressed through game-theoretic learning.

This work provides a clear picture of the strengths and challenges of adopting novel game-theoretic learning methods within the context of network systems. In this article, complete-information games are the basis of the subject, for which a brief introduction to both static and dynamics games is provided. More comprehensive treatments on this topic as well as other game models such as incomplete information games can be found in [17] and [19]. As most of the network topologies can be characterized by the structure of the utility function of the game [1], [20], we do not articulate the influence of network topologies on the game itself. Instead, we focus on its influence on the learning process in games (where players' information feedback depends on the network structures) and present representative network applications to showcase this influence. Refer to [1] and [20] for further information on games over various networks.

The discussions are structured as follows. The "Noncooperative Game Theory" section introduces noncooperative games and associated solution concepts, including the Nash equilibrium and its variants, which capture the strategic interactions of self-interested players. The "Learning in Games" section moves to the main focus of this article: learning dynamics in games that converge to the Nash equilibrium. Within the stochastic approximation framework, a unified description of various dynamics is provided, and the analytical properties can be studied using ordinary differential equation (ODE) methods. The "Game-Theoretic Learning Over Networks" section discusses applications of these learning algorithms in networks, leading to distributed and learning-based controls for network systems. Finally, the "Conclusion" section closes the article. For the reader's convenience, notations that are frequently used are summarized in Table 1.

NONCOOPERATIVE GAME THEORY

Game theory constitutes a mathematical framework with two main branches: noncooperative and cooperative game theory. Noncooperative game theory focuses on the strategic decision-making process of independent entities or players that aim to optimize their distinct objective functions without any external enforcement of cooperative behaviors. The term *noncooperative* does not necessarily mean that players are not engaged in cooperative behaviors. Induced cooperative or coordinated behaviors do arise in noncooperative circumstances within the context of the Nash equilibrium, a solution concept of noncooperative games. However, such coordination is self-enforcing and arises from decentralized decision-making processes of self-interested players. This will be further discussed in the "Game-Theoretic Learning Over Networks" section where

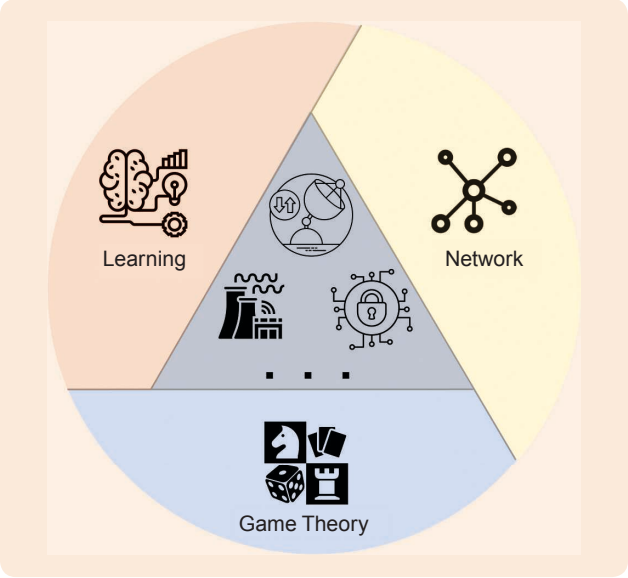


FIGURE 1 The confluence of networks, games, and learning. The combination of game-theoretic modeling and learning theories leads to resilient and agile network controls for various networked systems.

TABLE 1 The table of notations.

Symbol	Meaning
$\mathcal{N}$	The set of players
$i, j \in \mathcal{N}$	Subscript index denoting players
$\mathcal{N}(i)$	The set of neighbors of player i
$\mathcal{A}_i$	The set of actions available to player i
$\Delta(\mathcal{A}_i)$	The set of Borel probability measures (The probability simplex in $\mathbb{R}^{\mathcal{A}_i}$ for finite action set $\mathcal{A}_i$)
$s \in \mathcal{S}$	State variable
$u_i : \Pi_{j\in\mathcal{N}} A_j \to \mathbb{R}$	Player i's utility function
$a_i \in \mathcal{A}_i$	Action of player i
$a_{-i} \in \Pi_{j\in\mathcal{N}, j\neq i} A_j$	Joint actions of players other than i
$\mathbf{a} \in \Pi_{i\in\mathcal{N}} \mathcal{A}_i$	Joint actions of all players
$\pi_i \in \Delta(\mathcal{A}_i)$	Strategy of player i
$\pi_{-i} \in \Pi_{j\in\mathcal{N}, j\neq i} \Delta(\mathcal{A}_j)$	Joint strategy of players other then i
$\mathbf{u}_i(\pi_{-i})$ or $\mathbf{u}_i \in \mathbb{R}^{\lvert\mathcal{A}_i\rvert}$	Player i's utility vector in finite games
$D_i(\mathbf{a})$	The individual payoff gradient of player i
$D(\mathbf{a})$	The concatenation of $\{D_i(\mathbf{a})\}_{i\in\mathcal{N}}$
I_i^k	The feedback of player i at time k
$U_i^k \in \mathbb{R}$	The payoff feedback received by player i at time k
$\hat{\mathbf{u}}_i^k \in \mathbb{R}^{\lvert\mathcal{A}_i\rvert}$	Estimated utility vector at time k
$\hat{\mathbf{U}}_i^k \in \mathbb{R}^{\lvert\mathcal{A}_i\rvert}$	Estimator of $\mathbf{u}_i(\pi_{-i}^k)$ at time k
BR_i	Best response mapping for player i
QR^ϵ	Regularized best response or quantal response

Information **in games refers to the structure regarding the knowledge players acquire about the game and its history when they decide their moves.**

game-theoretic methods for distributed machine learning (ML) are introduced.

As briefly discussed, noncooperative game theory naturally characterizes the decision-making process of heterogeneous entities acting independently over networks, which is the main focus of this article. The following introduces various game models and related solution concepts, including the Nash equilibrium and its variants. Generally speaking, a game involves the following elements: decision makers' (players') choices available to each player (actions), knowledge that a player acquires for making decisions (information), and each player's preference ordering among its actions (utilities or cost). The following is a short list of these concepts, which will be further discussed and explained in this section:

- *Players* are participants in a game, where they compete for their own good. A player can be an individual or encapsulation of a set of individuals.
- *Actions* of a player, in the terminology of control theory, are the implementations of the player's control.
- *Information* in games refers to the structure regarding the knowledge players acquire about the game and its history when they decide their moves. The information structure can vary considerably. For some games, the information is *static* and does not change during the play. For other games, new information will be revealed after players' moves as the "state" of the game (a concept to be elucidated later) is determined by players' actions during the play. In the latter case, the information is *dynamic*. Both types of games are addressed in this article.
- *A strategy* is a mapping that associates a player's move with the information available to him or her at the time when he or she decides which move to choose.
- *A utility or payoff* is often a real-valued function capturing a player's ordering preference among possible outcomes of the game. Using the terminology in control theory, this can also be viewed as a cost function for the player's controls.

This list refers to elements of games in relatively imprecise common language terms, and more formal definitions are presented in the next section. To facilitate this discussion, noncooperative games are categorized into two main classes: static and dynamic games, based on the nature of the information structure.

Static Games

Static games are one shot, where players make decisions simultaneously based on prior information on the games, such as sets of players' actions and their payoffs. In such games, each player's knowledge about the game is static and does not evolve during the play. A static noncooperative game is mathematically defined as follows.

Definition 1 (Static Games)

A static game is defined by a triple $G := \langle \mathcal{N}, (\mathcal{A}_i)_{i\in\mathcal{N}}, (u_i)_{i\in\mathcal{N}} \rangle$, where

- $\mathcal{N} = \{1, 2, \ldots, N\}$ is a finite set of players.
- $\mathcal{A}_i$ with some specified topology denotes the set of actions available to the player $i \in \mathcal{N}$.
- $u_i : \Pi_{j\in\mathcal{N}} \mathcal{A}_j \to \mathbb{R}$ defines player i's utility, and $u_i(a_i, a_{-i})$ gives the payoff of player i when taking action a_i, given other players' actions $a_{-i} := (a_j)_{j\in N, j\neq i}$.

In static games, each player develops its strategy (a probability distribution over his or her action set) with the objective of maximizing the expected value of its own utility. If players have finite action sets, then such a static game is called a *finite* one. In this case, a strategy is a finite-dimensional vector in the probability simplex over the action set, that is, $\pi_i \in \Delta(\mathcal{A}_i) := \{\pi \in \mathbb{R}^{|\mathcal{A}_i|} \mid \pi(a) \geq 0, \forall a \in \mathcal{A}_i, \Sigma_{a\in\mathcal{A}_i}\pi(a) = 1\}$. If π_i is a unit vector $e_a, a \in \mathcal{A}_i$ with the ath entry being one and zero for others, then it is a pure strategy (selecting action a with probability one); otherwise, it is a mixed strategy (choosing actions randomly under the selected probability distribution). Similarly, for infinite action sets, the strategy is defined as a *Borel probability measure* over the action set, with a Dirac measurement being the pure strategy. By a possible abuse of notation, denote the set of Borel probability measurements over $\mathcal{A}_i$ by $\Delta(\mathcal{A}_i)$. Unless specified otherwise, the static games considered in this article are all assumed to be finite, where the player set and the action sets are all finite.

As a special case of games with infinite actions, the mixed extension of finite games is introduced in the sequel. Consider a two-player finite game $G := \langle \mathcal{N}, (\mathcal{A}_i)_{i\in\mathcal{N}}, (u_i)_{i\in\mathcal{N}} \rangle$, where $\mathcal{N} = \{1, 2\}$, and the action sets are finite $|\mathcal{A}_i| < \infty, i \in \mathcal{N}$. Given the mixed strategies of players, $\pi_i \in \Delta(\mathcal{A}_i)$, the expected utility of player i is $\mathbb{E}_{a_1\sim\pi_1, a_2\sim\pi_2}[u_i(a_1, a_2)]$. With a slight abuse of notation, denote this expected utility by $u_i(\pi_1, \pi_2) := \mathbb{E}_{a_1\sim\pi_1, a_2\sim\pi_2}[u_i(a_1, a_2)]$. Then, studying the players' strategic interactions is equivalent to considering the following infinite game $G^\infty = \langle \mathcal{N}, (\Delta(\mathcal{A}_i))_{i\in\mathcal{N}}, (u_i)_{i\in\mathcal{N}} \rangle$, where u_i denotes the expected utility. In G^∞, an action is a vector from the corresponding probability simplex, a convex and compact set with a continuum of elements. Similar to the notations used in the definition, for the mixed extension G^∞, the joint action of players other than i is denoted by

$\pi_{-i} := (\pi_j)_{j\in\mathcal{N}, j\neq i}$. Furthermore, let $\mathbf{u}_i(\pi_{-i}) \in \mathbb{R}^{|\mathcal{A}_i|}$ be the utility vector of player i, given other players' strategy profiles π_{-i}, whose ath entry is defined as $\mathbf{u}_i(\pi_{-i})(a) := u_i(e_a, \pi_{-i})$. Due to the definition of expectation, $u_i(\pi_i, \pi_{-i})$ can be expressed as an inner product $\langle \pi_i, \mathbf{u}_i(\pi_{-i})\rangle$, which will be used frequently later when discussing learning algorithms in finite games. This mixed extension provides a geometric characterization to the Nash equilibria of finite games, based on variational inequalities, as discussed in the "Solution Concepts" section. Meanwhile, this inner product expression connects learning theory in finite games with online linear optimization [21], where the generic player's decision variable is π_i and the loss function specified by $\langle \cdot, \mathbf{u}_i(\pi_{-i})\rangle$ is linear in π_i.

Even though widely applied in modeling behaviors of self-interested players, the static game model is far from sufficient to cover multiagent decision-making problems arising in different fields. For instance, when playing poker games, new information will be revealed during game play (such as cards played at each round) based on which players can adjust their moves. There are many games where players' information about the game changes over time, which cannot be suitably described by static games. Therefore, dynamic game models are needed to capture information changes.

Dynamic Games

To explicitly represent the dynamic nature of the decision-making process, system theory terminology and the state of the game should describe its evolution over a period of time (which could be finite or infinite). Roughly speaking, the current state specifies the current situation of the dynamic game, including the set of players who are about to take actions, actions available to them, and their utilities at this time. The fundamental difference between static and dynamic games is that, for the latter, the game changes over time as players implement their sequences of actions during the play. Hence, players' knowledge regarding the game also evolves as players can fully or partially observe the current state.

In the following, a subclass of Markov games is introduced as an example of dynamic games, which is a very popular game model for studies on multiagent sequential decision making under uncertainties (such as multiagent reinforcement learning [22]).

Definition 2 (Markov Games)

An N-person discrete-time infinite horizon discounted Markov game consists of

- a player set $\mathcal{N} = \{1, 2, \ldots, N\}$
- a discrete time set $\mathbb{N}_+ := \{1, 2, \ldots\}$, with actions by players taken at each $k \in \mathbb{N}_+$
- a set $\mathcal{A}_i$ with some specified topology (defined for each $i \in \mathcal{N}$), corresponding to the set of actions or controls available to player i
- a set $\mathcal{S}$ with some specified topology, denoting the state space of the game, where $s^k \in \mathcal{S}$, $k \in \mathbb{N}_+$ represent the state of the game at time k;
- a transition kernel $T : \mathcal{S} \times \Pi_{i\in\mathcal{N}} \mathcal{A}_i \to \Delta(\mathcal{S})$, according to which the next state is sampled; $s^{k+1} \sim T(s^k, \mathbf{a}^k)$, where $\mathbf{a}^k = (a_1^k, \ldots, a_N^k)$ is the N-tuple of actions at time $k \in \mathbb{N}_+$, and $s^1 \in \mathcal{S}$ is sampled from an initial distribution
- an instantaneous payoff: $u_i : \mathcal{S} \times \Pi_i \mathcal{A}_i \to \mathbb{R}$, defined for each $i \in \mathcal{N}$ and $k \in \mathbb{N}_+$, determining the payoff $u_i(s^k, \mathbf{a}^k)$ received by player i at time k;
- a discounting factor γ. Given $\{s^1, \ldots, s^k, \ldots; \mathbf{a}^1, \ldots, \mathbf{a}^k, \ldots\}$, the discounted cumulative payoff for player i is $\Sigma_{k=1}^{\infty} \gamma^k u_i(s^k, \mathbf{a}^k)$.

This definition characterizes only one special case of dynamic games. Based on this definition, many other game models can be derived. For example, state transitions can be independent of players' actions as well as the current state, yielding a special case of stochastic games (which will be further discussed in another article in this special issue of *IEEE Control Systems* [23]). We can also consider continuous-time dynamic games where the transition is described by a differential equation, leading to a differential game model. For extensive coverage of dynamic game models, refer to [17].

With full observation of states, consider the stationary strategy $\pi_i : \mathcal{S} \to \Delta(\mathcal{A}_i)$, by which players plan their moves based only on the current state $s \in \mathcal{S}$. In this case, the state variable s characterizes players' knowledge of the game as the actions, utilities, and next possible states are all determined by the current state. For dynamic games under partial observation and/or non-Markovian transition, refer to [17] as these topics are beyond the scope of this article.

Solution Concepts

The solution or outcome of any given game is more or less a matter of understanding game rules and relationships between players. However, besides these concrete matters, there exist general principles that dictate players' behaviors and apply to all games. These principles revolve around the notion of rationality, based on which we introduce the solution concept of Nash equilibrium and some of its variants. Mathematically, a solution to an N-person game is a collection of all players' strategies, which has attractive properties expressed in terms of payoffs received by the players. In addition, players can admit different strategies depending on how the game is defined and, in particular, the information that players acquire. The following discussion on solution concepts begins with static games, where the information structure is relatively simple.

Compared with single-agent optimization problems, the analysis of games is more involved as each player's utility is determined not only by its own decision but also by others' moves. Hence, when a player takes an action, it must consider possible moves of the other players (which leads to the

notion of best response). To introduce "best response," for clarity, but without any loss of conceptual generality, we focus on games with two players. For player 1, given the other player's strategy π_2, the optimal choice is

$$\pi_1 \in BR_1(\pi_2) := \underset{\pi \in \Delta(\mathcal{A}_1)}{\operatorname{argmax}} \{\langle \pi, \mathbf{u}_1(\pi_2)\rangle\}, \tag{1}$$

which is referred to as a *best response* of player 1 to player 2's strategy π_2. $BR_1(\cdot)$ is the best response set of player 1. Similarly, given player 1's strategy π_1, a best response of player 2 is $\pi_2 \in BR_2(\pi_1) := \operatorname{argmax}_{\pi \in \Delta(\mathcal{A}_2)} \{\langle \pi, \mathbf{u}_2(\pi_1)\rangle\}$. Hence, a point-to-set mapping $BR: \Delta(\mathcal{A}_1) \times \Delta(\mathcal{A}_2) \to 2^{\Delta(\mathcal{A}_1)\times\Delta(\mathcal{A}_2)}$ can be defined as the concatenation of BR_1 and BR_2. Given a joint strategy profile $\pi = (\pi_1, \pi_2)$,

$$BR(\pi) := \{(\pi_1', \pi_2') | \pi_1' \in BR_1(\pi_2), \pi_2' \in BR_2(\pi_1)\}. \tag{2}$$

If a fixed point of this best-response mapping $\pi^* = (\pi_1^*, \pi_2^*)$ can be found [that is, $\pi^* \in BR(\pi^*)$], then when both players adopt the corresponding strategy in this profile, they could do no better by unilaterally deviating from the current strategy. In other words, this fixed point corresponds to an equilibrium outcome of the game, which further leads to the definition of Nash equilibrium.

Definition 3 (Nash Equilibrium)

For a static game $\langle \mathcal{N}, (\mathcal{A}_i)_{i\in\mathcal{N}}, (u_i)_{i\in\mathcal{N}}\rangle$, the Nash equilibrium is a strategy profile $\pi^* = (\pi_i^*, \pi_{-i}^*)$ with the property that for all $i \in \mathcal{N}$,

$$u_i(\pi_i^*, \pi_{-i}^*) \geq u_i(\pi_i, \pi_{-i}^*), \tag{3}$$

where π_i is an arbitrary strategy of player i, and $\pi_{-i}^* = (\pi_j^*)_{j\in\mathcal{N}, j\neq i}$ denotes the joint strategy profile of the other players. If the inequality holds strictly for all $\pi_i \neq \pi_i^*$, then it is referred to as a *strict Nash equilibrium*.

Note that the preceding definition naturally carries over to games with infinite action sets; refer to [17, Ch. 4] for more details. Furthermore, for infinite games, if some topological structures are imposed on the action sets and regularity conditions on the utility functions, then a geometric interpretation of the Nash equilibrium is derived from the inequality in (3). Toward that end, consider a (static) game with compact and convex action sets $(\mathcal{A}_i)_{i\in\mathcal{N}}$ and smooth, concave utilities:

$$u_i(a_i, a_{-i}) \text{ is concave in } a_i \text{ for all } a_{-i} \in \prod_{j\in\mathcal{N}, j\neq i} \mathcal{A}_j, \; i \in \mathcal{N}.$$

In such a game, the number of actions available to each player is a continuum, and the utility function is continuous; these games are referred to as *continuous-kernel* or *continuous games*. In this case, a pure-strategy Nash equilibrium $\mathbf{a}^* = (a_i^*, a_{-i}^*) \in \prod_{i\in\mathcal{N}} \mathcal{A}_i$ is defined by the inequality

$$u_i(a_i^*, a_{-i}^*) \geq u_i(a_i, a_{-i}^*), \quad \text{for all } a_i \in \mathcal{A}_i \text{ and all } i \in \mathcal{N}. \tag{4}$$

Further assuming that $u_i(a_i, a_{-i})$ is continuously differentiable in $a_i \in \mathcal{A}_i$ for all a_{-i}, by the first-order condition, the Nash equilibrium in (4) can be characterized by

$$\langle D_i(\mathbf{a}^*), a_i - a_i^* \rangle \leq 0, \quad \text{for all } a_i \in \mathcal{A}_i, i \in \mathcal{N},$$

where $D_i(\mathbf{a}) := \nabla_{a_i} u_i(a_i, a_{-i})$ denotes the individual payoff gradient of player i, and $\nabla_{a_i} u_i(a_i, a_{-i})$ represents differentiation with respect to the variable a_i. Rewriting the aforementioned inequality in a more compact form yields the following variational characterization of the Nash equilibrium

$$\langle D(\mathbf{a}^*), \mathbf{a} - \mathbf{a}^* \rangle \leq 0, \quad \text{for all } \mathbf{a} \in \prod_{i\in\mathcal{N}} \mathcal{A}_i, \tag{5}$$

where $D(\mathbf{a})$ is the concatenation of $\{D_i(\mathbf{a})\}_{i\in\mathcal{N}}$, that is, $D(\mathbf{a}) = (D_1(\mathbf{a}), \ldots, D_N(\mathbf{a}))$. Geometrically, (5) states that for concave games, $\mathbf{a}^*$ is a Nash equilibrium if and only if $D(\mathbf{a}^*)$ lies within the polar cone of the set $\prod_{i\in\mathcal{N}} \mathcal{A}_i - \mathbf{a}^* := \{\mathbf{a} - \mathbf{a}^* | \mathbf{a} \in \prod_{i\in\mathcal{N}} \mathcal{A}_i\}$, as shown in Figure 2.

In addition to concave games, such a variational inequality characterization has been studied in much broader contexts, such as in monotone games [24], which bridges the gap between the theory of monotone operators and Nash equilibrium seeking. For a detailed discussion, the reader is referred to another article in this special issue of *IEEE Control Systems* [25]. The variational inequality (5) is denoted as the *Stampacchia-type variational inequality* (*SVI*) [26], and a similar variational inequality of this kind can also be derived in the context of the mixed extension. As a special case of continuous games, the mixed extension of finite games also satisfies regularity conditions: The action

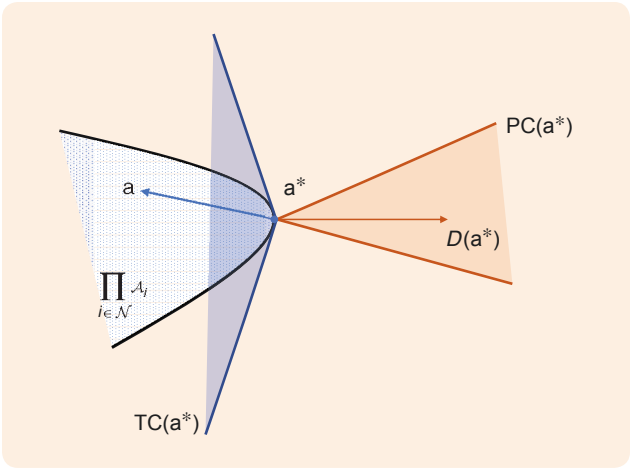


FIGURE 2 The variational characterization of a Nash equilibrium $\mathbf{a}^*$ in concave games. TC($\mathbf{a}^*$) and PC($\mathbf{a}^*$) denote the tangent and the polar cone, respectively, of $\prod_{i\in\mathcal{N}} \mathcal{A}_i - \mathbf{a}^*$. According to the variational inequality (5), $\mathbf{a}^*$ is a Nash equilibrium if and only if the payoff gradient $D(\mathbf{a}^*)$ lies in the polar cone.

spaces are probability-simplex regions (which are compact and convex), and the utility function is naturally smooth and concave (due to its linearity with respect to any player's mixed strategy). Therefore, the mixed-strategy Nash equilibrium can be characterized by a variational inequality as well. Thanks to the inner product expression of the utility in the mixed extension, the individual payoff gradient is simply $\mathbf{u}_i(\pi_{-i})$, and denote the concatenation of $\{\mathbf{u}_i\}_{i\in\mathcal{N}}$ by $\mathbf{u}(\pi) := [\mathbf{u}_1(\pi_{-1}(t)), \mathbf{u}_2(\pi_{-2}(t)), \ldots, \mathbf{u}_N(\pi_{-N}(t))]$, which is also referred to as the *joint utility vector* under the strategy profile π. Similar to (5), a strategy profile π^* is the Nash equilibrium of the underlying finite game if and only if the following SVI holds

$$\langle \mathbf{u}(\pi^*), \pi - \pi^* \rangle \le 0, \quad \text{for all } \pi \in \prod_{i\in\mathcal{N}} \Delta(\mathcal{A}_i). \tag{SVI}$$

As shown in the "Nash Equilibrium and Lyapunov Stability" section, this variational characterization of Nash equilibrium bridges the equilibrium concept of games and the equilibrium concept of dynamical systems induced by learning algorithms.

In the same spirit of (3), Nash equilibrium in dynamic games can also be defined accordingly. For Markov games, given players' stationary strategy profile π, the cumulative expected utility of player i (starting from the initial state $s^1 = s$) is

$$V_i^{\pi}(s) := \mathbb{E}_{s^{k+1}\sim T, \mathbf{a}^k\sim\pi}\left[\sum_{k=1}^{\infty} \gamma^k u_i(s^k, \mathbf{a}^k) | s^1 = s\right], \tag{6}$$

which is referred to as the *state-value function* in a Markov decision process [27]. V_i^{π} is the utility under the strategy profile π, and following (3), the Nash equilibrium is defined for the Markov game, where the inequality holds for every state. In other words, regardless of the previous play, as long as players follow π^* from the current state s, they achieve the best outcome for the rest of the game, and no player has any incentive to deviate from the strategy dictated by π^*. Hence, this kind of Nash equilibrium is referred to as a *subgame perfect Nash equilibrium*, which is widely used in the study of dynamic games [28], [29].

The Nash equilibrium serves as a building block for noncooperative games. One of its major advantages is that it characterizes the stable state of a noncooperative game, in which no rational player has an incentive to move unilaterally. This stability idea will be further discussed in the "Nash Equilibrium and Lyapunov Stability" section, which relates the stability theory of differential equations to the convergence of learning algorithms in Nash equilibrium seeking.

LEARNING IN GAMES

Learning in games refers to a long-run nonequilibrium process of learning, adaptation, and/or imitation that leads to some equilibrium [30]. Unlike pure equilibrium analysis based on the definition, learning in games accounts for how players behave adaptively during repeated game play under uncertainties and partial observations. Computationally, computing the Nash equilibrium based on equilibrium analysis is challenging due to the computational complexity [31], which hardly accounts for the decision-making process in practice (where players have limited computation power and information). Hence, learning models are needed to describe how less than fully rational players behave to reach equilibrium. Equilibrium seeking or computation motivates learning in games [29].

If the learning process is viewed as a dynamical system, then the learning model can predict how each player adjusts its behavior in response to other players over time to search for strategies that will lead to higher payoffs. From this perspective, a Nash equilibrium can also be interpreted as the steady state of the learning process, which serves as a prediction of the limiting behavior of the dynamical system induced by the learning model. This viewpoint has been widely adopted in the study of population biology and evolutionary game theory, as shown more clearly when discussing reinforcement learning and replicator dynamics [32].

In this section, various learning dynamics are presented in the context of infinitely repeated games for Nash equilibrium seeking. Consider a number of players repeatedly playing the game $\langle \mathcal{N}, (\mathcal{A}_i)_{i\in\mathcal{N}}, (u_i)_{i\in\mathcal{N}} \rangle$ infinitely many times. At time k, players determine their moves based on their observations up to time $k-1$. They then receive feedback from the environment, which provides information on past actions. In finite games, based on the information available to it, player i constructs a mixed strategy $\pi_i^k \in \Delta(\mathcal{A}_i)$, from which it samples an action a_i^k and implements it. It will then receive a payoff feedback related to $u_i(a_i^k, a_{-i}^k)$, which evaluates the performance of a_i^k and helps the player shape its strategy for future plays. In such a repeated game, the amount of information that players acquire in repeated plays directly determines how players plan their moves at each round and further influences the resulting learning dynamics. Besides theoretical importance, information feedback in the learning process (such as players' observations of their opponents' moves) is also of vital importance in designing learning-based methods for solving network problems. As shown more clearly in "Game-Theoretic Learning Over Networks," networked agents observe only their surroundings in many network applications, without any access to global information regarding the whole network. Therefore, due to its significance in learning processes, existing feedback structures are first discussed in the following section.

Feedback Structures in Learning

The feedback structure for a player in a repeated game includes its observations regarding the game and repeated plays, which is a subset of every player's histories of plays and payoffs. To make the discussion more concrete, the

following notation is introduced. Let I_i^k be the feedback of player i up to time k. Denote the payoff received by player i at the kth round by $u_i^k := u_i(a_i^k, a_{-i}^k)$ and the sequence of payoffs received up to time k by $u_i^{1:k} := \{u_i^1, \ldots, u_i^k\}$.

The simplest feedback structure is called the *perfect global feedback*, where

$$I_i^k = \left\{\{u_j^{1:k}\}_{j\in\mathcal{N}}, \{a_j^{1:k}\}_{j\in\mathcal{N}}\right\},$$

indicating completeness of feedback from both the temporal and spatial senses. Furthermore, consider the noisy feedback of payoffs, U_i^k, defined as

$$U_i^k = u_i(a_i^k, a_{-i}^k) + \xi_i^k,$$

where ξ_i^k is a zero-mean martingale noise process with a finite second moment, that is, $\mathbb{E}[\xi_i^k | \mathcal{F}^{k-1}] = 0, \mathbb{E}[(\xi_i^k)^2 | \mathcal{F}^{k-1}]$ is less than a constant, and the expectation is taken with respect to the σ-field $\mathcal{F}^{k-1}$ generated by the history of play up to time $k-1$. Simply put, the noisy feedback U_i^k is a conditionally unbiased estimator of u_i^k with respect to the history, which is a standing assumption when addressing the convergence of learning dynamics in games. For noisy feedback in general, or equivalently, ξ_i^k being a generic random variable, the discussion will be implemented in a different context. In that case, a system state should be introduced, which accounts for the uncertainty in the environment, and the learning problem becomes Nash equilibrium seeking in stochastic games (see Definition 2). For more detailed discussions, the reader is referred to another article in this special issue of *IEEE Control Systems* [23].

Perfect global feedback is of limited use in practice when designing learning algorithms as global information is difficult or even impossible to acquire for individuals in large-scale network systems. For example, in distributed or decentralized learning over heterogeneous networks, players may have no access to others' utilities due to physical limitations. Therefore, we are interested in the scenario where players only have direct or indirect access to their own utilities as well as their neighbors', and hence players' feedback can be dependent on the topological structure of the underlying network that connects them.

Consider a repeated game over a graph $\mathcal{G} := (\mathcal{N}, \mathcal{E})$, where $\mathcal{N} = \{1, 2, \ldots, N\}$ is the set of nodes representing the players in the game who are connected via the edges in $\mathcal{E} = \{(i, j) \,|\, i, j \text{ are connected}\}$. To simplify the exposition, assume that the graph is undirected. Note that the direction of the edges does not affect the discussion as long as the neighborhood is properly defined. For example, in a directed graph, when in neighbors or out neighbors specify to whom the player in question can pass information, the following characterizations of feedback structures still apply. For a more comprehensive treatment of games over networks, refer to [20].

Each player is allowed to exchange payoff feedback with its neighbors through the edges and observe their actions during the repeated play, whereas the information regarding the rest is hidden from him or her. In this case, the feedback structure for player i is

$$I_i^k = \left\{\{u_j^{1:k}\}_{j\in\{i\}\cup\mathcal{N}(i)}, \{a_j^{1:k}\}_{j\in\{i\}\cup\mathcal{N}(i)}\right\}, \quad \mathcal{N}(i) := \{j \,|\, (i, j) \in \mathcal{E}\}.$$

Note that the player's feedback regarding payoffs and actions may not be consistent. For example, in a multiagent robotic system where only the sensors network is effective, each agent can observe only its neighbors' movements through sensors. In this case, without any information of others' utilities, the information feedback of agent i reduces to $I_i^k = \left\{\{u_i^{1:k}\}, \{a_j^{1:k}\}_{j\in\{i\}\cup\mathcal{N}(i)}\right\}$. In summary, if the players can receive feedback only from their neighbors, then players' feedback structures are related to the underlying topology, which is referred to as *local feedback*. In accordance with this, an extreme case of local feedback is one where the player is isolated in the network, and no information other than its own payoff feedback and actions are available to it. This extreme case is referred to as *individual feedback*, which is a typical information feedback considered in fully decentralized learning and will be further elaborated on when discussing specific learning dynamics later in this section.

In addition to refinements from the spatial side, consider feedback with various temporal structures. If the player has perfect recall of previous plays, the resulting feedback is said to be *perfect*, and the feedback structures introduced previously all fall within this class. Otherwise, players have access to *imperfect feedback*, and two common cases of imperfect information feedback are discussed in the following: windowed and delayed feedback.

For simplicity, perfect feedback $I_i^k = \{u_i^{1:k}, a_i^{i:k}\}$ is used as a baseline to illustrate that different missing parts of I_i^k lead to different kinds of imperfect feedback. If the head of $u_i^{1:k}$ and/or $a_i^{1:k}$ is not available to the player (that is, there exists a window $0 < m < k$ such that the player recalls only $u_i^{(k-m):k}, a_i^{(k-m):k}$), then the corresponding feedback $I_i^{(k-m):k} = \{u_i^{(k-m):k}, a_i^{(k-m):k}\}$ is called *windowed feedback*, with a window size of m. Similarly, if the tail of $u_i^{1:k}$ and/or $a_i^{1:k}$ is not available (that is, the player recalls only $u_i^{1:(k-m)}, a_i^{1:(k-m)}$), then the imperfect information feedback is $I_i^{1:(k-m)} = \{u_i^{1:(k-m)}, a_i^{1:(k-m)}\}$, which is called m-step delayed feedback.

For learning in games, each player learns to select actions by updating the strategy based on the available feedback at each round. To describe this in mathematical terms, let F_i^k be the strategy learning policy of player i. The learning policy produces a new strategy π_i^{k+1} for the next play according to

$$\pi_i^{k+1} = (1 - \lambda_i^k)\pi_i^k + \lambda_i^k F_i^k(I_i^k), \tag{7}$$

where λ_i^k is the learning rate, indicating the player's capabilities of information retrieval. Different feedback structures

lead to different learning dynamics in repeated games. Under the global or local feedback structure, each player's feedback is influenced by its opponents' actions and/or payoffs, which makes the players' learning processes coupled (as shown in Figure 3).

In the case of fully decentralized learning under individual information feedback, players learn to play the game independently, and such a learning process is said to be uncoupled. Uncoupled learning processes are of great significance in both theoretical studies [33] and practical applications. Theoretically, learning with such limited information feedback is much more transferable in the sense that learning algorithms under this feedback also apply to online optimization problems, where the online decision-making process is viewed as a repeated game played between a player and the environment [21].

Considering its theoretical importance, we focus on learning with individual feedback in the sequel, and the reader is referred to [34] for a survey on learning methods under other kinds of feedback. We first present reinforcement learning for finite games, where the learning algorithms are characterized into two main classes due to their distinct nature in exploration. We then proceed to gradient play for infinite games and elaborate on its connection to reinforcement learning. The convergence results of presented algorithms are discussed in the "Convergence of Learning in Games" section based on stochastic approximation [35], [36] and Lyapunov stability theory.

Reinforcement Learning

Reinforcement learning has been studied in many disciplines and become a catch-all term for learning in sequential decision-making processes where the players' future choices of actions are shaped by feedback. In general, reinforcement learning consists of two functions: the *score function* (evaluating the performance of actions) and the *choice mapping* (determining the next move). Note that in ML literature [37], the score function and the choice mapping are also called the *critic* and the *actor*, respectively. Different score functions and choice mappings lead to different reinforcement learning algorithms. We first provide a generic description of the score function and choice mapping in reinforcement learning from a dynamic system viewpoint, then give a characterization of various reinforcement learning algorithms based on different natures in choice mappings. Finally, relations among introduced reinforcement learning algorithms are discussed.

We first show how the score function can be constructed using the information feedback recursively. As the player has no direct access to its utility function in this case, it can construct an estimator $\hat{\mathbf{u}}_i^k \in \mathbb{R}^{|\mathcal{A}_i|}$ based on I_i^k to evaluate actions $a \in \mathcal{A}_i$. Using this estimator, the player can compare its actions and choose the one that can achieve higher payoffs in the next round. In mathematical terms, the estimator (score function) is given by the following discrete-time dynamical system:

$$\hat{\mathbf{u}}_i^{k+1} = (1-\mu_i^k)\hat{\mathbf{u}}_i^k + \mu_i^k G_i^k(\pi_i^k, \hat{\mathbf{u}}_i^k, U_i^k, a_i^k), \tag{8}$$

where $G_i^k : \Delta(\mathcal{A}_i) \times \mathbb{R}^{|\mathcal{A}_i|} \times \mathbb{R} \times \mathcal{A}_i \to \mathbb{R}^{|\mathcal{A}_i|}$ is the learning policy for utility learning, π_i^k is the policy employed at time k, and μ_i^k is the learning rate. Based on the score function, the player can modify its strategy accordingly in the sense that better actions shall be played more frequently in the future. With a slight abuse of notations, the strategy update is

$$\pi_i^{k+1} = (1-\lambda_i^k)\pi_i^k + \lambda_i^k F_i^k(\pi_i^k, \hat{\mathbf{u}}_i^{k+1}, U_i^k, a_i^k), \tag{9}$$

where $F_i^k : \Delta(\mathcal{A}_i) \times \mathbb{R}^{|\mathcal{A}_i|} \times \mathbb{R} \times \mathcal{A}_i \to \Delta(\mathcal{A}_i)$ is the learning policy for strategy learning, yielding a new policy for the next play. Compared with (7), the preceding discrete-time systems [(8) and (9)] explicitly show how feedback shapes the player's future play. According to (8), the player recursively updates its estimate of the utility function based on the feedback it receives after playing π_i^k and determines its move in the next round following (9). Intuitively, $(\pi_i^k, \hat{\mathbf{u}}_i^{k+1})$ can be viewed as the information extracted from I_i^k for updating the player's strategy.

In reinforcement learning, the choice mapping plays an important role in achieving a balance between exploitation and exploration. On one hand, the player would like to choose the best action that is supposed to incur the highest payoff based on the score function. However, this pure exploitation often leads to myopic behaviors as the score function may return a poor estimate of the utility function at the beginning of the learning process. Hence, to gather more information for a better estimator, the player also needs some experimental moves for exploration, where suboptimal actions are implemented. In summary, the tradeoff between exploitation and exploration is of vital importance to the success of reinforcement learning, and it depends on construction of the choice mapping. Different choice mappings result in different reinforcement learning algorithms. Based on their distinct natures in exploration, the algorithms can be categorized into two

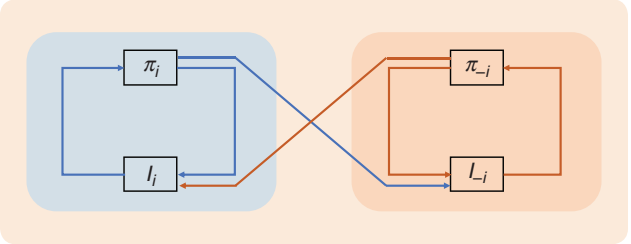


FIGURE 3 A player's strategy learning with the corresponding feedback. Under the global or local feedback structure, players' learning processes are coupled as their feedback is influenced by their opponents' moves. In contrast, players learn to play the game independently under individual feedback.

In general, reinforcement learning consists of two functions: the *score function* (evaluating the performance of actions) and the *choice mapping* (determining the next move).

main classes: *exploitative reinforcement learning* and *exploratory reinforcement learning*.

Recall that in strategy learning (9), the next strategy produced by the corresponding choice mapping is

$$\pi_i^{k+1} = (1-\lambda_i^k)\pi_i^k + \lambda_i^k F_i^k(\pi_i^k, \hat{\mathbf{u}}_i^{k+1}, U_i^k, a_i^k),$$

where $(1-\lambda_i^k)\pi_i^k$ is referred to as the *cognitive inertia* (or simply, *inertia*), describing the player's tendency to repeat previous choices independent of the outcome. When determining its next move π_i^{k+1}, the player considers both its previous strategy π_i^k and the increment update using the strategy learning policy F_i^k. Therefore, players' exploration at $(k+1)$-th round stems either from this inertia or the strategy learning policy F_i^k. The former is called *passive exploration* (as it relies on the player's tendency to repeat previous choices), while the latter is referred to as *active exploration* (as the player deliberately tries actions based on what was learned from previous plays).

As the new strategy is a convex combination of the inertia term π_i^k and the learned incremental update $F_i^k(\pi_i^k, \hat{\mathbf{u}}_i^{k+1}, U_i^k, a_i^k)$, there is no clear-cut boundary between passive and active exploration. In fact, reinforcement learning is a continuum of learning algorithms. The following illustrates such a continuum by three prominent learning schemes. The first is the best response dynamics (BR-d) (located on the left endpoint), which is an example of exploitative reinforcement learning. Solely relying on the inertia for passive exploration, BR-d adopts a purely exploitative learning policy: the best response mapping in (1). In contrast to the exploitative one, dual-averaging dynamics (DA-d) is an example of exploratory reinforcement learning, which only leverages the learning policy for exploring suboptimal actions without any cognitive inertia. In between, there lies the smoothed BR-d (SBR-d), where both the inertia and strategy learning policy are used to achieve a balance between exploration and exploitation.

Exploitative Reinforcement Learning

For exploitative reinforcement learning, the strategy learning policy always outputs the best strategy based on the score function, which can be viewed as a natural extension of the best response idea in the context of a Nash equilibrium (1). In the repeated-play scenario, given the opponent's strategy at the kth round π_{-i}^k, from player i's standpoint, the best he or she can do is to choose the best response $BR_i(\pi_{-i}^k) := \operatorname{argmax}_{\pi\in\Delta(\mathcal{A}_i)}\{\langle \pi, \mathbf{u}_i(\pi_{-i}^k)\rangle\}$ (which is purely exploitative). In this case, the strategy learning scheme becomes

$$\pi_i^{k+1} \in (1-\lambda_i^k)\pi_i^k + \lambda_i^k BR_i(\pi_{-i}^k). \tag{10}$$

In general, the best response mapping is a point-to-set mapping and, differential inclusion theory [36] is needed to analyze the associated learning dynamics, which makes the convergence analysis more involved, as discussed in the "Nash Equilibrium and Lyapunov Stability" section.

Under the noisy feedback $I_i^k = \{U_i^{1:k}, a_i^{1:k}\}$, the score function of player i is the estimated utility $\hat{\mathbf{u}}_i^k$, which is updated according to the following moving average scheme [38]:

$$\hat{\mathbf{u}}_i^{k+1}(a) = (1-\mu_i^k)\hat{\mathbf{u}}_i^k(a) + \mu_i^k \frac{\mathbb{1}_{\{a=a_i^k\}}}{\pi_i^k(a)} U_i^k, \quad a\in\mathcal{A}_i, \tag{11}$$

where $\mathbb{1}_{\{\cdot\}}$ is an indicator function. Note that in (11), the importance sampling technique (which is common in bandit algorithms [21]) is utilized to construct an unbiased estimator of $\mathbf{u}_i(\pi_{-i}^k)$. To see this, define a vector $\hat{\mathbf{U}}_i^k \in \mathbb{R}^{|\mathcal{A}_i|}$ whose a-th entry is $\hat{\mathbf{U}}_i^k(a) := \mathbb{1}_{\{a=a_i^k\}} U_i^k / \pi_i^k(a)$, and then, $\mathbb{E}[\hat{\mathbf{U}}_i^k(a) \,|\, \mathcal{F}^{k-1}] = u_i(a, \pi_{-i}^k)$. Hence, (11) can be rewritten as

$$\hat{\mathbf{u}}_i^{k+1} = (1-\mu_i^k)\hat{\mathbf{u}}_i^k + \mu_i^k \hat{\mathbf{U}}_i^k, \tag{12}$$

and $\hat{\mathbf{u}}_i^{k+1}(a)$ gives the averaged payoff incurred by a in the first k rounds. This importance sampling technique can be viewed as compensating for the fact that actions played with a low probability do not receive frequent updates of the corresponding estimates so that when they are played, any estimation error $U_i^k - \hat{\mathbf{u}}_i^k(a_i^k)$ must have a greater influence on the estimated value than if frequent updates occur. Refer to [21] and [30] for more details on importance sampling and its use in learning processes.

With a slight abuse of the notation of best response mapping in (2), define the corresponding best response under the noisy feedback as

$$BR_i(\hat{\mathbf{u}}_i^k) := \operatorname*{argmax}_{\pi\in\Delta(\mathcal{A}i)} \{\langle \pi, \hat{\mathbf{u}}_i^k\rangle\}. \tag{13}$$

The strategy learning scheme under the noisy feedback [30] follows

$$\pi_i^{k+1} \in (1-\lambda_i^k)\pi_i^k + \lambda_i^k BR_i(\hat{\mathbf{u}}_i^k). \tag{14}$$

The resulting dynamical system under the noisy feedback is a coupled system as

$$\hat{\mathbf{u}}_i^{k+1} = (1-\mu_i^k)\hat{\mathbf{u}}_i^k + \mu_i^k \hat{\mathbf{U}}_i^k,$$
$$\pi_i^{k+1} \in (1-\lambda_i^k)\pi_i^k + \lambda_i^k BR_i(\hat{\mathbf{u}}_i^k). \qquad \text{(BR-d)}$$

Originally proposed as a computational method for Nash equilibrium seeking [38], [39], the BR-d is built directly upon the best response idea and has been widely applied to evolutionary game problems [40]. One prominent example of (BR-d) is fictitious play [41], where a player's empirical play follows (BR-d); and more details are included in "Fictitious Play." As shown, (BR-d) adopts passive exploration, and the best response mapping $BR_i(\cdot)$ encourages greedy actions that might be myopic. As a result, exploitative reinforcement learning may fail to converge [30], [42].

Exploratory Reinforcement Learning

In contrast to the inertia-based passive exploration in (BR-d), dual averaging (as introduced in this section) relies only on the strategy learning policy F_i^k for exploring suboptimal actions to avoid myopic behaviors due to poor estimates of the utility function. In dual averaging, given the player's utility vector $\mathbf{u}_i$, the strategy learning policy is a regularized best response [43] defined as

$$QR^\epsilon(\mathbf{u}_i) := \operatorname*{argmax}_{\pi_i \in \Delta(\mathcal{A}_i)} \{\langle \pi_i, \mathbf{u}_i \rangle - \epsilon h(\pi_i)\}, \qquad (15)$$

where $h(\cdot)$ is a penalty function or regularizer and ϵ is the regularization parameter. According to [44], a proper regularizer $h(\cdot)$ defined on the probability simplex should be continuous over the simplex and smooth on the relative interior of every face of the simplex. Moreover , h should be a strongly convex function, and these assumptions ensure that $QR^\epsilon(\cdot)$ always returns a unique maximizer. The QR^ϵ mapping is referred to as a *quantal response mapping* [45], which allows players to choose suboptimal actions with positive probability. To see how this regularization contributes to active exploration, consider the entropy regularizer $h(x) = \Sigma_{x_i} x_i \log x_i$. In this case, QR^ϵ is

$$QR^\epsilon(\mathbf{u}_i)(a) := \frac{\exp\left(\frac{1}{\epsilon}u_i(a, \pi_{-i})\right)}{\sum_{a' \in \mathcal{A}_i} \exp\left(\frac{1}{\epsilon}u_i(a', \pi_{-i})\right)}, \quad a \in \mathcal{A}_i, \qquad (16)$$

which is also known as the *Boltzmann–Gibbs strategy mapping* [46] or the *soft-max function parameterized by* $\epsilon > 0$. On the one hand, the Boltzmann–Gibbs mapping produces a strategy that assigns more weight to the actions leading to higher payoffs, that is, the larger $\mathbf{u}_i(a) = u_i(a, \pi_{-i})$ is, the larger $QR^\epsilon(\mathbf{u}_i)(a)$ becomes. On the other hand, it always retains positive probabilities for every action when $\epsilon > 0$. Note that QR^ϵ can induce different levels of exploration by adjusting the parameter ϵ. When ϵ tends to zero, the strategy (16) simply returns the action that yields the highest payoff, implying that QR^ϵ reduces to the best response mapping $BR_i(\cdot)$ in (2). As ϵ gets larger, $1/\epsilon$ tends to zero, and the strategy does not distinguish among actions, leading to equal weights for all actions.

Similar to the previous argument, with the noisy feedback, replace $\mathbf{u}_i$ by the estimator $\hat{\mathbf{u}}_i^k$, and the definition of quantal response mapping is then modified accordingly as

$$QR^\epsilon(\hat{\mathbf{u}}_i^k)(a) := \frac{\exp\left(\frac{1}{\epsilon}\hat{\mathbf{u}}_i^k(a)\right)}{\sum_{a' \in \mathcal{A}_i} \exp\left(\frac{1}{\epsilon}\hat{\mathbf{u}}_i^k(a')\right)}, \quad a \in \mathcal{A}_i.$$

Due to the active exploration brought up by QR^ϵ, consider an inertia-free reinforcement learning scheme where the choice map is simply the strategy learning policy QR^ϵ. The corresponding strategy learning scheme is then

$$\pi_i^{k+1} = QR^\epsilon(\hat{\mathbf{u}}_i^{k+1}),$$

where the score function $\hat{\mathbf{u}}_i^k$ is updated according to the following [47]:

$$\hat{\mathbf{u}}_i^{k+1} = \hat{\mathbf{u}}_i^k + \mu_i^k \hat{\mathbf{U}}_i^k. \qquad (17)$$

Fictitious Play

Consider the repeated play between two players, with each player knowing their own utility function. Further, each player can observe actions of the other player and choose an optimal action based on the empirical frequency of these actions.

In fictitious play, from player 1's viewpoint, player 2's strategy at time k can be estimated as $\pi_2^k(a) = \Sigma_{s=1}^k \mathbb{1}_{\{a_2^s = a\}}/k$, $a \in \mathcal{A}_2$, which is the empirical frequency of actions player 2 has implemented up to that point. π_2^k can be computed by a moving average scheme:

$$\pi_2^k = \left(1 - \frac{1}{k}\right)\pi_2^{k-1} + \frac{1}{k}e_{a_2^k}.$$

Using this, player 1 chooses the best response: $a_1^{k+1} = \operatorname{argmax}_{a \in \mathcal{A}_1} u_1(a, \pi_2^k)$ for the next play. The empirical frequency of player 1's implemented actions is updated according to

$$\pi_1^{k+1} = \left(1 - \frac{1}{k+1}\right)\pi_1^k + \frac{1}{k+1}e_{a_1^{k+1}},$$

where $e_{a_1^{k+1}} \in \Delta(\mathcal{A}_1)$ is exactly given by $BR_1(\pi_2^k)$, and the equation is the same as the one in (10), with the learning rate being $\lambda_1^k = 1/k + 1$. Hence, in fictitious play, a player's empirical play follows best response dynamics. Furthermore, if the best response mapping BR is replaced with the quantal response QR^ϵ, an important variant is obtained: stochastic fictitious play [30].

To recap, the learning algorithm operates in the following fashion: At each time k, an unbiased estimator $\hat{\mathbf{U}}_i^k$ is constructed as introduced in (11) using importance sampling, and the score function is updated according to (17). Then, the next strategy is produced by the mapping QR^ϵ, acting on the score function $\hat{\mathbf{u}}_i^{k+1}$, as shown in

$$\begin{aligned}\hat{\mathbf{u}}_i^{k+1} &= \hat{\mathbf{u}}_i^k + \mu_i^k \hat{\mathbf{U}}_i^k,\\ \pi_i^{k+1} &= QR^\epsilon(\hat{\mathbf{u}}_i^{k+1}).\end{aligned} \tag{DA-d}$$

(DA-d) is also referred to as *dual averaging*, pioneered by Nesterov [47], which was originally proposed as a variant of gradient methods for solving convex programming problems. We elucidate the term *dual averaging* later when discussing the relationship between dual averaging and gradient play, where it is demonstrated that (DA-d) can be viewed as a gradient-based algorithm in finite games, with $\hat{\mathbf{u}}_i^k$ being the gradient. Finally, note that in (DA-d), the score function is updated in a manner different than in (BR-d). However, this is merely a matter of presentation. By selecting a proper ϵ, the moving averaging scheme (12) is essentially the same as the discounted accumulation (17) [47], [48]. By adopting the discounted accumulation (17), a connection can later be drawn between dual averaging and gradient play.

Apparently, (DA-d) does not depict how $\pi_i(t)$ evolves in $\Delta(\mathcal{A}_i)$, and it is not straightforward to tell how those good actions bringing higher payoffs are "reinforced" in the sense that probabilities of choosing them increases as the learning process proceeds. "Replicator Dynamics" presents that when choosing entropy regularization, (DA-d) is equivalent to the replicator dynamics (one of the well-known evolutionary dynamics [49]–[51]), which explicitly displays a gradual adjustment of strategies based on the quality of each action. Using an example of population games, it is shown that this connection brings

Replicator Dynamics

Recall that continuous-time learning dynamics under dual averaging is

$$\begin{aligned}\frac{d\hat{\mathbf{u}}_i(t)}{dt} &= \mathbf{u}_i(\pi_{-i}(t)),\\ \pi_i(t) &= QR^\epsilon(\hat{\mathbf{u}}_i(t)).\end{aligned}$$

Now consider the entropy regularizer $h(x) = \Sigma_{x_i} x_i \log x_i$ and let $\epsilon = 1$ for simplicity. Differentiate the strategy $\pi_i(t)$ with respect to time variable in the continuous-time version of DA-d (DA-c), arriving at

$$\begin{aligned}\frac{d\pi_{i,a}(t)}{dt} &= \frac{1}{\left(\sum_{a'} e^{\hat{\mathbf{u}}_{i,a}(t)}\right)^2}\left(\frac{d\hat{\mathbf{u}}_{i,a}(t)}{dt} e^{\hat{\mathbf{u}}_{i,a}(t)} \sum_{a'} e^{\hat{\mathbf{u}}_{i,a'}(t)}\right.\\ &\quad \left. - e^{\hat{\mathbf{u}}_{i,a}(t)} \sum_{a'} e^{\hat{\mathbf{u}}_{i,a'}(t)} \frac{d\hat{\mathbf{u}}_{i,a'}(t)}{dt}\right)\\ &= \pi_{i,a}(t)\left(\frac{d\hat{\mathbf{u}}_{i,a}(t)}{dt} - \sum_{a'} \pi_{i,a'}(t) \frac{d\hat{\mathbf{u}}_{i,a'}(t)}{dt}\right)\\ &= \pi_{i,a}(t)[u_i(a, \pi_{-i}(t)) - u_i(\pi_i(t), \pi_{-i}(t))].\end{aligned} \tag{S1 (RD)}$$

From this equation, it is shown that for a certain action a, if its outcome $u_i(a, \pi_{-i}(t))$ is above the average $u_i(\pi_i(t), \pi_{-i}(t))$, then it will be "reinforced" in the sense that the probability of choosing a gets higher as time evolves. Equation (S1) is referred to as *replicator dynamics* (*RD*) and has been widely used in evolutionary game theory to understand natural selection and population biology. Consider a two-population system and reinterpret the elements in the two-player game using population biology language. For population 1, there are $|\mathcal{A}_1|$ types, and each type is specified by an element, $a \in \mathcal{A}_1$. Let $\pi_{1,a}(t)$ be the percentage of type a in population 1 at time t, and assume that $\pi_1(t)$ is differentiable with respect to time t as the population (which is infinitely large) interacts with the other population in a continuous-time manner.

Population 2 has similar notions. If individuals from the two population meet randomly, then they engage in a competition or a game with a payoff dependent on their types. For example, if type a_1 from population 1 competes with type a_2 from population 2, then payoffs for the two types are given by $u_1(a_1, a_2)$ and $u_2(a_1, a_2)$, respectively. For population i, if it is assumed that the per capita rate of growth is given by the difference between the payoff for type a and the average payoff in the population (a rule studied in [49]), then the percentage of different types within a population is precisely described by

$$\frac{1}{\pi_{i,a}} \frac{d\pi_{i,a}(t)}{dt} = u_i(a, \pi_i(t)) - u_i(\pi_i(t), \pi_{-i}(t)),$$

which is exactly the RD (S1). In addition, as shown in [44], different regularizers lead to different learning dynamics, which display different asymptotic behavior accounts for the evolutionary process under different circumstances.

With (S1) and other related evolutionary dynamics, biologists can predict the evolutionary outcome of the multipopulation system by examining the Nash equilibrium of the underlying game, which brings strategic reasoning into population biology and has a profound influence on evolutionary game theory [50], [51]. Moreover, the Nash equilibrium in this population game, characterized by the limiting behavior of the dynamics under proper conditions [51], represents an evolutionarily stable state of the population (which is an important refinement of the Nash equilibrium). When this stable state is reached, natural selection alone is sufficient to prevent the population from being influenced by mutation [40], [50]. For more details on this refinement and its application in biology, refer to [17], [40], [50], and [51].

learning in games to the broader context of evolutionary game theory [40], [50].

As mentioned previously, reinforcement learning is a continuum of learning algorithms, and (BR-d) and (DA-d) are the two endpoints of the continuum. Naturally, reinforcement learning methods with a blend of both passive and active exploration can be considered, where the exploration stems from both the inertia term and the strategy learning policy, as presented in the following.

Instead of choosing actions greedily, replace the best response $BR_i(\cdot)$ in (14) with $QR^\epsilon(\cdot)$, the quantal response for active exploration. We then obtain the strategy learning scheme [30]

$$\pi_i^{k+1} = (1-\lambda_i^k)\pi_i^k + \lambda_i^k QR^\epsilon(\hat{\mathbf{u}}_i^k).$$

Similar to (BR-d), if utility learning follows the moving average scheme in (11), the resulting reinforcement learning has the following discrete-time learning dynamics

$$\begin{aligned}\hat{\mathbf{u}}_i^{k+1} &= (1-\mu_i^k)\hat{\mathbf{u}}_i^k + \mu_i^k \hat{\mathbf{U}}_i^k,\\ \pi_i^{k+1} &= (1-\lambda_i^k)\pi_i^k + \lambda_i^k QR^\epsilon(\hat{\mathbf{u}}_i^k).\end{aligned} \qquad \text{(SBR-d)}$$

Considering its similarity to (BR-d), (SBR-d) is referred to as (*SBR-d*) in [30] and [52]. Specifically, if the entropy regularizer is adopted, the resulting learning process is called *Boltzmann–Gibbs reinforcement learning* [53] or entropic reinforcement learning, which has been extensively studied in the context of Markov decision processes [54].

Relationships Among Reinforcement Learning Algorithms

Before concluding the discussion of reinforcement learning in finite games, we examine relationships among the introduced learning algorithms. Note that reinforcement learning corresponds to a continuum of learning algorithms, where one algorithm can be converted into the other by adjusting the learning rate λ_i^k in strategy learning (7) and/or the exploration parameter ϵ. The diagram of such a conversion is presented in Figure 4. The discussion associated with this diagram revolves around the learning rate λ_i^k and the exploration parameter ϵ. For simplicity, suppress the subscript and superscript of the learning rate and denote them by λ.

We begin the discussion with the learning rate λ. Unlike (DA-d), the (BR-d) and (SBR-d) are actor-critic learning [38], [55], [56] due to a positive learning rate $\lambda > 0$. Under the actor-critic framework such as (BR-d)-(SBR-d), the player maintains two recursive schemes for updating the estimated utility vector and strategy, respectively. The recursive schemes lead to coupled dynamical systems of $\hat{\mathbf{u}}_i^k$ and π_i^k. In contrast, even though (DA-d) also consists of both the updating schemes for estimated utility vector and the strategy, as the learning rate is zero, there is only one effective dynamical system: the one induced by the estimation of utility vector (17). Another way to see the difference between actor-critic learning (BR-d)-(SBR-d) and (DA-d) is through the corresponding continuous-time learning dynamics in the "Learning Dynamics and Stochastic Approximation" section.

Even though (DA-d) is not an actor-critic learning, its trajectory is closely related to that of (BR-d)-(SBR-d)'s. Intuitively speaking, (DA-d) only differs from the smoothed best response in that (DA-d) does not acquire an inertia term, as the learning rate is zero. Hence, π_i^k in (SBR-d) can be seen as the moving average of $QR^\epsilon(\hat{\mathbf{u}}_i^k)$ in (DA-d). Therefore, it is reasonable to expect that the time average of the trajectory produced by (DA-d) is related to the one produced by the smoothed best response. This intuition is verified in [44] and [57], where it is shown that the time-averaged trajectory of (DA-d) follows (SBR-d) with a time-dependent perturbation $\epsilon(t)$.

Apart from the difference in the learning rates, learning algorithms also display distinct asymptotic behavior due to the difference in the exploration parameter. The exploration parameter ϵ has less drastic consequence under (DA-d) than under the actor-critic learning (BR-d)-(SBR-d). As observed in [44], adding a positive ϵ is equivalent to rescaling the regularizer [that is, replacing $h(\cdot)$ with $\epsilon h(\cdot)$]. As long as $\epsilon > 0$, the regularization $\epsilon h(\cdot)$ is still proper (15). This implies that even though the choice of ϵ affects the speed at which (DA-d) evolves, the qualitative results remain the same. The reader is referred to [44] and [58] for a detailed discussion. There is no exploration or inertia for (DA-d) when $\epsilon = 0$, and in this case, players always choose their actions greedily according to the best response mapping

$$\pi_i^{k+1} = \operatorname*{argmax}_{\pi \in \Delta(\mathcal{A}_i)} \{\langle \pi, \hat{\mathbf{u}}_i^k \rangle\}, \qquad \text{(FTL)}$$

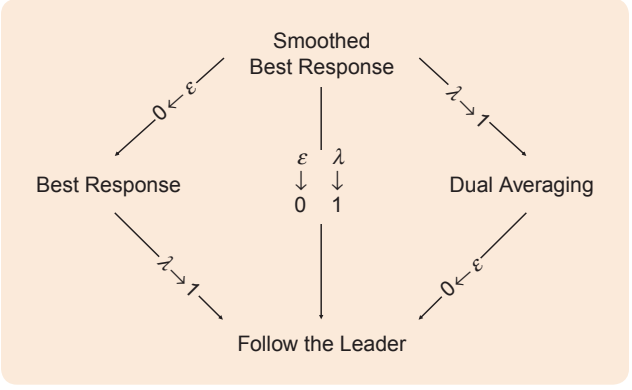


FIGURE 4 The relationships of reinforcement learning algorithms. For $0 < \lambda < 1$ and $\epsilon > 0$, we obtain exploratory reinforcement learning: smoothed best response dynamics (SBR-d), where exploration arises from both the inertia and learning policy. If the active exploration vanishes as ϵ goes to zero, SBR-d reduces to BR-d, an example of exploitative reinforcement learning. In contrast, dual-averaging dynamics is obtained if λ tends to one. Finally, if ϵ goes to zero while λ tends to one, players always choose their actions greedily according to follow the leader.

where $\hat{\mathbf{u}}_i^k$ is the score function of player i (based on its history of play up to round k) that can be updated following (11) or (17). In the online learning literature [21], the aforementioned greedy policy is known as *follow the leader* (*FTL*) and can also be obtained by eliminating the inertia term in (BR-d). Due to a lack of exploration, FTL is too aggressive and can be exploited by the adversary, resulting in a positive, nondiminishing regret [21]. The regret is a measurement of the performance gap between the cumulative payoffs of current-policy FTL and that of the best policy in hindsight.

The exploration parameter plays a more important role in the actor-critic learning, which balances exploration and exploitation [37]. (SBR-d), which is a perturbed version of the best response, can only use the regularization $\epsilon h(\cdot)$ for encouraging active exploration. Thanks to the positive exploration parameter, (SBR-d) enjoys an ϵ-no-regret property (a weak form of external consistency studied in [57] and [59]), which is desired in an adversarial environment [21]. In contrast, (BR-d), due to the myopic nature of the best response mapping (2), does not possess similar properties.

Gradient Play

Thus far, discussions have been limited to learning processes in finite games, where the score function (8) and the choice mapping (9) act on finite-dimensional vectors. For continuous-kernel games, it is not straightforward to extend reinforcement learning as a suitable score function is required to evaluate a continuum of actions, and constructing such a score function can be very challenging. Even though function approximators such as linear [60], [61] or nonlinear [62] ones can be of some help, there is a mathematically more elegant way of leveraging the reinforcement idea based on gradients of utility functions. In other words, instead of seeking the maximizers, we seek a better response by searching along the gradient direction. Such gradient-based learning algorithms, referred to as *gradient play*, are popular in a variety of multiagent settings due to their versatility, ease of implementation, and dependence on local information.

For simplicity, we restrict the discussion to a pure-strategy Nash equilibrium in continuous games [see (4) for the definition and (5) for its variational characterization], to avoid measure-theoretic issues when studying the mixed-strategy case. Further assume that utilities are smooth functions and perfect feedback is available to players, implying that each player can compute the gradient of the utility function given current iterates: $D_i^k = \nabla_{a_i} u_i(a_i^k, a_{-i}^k)$. Even though perfect feedback is assumed here, it is purely for the simplicity of exposition. It is viable for players to estimate the gradient based on the realized payoff under noisy individual feedback by simultaneous perturbation stochastic approximation [63], [64]. Based on this gradient, players update their actions according to

$$\begin{aligned} a_i^{k+1} &= \operatorname{proj}_{\mathcal{A}_i}[a_i^k + \mu_i^k D_i^k], \\ &:= \operatorname*{argmin}_{a \in \mathcal{A}_i} \left\{ \left\| a_i^k + \mu_i^k D_i^k - a \right\|_2^2 \right\}, \end{aligned} \tag{GD}$$

where $\operatorname{proj}_{\mathcal{A}_i}(\cdot)$ is the Euclidean projection operator, and (GD) is the online gradient descent or projected gradient descent [48]. One extensively studied variant of (GD) [48], [65] is

$$\begin{aligned} Y_i^{k+1} &= Y_i^k + \mu_i^k D_i^k, \\ a_i^{k+1} &= \operatorname{proj}_{\mathcal{A}_i}(Y_i^{k+1}), \end{aligned} \tag{LGD}$$

where Y_i^k is an auxiliary variable that aggregates the gradient steps. Such an algorithm is referred to as the *lazy gradient descent (LGD)* [47] because the algorithm aggregates the gradient steps "lazily," without transporting them to the action space as (GD) does. The difference between the two algorithms is illustrated in Figure 5. Note that based on the gradient descent idea, (LGD) and (GD) share the same asymptotic behavior [21], and the two coincide when $\mathcal{A}_i$ is an affine subspace of $\mathbb{R}^n$.

Unlike a purely primal-based algorithm such as (GD), where the trajectory of the algorithm evolves only in the primal space (the action space), (LGD) is a primal-dual scheme, and the interplay between primal variables (actions a_i^k) and dual variables [gradients $D_i(\mathbf{a}^k)$] is of great significance. The main idea of (LGD) is as follows. At the kth round, each player computes the gradient $D_i(\mathbf{a}^k)$ based on the knowledge of utility functions and observations of the opponent's move. Subsequently, players take a step along this gradient in the dual space (where gradients live) and "mirror" the output back to the primal space (the action space) using the Euclidean projection.

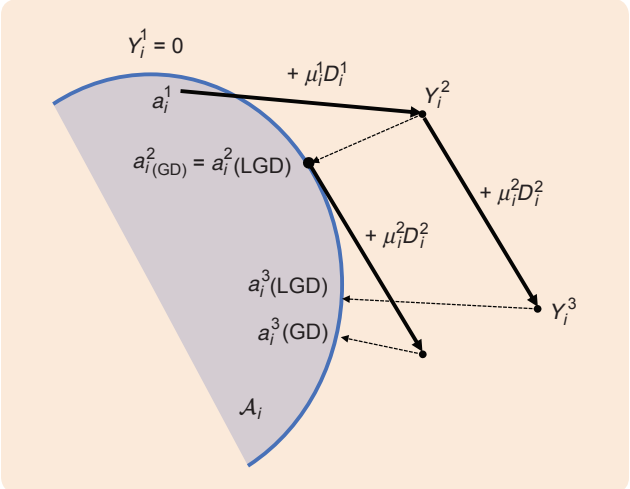


FIGURE 5 An illustration of the difference between gradient descent (GD) (18) and lazy GD (LGD). $a_{i(\mathrm{GD})}^k$ and $a_{i(\mathrm{LGD})}^k$ denote the iterates generated by (18) and LGD, respectively. The LGD first aggregates the gradient steps and then projects the aggregation onto the primal space to generate a new gradient step.

Gradient-based learning algorithms are further investigated in another article in this special issue in the context of generalized Nash equilibrium seeking [25]. The following presents a generalization of (LGD): mirror descent (MD) [47], [65]. Starting with some arbitrary initialization Y_i^1, the MD scheme can be described via the recursion,

$$\begin{aligned} Y_i^{k+1} &= Y_i^k + \mu_i^k D_i(\mathbf{a}^k), \\ a_i^{k+1} &= QR^\epsilon(Y_i^{k+1}), \end{aligned} \tag{MD}$$

where QR^ϵ is the quantal response mapping in the context of the continuous game, defined as

$$QR^\epsilon(Y) = \operatorname*{argmax}_{a\in\mathcal{A}_i}\{\langle Y, a\rangle - \epsilon h(a)\}.$$

When choosing the Euclidean norm as the regularizer (that is, $h(x) = (1/2)\|x\|_2^2$ and $\epsilon = 1$), QR^ϵ reduces to the projection operator $\text{proj}_{\mathcal{A}_i}$. Geometrically, the gradient search step is performed in the dual space, and the primal update is produced by the mapping QR^ϵ. As QR^ϵ "mirrors" the gradient update in the dual space back to the primal space, it is also referred to as the *mirror map* in the online optimization literature [21].

Mirror Descent as Reinforcement Learning in Continuous Games

MD and reinforcement learning (DA-d) share the same choice map, and they are closely connected. It is demonstrated in the following that as a gradient-based algorithm, (MD) can also be cast as a reinforcement learning scheme in continuous games, with Y_i^k being the "score function."

To evaluate a certain action $a \in \mathcal{A}_i$ at time k, consider $\Sigma_{\tau=1}^k(a, a_{-i}^\tau)$ (the cumulative payoff had player i implemented a in the past). The higher the sum, the better action a is as playing a could have resulted in higher payoffs. Hence, the player can choose the next action that is optimal in hindsight:

$$a_i^{k+1} = \operatorname*{argmax}_{a\in\mathcal{A}_i}\left\{\sum_{\tau=1}^k u_i(a, a_{-i}^\tau) - \epsilon h(a)\right\}, \tag{FTRL}$$

where $\epsilon h(\cdot)$ is the regularization introduced in (15), encouraging exploration in the learning process. Based on the optimality in hindsight, this action selection is known as *follow the regularized leader* (*FTRL*) [66]. Moreover, if u_i is well behaved in the sense that it can be approximated by the first-order Taylor expansion (that is, $u_i(a, a_{-i}^\tau) \approx u_i(a_i^\tau, a_{-i}^\tau) + \langle D_i(\mathbf{a}^\tau), a - a_i^\tau\rangle$), then FTRL is equivalent to

$$\begin{aligned} a_i^{k+1} &= \operatorname*{argmax}_{a\in\mathcal{A}_i}\left\{\sum_{\tau=1}^k \langle D_i(\mathbf{a}^\tau), a\rangle - \epsilon h(a)\right\} \\ &= \operatorname*{argmax}_{a\in\mathcal{A}_i}\left\{\left\langle \sum_{\tau=1}^k D_i(\mathbf{a}^\tau), a\right\rangle - \epsilon h(a)\right\} \\ &= QR^\epsilon\left(\sum_{\tau=1}^k D_i(\mathbf{a}^\tau)\right), \end{aligned}$$

which is exactly (MD), despite using an auxiliary variable Y_i^k to aggregate these gradients weighted by the learning rates μ_i^k. In other words, using the first-order expansion, the sum of gradients living in the dual space serves a linear functional for evaluating the quality of the actions. Hence, the sum (or equivalently, Y_i^k) can be treated as a "score function," based on which the mirror map outputs a better action in hindsight, yielding a reinforcement procedure.

Reinforcement Learning as Mirror Descent in Finite Games

In the aforementioned discussion, (MD) is interpreted as "reinforcement learning" in continuous games. This section further shows that the idea of MD can also be employed in finite games, and the resulting learning dynamics is in fact the exploratory reinforcement learning scheme (DA-d).

In finite games, the utility function is not differentiable with respect to the action, as action sets are finite. To leverage gradient play, consider the mixed extension of finite games and the expected utility $u_i(\pi_i, \pi_{-i}) = \langle \pi_i, \mathbf{u}_i(\pi_{-i})\rangle$. Then the gradient of the expected utility with respect to player i's strategy π_i is given by $\mathbf{u}_i(\pi_{-i})$. Naturally, (MD) can be applied to this mixed extension without difficulty. Furthermore, if the gradient is not directly available (for example, learning under noisy feedback), we rely on the unbiased estimator of $\mathbf{u}_i(\pi_{-i}^k)$, $\hat{\mathbf{U}}_i^k$, which can be viewed as an estimator of the payoff gradient D_i in (MD). It can be seen that (MD) for this induced continuous game reduces to the exploratory reinforcement learning in (DA-d). Consequently, the learning scheme (DA-d) is called *dual averaging*: the dual variables, the gradients $\hat{\mathbf{U}}_i^k$, are aggregated first within the dual space and are then "mirrored" back to the primal space by the mirror mapping [47]. A schematic representation of dual averaging is provided in Figure 6.

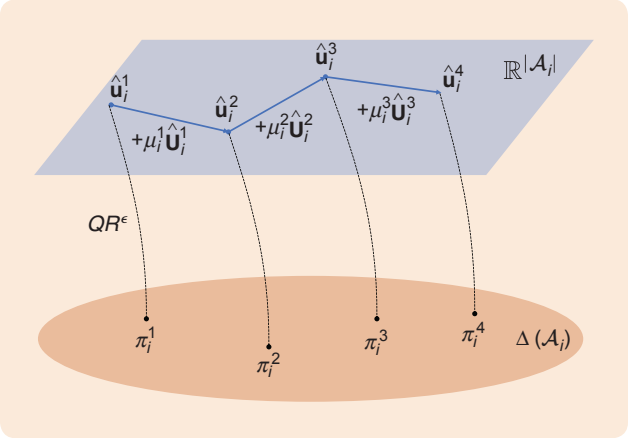


FIGURE 6 A schematic representation of dual averaging. There are no explicit dynamics in the primal space $\Delta(\mathcal{A}_i)$. Instead, the dual variables $\hat{\mathbf{U}}_i^k$ are first aggregated within the dual space $\mathbb{R}^{|\mathcal{A}_i|}$ and are then "mirrored" back to the primal space via the mirror mapping QR^ϵ.

Convergence of Learning in Games

This section examines the asymptotic behavior of learning algorithms introduced in the previous section, with the focus on convergence results of the introduced learning algorithms. Due to the close connection between gradient play in continuous games and reinforcement learning in finite games, the scope is limited to reinforcement learning algorithms in finite games. The reader is referred to [43], [64], and [67]–[69] for the treatment in continuous games. In this section, the discussion is primarily based on stochastic approximation and Lyapunov stability theories [36], [70]. A generic procedure of applying such analytical tools consists of three steps: 1) develop mean-field continuous-time dynamics using stochastic approximation theory; 2) study continuous-time learning dynamics using ODE methods, relating its Lyapunov stability to Nash equilibria of the underlying game; 3) derive convergence results of discrete-time algorithms using asymptotic convergence of corresponding continuous-time dynamics. As the third step is a direct corollary of the results of the first and second steps, the first two steps are articulated in the sequel. Refer to "Stochastic Approximation Theory" and references therein for details on the relationship between discrete-time trajectory and its continuous counterpart.

Learning Dynamics and Stochastic Approximation

With proper F_i^k and G_i^k, learning algorithms allow the players to reach the Nash equilibrium of the game in the limit. Hence, the problem reduces to analyzing limiting behavior of discrete-time systems (BR-d)-(DA-d)-(SBR-d), that is, whether its global attractor comprises equilibria. Direct investigations into such learning dynamics are challenging as stochasticity enters updating rules. For example, the action at time k, a_i^k, is sampled from the strategy π_i^k, and the payoff feedback U_i^k also incurs randomness.

The celebrated stochastic approximation theory allows for shifting focus to the continuous counterpart of the discrete-time dynamics: an ODE whose trajectory enjoys the same asymptotic property. From a technical standpoint, continuous-time dynamics often produce a more comprehensible picture for analysis with fruitful tools. One of the most powerful tools is Lyapunov stability theory. Such a continuous-time framework also allows for connecting learning theory with extensive literature on game dynamics in biology and evolutionary theory [30], where the time interval between two repetitions of the game is infinitesimally small.

Recall that reinforcement learning adopts two coupled, discrete-time, dynamical systems: one for the score function (8) and the other for choice mapping (9)

$$\hat{\mathbf{u}}_i^{k+1} = (1-\mu_i^k)\hat{\mathbf{u}}_i^k + \mu_i^k G_i^k(\pi_i^k, \hat{\mathbf{u}}_i^k, U_i^k, a_i^k),$$
$$\pi_i^{k+1} = (1-\lambda_i^k)\pi_i^k + \lambda_i^k F_i^k(\pi_i^k, \hat{\mathbf{u}}_i^{k+1}, U_i^k, a_i^k).$$

In the following, the continuous-time dynamics associated with (8) and (9) is obtained via stochastic approximation, which paves the way for the ODE-based convergence analysis. We begin with a generic description of learning dynamics under reinforcement learning, and then specify the learning dynamics corresponding to (BR-d)-(DA-d)-(SBR-d). For more details regarding stochastic approximation, refer to "Stochastic Approximation Theory" and references therein.

For simplicity in exposition, assume that the learning policies in (8) and (9) are time invariant (denoted by F_i and G_i, respectively). When the learning policies are time variant, stochastic approximation theory still applies (refer to [53] for more details). Let the mean-field components of (8) and (9) be denoted by $f_i(\pi_i^k, \hat{\mathbf{u}}_i^{k+1}) = \mathbb{E}[F_i(\pi_i^k, \hat{\mathbf{u}}_i^{k+1}, U_i^k, a_i^k) \,|\, \mathcal{F}^{k-1}]$ and $g_i(\pi_i^k, \hat{\mathbf{u}}_i^k) = \mathbb{E}[G_i(\pi_i^k, \hat{\mathbf{u}}_i^k, U_i^k, a_i^k) \,|\, \mathcal{F}^{k-1}]$, respectively. Then note the coupled differential equations

$$\frac{d\hat{\mathbf{u}}_i(t)}{dt} = g_i(\pi_i(t), \hat{\mathbf{u}}_i(t)),$$
$$\frac{d\pi_i(t)}{dt} = f_i(\pi_i(t), \hat{\mathbf{u}}_i(t)),$$

which are closely related to (8) and (9). Using stochastic approximation theory (see "Stochastic Approximation Theory"), the linear interpolations of sequences $\{\pi_i^k\}$ and $\{\hat{\mathbf{u}}_i^k\}$ are the perturbed solutions to the aforementioned differential equations, which are arbitrarily close to the true solution as time approaches infinity. In other words, the convergence results of (8) and (9) can be obtained by studying limiting behavior of the associated differential equations.

Following the same argument, the learning dynamics of (BR-d) can be written as

$$\frac{d\hat{\mathbf{u}}_i(t)}{dt} = \mathbf{u}_i(\pi_{-i}(t)) - \hat{\mathbf{u}}_i(t),$$
$$\frac{d\pi_i(t)}{dt} \in BR_i(\hat{\mathbf{u}}_i(t)) - \pi_i(t). \qquad \text{(BR-c)}$$

If (BR-d) is adopted by every player, then continuous-time dynamics of the strategy profile of all players $\pi(t) = [\pi_1(t), \pi_2(t), \ldots, \pi_N(t)]$ can be studied under the joint best response [see (2)]. Denote the joint utility vector by $\mathbf{u}(\pi(t)) := [\mathbf{u}_1(\pi_{-1}(t)), \mathbf{u}_2(\pi_{-2}(t)), \ldots, \mathbf{u}_N(\pi_{-N}(t))]$, and similarly, joint estimated utility vector by $\hat{\mathbf{u}}(t) := [\hat{\mathbf{u}}_1(t), \hat{\mathbf{u}}_2(t), \ldots, \hat{\mathbf{u}}_N(t)]$. Then, for the strategy profile $\pi(t)$, continuous-time learning dynamics under the best response algorithm is

$$\frac{d\hat{\mathbf{u}}(t)}{dt} = \mathbf{u}(\pi(t)) - \hat{\mathbf{u}}(t), \qquad (18)$$
$$\frac{d\pi(t)}{dt} \in BR(\hat{\mathbf{u}}(t)) - \pi(t). \qquad (19)$$

From its associated learning dynamics, (BR-d) [or equivalently, its continuous-time mean-field dynamics (BR-c)] is an actor-critic learning [37], where the approximation $\hat{\mathbf{u}}(t)$ given by (18) serves as the actor evaluating the performance

Stochastic Approximation Theory

Following the multiple timescale stochastic approximation framework developed in [35] and [S1], (8) and (9) can be written using discrete-time stochastic approximation

$$\begin{aligned}\pi_i^{k+1}-\pi_i^k&=\bar{\lambda}_i^k(f_i(\pi_i^k,\hat{\mathbf{u}}_i^{k+1})+M_i^{k+1}),\\ \hat{\mathbf{u}}_i^{k+1}-\hat{\mathbf{u}}_i^k&=\bar{\mu}_i^k(g_i(\pi_i^k,\hat{\mathbf{u}}_i^k)+\Gamma_i^{k+1}),\end{aligned}\tag{S2}$$

where $f_i(\pi_i^k,\hat{\mathbf{u}}_i^{k+1})$ and $g_i(\pi_i^k,\hat{\mathbf{u}}_i^k)$ are the mean-field components of (8) and (9), respectively, and are defined as

$$\begin{aligned}f_i(\pi_i^k,\hat{\mathbf{u}}_i^{k+1})&=\mathbb{E}[F_i(\pi_i^k,\hat{\mathbf{u}}_i^{k+1},U_i^{k+1},a_i^{k+1})\,|\,\mathcal{F}^{k-1}],\\ g_i(\pi_i^k,\hat{\mathbf{u}}_i^k)&=\mathbb{E}[G_i(\pi_i^k,\hat{\mathbf{u}}_i^k,U_i^{k+1},a_i^{k+1})\,|\,\mathcal{F}^{k-1}].\end{aligned}$$

With the mean-field part defined as the aforementioned equation, $M_i^{k+1}=F_i(\pi_i^k,\hat{\mathbf{u}}_i^{k+1},U_i^{k+1},a_i^{k+1})-f_i(\pi_i^k,\hat{\mathbf{u}}_i^{k+1})$ and Γ_i^{k+1} take a similar form. $\bar{\lambda}_i^k,\bar{\mu}_i^k$ are time-scaling factors dependent on the learning rates λ_i^k,μ_i^k, which account for adjustment of the original step sizes in asynchronous schemes [35], [70]. In synchronous cases, time-scaling factors coincide with original step sizes. Similar to the discussion in the main text [see (18) and (19)], consider the dynamical system of the joint strategy profile π^k and utility vector $\hat{\mathbf{u}}^k$

$$\begin{aligned}\pi^{k+1}-\pi^k&=\bar{\lambda}^k(f(\pi^k,\hat{\mathbf{u}}^{k+1})+M^{k+1}),\\ \hat{\mathbf{u}}^{k+1}-\hat{\mathbf{u}}^k&=\bar{\mu}^k(g(\pi^k,\hat{\mathbf{u}}^k)+\Gamma^{k+1}),\end{aligned}\tag{DSA}$$

where f and g are concatenations of $\{f_i\}_{i\in\mathcal{N}}$ and $\{g_i\}_{i\in\mathcal{N}}$, respectively. $\bar{\lambda}^k,\bar{\mu}^k$ and M^k,Γ^k take similar forms.

As discussed in the "Convergence of Learning in Games" section, to obtain an approximately accurate score function, the two coupled discrete-time systems in (DSA) should operate on different timescales: the score function $\hat{\mathbf{u}}^k$ should be updated sufficiently many times until near convergence before updating the strategy. This two-timescale iteration can be achieved by adjusting the time-scaling factors: $\bar{\lambda}^k$ and $\bar{\mu}^k$ are chosen so that $\lim_{k\to\infty}\bar{\lambda}^k/\bar{\mu}^k=0$. To understand this timescale system, it is instructive to consider a coupled, continuous-time dynamical system, as suggested in [35]:

$$\begin{aligned}\frac{d\pi(t)}{dt}&=f(\pi(t),\hat{\mathbf{u}}(t)),\\ \frac{d\hat{\mathbf{u}}(t)}{dt}&=\frac{1}{\varepsilon}g(\pi(t),\hat{\mathbf{u}}(t)),\end{aligned}\tag{S3}$$

where ε tends to zero. Hence, $\hat{\mathbf{u}}(t)$ is fast transient while $\pi(t)$ is slow. The long-run behavior of the aforementioned coupled system can then be analyzed as if the fast process is always fully calibrated to the current value of the slow process. This suggests investigating the ordinary differential equation

$$\frac{d\hat{\mathbf{u}}(t)}{dt}=g(\pi,\hat{\mathbf{u}}(t)),\tag{S4}$$

where π is held fixed as a constant parameter. Suppose (S4) has a globally asymptotically stable equilibrium $\Lambda(\pi)$, where the mapping $\Lambda(\cdot)$ satisfies regularity conditions specified in [36] and [70]. Then, it is reasonable to expect $\hat{\mathbf{u}}(t)$ given by (S4) to closely track $\Lambda(\pi)$. In turn, this suggests that the investigation into the coupled system (S3) is equivalent to the study of the single-timescale one

$$\frac{d\pi(t)}{dt}=f(\pi(t),\Lambda(\pi(t))),\tag{S5}$$

which would capture the long-run behavior of $\pi(t)$ in (S3) to a good approximation [35].

Informally speaking, to study the convergence of (DSA), its discrete-time trajectory can be related to that of (S3), which is further equivalent to $(\pi(t),\Lambda(\pi(t)))$ specified by (S5). Therefore, Lyapunov stability theory can be applied to (S5) to derive convergence results of the original discrete-time algorithm. We begin with the linear interpolation process of the discrete-time trajectory, which connects (DSA) and its continuous-time counterpart (S3), (S5). Under some regularity conditions [36], for $\{\pi^k\}$ [the sequence generated by (DSA)], the continuous-time process $\bar{\pi}(t):\mathbb{R}_+\to\Delta(\mathcal{A})$ is constructed based on the linear interpolation of $\{\pi^k\}$. Letting $\tau^0=0$ and $\tau^k=\Sigma_{s=1}^k\bar{\lambda}^s$, define

$$\bar{\pi}(t):=\pi^k+(t-\tau^k)\frac{\pi^{k+1}-\pi^k}{\tau^{k+1}-\pi^k},\quad t\in[\tau^k,\tau^{k+1}).$$

Similarly, define a continuous-time process $\bar{\mathbf{u}}(t)$ corresponding to $\{\hat{\mathbf{u}}^k\}$.

As shown in [36] and [70], such a linearly interpolated process $(\bar{\pi}(t),\bar{\mathbf{u}}(t))$ is closely related to flow of the differential equations

$$\begin{aligned}\frac{d\pi(t)}{dt}&=f(\pi(t),\hat{\mathbf{u}}(t)),\\ \frac{d\hat{\mathbf{u}}(t)}{dt}&=g(\pi(t),\hat{\mathbf{u}}(t)).\end{aligned}\tag{S6}$$

Note that (S6) is defined for ease of presentation, and the actual differential inclusion systems involve rearrangement of several terms (refer to [70] for more details). Further, denote the flow of (S6) by

$$\begin{aligned}\Phi_t(\pi^0,\mathbf{u}^0):=\{&(\pi(t),\hat{\mathbf{u}}(t))\,|\,(\pi(t),\hat{\mathbf{u}}(t))\text{ is a solution to (S6),}\\ &\text{with }\pi(0)=\pi^0,\hat{\mathbf{u}}(0)=\mathbf{u}^0\}.\end{aligned}$$

The key to the stochastic approximation theory lies in the fact that in the presence of a global attractor for (S6), the continuous-time process $(\bar{\pi}(t),\bar{u}(t))$ asymptotically tracks the flow with arbitrary accuracy over windows of arbitrary length [36],

$$\lim_{t\to\infty}\sup_{s\in[0,T]}\operatorname{dist}\{(\bar{\pi}(t+s),\bar{\mathbf{u}}(t+s)),\Phi_s(\bar{\pi}(t),\bar{\mathbf{u}}(t))\}=0,$$

where $dist\{\cdot,\cdot\}$ denotes a distance measure on $\Delta(\mathcal{A})\times\mathbb{R}^{\mathcal{A}}$. Refer to $(\bar{\pi}(t),\bar{\mathbf{u}}(t))$ as an asymptotic pseudotrajectory of the dynamics (S6). In other words, to study the convergence of (DSA), the convergence analysis of (S6) is used, which can be addressed by Lyapunov stability theory, as depicted in [36] and [70]. The key conclusion is that if there is a global attractor A for (S5), then the interpolated process $(\bar{\pi}(t),\bar{\mathbf{u}}(t))$ [or simply, $(\pi^k,\bar{\mathbf{u}}^k)$] converges almost surely to $(A,\Lambda(A))$.

REFERENCE

[S1] H. Kushner and G. G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*. New York, NY, USA: Springer Science & Business Media, 2003, vol. 35.

To analyze the convergence of two-timescale dynamics, one can study its equivalent single-timescale dynamics.

of the current strategy profile, while the strategy update (19) is the critic that improves the strategy.

As observed in the literature [37], the performance of the actor-critic learning relies on the quality of evaluation from the actor. One approach to obtain a satisfying actor in learning is to leverage the two-timescale idea [35], according to which (18) should operate at a faster timescale than (19). Intuitively speaking, to obtain a $\hat{\mathbf{u}}(t)$ that can approximately evaluate the current strategy profile $\pi(t)$, the player must wait until $\hat{\mathbf{u}}(t)$ nearly converges before it updates the strategy using (19). To analyze the convergence of two-timescale dynamics, one can study its equivalent single-timescale dynamics. As the actor (18) runs at a faster timescale, the system (18) and (19) can be "decoupled" in the following way: By fixing $\pi(t) = \pi$, the faster timescale update (18) converges to $\mathbf{u}(\pi)$, where π is viewed as a parameter, Then, after the convergence of the fast dynamics to an equilibrium $\mathbf{u}(\pi)$, the slow dynamics (19) is set in motion, where $\hat{\mathbf{u}}(t)$ is replaced by its equilibrium point $\mathbf{u}(\pi(t))$ and the resulting learning dynamics is

$$\frac{d\pi(t)}{dt} \in BR(\pi(t)) - \pi(t). \tag{20}$$

As illustrated in "Stochastic Approximation Theory," the coupled dynamics (18), (19), and the single-timescale (20) share similar asymptotic behaviors. Hence, we can focus on the much simplified one (20) for the derivation of the convergence results. For more details about the two-timescale learning and the derivation of the equivalent dynamics, refer to "Stochastic Approximation Theory" and references therein.

Applying the same argument to (SBR-d) yields

$$\begin{aligned} \frac{d\hat{\mathbf{u}}_i(t)}{dt} &= \mathbf{u}_i(\pi_{-i}(t)) - \hat{\mathbf{u}}_i(t), \\ \frac{d\pi_i(t)}{dt} &= QR^{\epsilon}(\hat{\mathbf{u}}_i(t)) - \pi_i(t), \end{aligned} \tag{SBR-c}$$

and its equivalent dynamics regarding the joint strategy profile is

$$\frac{d\pi(t)}{dt} = QR^{\epsilon}(\mathbf{u}(\pi(t))) - \pi(t). \tag{21}$$

Unlike (BR-d) and (SBR-d), (DA-d) does not belong to the class of actor-critic methods. To see this, note its continuous-time dynamics

$$\begin{aligned} \frac{d\hat{\mathbf{u}}_i(t)}{dt} &= \mathbf{u}_i(\pi_{-i}(t)), \\ \pi_i(t) &= QR^{\epsilon}(\hat{\mathbf{u}}_i(t)). \end{aligned} \tag{DA-c}$$

Similar to the previous argument, learning dynamics for the strategy profile is

$$\begin{aligned} \frac{d\hat{\mathbf{u}}(t)}{dt} &= \mathbf{u}(\pi(t)), \\ \pi(t) &= QR^{\epsilon}(\hat{\mathbf{u}}(t)), \end{aligned} \tag{DA}$$

where the dynamics regarding $\hat{\mathbf{u}}(t)$ does not produce an approximation of $\mathbf{u}(\pi(t))$. Instead, it gives the cumulative payoff: $\hat{\mathbf{u}}(t) = \int_0^t \mathbf{u}(\pi(\tau))d\tau + \hat{\mathbf{u}}(0)$. It is straightforward to see that as there is only one differential equation in (DA), the resulting autonomous dynamical system is related only to $\hat{\mathbf{u}}(t)$. Hence, there is no additional dynamics regarding the strategy update, which makes (DA) fundamentally different from (BR-c) and (SBR-c).

Nash Equilibrium and Lyapunov Stability

As the various learning algorithms belong to different classes, discussions regarding the convergence results of the introduced learning dynamics are organized in the following way. We begin with (DA-d) [or equivalently, its continous-time dynamics (DA)], a type of gradient-based dynamics, then proceed to (BR-c) and the (SBR-c).

Dual Averaging

Consider learning dynamics of the joint strategy profile and the estimated utility vector under (DA)

$$\begin{aligned} \frac{d\hat{\mathbf{u}}(t)}{dt} &= \mathbf{u}(\pi(t)), \\ \pi(t) &= QR^{\epsilon}(\hat{\mathbf{u}}(t)). \end{aligned} \tag{DA}$$

This compact form implies that (DA) is an autonomous system evolving in the dual space. Similar to the discussion in the "Gradient Play" section, the terminology in [47] and [48] is adopted, where the gradient $\mathbf{u}(\pi(t))$ is the dual variable and the corresponding space is termed *dual space*. As shown in [44], (DA) is a well-posed dynamical system in the dual space in that it admits a unique global solution for every initial $\hat{\mathbf{u}}(0)$. Furthermore, it can be shown that the dynamics of $\pi(t)$ on the game's strategy space induced by (DA) under steep regularizers is also well posed [44], [58]. However, well posedness of the induced dynamics under generic regularizers remains unclear [44]. The reason lies

in the fact that under steep regularizers such as the entropy regularizer, the projected dynamics regarding $\pi(t)$ evolves within the interior of the simplex, and the resulting ODE is also well posed in the primal space (which need not hold for nonsteep regularizers). For more generic choices of QR and related stability analysis, refer to [44].

Even though studying stability of the induced dynamics in the primal space may not be viable due to the well-posedness issue, the asymptotic behavior of $\pi(t)$ can be characterized by investigating its dual $\hat{\mathbf{u}}(t)$. Toward that end, $\pi(t) = QR^{\epsilon}(\hat{\mathbf{u}}(t))$ is referred to as the *induced orbit of* (*DA*) (or simply, *orbit*), and the following notions regarding the stability and stationarity of $\pi(t)$ are introduced (which are adapted from [44]).

Definition 4

Denote by $\operatorname{im}(QR^{\epsilon})$ the image of QR^{ϵ}. For $\pi(t) = QR(\mathbf{u}(t))$, an orbit of (DA), a fixed $\pi^* \in \prod_{i\in\mathcal{N}} \Delta(\mathcal{A}_i)$ is

- *stationary*, if $\pi(t) = \pi^* \in \operatorname{im}(QR^{\epsilon})$ for all $t \geq 0$, whenever $\pi(0) = \pi^*$
- *Lyapunov stable*, if for every neighborhood U of π^*, there exists a neighborhood U' of π^* such that $\pi(t) \in U$ for all $t \geq 0$, whenever $\pi_0 \in U' \cap \operatorname{im}(QR^{\epsilon})$
- *attracting*, if there exists a neighborhood U such that $\pi(t) \to \pi^*$ as $t \to \infty$, whenever $\pi_0 \in U \cap \operatorname{im}(QR^{\epsilon})$
- *globally attracting*, if π^* is attracting, with the attracting basin being the entire image $\operatorname{im}(QR^{\epsilon})$
- *asymptotically stable*, if π^* is both attracting and Lyapunov stable
- *globally asymptotically stable*, if π^* is both globally attracting and Lyapunov stable.

Similar to the Folk theorem of evolutionary game theory [40], there is an equivalence between the stationary points of (DA) and the Nash equilibria [40], [44]: any stationary point is a Nash equilibrium, and conversely, every Nash equilibrium that is within the image of the mirror map (15) is a stationary point. In addition to the relationship between the Nash equilibrium and the stationary point, another important question is *"Are Nash equilibria of the underlying game (globally) asymptotically stable under (DA)?"*

Answering this question requires revisiting a variational characterization of the Nash equilibrium, which bridges the equilibrium concepts associated with two different mathematical models: games and dynamical systems. Recall that the Nash equilibrium is equivalent to the solution of the variational inequality

$$\langle \mathbf{u}(\pi^*), \pi - \pi^* \rangle \leq 0, \quad \text{for all } \pi \in \prod_{i\in\mathcal{N}} \Delta(\mathcal{A}_i). \tag{SVI}$$

As the utility function $u_i(\pi_i, \pi_{-i})$ is linear in π_i, the SVI is equivalent to the Minty-type variational inequality

$$\langle \mathbf{u}(\pi), \pi - \pi^* \rangle \leq 0, \quad \text{for all } \pi \in \prod_{i\in\mathcal{N}} \Delta(\mathcal{A}_i), \tag{MVI}$$

which implies that the Nash equilibrium π^* is the solution to the (MVI) [26]. Then, to answer the question of interest, it suffices to investigate whether the solution to the (MVI) is attracting under (DA). As discussed in [67], the answer is negative: not every Nash equilibrium of an N-player, general-sum game is attracting. To ensure the convergence of (DA), an additional condition must be imposed on the (MVI).

Definition 5 (Variational Stability) [44]

π^* is said to be variationally stable if there exists a neighborhood U of π^* such that

$$\langle \mathbf{u}(\pi), \pi - \pi^* \rangle \leq 0, \quad \text{for all } \pi \in U, \tag{VS}$$

where equality holds if and only if $\pi^* = \pi$. In particular, if $U = \prod_{i\in\mathcal{N}} \Delta(\mathcal{A}_i)$, π^* is said to be globally variationally stable.

The definition of variational stability (VS) can be extended to sets [44]. Let a subset $\Pi^* \subset \prod_{i\in\mathcal{N}} \Delta(\mathcal{A}_i)$ be closed and nonempty. Π^* is said to be variationally stable if there exists a neighborhood U of Π^* such that

$$\langle \mathbf{u}(\pi), \pi - \pi^* \rangle \leq 0, \quad \text{for all } \pi \in U, \pi^* \in \Pi^*, \tag{22}$$

where equality holds for a given $\pi^* \in \Pi^*$ if and only if $\pi \in \Pi^*$.

"VS" is proposed in [44] as a relaxation of the monotonicity condition of the pseudogradient mapping of the game, such as $\mathbf{u}(\pi)$ in the mixed extension of finite games or $D(a)$ in continuous games. VS alludes to the seminal notion of evolutionary stability introduced in [49], which is in a spirit similar to the variational characterization of the evolutionarily stable state studied in [40]. An equivalent notion is developed in the work on gradient-based learning [67], named *locally asymptotically stable Nash equilibria (LASNE).* As its name suggests, Nash equilibria satisfying the VS are asymptotically stable under gradient-based dynamics. Likewise, the equilibria satisfying global VS are globally asymptotically stable Nash equilibria (GASNE). Refer to [67] and references therein for more details about this characterization of Nash equilibria.

What is presented in this section provides a generic criterion for examining the convergence of gradient-based dynamics (DA). Based on the notion of VS, the following discusses some concrete cases where the learning dynamics converge, either locally or globally, to Nash equilibria. As shown in [43], for any finite games, every strict Nash equilibrium satisfies (VS) and hence is LASNE. Therefore, every strict Nash equilibrium in finite games is locally attractive. On the other hand, to ensure global convergence, the underlying Nash equilibrium must be GASNE or equivalently satisfy the global VS. For finite games, the existence of a potential implies monotonicity, which further implies the existence of globally variationally stable Nash equilibria [43]. Hence, for potential games [44], [71] and monotone

games [43], [72], regardless of the initial points, the orbit of (DA) always converges to the set of Nash equilibria. These discussions are summarized in the following, where 1) and 2) are direct extensions of the folk theorem of evolutionary dynamics [40], while 3)–5) are corollaries of variational characterization of Nash equilibria in [44] and [67].

For every finite game, the Nash equilibrium can be characterized using the language of Lyapunov stability [44], [58]. For a fixed $\pi^* \in \Pi_{i\in N}\Delta(\mathcal{A}_i)$,

1) if π^* is stationary, it is a Nash equilibrium.
2) if π^* is Lyapunov stable, then π^* is a Nash equilibrium.
3) if π^* is a Nash equilibrium and it falls within the image of the mirror map, then it is stationary.
4) if π^* is a strict Nash equilibrium, it is asymptotically stable.
5) if π^* is a Nash equilibrium of a potential game or a monotone game, it is globally asymptotically stable.

(BR-d)

The analysis of (BR-d) [or equivalently, (20)] is more involved than that of (DA-d) [or equivalently, (DA)]. The theoretical challenge is mainly due to the discontinuous, set-valued nature of the best response mapping (2). As a differential inclusion, (20) typically admits nonunique solutions through every initial point [36]. Early works have established the convergence results on (BR-d) for games with special structures: (BR-d) converges to the Nash equilibrium in zero-sum games [39], [68], [73] (where the Nash equilibrium is essentially a saddle point), two-player strictly supermodular games [50], and finite potential games [36], [39]. However, although most of these works still rely on the Lyapunov argument [36], [39], [68], [73], they do not directly reveal any generic relationship between Lyapunov stability and Nash equilibrium in general multiplayer nonzero-sum games and are mostly on an ad hoc basis.

Recent endeavors on the study of (BR-d) have helped shed some light on the asymptotic behavior of (BR-d) by relating the best response vector field $BR(\pi) - \pi$ to the gradient field $\mathbf{u}(\pi)$, which renders (BR-d) in some potential games [74], [75] as an approximation of the gradient-based dynamical system [74]. For the finite potential games considered in [74], additional regularity conditions are imposed (which are closely related to the notion of VS introduced in the previous section). Therefore, variational characterization of the Nash equilibrium and VS becomes relevant under (BR-d). Following this line of reasoning, it is shown in [74] that in regular potential games, (BR-d) is well posed for almost every initial condition and converges to the set of Nash equilibria.

Smoothed Best Response

As shown from the explicit expression, (SBR-d) only differs from (BR-d) in the operator $QR^{\epsilon}(\cdot)$, which serves as a perturbed best response [76]. The perturbation is determined by ϵ [57]. Hence, if ϵ tends to zero, it is straightforward to see that (SBR-d) will enjoy the same asymptotic property as (BR-d), which implies that identical results should also be achievable for smoothed best response with vanishing exploration. This intuition is verified in [52], [69], where (SBR-d) [or equivalently, (21)] is shown to converge in zero-sum, potential, and supermodular games.

On the other hand, with a constant ϵ, it is not realistic to expect (SBR-d), essentially a fixed-point iteration, to always converge to the exact Nash equilibrium. Hence, a new equilibrium concept is introduced in the literature, which is termed the *perturbed Nash equilibrium* in [77] and [78] or Nash distribution in [38] and [56]. The new equilibrium is defined as the *fixed point of the regularized best response mapping* in (15). This article does not include detailed discussions on this topic as the convergence analysis still rests on the standard Lyapunov argument, and the epistemic justification of such equilibrium [30], [39] is beyond the scope of this article. The reader is referred to [30], [56], [69], and [78] for a rigorous treatment of this new equilibrium.

Beyond Stochastic Approximation

In addition to stochastic approximation and related ODE methods, another class of widely applied learning algorithms is built upon Markov chain (MC) theory [79], which is termed *learning by trial and error* (*LTE*) [80]. Even though the name of the proposed learning suggests its similarity to reinforcement learning, the learning process is quite different in the sense that there are no explicit score functions or choice mappings in the proposed method. In LTE, there are two basic rules: 1) players occasionally experiment with alternative strategies and keep the new strategy if and only if it leads to a strict increase in payoff and 2) if the player experiences a payoff decrease due to a strategy change by someone else, it starts a random search for a new strategy. Eventually, it settles on a new strategy with a probability that increases monotonically with its realized payoff. In other words, the "error" part relies on the realized payoff, and no advanced device (such as score functions like Q-functions [110] or estimated utilities) is needed, while the "trial" part is a random search procedure implemented according to the two basic rules. A novel feature of the process is that different search procedures are triggered by different psychological states or moods, where mood changes are induced by the relationship between players' realized payoffs and their current payoff expectations. To be specific, there are four moods: *content* (*C*), *hopeful* (*H*), *watchful* (*W*), and *discontent* (*D*), and different moods lead to different random search procedures. Briefly, players will explore new strategies with high probabilities when in W and D, while keeping the current one with high probabilities if the mood is C or H. The details can be found in [80], and a concise summary is provided in [81].

This mood-based trial and error is different from reinforcement learning introduced in the previous section,

where the exploration is not determined explicitly by the score function and the choice mapping. Hence, LTE does not fit the stochastic approximation framework introduced in the previous section. Instead, the associated convergence proof relies on perturbed MC theory [79], [82]. It is shown in [80] that in a two-player finite game, if there exists at least a pure Nash equilibrium, then LTE guarantees that pure Nash equilibrium is played at least $1-\epsilon$ of the time (where ϵ is the probability of exploring new strategies). For an N-player finite game, if the game is *interdependent* [80] and there exists at least one pure Nash equilibrium, the same theoretical guarantee for the two-player case also holds. It is not surprising that LTE does not achieve convergence in conventional ways (that is, almost sure convergence and convergence in the mean as players will always explore new strategies with positive probability at least ϵ). The proposed learning method and its variants have also been applied to learning efficient equilibrium [83] (Pareto dominant, maximizing social welfare), learning efficient correlated equilibria [84], achieving Pareto optimality [85], and other related works in engineering applications, such as cognitive radio problems [34].

The idea of trial and error in LTE leads to many important variants, such as sample experimentation dynamics in [82] and optimal dynamical learning [81], [85], which also rely on perturbed Markov processes for equilibrium seeking. Even though the convergence results of these algorithms all rest on MC theory [79], analysis of their performance remains unclear due to computation complexity of the inherent MC generated by these algorithms. To circumvent the dimensionality issue regarding the number of states in the original MC, an approximation-based, dimension-reduction method is proposed in [81], which allows numerical convergence analysis for LTE and its variants based on Monte Carlo simulations. Also note that a simplified trial-and-error algorithm is theoretically analyzed in [86], where the optimal exploration rate is identified and the associated convergence rate is discussed. It is not unrealistic to expect that a similar argument may apply to LTE and its variants. However, technical challenges regarding the dimensionality should not be downplayed.

Resurgence of Learning in Games

With ML algorithms being increasingly deployed in real-world applications, there is a resurgence in research endeavors on multiagent learning and learning in games [87]. In addition to the line of research driven by evolutionary dynamics dating back to 1950s [40], [50], the current wave of learning theory development is mainly driven by a desire to better understand and improve the performance of ML algorithms in a competitive environment. In general, there are two possible roles that game-theoretic methods can play in ML study: 1) Game-theoretic methods are an add-on for improving the performance of ML algorithms. 2) Certain ML problems manifest the game features, which call for game-theoretic tools. For supervised learning, the recent interest in adversarial learning techniques serves as an example of how game-theoretic models and learning methods can be used to robustify ML [88], [89], where potential attacks or disturbances are viewed as strategic moves of an opponent. On the other hand, there are problems in unsupervised learning where game-theoretic models are no longer tools for solving the problem but the problem itself. Generative adversarial networks (GANs) [90] are an approach to generative modeling using deep learning methods, involving automatically discovering and learning the patterns of input data in such a way that newly generated examples output by the generative model (generator) cannot be distinguished from the input. In game-theoretic language, the training process of a GAN is essentially a learning process in a zero-sum game between the generator and the discriminator, where the generator tries to generate new samples that plausibly could have been drawn from the original data set, while the discriminator tries to select those fake ones produced by the generator. We do not intend to provide a comprehensive survey for these ML applications; instead, the reader is referred to [87] and [88].

Despite different contexts under which the learning theory is studied, recent research efforts mainly revolve around the following three aspects:

1) learning dynamics in general multiplayer repeated games
2) learning dynamics in repeated games with acceleration design
3) learning dynamics in dynamic games in a decentralized manner.

The first research direction is a natural follow up to the study of evolutionary dynamics [40], [50], which aims to bring learning in games to a broad range of ML applications because in ML, the game structure is specified by the underlying data and may not enjoy any desired properties. Recall that convergence results and asymptotic behaviors regarding the three dynamics, (BR-c)-(SBR-c)-(DA-c), are discussed with the assumption that the underlying game acquires special structures, such as potential games, supermodular games, and zero-sum games. However, for games with fewer assumptions on the utility function, there is still a lack of understanding of the dynamics and the limiting behavior of learning algorithms. One of the central questions of this direction is "*What are the relationships between Nash equilibria and stationary points as well as attracting sets under the learning dynamics?*" Recent attempts try to answer this question from a variational perspective [91] and provide various characterizations of Nash equilibria with desired properties under gradient-based dynamics [58], [67], [92]. Furthermore, considering its applications in ML problems, learning algorithms in stochastic settings are of great significance in recent studies. Refer to [67] and [93] for more details and [23] for an introduction to stochastic Nash equilibrium seeking.

The second research direction (which attracts attention from the ML, optimization, and control communities) is directly related to the design of ML algorithms. The goal is to develop acceleration techniques that improve the performance of learning algorithms. Based on the understanding of first-order, gradient-based dynamic games such as (18) (LGD), recent research efforts have focused on high-order gradient methods (which can be dated back to Nesterov's momentum idea [48]), with researchers endeavoring to propose a general framework that generalizes the momentum for generation of accelerated gradient-based algorithms [91]. On account of the close relationships among Nash equilibrium, variational problems, and dynamical systems [26], one approach for developing acceleration is to generalize the concept momentum by formulating the equilibrium seeking as a variational (optimization) problem [26], [94], and then investigate acceleration methods within the optimization context using, for example, variational analysis [91], extragradient [94], and differential equations [95]. In addition to these works, the reader is referred to [25] for a review on the optimization-based approach. On the other hand (as depicted in Figure 3), a learning process, in general, is a feedback system, and it is not surprising that control theory can play a part in designing the acceleration. For example, recent studies on reinforcement learning demonstrate that passivity-based control theory can be leveraged in designing high-order learning algorithms [96], [72], where the learning rule is the control law to be designed. In [97], the use of memory in best response maps is promoted to accelerate convergence in Nash seeking and demonstrates substantial improvements. In addition to the previously mentioned references, the reader is referred to [98] for a review on control-theoretic approaches on distributed Nash equilibrium seeking, and [99] for the use of extreme seeking in the learning process.

The recent advance in the third research direction is, in part, driven by multiagent reinforcement learning and its applications such as multiagent robotic control [100]–[102]. Unlike the first two directions, where the learning dynamics are primarily studied in the context of repeated games, the third research direction focuses on games with dynamic information (see the "Dynamic Games" section). In this context, the appropriate learning objective, out of practical consideration [22], is to obtain stationary strategies that are subgame perfect [103] (see the "Dynamic Games" section for the definition of subgame perfectness). Unlike the first two, where the change to payoffs resulting from a certain action completely stems from the opponents' move, the feedback each player receives in dynamic games not only depends on other players' moves but also the dynamic environment. Moreover, when making decisions at each state, players must trade off current stage payoffs for estimated future payoffs while forming predictions on the opponent's strategies. A dynamic tradeoff makes the analysis of learning in stochastic games potentially challenging [104].

The earlier works for such Markov perfect Nash equilibria are largely based on dynamic programming [105], [106], which requires global information feedback (a restrictive assumption in practice). The recent efforts focus on various approaches to lessen this requirement. Currently, there are mainly three areas of research regarding learning in dynamic games. The first approach is to extend learning dynamics in repeated games to dynamic ones. Built upon similar ideas in (BR-d), two-timescale dynamics for zero-sum Markov games is considered in [104] and [107]. Meanwhile, gradient play is also investigated in linear-quadratic, zero-sum games [67], [108], [109]. The key challenge in the approach, particularly in the case of Markov games, is to properly construct the score function (which balances current stage and future payoffs). Refer to the mentioned references for more details and [87] for an overview.

The second approach is to extend learning methods in single-agent Markov decision processes to Markov games. However, the direct extension of methods such as Q-learning [110], policy gradient [37], and actor-critic [55] often fail to deliver desired results due to the nonstationarity issue [111]. One natural way to overcome the nonstationarity issue is to allow players to exchange information with neighbors [112], [113], enabling players to jointly identify nonstationarity created by the dynamic environment. For more details regarding this approach, refer to recent reviews [87], [111]. Finally, the third approach considers a unilateral viewpoint of dynamic games. Unlike the first two approaches where learning processes are still investigated in a competitive environment, the third one interprets learning in Markov games as an online optimization problem [114], [115], where players make decisions independently based on the received feedback. This approach accounts for fully decentralized learning where, from each player's perspective, other players are considered as a part of the environment. The key idea of this approach is to leverage the regret-minimization technique [21], which leads to many successes in solving extensive-form games of incomplete information [116]. Despite recent advances regarding the first two approaches [67], [87], [104], [107], [117] and positive results for the last one [114], [115], [118], we still lack a unified framework and thorough understanding regarding the learning process in general Markov games. Decentralized learning in dynamic games remains an open area for researchers from diverse communities.

GAME-THEORETIC LEARNING OVER NETWORKS

Learning in games is not only intellectually interesting but also practically useful. When combined with game-theoretic modeling, such learning methods (thanks to their decentralized and adaptive nature) provide a comprehensive tool kit for designing resilient, agile, and computationally efficient controls or mechanisms for diverse applications of networks.

This section demonstrates that such a combination of game-theoretic models and associated learning dynamics, referred to as *game-theoretic learning*, has become indispensable for modern network problems. On the one hand, these networks often admit complex topological structures and heterogeneous nodes, resulting in large-scale complex systems (making centralized controls or mechanisms either impractical or costly). In contrast, game-theoretic models treat each node in the network as a rational and self-interested player. The heterogeneous nature is captured by players' distinct utilities and action sets as well as information available to them, leading to a bottom-up approach for designing decentralized and scalable mechanisms and controls. On the other hand, modern networked systems (such as wireless communication networks and smart grids) operate in a dynamic or adversarial environment, calling for learning-based mechanisms that are responsive to changes in the environment or malicious attacks from adversaries. As shown in the previous section, game-theoretic learning provides a self-adaptive procedure for each player in the system, according to which players adjust their moves based on feedback from the environment (resulting in desired collective behaviors).

Thanks to its advantageous features over the centralized approach, game-theoretic learning has gained popularity among researchers working on multiagent systems and network applications. There have been numerous encouraging successes in many fields, ranging from wireless and Internet of Things (IoT) communication networks [119]–[123], smart grid and power networks [3], [4], [124], [125], and infrastructure systems [126]–[129], to cybersecurity applications [130]–[134]. In the following, some representative works in these fields are presented. The section focuses on the applications of learning methods in wireless communications, smart grids, and distributed ML. Other related applications will be briefly discussed at the end of the section.

Next-Generation Wireless Networks

Next-generation wireless communication technologies offer an accommodating and adaptive solution that meets the requirements of a diverse range of use cases within a common network infrastructure, providing the necessary flexibility for service heterogeneity and compatibility [7]. Such architecture, as noted in [135], aims to meet the following demands:

- increased indoor and small cell/hot spot traffic (which will comprise the majority of mobile traffic volume), leading to complex network structures
- higher numbers of connected heterogeneous devices (stemming from the IoT), which will support massive machine-to-machine communications and applications
- improved energy consumption or efficient power control for reducing carbon footprint.

From a system science perspective, these requirements impose a large-scale, time-variant, and heterogeneous network topology on modern wireless communication systems, as shown in Figure 7. Hence, it is impractical to manage/secure the wireless communications network in a centralized fashion. To address this challenge, game-theoretic learning provides a scalable, distributed solution with adaptive attributes. In the following, the dynamic secure routing mechanism is used to illustrate how game-theoretic learning contributes to a resilient and agile communication system.

Security of routing in a distributed cognitive radio (CR) network is a prime issue, as the routing may be compromised by unknown attacks, malicious behaviors, and unintentional misconfigurations (which makes it inherently fragile). Even with appropriate cryptographic techniques, routing in CR networks is still vulnerable to attacks in the physical layer, which can critically compromise performance and reliability. Most of the existing work focuses on resource-allocation perspectives, which fail to capture a user's lack of knowledge of the attacker due to the distributed mechanism. To address these issues, [120] provides a learning-based secure scheme that allows the network to defend against unknown attacks with a minimum level of deterioration in performance.

Consider $\mathcal{G}_w := (\mathcal{N}_w, \mathcal{E}_w)$, which is a topology graph for a multihop CR network, where $\mathcal{N}_w = \{n_1, n_2, \ldots, n_N\}$ is a set of secondary users, and $\mathcal{E}_w$ is a set of links connecting these users. The system state s indicates whether nodes are occupied by the primary users. The objective of the secondary user is to find an optimal path to its destination. In multihop routing, a secondary user n_i starts with an exploration of neighboring nodes that are not occupied and then chooses a node among them, to which the user routes data. The selected node initializes another exploration process for discovering the next node, and the same process is repeated until the destination is reached.

Let $\mathcal{P}_i(0, L_i) := \{(n_i, l_i), l_i \in \{0, 1, 2, \ldots, L_i\}\}$ be the multihop path from the node n_i to its destination, where L_i is the total number of explorations until it reaches its destination. Suppose there are J jammers in the network, the set of these jammers is given by $\mathcal{J} := \{1, 2, \ldots, J\}$. Let $\mathcal{R}_j$, $j \in \mathcal{J}$, be the set of nodes that are under the influence of jammer j. Denote the joint action of the jammers by $\mathbf{r} = [r_j]_{j \in \mathcal{J}}$, where $r_j \in \mathcal{R}_j$. A zero-sum game formulation is proposed in [120], where secondary users aim to find an optimal routing path by selecting $\mathcal{P}_i(0, L_i)$, while the jammers aim to compromise the data transmission by choosing $\mathbf{r}$. The expected utility function is

$$\mathbb{E}_s[u_i(s, \mathcal{P}_i(0, L_i), \mathbf{r})] = -\mathbb{E}_s\left[\sum_{l_i=1}^{L_i}\left(\ln q^{(n_i,l_i)}_{(n_i,l_i-1)} + \lambda \tau^{(n_i,l_i)}_{(n_i,l_i-1)}\right)\right],$$

where $q^{(n_i,l_i)}_{(n_i,l_i-1)}$ is the probability of successful transmissions from node $(n_i, l_i - 1)$ to node (n_i, l_i), and $\lambda^{(n_i,l_i)}_{(n_i,l_i-1)}$ is the

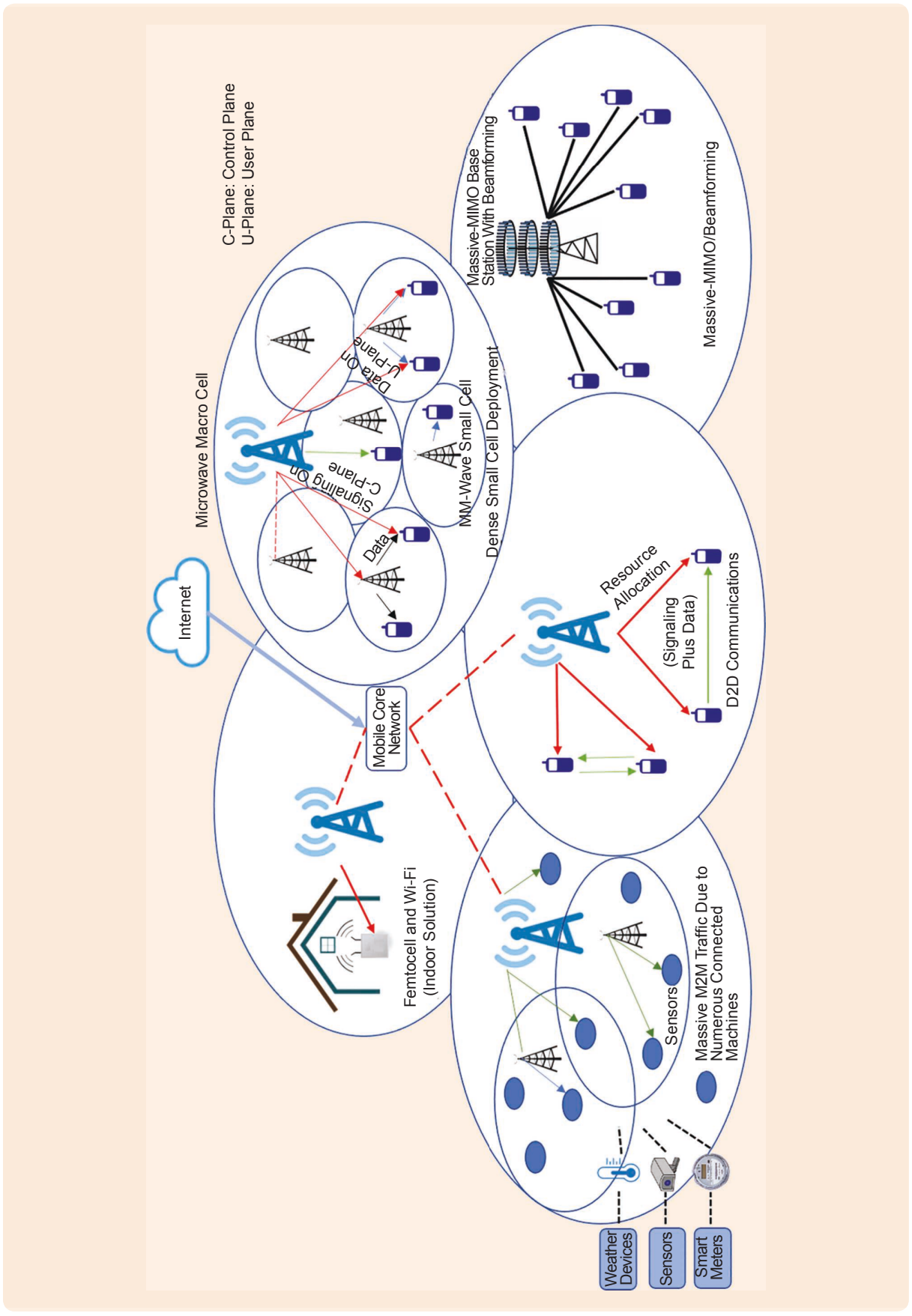


FIGURE 7 The next generation of communication networks: macrocells (bands <3 GHz); small cells [millimeter-wave (mm-wave)]; femtocells and Wi-Fi (mm-wave); massive multiple-input, multiple-output (MIMO) with beamforming; and device-to-device (D2D) and machine-to-machine (M2M) communications. The solid arrows indicate wireless links, whereas the dashed arrows indicate backhaul links.

transmission delay between these two nodes. Here, the expectation $\mathbb{E}_s[\cdot]$ is taken over all the possible system states.

Due to the lack of complete knowledge of adversaries and payoff structures, Boltzmann–Gibbs reinforcement learning (SBR-d) is utilized to find the optimal path because of its capability of estimating the expected utility. The resulting secure routing algorithm can spatially circumvent jammers along the routing path and learn to defend against malicious attackers as the state changes. As shown in Figure 8, the routing path generated from the proposed routing algorithm in [120] and [136] can avoid the nodes that are compromised by the jammers. Thus, the routing algorithm stemmed from the proposed game-theoretic formulation provides more resilience, security, and agility than the ad hoc on-demand distance vector (AODV) algorithm as AODV fails to dynamically adjust the routing path in the case of a malicious attack. Moreover, the proposed routing algorithm can reduce the delay time incurred by the attack due to its adaptive and dynamical feature (and thus is more efficient than the AODV).

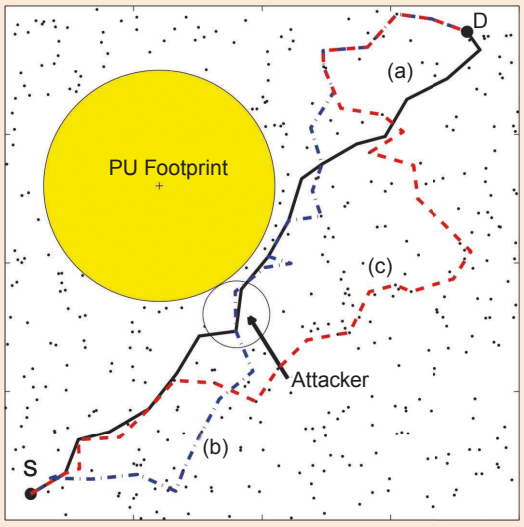


FIGURE 8 An illustration of a random network topology for 500 secondary users with a source (S) and a destination (D), and routes generated by an ad hoc on-demand distance vector (AODV) algorithm and the proposed secure routing algorithm in a 2-km × 2-km area. The primary user (PU) footprint denotes the set of nodes not available to secondary users. Without an attacker, the AODV establishes (a) the route path described by the solid line, while (b) the route path (the blue dashed line) is generated by the Boltzmann–Gibbs learning method. Even though the AODV path is the shortest path between the source and the destination, it is disrupted by the presence of malicious attacks. In contrast, the learning method can develop (c) a new route path that circumvents jammers, leading to a resilient routing mechanism.

The Smart Grid

The gradual replacement of conventional energies with renewable energies greatly helps with the reduction of greenhouse gases and mitigation of climate change. Currently, more microgrids are being integrated with the main power grid, which are green systems that rely on renewable distributed resources such as wind turbines and fuel cells. As displayed in Figure 9, the integration of microgrids can enhance the stability, resiliency, and reliability of the power system as they can operate independently from the main power grid in an autonomous manner. Such integration (together with smart meters and appliances) leads to the smart grid, a modern infrastructure for reliable delivery of electricity.

The future smart grid is envisioned as a large-scale cyberphysical system comprising advanced power, communications, control, and computing technologies. To accommodate these technologies employed by different parties in the grid (and to ensure an efficient and robust operation of such heterogeneous and large-scale cyberphysical systems), game-theoretic methods have been widely employed in smart grid

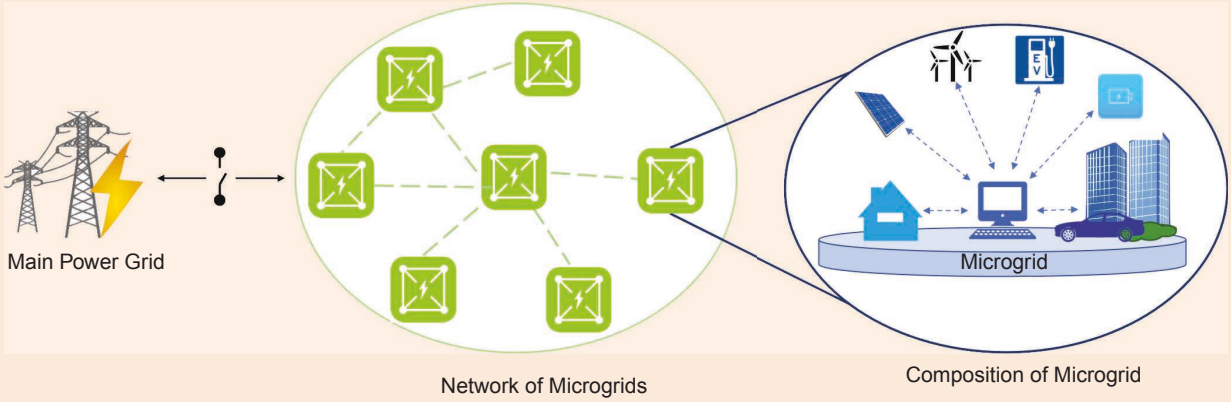


FIGURE 9 The integration of microgrids. A microgrid consists of a controller, consumers, generators, and energy storage. In the grid, microgrids can either be connected to the main grid or other microgrids, and these networked microgrids can operate, communicate, and interact autonomously to efficiently deliver power and electricity to their consumers.

management problems. Microgrids are modeled as self-interested players that can operate, communicate, and interact autonomously to efficiently deliver power and electricity to their consumers. Here, a microgrid management mechanism developed in [124] is presented. Such a mechanism is built on game-theoretic learning and enables autonomous management of renewable resources.

The system model considered in [124] includes generators, microgrids, and communications. As illustrated in Figure 10, generators in the upper layer determine their amount of power to be generated and the electricity price, then send them to the bottom layer. A microgrid can generate renewable energies and make decisions by responding to the strategies of the generators and other microgrids to optimize their payoffs, which is specified in the following game-theoretic model.

Let $\mathcal{N}_d = \{r, 1, 2, \ldots, N_d\}$ be the set of $N_d + 1$ buses in a power grid, where r denotes the slack bus. Assume that a smart grid is composed of load and generator buses, and let p_i^g, p_i^l, and θ_i be the power generation, power load, and voltage angle, respectively, at the ith bus. Note that the active power injection at the ith bus satisfies

$$p_i = p_i^g - p_i^l, \quad \forall\, i \in \mathcal{N}_d,$$

while the balance of the grid gives $\Sigma_{i\in\mathcal{N}_d} p_i^g = \Sigma_{i\in\mathcal{N}_d} p_i^l$. Let $\mathcal{N} := \{1, 2, \ldots, N\} \subseteq \mathcal{N}_d$ be the set of N buses that can generate renewable energies, such as wind and solar power.

In the game considered in [124], the utility function of the ith bus not only measures economic factors related to power generation but also the efficiency of the microgrids. Before giving the mathematical definition of the utility function, we first introduce the following notations. Let c_i be the unit cost of generated power for the ith player, and c the unit price of renewable energy for sale defined by the power market. c_i, c are quantities relevant to the profit gained by the bus. For efficiency, denote by r_i a weighting parameter that is a measurement of the importance of regulations of voltage angle at the ith bus. Further, $[s_{ij}]_{i,j\in\mathcal{N}_d} = -[b_{ij}]^{-1}_{i,j\in\mathcal{N}_d}$, where b_{ij} is the imaginary part of the element (i, j) in the admittance matrix of the power grid. Moreover, each microgrid has a maximum generation, which is denoted by $\bar{p}_i^g$. Finally, note that as a physical constraint, $[s_{ij}]$ and $[p_i]$ satisfy (23) due to the power flow equation [124]

$$\sum_{j\in\mathcal{N}_d\setminus\mathcal{N}} s_{ij}p_j + \sum_{j\neq i\in\mathcal{N}} s_{ij}p_j = \theta_i - s_{ii}p_i, \forall i \in \mathcal{N}, \tag{23}$$

where θ_i is the voltage angle of the ith bus. Using the aforementioned notations, the utility function of the ith bus is defined as

$$u_i(p_i^g, p_{-i}^g) := -c_i p_i^g - c(p_i^l - p_i^g) - \frac{1}{2} r_i^2 \Big(\sum_{j\in\mathcal{N}_d} s_{ij}p_j \Big),$$
$$0 \le p_i^g \le \bar{p}_i^g, \quad i \in \mathcal{N}.$$

To seek the Nash equilibrium, three learning methods are proposed, all of which are based on (BR-d). The first two algorithms are the parallel-update algorithm (PUA) and random-update algorithm (RUA) studied in [119]. PUA is essentially (BR-d), with the learning rate λ_i^k being zero for all i, and all players updating their strategies in parallel. As its name suggests, RUA incorporates randomness into (BR-d), resulting in an ϵ-greedy algorithm: players update their strategies according to (10) with probability $1-\epsilon$, with $\epsilon \in (0, 1)$ and retain their previous strategies otherwise. Players always update their strategies in every round when $\epsilon = 0$: in this case, RUA reduces to PUA.

However, [as special cases of (10)] PUA and RUA require global information regarding the grid, including the specific generated power of generators as well as other players› active power injections (which are assumed to be private in practice). Hence, implementing these algorithms requires communication networks to broadcast information to players, which is costly and not confidential. In this case, incorporating utility estimation is a possible remedy, and (SBR-d) can be applied as in the wireless setting introduced in the previous section. Another simpler approach, shown in [124], is to modify (BR-d) using power flow equations in the smart grid. Based on a phasor measurement unit (PMU), the third algorithm [termed a *PMU-enabled*

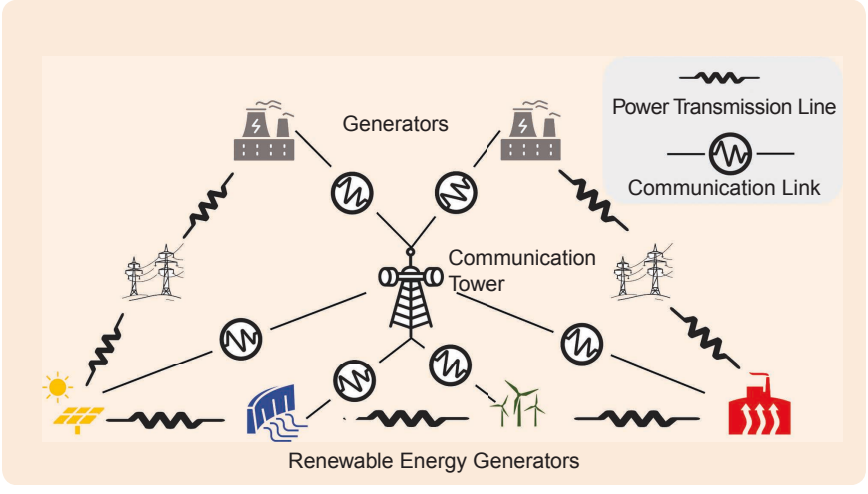


FIGURE 10 The smart grid hierarchy model. The upper layer containing conventional generators forms a generator network, and the distributed renewable energy generators in the bottom layer constitute the microgrid network. The information exchange (such as the electricity market price and amount of power generation) between two layers is through the communication network layer in the middle.

distributed algorithm (*PDA*)] enables each player to compute the aggregation of others' actions, with the player's voltage angle θ_i being the only information needed. Therefore, by considering the power flow equation (23), a player does not need other players' private information of active power injection when using a PDA (as shown in Figure 11). Compared with the other two, a PDA requires much less information and is more self-dependent as players need only their real-time voltage angles θ_i and common knowledge of the electricity price.

As indicated in [124], effectiveness and resiliency of the algorithm have been validated via case studies based on the IEEE 14-bus system: the game-theory-based distributed algorithm not only converges to the unique Nash equilibrium but also provides strong resilience against fault models (generator breakdown, microgrid turn off, and open circuit of the transmission line) and attack models (data-injection attacks, unavailability of PMU data, and jamming attacks). Strong resilience enables the microgrids to operate properly in unanticipated situations. Moreover, the distributed algorithm enables autonomous management of renewable resources and plug-and-play feature of the smart grid. The proposed learning algorithm only requires the players to have common knowledge without revealing their private information, which increases security and privacy and reduces communication overhead.

Distributed ML Over Networks

The rise of big data has led to new demands for large-scale ML systems that promise adequate capacity to digest massive data sets and offer powerful predictive analytics. With the unrestrainable growth of data, large-scale ML must address new challenges regarding the scalability and efficiency of learning algorithms with respect to computational and memory resources. Compared with classical ML approaches that are designed to learn from a single integrated data set, one of the promising research areas of large-scale ML is distributed ML over networks (DMLONs), which aims to develop efficient and scalable algorithms with reasonable requirements of memory computation resources by allocating the learning processes among several networked computing units with distributed data sets.

The key feature of DMLONs is that data sets are stored and processed locally on these computing units, which enables distributed and parallel computing schemes in large-scale ML systems. Compared with centralized approaches, distributed ML not only avoids maintaining and mining a central data set but also preserves data privacy as these networked units exchange knowledge about learned models without the exchange of raw, private data.

Based on the idea of "local learning and global integration," DMLONs utilizes different learning processes to train several models from distributed data sets, then produces an integration of learned models that increases the possibility of achieving higher accuracy (especially on a large-size domain). For example, in federated learning [137], global integration is created by a third-party coordinator other than computing units, which makes networked computing units collaboratively train an ML model using their data in security. On the other hand (as indicated in [138]), such a global integration can also stem from the collective patterns of local learning without external

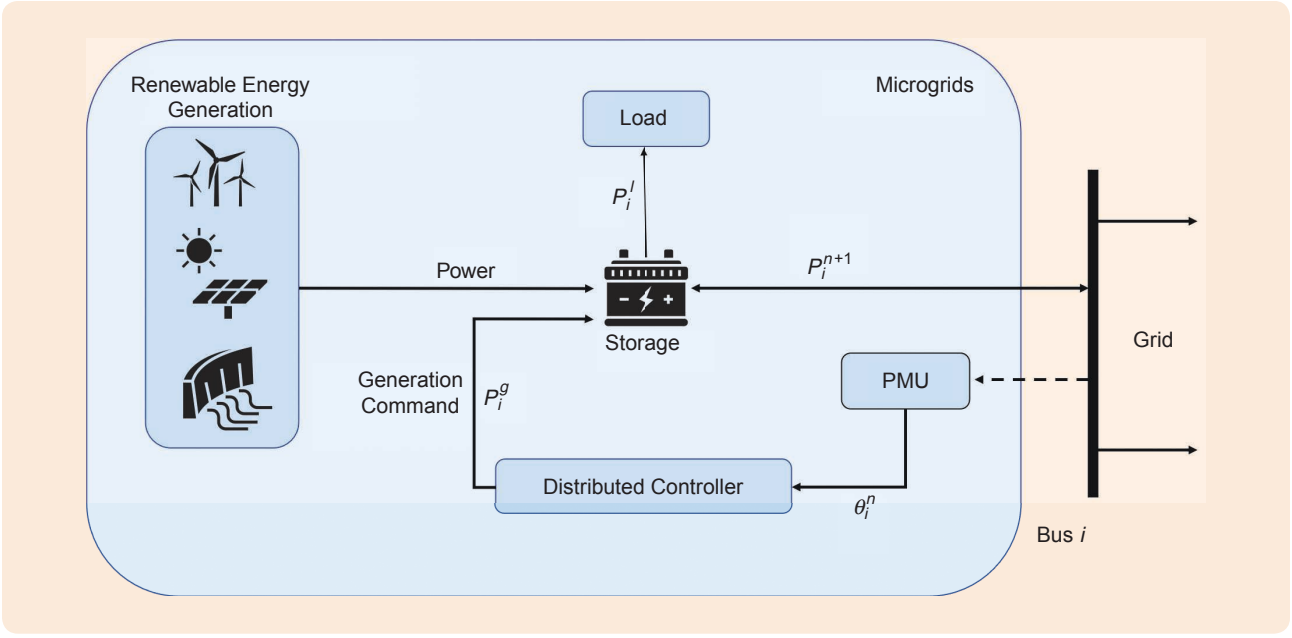


FIGURE 11 The framework used to implement the phasor-measurement-unit (PMU)-enabled distributed algorithm. A PMU measures the voltage angle at the bus, and the controller generates a command regarding the amount of microgrid-renewable energy injected from the local storage to the grid based on the received voltage angle.

enforcement. The key behind this bottom-up integration is that each computing unit is modeled as a self-interested player that learns the learning model based on the local data set as well as the feedback from its neighbors. It is shown in [138] that by modeling DMLONs as a noncooperative game, game-theoretic learning methods lead to communication-efficient, distributed ML where the global outcome is characterized by the Nash equilibrium resulting from players' self-adaptive behaviors.

Specifically, the networked system of computing units is described by a graph with the set of nodes $\mathcal{N}_m := \{1, 2, \ldots, N\}$ representing these units. Each node $i \in \mathcal{N}_m$ possesses local data that cannot be transferred to other nodes. In the game model considered in [138], instead of fixing the network topology, nodes can determine connectivity of the network based on their attributes when they perform learning tasks (which results in a network-formation game). In mathematical terms, the action of node i consists of two components: the learning parameter $\theta_i \in \mathbb{R}^d$ and network-formation parameter $e_i \in \mathbb{R}^{N-1}$. The first component θ_i corresponds to weights or parameters of the ML model that capture the local learning process at node i. The corresponding empirical loss (given the local data) is denoted by $L_i(\theta_i)$. In addition to this learning parameter θ_i, the network-formation parameter e_i plays an important role in the global integration. The parameter $e_i := (e_i^j)_{j \neq i, j \in \mathcal{N}} \in [0, 1]^{N-1}$ denotes the concatenation of weights on the directed edges from node i to other nodes, where e_i^j can be interpreted as the attention node i pays to the local learning at node j (which further influences communication among nodes). During the distributed learning process, each node can choose to communicate with its neighbors to exchange learning parameters if their objectives are aligned. Otherwise, the corresponding edge weight e_i^j is set to zero. For node i, the communication cost is $C_i(\theta_i, \theta_{-i}, e_i)$. In the game considered in [138], each node aims to maximize its utility function, defined as

$$u_i(\theta_i, \theta_{-i}, e_i, e_{-i}) := -L_i(\theta_i) - C_i(\theta_i, \theta_{-i}, e_i).$$

In this definition, the first term $L_i(\theta_i)$ captures the local learning process at node i, whereas the second term $C_i(\theta_i, \theta_{-i}, e_i)$ depicts interactions among nodes. The objective of each node is to improve the performance of learning while reducing communication overhead.

A two-layer learning approach is proposed in [138] to find the Nash equilibrium of the game, and a schematic representation is provided in Figure 12. The outer layer corresponds to network-formation learning, where each node decides its network-formation parameter e_i with the learning parameter fixed, and the joint parameters of all nodes $e = (e_i)_{i \in \mathcal{N}_m}$ give rise to a new network topology (leading to efficient communication). In network-formation learning, each node decides its optimal parameter e_i using gradient play (18). Computing the individual payoff gradient $\nabla_{e_i} u_i(\theta_i, \theta_{-i}, e_i, e_{-i})$ relies on the stabilized learning

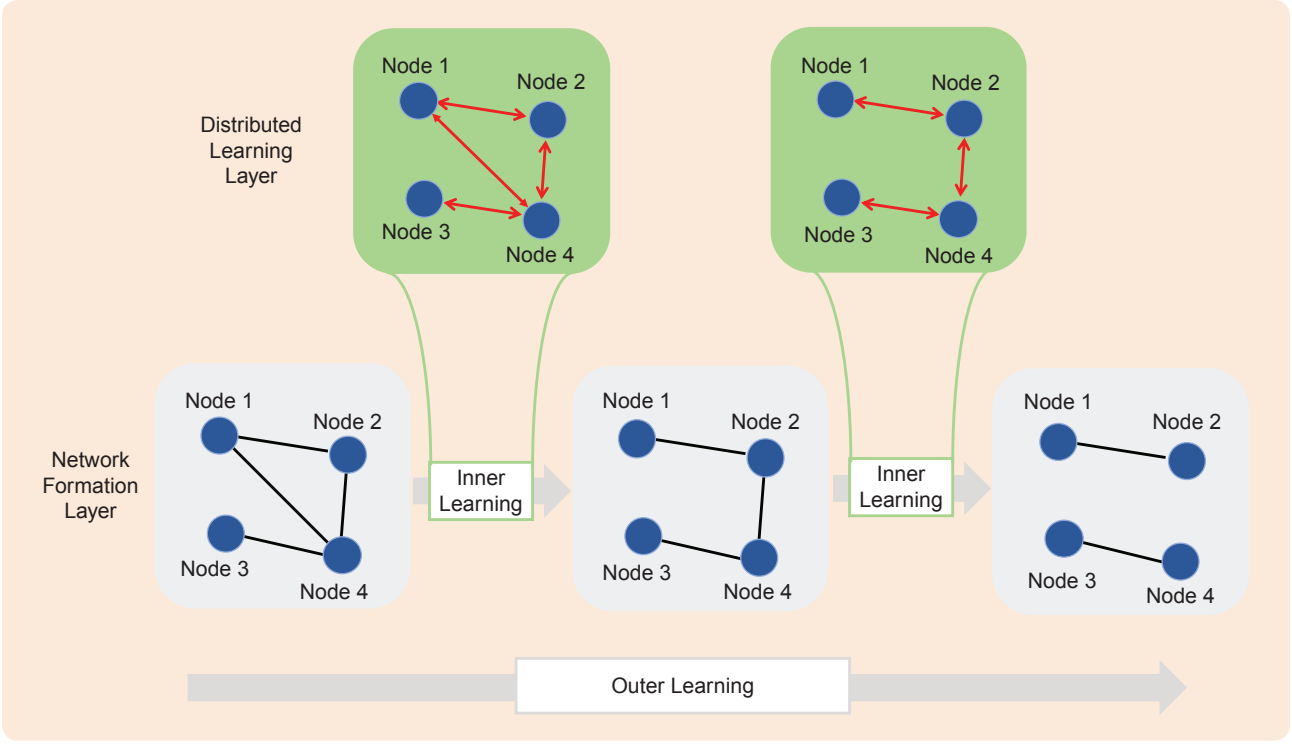


FIGURE 12 A schematic representation of two-layer learning. The directed red lines represent communication between nodes. In the network-formation layer, the nodes learn to eliminate/establish links with other nodes to achieve efficient communication. In the distributed ML layer, the nodes communicate their parameters with their neighbors and perform their own learning tasks.

parameters θ_i, θ_{-i} given by the inner layer: the distributed learning layer. In this inner learning, the network-formation parameter is fixed, and each node implements online MD for seeking the Nash equilibrium with the local feedback under the current network topology (as the networked nodes can exchange information with their neighbors).

Compared with existing works on distributed ML, the game-theoretic method studied in [138] enables distributed ML over strategic networks. On the one hand, the global outcome characterized by the Nash equilibrium is self-enforcing, resulting from the coordinated behaviors of independent computing units. This bottom-up approach, compared with the external enforcing one in federated learning, scales efficiently when additional computing units are introduced into the system. On the other hand, strategic interactions over the network (described by the network-formation decision of each node) create a network intelligence that allows each computing unit to adaptively adjust the underlying topology, resulting in a desired distributed learning pattern that minimizes communication costs during the learning process.

Emerging Network Applications

The aforementioned examples demonstrate that game-theoretic learning provides a natural, scalable design framework that creates network intelligence for autonomous control, management, and coordination of large-scale complex network systems with heterogeneous parties. The following offers thoughts regarding various applications of game-theoretic learning in a broader context, showing that such a design framework is pervasive for diverse network problems.

Interdependent infrastructure networks (including wireless communication networks and smart grids) play a significant role in modern society, where IoT devices are massively deployed and interconnected. These devices are connected with each other and to cellular/cloud networks, creating multilayer networks (referred to as *networks of networks* [139]). The smart grid is one prominent example, where wireless sensors collect the data of buses and power transmission lines, forming a sensor network built on power networks for grid monitoring and decision-planning purposes [140]. The networks-of-networks model has also been extensively studied in other infrastructure networks. For instance, in an intelligent transportation network, apart from vehicle-to-vehicle (V2V) communications, vehicles can also communicate with roadside infrastructures or units (which belong to one or several service providers) to exchange various types of data related to different applications, such as GPS navigation. In this case, the vehicles form one network while the infrastructure nodes form another. The interconnections between two networks lead to intelligent management and operation of modern transportation networks.

Due to heterogeneous and multitier features of interdependent networks, the required management mechanisms or controls vary for different networks. For example, the connectivity of sensor networks in smart grids or V2V communication networks requires higher security levels than infrastructure networks because cyberspace is more likely to be targeted by adversaries [122]. Therefore, to manage and secure interdependent infrastructure networks, game-theoretic learning methods (especially heterogeneous learning [46], [53]) can be used to design decentralized and resilient mechanisms that are responsive to attacks and adaptive to the dynamic environment (as different parties in interdependent infrastructure networks may acquire different information). For further information on this topic, refer to [53], [139]. and references therein.

Similar to distributed optimization and ML based on game-theoretic learning, the control of autonomous mobile robots can also be cast as a Nash-equilibrium-seeking problem over networks, where the equilibrium is viewed as the desired coordination of all robots [101], [102]. For applications of this kind (where the nature of robot movements determines the network topologies), dynamic games over networks are considered, and corresponding learning algorithms are employed. Based on their observations of the surroundings, robots rely on game-theoretic learning (such as reinforcement learning) to develop self-rule policies, leading to a decentralized operation of multiagent robotic systems. Moreover, when combined with powerful function approximators such as deep neural networks, reinforcement learning has proven to be effective for real-world, multiagent robotic controls. This area of research, termed *deep multiagent reinforcement learning* [87], [141], is growing rapidly and attracting the attention of researchers from ML, robotics, and control communities.

In addition to these prescriptive mechanisms in engineering practices, game-theoretic learning also provides a descriptive model for studying human decision making and strategic interactions in epidemiology and social sciences, where the Nash equilibrium represents a stable state of the underlying noncooperative game. For example, a differential game model is proposed in [142] to study viruses or diseases spreading over the network, and authors develops a decentralized mitigation mechanism for controlling the spread. Such an approach is further explored in [143], where an optimal quarantining strategy is proposed to suppress two interdependent epidemics spreading over complex networks. Furthermore, such a strategy is shown to be robust against random changes in network connections [143].

CONCLUSION

This article provided a comprehensive overview of game theory basics and related learning theories, which serve as building blocks for a systematic treatment of multiagent decision making over networks. We elaborated on game-theoretic learning methods for network applications drawn from spanning emerging areas such as next-generation wireless networks, smart grids, and networked ML. In each

area, we identified the main technical challenges and discussed how game theory can be applied to address them using a bottom-up approach.

From the surveyed works, it was demonstrated that noncooperative game theory is one of the cornerstones of decentralized mechanisms for large-scale complex networks with heterogeneous entities, where each node is modeled as an independent decision maker. The resulting collective behaviors of these rational decision makers over the network can be mathematically depicted by the solution concept: the Nash equilibrium. In addition to various game models, learning in games is of great significance for creating distributed network intelligence, which enables each entity in the network to respond to unanticipated situations (such as malicious attacks from adversaries in cyber-physical systems [140]). Under local or individual feedback, the introduced learning dynamics leads to a decentralized and self-adaptive procedure, yielding desired collective behavior patterns without any external enforcement.

Beyond the existing successes of game-theoretic learning (which mainly focus on learning in static repeated games), it is also of interest to investigate dynamic game models and associated learning dynamics to better understand the decision-making process in dynamic environments. The motivation for studying dynamic models and related learning theory stems, on the one hand, from the pervasive presence of time-varying network structures such as generation and demand in the smart grid [124]. On the other hand, by defining auxiliary state variables, the problem of decision making under uncertainties can be modeled as a dynamic game, where the state of the game includes the hidden information players do not have access to when making decisions. For example, the state variable can capture uncertainty of the environment (as discussed in the context of the dynamic routing problem [120]) or global status of the entire system (as shown in the example of distributed optimization [144]). The dynamic game models not only simplify construction of players' utilities and actions (providing a clear picture of the strategic interactions under uncertainties in the dynamic environment) but also offer a scalable design framework for prescribing players' self-adaptive behaviors, which leads to equilibrium states under various feedback structures.

This article presents a comprehensive overview of game-theoretic learning and its potential for tackling challenges emerging from network applications. The combination of game-theoretic modeling and related learning theories constitutes a powerful tool for designing future data-driven network systems with distributed intelligent entities, which serve as the bedrock and a key enabler for resilient and agile control of large-scale artificial intelligence systems in the near future.

ACKNOWLEDGMENT

This work is partially supported by grants SES-1541164, ECCS-1847056, CNS-2027884, and BCS-2122060 from the National Science Foundation, grant 20-19829 from DOE-NE, and grant W911NF-19-1-0041 from the Army Research Office.

AUTHOR INFORMATION

Tao Li (tl2636@nyu.edu) received the B.S. degree in mathematics from Xiamen University, Xiamen, China, in 2018. After a short visit to the University of Alberta in Canada, he is pursuing the Ph.D. degree in electrical engineering at New York University, New York, 11201, USA. His research focuses on game theory, multiagent decision making, online optimization, and learning theory.

Guanze Peng received the B.Eng. degree in electrical engineering from Fudan University, Shanghai, China. He is currently working toward the Ph.D. degree at New York University, New York, 11201, USA. His research interests include optimal control, sequential decision theory, and game theory.

Quanyan Zhu received the B. Eng. degree in electrical engineering (with honors) from McGill University in 2006, the M. A. Sc. degree from the University of Toronto in 2008, and the Ph.D. from the University of Illinois at Urbana-Champaign in 2013. After stints at Princeton University, he is currently an associate professor in the Department of Electrical and Computer Engineering, New York University, New York, 11201, USA (NYU). He is an affiliated faculty member of the Center for Urban Science and Progress and Center for Cyber Security at NYU. He spearheaded and chaired the Midwest Workshop on Control and Game Theory, the IEEE International Conference on Robotics and Automation workshop on Security and Privacy of Robotics, and IEEE Control Systems Society Technical Committee on Security and Privacy. He is a coauthor of four recent books: *Cyber-Security in Critical Infrastructures: A Game-Theoretic Approach* (with S. Rass, S. Schauer, and S. König; Springer 2020), *A Game- and Decision-Theoretic Approach to Resilient Interdependent Network Analysis and Design* (with J. Chen; Springer), *Cross-Layer Design for Secure and Resilient Cyber-Physical Systems: A Decision and Game Theoretic Approach* (with Z. Xu; Springer), and *Game Theory for Cyber Deception* (with J. Pawlick; Birkhäuser).

Tamer Başar received the B.S.E.E. degree from Robert College, Istanbul, and the M.S., M.Phil., and Ph.D. degrees in engineering and applied science from Yale University, from which he received the Wilbur Cross Medal in 2021. He is currently with the University of Illinois Urbana-Champaign, Illinois, 61801, USA, as the Swanlund Endowed Chair Emeritus, CAS Professor Emeritus of Electrical and Computer Engineering, and research professor with the Coordinated Science Laboratory and the Information Trust Institute. He has authored/coauthored nearly 1000 publications in systems, control, communications, optimization, networks, and dynamic games, including books on noncooperative dynamic game theory, robust control, network security, wireless and communication networks, and stochastic networked control. His current research interests

include stochastic teams, games, and networks; multiagent systems and learning; data-driven distributed optimization; epidemics modeling and control over networks; security and trust; energy systems; and cyberphysical systems. He is a member of the U.S. National Academy of Engineering. He has served as president of the IEEE Control Systems Society (CSS), International Society of Dynamic Games (ISDG), and American Automatic Control Council (AACC). He was a recipient of several awards and recognitions over the years, including the IEEE Control Systems Award, the highest awards of the CSS, International Federation of Automatic Control, AACC, and ISDG, and a number of international honorary doctorates and professorships. He was the editor-in-chief of *Automatica* from 2004 to 2014 and is currently the editor of several book series. He is a Fellow of IEEE, IFAC, and the Society for Industrial and Applied Mathematics.

REFERENCES

[1] M. O. Jackson, *Social and Economic Networks*. Princeton, NJ, USA: Princeton Univ. Press, 2010.

[2] D. Easley and J. Kleinberg, *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge, U.K.: Cambridge Univ. Press, 2010.

[3] S. Maharjan, Q. Zhu, Y. Zhang, S. Gjessing, and T. Başar, "Dependable demand response management in the smart grid: A Stackelberg game approach," *IEEE Trans. Smart Grid*, vol. 4, no. 1, pp. 120–132, 2013, doi: 10.1109/TSG.2012.2223766.

[4] Q. Zhu, Z. Han, and T. Başar, "A differential game approach to distributed demand side management in smart grid," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2012, pp. 3345–3350, doi: 10.1109/ICC.2012.6364562.

[5] Z. Han, D. Niyato, W. Saad, T. Başar, and A. Hjørungnes, *Game Theory in Wireless and Communication Networks: Theory, Models, and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2012.

[6] Q. Zhu, Z. Yuan, J. B. Song, Z. Han, and T. Başar, "Interference aware routing game for cognitive radio multi-hop networks," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 10, pp. 2006–2015, 2012, doi: 10.1109/JSAC.2012.121115.

[7] Z. Han, D. Niyato, W. Saad, and T. Başar, *Game Theory for Next Generation Wireless and Communication Networks: Modeling, Analysis, and Design*. Cambridge, U.K.: Cambridge Univ. Press, 2019.

[8] M. H. Manshaei, Q. Zhu, T. Alpcan, T. Başar, and J.-P. Hubaux, "Game theory meets network security and privacy," *ACM Comput. Surveys (CSUR)*, vol. 45, no. 3, pp. 1–39, 2013, doi: 10.1145/2480741.2480742.

[9] Q. Zhu and T. Başar, "Game-theoretic methods for robustness, security, and resilience of cyberphysical control systems: Games-in-games principle for optimal cross-layer resilient control systems," *IEEE Control Syst. Mag.*, vol. 35, no. 1, pp. 46–65, 2015, doi: 10.1109/MCS.2014.2364710.

[10] J. Pawlick and Q. Zhu, *Game Theory for Cyber Deception: From Theory to Applications*. Cham, Switzerland: Springer International Publishing, 2021.

[11] S. Rass, S. Schauer, S. König, and Q. Zhu, *Cyber-Security in Critical Infrastructures*. Berlin, Germany: Springer-Verlag, 2020.

[12] V. M. Bier, L. A. Cox, and M. N. Azaiez, "Why both game theory and reliability theory are important in defending infrastructure against intelligent attacks," in *Game Theoretic Risk Analysis of Security Threats*. Berlin, Germany: Springer-Verlag, 2009, pp. 1–11.

[13] L. Huang, J. Chen, and Q. Zhu, "Factored Markov game theory for secure interdependent infrastructure networks," in *Game Theory for Security and Risk Management*. Berlin, Germany: Springer-Verlag, 2018, pp. 99–126.

[14] L. Huang and Q. Zhu, "A dynamic games approach to proactive defense strategies against advanced persistent threats in cyber-physical systems," *Comput. Security*, vol. 89, p. 101,660, Feb. 2020, doi: 10.1016/j.cose.2019.101660.

[15] Q. Zhu and Z. Xu, *Cross-Layer Design for Secure and Resilient Cyber-Physical Systems: A Decision and Game Theoretic Approach*, vol. 81. Springer Nature, 2020.

[16] N. Groot, B. De Schutter, and H. Hellendoorn, "Toward system-optimal routing in traffic networks: A reverse Stackelberg game approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 29–40, 2014, doi: 10.1109/TITS.2014.2322312.

[17] T. Başar and G. J. Olsder, *Dynamic Noncooperative Game Theory*, 2nd ed. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1998.

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

[19] M. Maschler, E. Solan, and S. Zamir, *Game Theory*. Cambridge, U.K.: Cambridge Univ. Press, 2013.

[20] M. O. Jackson and Y. Zenou, "Chapter 3 – Games on networks," *Handbook Game Theory Econ. Appl.*, vol. 4, P. Young and S. Zamir Eds., Oxford, UK: Elsevier Science, 2015, pp. 95–163.

[21] S. Shalev-Shwartz, "Online learning and online convex optimization," *Found. Trends Mach. Learn.*, vol. 4, no. 2, pp. 107–194, 2011, doi: 10.1561/2200000018.

[22] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *Proc. 11th Int. Conf. Mach. Learn. (ICML'94)*, 1994, pp. 157–163.

[23] J. Lei and U. V. Shanbhag, "Stochastic Nash equilibrium problems: Models, analysis, and algorithms," *IEEE Control Syst. Mag.*, to be publihed.

[24] J. B. Rosen, "Existence and uniqueness of equilibrium points for concave N-person games," *Econometrica*, vol. 33, no. 3, pp. 520–534, 1965, doi: 10.2307/1911749.

[25] G. Belgioioso, P. Yi, S. Grammatico, and L. Pavel, "Distributed generalized Nash equilibrium seeking: An operator theoretic perspective," *IEEE Control Syst. Mag.*, to be publihed.

[26] E. Cavazzuti, M. Pappalardo, and M. Passacantando, "Nash equilibria, variational inequalities, and dynamical systems," *J. Optim. Theory Appl.*, vol. 114, no. 3, pp. 491–506, 2002, doi: 10.1023/A:1016056327692.

[27] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, 1st ed. Hoboken, NJ, USA: Wiley, 1994.

[28] R. Selten, "Reexamination of the perfectness concept for equilibrium points in extensive games," *Int. J. Game Theory*, vol. 4, no. 1, pp. 25–55, 1975, doi: 10.1007/BF01766400.

[29] T. Başar, "Time consistency and robustness of equilibria in non-cooperative dynamic games," in *Dynamic Policy Games in Economics* (Contributions to Economic Analysis, vol. 181). Amsterdam, The Netherlands: Elsevier, 1989, pp. 9–54.

[30] D. Fudenberg, *The Theory of Learning in Games*. Cambridge, MA, USA: MIT Press, 1998.

[31] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou, "The complexity of computing a Nash equilibrium," *SIAM J. Comput.*, vol. 39, no. 1, pp. 195–259, 2009, doi: 10.1137/070699652.

[32] P. D. Taylor and L. B. Jonker, "Evolutionary stable strategies and game dynamics," *Math. Biosci.*, vol. 40, nos. 1–2, pp. 145–156, 1978, doi: 10.1016/0025-5564(78)90077-9.

[33] S. Hart and A. Mas-Colell, "Uncoupled dynamics do not lead to Nash equilibrium," *Amer. Econ. Rev.*, vol. 93, no. 5, pp. 1830–1836, 2003, doi: 10.1257/000282803322655581.

[34] J. R. Marden and J. S. Shamma, "Chapter 16 – Game theory and distributed control," *Handbook Game Theory Econ. Appl.*, vol. 4, P. Young and S. Zamir, Eds., Oxford, UK: Elsevier Science, 2015, pp. 861–899.

[35] V. S. Borkar, *Stochastic Approximation, A Dynamical Systems Viewpoint*. Berlin, Germany: Springer-Verlag, 2008, vol. 48.

[36] M. Benaïm, J. Hofbauer, and S. Sorin, "Stochastic approximations and differential inclusions," *SIAM J. Control Optim.*, vol. 44, no. 1, pp. 328–348, 2005, doi: 10.1137/S0363012904439301.

[37] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[38] D. S. Leslie and E. J. Collins, "Convergent multiple-timescales reinforcement learning algorithms in normal form games," *Ann. Appl. Probability*, vol. 13, no. 4, pp. 1231–1251, Nov. 2003, doi: 10.1214/aoap/1069786497.

[39] C. Harris, "On the rate of convergence of continuous-time fictitious play," *Games Econ. Behav.*, vol. 22, no. 2, pp. 238–259, 1998, doi: 10.1006/game.1997.0582.

[40] J. Hofbauer and K. Sigmund, "Evolutionary game dynamics," *Bull. Amer. Math. Soc.*, vol. 40, no. 4, pp. 479–519, 2003, doi: 10.1090/S0273-0979-03-00988-1.

[41] G. W. Brown, "Iterative solution of games by fictitious play," *Activity Anal. Prod. Allocation*, vol. 13, no. 1, pp. 374–376, 1951.

[42] V. Krishna and T. Sjöström, "On the convergence of fictitious play," *Math. Oper. Res.*, vol. 23, no. 2, pp. 479–511, 1998, doi: 10.1287/moor.23.2.479.

[43] P. Mertikopoulos and Z. Zhou, "Learning in games with continuous action sets and unknown payoff functions," *Math. Program.*, vol. 173, nos. 1–2, pp. 465–507, 2018, doi: 10.1007/s10107-018-1254-8.

[44] P. Mertikopoulos and W. H. Sandholm, "Learning in games via reinforcement and regularization," *Math. Oper. Res.*, vol. 41, no. 4, pp. 1297–1324, 2016, doi: 10.1287/moor.2016.0778.

[45] R. D. McKelvey and T. R. Palfrey, "Quantal response equilibria for normal form games," *Games Econ. Behav.*, vol. 10, no. 1, pp. 6–38, 1995, doi: 10.1006/game.1995.1023.
[46] Q. Zhu, H. Tembine, and T. Başar, "Heterogeneous learning in zero-sum stochastic games with incomplete information," in *Proc. 49th IEEE Conf. Decis. Control (CDC)*, 2010, pp. 219–224, doi: 10.1109/CDC.2010.5718053.
[47] Y. Nesterov, "Primal-dual subgradient methods for convex problems," *Math. Program.*, vol. 120, no. 1, pp. 221–259, 2009, doi: 10.1007/s10107-007-0149-x.
[48] Y. Nesterov, *Introductory Lectures on Convex Optimization, a Basic Course* (Applied Optimization), New York, NY, USA: Springer, 2004.
[49] J. M. Smith and G. R. Price, "The logic of animal conflict," *Nature*, vol. 246, no. 5427, pp. 15–18, 1973, doi: 10.1038/246015a0.
[50] W. H. Sandholm, *Population Games and Evolutionary Dynamics*. Cambridge, MA, USA: MIT Press, 2010.
[51] R. Cressman and Y. Tao, "The replicator equation and other game dynamics," *Proc. Nat. Acad. Sci.*, vol. 111, no. 3, pp. 10,810–10,817, 2014, doi: 10.1073/pnas.1400823111.
[52] D. S. Leslie and E. Collins, "Generalised weakened fictitious play," *Games Econ. Behav.*, vol. 56, no. 2, pp. 285–298, 2006, doi: 10.1016/j.geb.2005.08.005.
[53] Q. Zhu, H. Tembine, and T. Başar, "Hybrid learning in stochastic games and its application in network security," in *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control*. Hoboken, NJ, USA: Wiley, 2012, pp. 303–329.
[54] G. Neu, A. Jonsson, and V. Gómez, "A unified view of entropy-regularized Markov decision processes," 2017, *arXiv:1705.07798*.
[55] V. R. Konda and V. S. Borkar, "Actor-critic–type learning algorithms for Markov decision processes," *SIAM J. Control Optim.*, vol. 38, no. 1, pp. 94–123, 1999, doi: 10.1137/S036301299731669X.
[56] D. S. Leslie and E. J. Collins, "Individual Q-learning in normal form games," *SIAM J. Control Optim.*, vol. 44, no. 2, pp. 495–514, 2005, doi: 10.1137/S0363012903437976.
[57] J. Hofbauer, S. Sorin, and Y. Viossat, "Time average replicator and best-reply dynamics," *Math. Oper. Res.*, vol. 34, no. 2, pp. 263–269, 2009, doi: 10.1287/moor.1080.0359.
[58] P. Mertikopoulos and W. H. Sandholm, "Riemannian game dynamics," *J. Econ. Theory*, vol. 177, pp. 315–364, Sep. 2018, doi: 10.1016/j.jet.2018.06.002.
[59] M. Benaïm, J. Hofbauer, and S. Sorin, "Stochastic approximations and differential inclusions, Part II: Applications," *Math. OR*, vol. 31, no. 4, pp. 673–695, Nov. 2006, doi: 10.1287/moor.1060.0213.
[60] A. Geramifard, T. J. Walsh, S. Tellex, G. Chowdhary, N. Roy, and J. P. How, "A tutorial on linear function approximators for dynamic programming and reinforcement learning," *Found. Trends Mach. Learn.*, vol. 6, no. 4, pp. 375–451, 2013, doi: 10.1561/2200000042.
[61] T. Li and Q. Zhu, "On convergence rate of adaptive multiscale value function approximation for reinforcement learning," in *Proc. IEEE 29th Int. Workshop on Mach. Learn. Signal Process. (MLSP)*, 2019, pp. 1–6, doi: 10.1109/MLSP.2019.8918816.
[62] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015, doi: 10.1038/nature14236.
[63] J. C. Spall, "A one-measurement form of simultaneous perturbation stochastic approximation," *Automatica*, vol. 33, no. 1, pp. 109–112, 1997, doi: 10.1016/S0005-1098(96)00149-5.
[64] M. Bravo, D. S. Leslie, and P. Mertikopoulos, "Bandit learning in concave N-person games," in *Proc. Adv. Neural Inf. Process. Syst.*, pp. 5666–5676, 2018.
[65] L. Xiao, "Dual averaging methods for regularized stochastic learning and online optimization," *J. Mach. Learn. Res.*, vol. 11, pp. 2543–2596, Dec. 2010.
[66] P. Mertikopoulos, C. Papadimitriou, and G. Piliouras, "Cycles in adversarial regularized learning," in *Proc. Annu. ACM-SIAM Symp. Discrete Algorithms*, 2018, pp. 2703–2717.
[67] E. Mazumdar, L. J. Ratliff, and S. S. Sastry, "On gradient-based learning in continuous games," *SIAM J. Math. Data Sci.*, vol. 2, no. 1, pp. 103–131, 2020, doi: 10.1137/18M1231298.
[68] J. Hofbauer and S. Sorin, "Best response dynamics for continuous zero-sum games," *Discrete Continuous Dyn. Syst. – B*, vol. 6, no. 1, pp. 215–224, 2006, doi: 10.3934/dcdsb.2006.6.215.
[69] J. Hofbauer and W. H. Sandholm, "On the global convergence of stochastic fictitious play," *Econometrica*, vol. 70, no. 6, pp. 2265–2294, 2002, doi: 10.1111/1468-0262.00376.
[70] S. Perkins and D. S. Leslie, "Asynchronous stochastic approximation with differential inclusions," *Stochastic Syst.*, vol. 2, no. 2, pp. 409–446, 2012, doi: 10.1287/11-SSY056.
[71] A. Heliou, J. Cohen, and P. Mertikopoulos, "Learning with bandit feedback in potential games," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6369–6378, vol. 30.
[72] B. Gao and L. Pavel, "On passivity, reinforcement learning, and higher order learning in multiagent finite games," *IEEE Trans. Autom. Control*, vol. 66, no. 1, pp. 121–136, 2019, doi: 10.1109/TAC.2020.2978037.
[73] E. N. Barron, R. Goebel, and R. R. Jensen, "Best response dynamics for continuous games," *Proc. Amer. Math. Soc.*, vol. 138, no. 3, pp. 1069–1069, 2010, doi: 10.1090/S0002-9939-09-10170-3.
[74] B. Swenson, R. Murray, and S. Kar, "On best-response dynamics in potential games," *SIAM J. Control Optim.*, vol. 56, no. 4, pp. 2734–2767, 2018, doi: 10.1137/17M1139461.
[75] B. Swenson, R. Murray, and S. Kar, "Regular potential games," *Games Econ. Behav.*, vol. 124, pp. 432–453, Nov. 2020, doi: 10.1016/j.geb.2020.09.005.
[76] M. Benaïm, J. Hofbauer, and S. Sorin, "Perturbations of set-valued dynamical systems, with applications to game theory," *Dyn. Games Appl.*, vol. 2, no. 2, pp. 195–205, 2012, doi: 10.1007/s13235-012-0040-0.
[77] J. C. Harsanyi, "Games with randomly disturbed payoffs: A new rationale for mixed-strategy equilibrium points," *Int. J. Game Theory*, vol. 2, no. 1, pp. 1–23, 1973, doi: 10.1007/BF01737554.
[78] J. Hofbauer and E. Hopkins, "Learning in perturbed asymmetric games," *Games Econ. Behav.*, vol. 52, no. 1, pp. 133–152, 2005, doi: 10.1016/j.geb.2004.06.006.
[79] H. P. Young, "The evolution of conventions," *Econometrica*, vol. 61, no. 1, pp. 57–84, 1993, doi: 10.2307/2951778.
[80] H. P. Young, "Learning by trial and error," *Games Econ. Behav.*, vol. 65, no. 2, pp. 626–643, 2009, doi: 10.1016/j.geb.2008.02.011.
[81] J. Gaveau, C. J. Le Martret, and M. Assaad, "Performance analysis of trial and error algorithms," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 6, pp. 1343–1356, 2020, doi: 10.1109/TPDS.2020.2964256.
[82] J. R. Marden, H. P. Young, G. Arslan, and J. S. Shamma, "Payoff-based dynamics for multiplayer weakly acyclic games," *SIAM J. Control Optim.*, vol. 48, no. 1, pp. 373–396, 2009, doi: 10.1137/070680199.
[83] B. S. Pradelski and H. P. Young, "Learning efficient Nash equilibria in distributed systems," *Games Econ. Behav.*, vol. 75, no. 2, pp. 882–897, 2012, doi: 10.1016/j.geb.2012.02.017.
[84] J. R. Marden, "Selecting efficient correlated equilibria through distributed learning," *Games Econ. Behav.*, vol. 106, pp. 114–133, Nov. 2017, doi: 10.1016/j.geb.2017.09.007.
[85] J. R. Marden, H. P. Young, and L. Y. Pao, "Achieving Pareto optimality through distributed learning," *SIAM J. Control Optim.*, vol. 52, no. 5, pp. 2753–2770, 2014, doi: 10.1137/110850694.
[86] Z. Hu, M. Zhu, P. Chen, and P. Liu, "On convergence rates of game theoretic reinforcement learning algorithms," *Automatica*, vol. 104, pp. 90–101, Jun. 2019, doi: 10.1016/j.automatica.2019.02.032.
[87] K. Zhang, Z. Yang, and T. Başar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," 2019, *arXiv:1911.10635*.
[88] Y. Zhou, M. Kantarcioglu, and B. Xi, "A survey of game theoretic approach for adversarial machine learning," *Wiley Interdisciplinary Rev., Data Mining Knowledge Discovery*, vol. 9, no. 3, p. e1259, 2019.
[89] K. Zhang, B. Hu, and T. Başar, "On the stability and convergence of robust adversarial reinforcement learning: A case study on linear quadratic systems," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 22,056–22,068.
[90] I. Goodfellow *et al.*, "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, 2020, doi: 10.1145/3422622.
[91] A. Wibisono, A. C. Wilson, and M. I. Jordan, "A variational perspective on accelerated methods in optimization," *Proc. Nat. Acad. Sci.*, vol. 113, no. 47, pp. E7351–E7358, 2016, doi: 10.1073/pnas.1614734113.
[92] E. V. Mazumdar, M. I. Jordan, and S. S. Sastry, "On finding local Nash equilibria (and only local Nash equilibria) in zero-sum games," 2019, *arXiv:1901.00838*.
[93] C. Jin, P. Netrapalli, R. Ge, S. M. Kakade, and M. I. Jordan, "On nonconvex optimization for machine learning: Gradients, stochasticity, and saddle points," 2019, *arXiv:1902.04811*.
[94] J. Diakonikolas, C. Daskalakis, and M. Jordan, "Efficient methods for structured nonconvex-nonconcave min-max optimization," in *Proc. 24th Int. Conf. Artif. Intell. Statist.*, Apr. 13–15, 2021, pp. 2746–2754.
[95] W. Su, S. Boyd, and E. J. Candès, "A differential equation for modeling Nesterov's accelerated gradient method: Theory and insights," *J. Mach. Learn. Res.*, vol. 17, no. 153, pp. 1–43, 2016.

[96] D. Gadjov and L. Pavel, "A passivity-based approach to Nash equilibrium seeking over networks," *IEEE Trans. Autom. Control*, vol. 64, no. 3, pp. 1077–1092, 2017, doi: 10.1109/TAC.2018.2833140.
[97] T. Başar, "Relaxation techniques and asynchronous algorithms for online computation of non-cooperative equilibria," *J. Econ. Dyn. Control*, vol. 11, no. 4, pp. 531–549, 1987, doi: 10.1016/S0165-1889(87)80006-4.
[98] G. Hu, Y. Pang, C. Sun, and Y. Hong, "Distributed Nash equilibrium seeking: Continuous-time control-theoretic approaches," *IEEE Control Syst. Mag.*, vol. 18, pp. 1075–1082, May 2020.
[99] P. Frihauf, M. Krstic, and T. Başar, "Nash equilibrium seeking in noncooperative games," *IEEE Trans. Autom. Control*, vol. 57, no. 5, pp. 1192–1207, 2012, doi: 10.1109/TAC.2011.2173412.
[100] P. Stone and M. Veloso, "Multiagent systems: A survey from a machine learning perspective," *Autonom. Robots*, vol. 8, no. 3, pp. 345–383, 2000, doi: 10.1023/A:1008942012299.
[101] G. A. Kaminka, D. Erusalimchik, and S. Kraus, "Adaptive multi-robot coordination: A game-theoretic perspective," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2010, pp. 328–334, doi: 10.1109/ROBOT.2010.5509316.
[102] W. Inujima, K. Nakano, and S. Hosokawa, "Multi-robot coordination using switching of methods for deriving equilibrium in game theory," in *Proc. 10th Int. Conf. Electr. Eng./Electron., Comput., Telecommun. Inf. Technol.*, 2013, pp. 1–6.
[103] W. He and Y. Sun, "Stationary Markov perfect equilibria in discounted stochastic games," *J. Econ. Theory*, vol. 169, pp. 35–61, May 2017, doi: 10.1016/j.jet.2017.01.007.
[104] M. O. Sayin, F. Parise, and A. Ozdaglar, "Fictitious play in zero-sum stochastic games," 2020, *arXiv:2010.04223*.
[105] R. Bellman, "The theory of dynamic programming," *Bull. Amer. Math. Soc.*, vol. 60, no. 6, pp. 503–515, 1954, doi: 10.1090/S0002-9904-1954-09848-8.
[106] J. Hu and M. P. Wellman, "Nash Q-learning for general-sum stochastic games," *J. Mach. Learn. Res.*, vol. 4, pp. 1039–1069, Nov. 2003.
[107] D. S. Leslie, S. Perkins, and Z. Xu, "Best-response dynamics in zero-sum stochastic games," *J. Econ. Theory*, vol. 189, p. 105,095, Sep. 2020, doi: 10.1016/j.jet.2020.105095.
[108] J. Bu, L. J. Ratliff, and M. Mesbahi, "Global convergence of policy gradient for sequential zero-sum linear quadratic dynamic games," 2019, *arXiv:1911.04672*.
[109] K. Zhang, X. Zhang, B. Hu, and T. Başar, "Derivative-free policy optimization for risk-sensitive and robust control design: Implicit regularization and sample complexity," 2021, *arXiv*.
[110] P. Dayan and C. J. Watkins, "Q-Learning," *Mach. Learn.*, vol. 8, no. 3/4, pp. 279–292, 1992, doi: 10.1023/A:1022676722315.
[111] P. Hernandez-Leal, M. Kaisers, T. Baarslag, and E. M. d Cote, "A survey of learning in multiagent environments: Dealing with non-stationarity," 2017, *arXiv:1707.09183*.
[112] H.-T. Wai, Z. Yang, Z. Wang, and M. Hong, "Multi-agent reinforcement learning via double averaging primal-dual optimization," 2018, *arXiv:1806.00877*.
[113] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Başar, "Fully decentralized multi-agent reinforcement learning with networked agents," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, vol. 80, pp. 5872–5881.
[114] I. A. Kash, M. Sullins, and K. Hofmann, "Combining no-regret and Q-learning," in *Proc. 19th Int. Conf. Autonom. Agents MultiAgent Syst. (AAMAS '20)*, 2020, pp. 593–601.
[115] T. Li, G. Peng, and Q. Zhu, "Blackwell online learning for Markov decision processes," 2020, *arXiv:2012.14043*.
[116] M. Zinkevich, M. Johanson, M. Bowling, and C. Piccione, "Regret minimization in games with incomplete information," in *Adv. Neural Inf. Process. Syst.*, 2008, vol. 20, pp. 1729–1736.
[117] K. Zhang, S. M. Kakade, T. Başar, and L. F. Yang, "Model-based multi-agent RL in zero-sum Markov games with near-optimal sample complexity," 2020, *arXiv:2007.07461*.
[118] V. Hakami and M. Dehghan, "Learning stationary correlated equilibria in constrained general-sum stochastic games," *IEEE Trans. Cybern.*, vol. 46, no. 7, pp. 1640–1654, 2016, doi: 10.1109/TCYB.2015.2453165.
[119] T. Alpcan, T. Başar, R. Srikant, and E. Altman, "CDMA uplink power control as a noncooperative game," *Wireless Netw.*, vol. 8, no. 6, pp. 659–670, 2002, doi: 10.1023/A:1020375225649.
[120] Q. Zhu, J. B. Song, and T. Başar, "Dynamic secure routing game in distributed cognitive radio networks," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM 2011)*, 2011, pp. 1–6.
[121] Q. Zhu, C. Fung, R. Boutaba, and T. Başar, "A game-theoretical approach to incentive design in collaborative intrusion detection networks," in *Proc. Int. Conf. Game Theory Netw.*, 2009, pp. 384–392.
[122] M. J. Farooq and Q. Zhu, "On the secure and reconfigurable multi-layer network design for critical information dissemination in the internet of battlefield things (IoBT)," *IEEE Trans. Wireless Commun.*, vol. 17, no. 4, pp. 2618–2632, 2018, doi: 10.1109/TWC.2018.2799860.
[123] M. J. Farooq and Q. Zhu, "Modeling, analysis, and mitigation of dynamic botnet formation in wireless IoT networks," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 9, pp. 2412–2426, 2019, doi: 10.1109/TIFS.2019.2898817.
[124] J. Chen and Q. Zhu, "A game-theoretic framework for resilient and distributed generation control of renewable energies in microgrids," *IEEE Trans. Smart Grid*, vol. 8, no. 1, pp. 285–295, 2016, doi: 10.1109/TSG.2016.2598771.
[125] S. Maharjan, Q. Zhu, Y. Zhang, S. Gjessing, and T. Başar, "Demand response management in the smart grid in a large population regime," *IEEE Trans. Smart Grid*, vol. 7, no. 1, pp. 189–199, 2015, doi: 10.1109/TSG.2015.2431324.
[126] J. Chen, C. Touati, and Q. Zhu, "A dynamic game approach to strategic design of secure and resilient infrastructure network," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 462–474, Jun. 2019, doi: 10.1109/TIFS.2019.2924130.
[127] L. Huang, J. Chen, and Q. Zhu, "A large-scale Markov game approach to dynamic protection of interdependent infrastructure networks," in *Proc. Int. Conf. Decis. Game Theory Security*, 2017, pp. 357–376, doi: 10.1007/978-3-319-68711-7_19.
[128] J. Chen and Q. Zhu, "Interdependent network formation games with an application to critical infrastructures," in *Proc. Amer. Control Conf. (ACC)*, 2016, pp. 2870–2875, doi: 10.1109/ACC.2016.7525354.
[129] J. Chen, C. Touati, and Q. Zhu, "Heterogeneous multi-layer adversarial network design for the IoT-enabled infrastructures," in *Proc. IEEE Global Commun. Conf. (GLOBECOM 2017)*, 2017, pp. 1–6.
[130] Z. Xu and Q. Zhu, "A game-theoretic approach to secure control of communication-based train control systems under jamming attacks," in *Proc. 1st Int. Workshop on Safe Control Connected Auton. Veh.*, 2017, pp. 27–34, doi: 10.1145/3055378.3055381.
[131] Q. Zhu, W. Saad, Z. Han, H. V. Poor, and T. Başar, "Eavesdropping and jamming in next-generation wireless networks: A game-theoretic approach," in *Military Commun. Conf. (2011-MILCOM 2011)*, 2011, pp. 119–124.
[132] Q. Zhu and T. Başar, "Game-theoretic approach to feedback-driven multi-stage moving target defense," in *Proc. Int. Conf. Decis. Game Theory Security*, 2013, pp. 246–263.
[133] Y. Huang, J. Chen, L. Huang, and Q. Zhu, "Dynamic games for secure and resilient control system design," *Nat. Sci. Rev.*, vol. 7, no. 7, pp. 1125–1141, 2020, doi: 10.1093/nsr/nwz218.
[134] Q. Zhu and S. Rass, "On multi-phase and multi-stage game-theoretic modeling of advanced persistent threats," *IEEE Access*, vol. 6, pp. 13,958–13,971, Mar. 2018, doi: 10.1109/ACCESS.2018.2814481.
[135] N. Al-Falahy and O. Y. Alani, "Technologies for 5G networks: Challenges and opportunities," *IT Professional*, vol. 19, no. 1, pp. 12–20, 2017, doi: 10.1109/MITP.2017.9.
[136] J. B. Song and Q. Zhu, "Performance of dynamic secure routing game," in *Game Theory for Networking Applications*, 2019, pp. 37–56.
[137] J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," 2016, *arXiv:1610.02527*.
[138] S. Liu, T. Li, and Q. Zhu, "Communication-efficient distributed machine learning over strategic networks: A two-layer game approach," 2020, *arXiv:2011.01455*.
[139] J. Chen and Q. Zhu, "A game- and decision-theoretic approach to resilient interdependent network analysis and design," *SpringerBriefs Elect. Comput. Eng.*, 2019, pp. 75–102.
[140] Q. Zhu, "Multilayer cyber-physical security and resilience for smart grid," in *Smart Grid Control*. Berlin, Germany: Springer-Verlag, 2019, pp. 225–239.
[141] J. Foerster, I. A. Assael, N. De Freitas, and S. Whiteson, "Learning to communicate with deep multi-agent reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, vol. 29, pp. 2137–2145.
[142] Y. Huang and Q. Zhu, "A differential game approach to decentralized virus-resistant weight adaptation policy over complex networks," *IEEE Trans. Control Netw. Syst.*, vol. 7, no. 2, pp. 944–955, 2020, doi: 10.1109/TCNS.2019.2931862.
[143] J. Chen, Y. Huang, R. Zhang, and Q. Zhu, "Optimal quarantining strategy for interdependent epidemics spreading over complex networks," 2020, *arXiv:2011.14262*.
[144] N. Li and J. R. Marden, "Designing games for distributed optimization," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 2, pp. 230–242, 2013, doi: 10.1109/JSTSP.2013.2246511.