Multiagent Reinforcement Learning for Nash Equilibrium Seeking in General-Sum Markov Games

Alireza Ramezani Moghaddam and Hamed Kebriaei, *Senior Member, IEEE*

***Abstract*—This article studies the problem of noncooperative multiagent reinforcement learning (MARL), where selfish agents play a general-sum Markov game. We consider the framework where no agent has explicit information on the model of dynamic environment, the model of other agents, and even on its own cost function. We propose an actor–critic MARL to learn the Nash equilibrium (NE) policy of the agents. The main contribution of this article is to extend the NE seeking methods to incomplete information stochastic nonzero sum games. Based on such formulation and under some conventional assumptions, we prove that by applying linear function approximators, the policies of agents converge to an approximation of the first-order NE point of the game. Finally, as a case study, the framework is applied to a Cloud Radio Access Network.**

***Index Terms*—Actor–critic, Markov game, reinforcement learning, stochastic policy gradient (SPG).**

I. Introduction

MARKOV Games represent one of the most important classes of multiagent decision problems over stochastic dynamic environment. Their distinctive feature is that the cost function of each individual agent is affected by other agents' actions. This structure arises in several application domains, such as resource allocation [1], social networks [2], electrical microgrids, and power systems [3].

The literature on Nash-seeking algorithms for noncooperative games is quite recent, e.g., best-response dynamics are studied in [4], policy iteration in [5], and subgradient dynamics in [6] (on myopic agents and deterministic environment). In most of the research on noncooperative games, the game is played in a static environment and as a result, the cost functions of the agents are only influenced by the actions of the neighbors. Nonetheless, in many applications, the agents interact in a dynamic environment, where the state transitions are typically modeled via a dynamical system whose inputs are the actions of the agents and the state of the system affects the cost functions of the agents [7].

The major objective of this article is to extend these results to cases where the agents interact in an unknown dynamic environment without having an explicit model of their cost functions and environment dynamics. There are some works studied the Nash equilibrium (NE) seeking problem in partially unknown dynamic games with known cost functions [8], [9]. Nevertheless, the problem becomes more challenging when the agents minimize their long-term discounted cost and also do not have an explicit model of their own cost functions.

One popular family of methods that is able to deal with unknown environments (including transition dynamic and cost function) in both cooperative [10] and noncooperative [11] settings is that of multiagent reinforcement learning (MARL). One of the major points that distinguishes our work from previous studies in MARL, is to present convergence proof in noncooperative games which is in sharp contrast to existing cooperative MARL studies that provide convergence proof to a global value function [12]. In noncooperative applications, each agent has his own objective, which in some cases is even in conflict with the goals of the others, and therefore, the problem shifts to the proof of convergence to the NE policy of the dynamic games with N different value functions.

The problem of developing MARL algorithms for noncooperative settings has gained increased attention in recent years. For example, In [13], online reinforcement learning is employed for a linear class of Markov games with quadratic cost functions which is proved to be converged both theoretically and practically. Guo et al. [14] addressed the off-policy Q-learning method in mean-field games with finite state space. Unfortunately, off-policy MARL requires that each agent knows the policies of other agents during the learning process, which is not a realistic assumption in competitive games. Also, the mentioned works rely only on linear quadratic games. However, one major challenge that restricts the use of conventional Nash-seeking methods is how to deal with massive or completely unknown dynamic environments in practical systems. One well-known family of algorithms that overcome this challenge is actor–critic algorithms, which are built upon value function approximation and parametrized policies [10].

Received 22 March 2024; revised 27 August 2024; accepted 6 September 2024. This work was supported in part by the Institute for Research in Fundamental Sciences (IPM) under Grant CS 1403-4-340. This article was recommended by Associate Editor L. C. Rego. *(Corresponding author: Hamed Kebriaei.)*

Alireza Ramezani Moghaddam is with the School of Electrical and Computer Engineering, College of Engineering, University of Tehran, Tehran 14155-6619, Iran (e-mail: a.ramezaney@ut.ac.ir).

Hamed Kebriaei is with the School of Electrical and Computer Engineering, College of Engineering, University of Tehran, Tehran 14155-6619, Iran, and also with the School of Computer Science, Institute for Research in Fundamental Sciences, Tehran 1953833511, Iran (e-mail: kebriaei@ut.ac.ir).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TSMC.2024.3462762.

Digital Object Identifier 10.1109/TSMC.2024.3462762

To address the Nash strategy seeking problem, Fox et al. [15] applied the MARL method for a class of games known as Markov potential games (MPGs) with continuous state-action spaces. Recent works of [16] and [17] suggest a class of independent policy gradient algorithms that analyze convergence for zero-sum MPGs. However, these works chiefly rely on a special class of zero-sum potential games which simplify the Nash-seeking problem. In the framework of two-player zero-sum stochastic games, Chen et al. [18] utilized a best-response learning-based algorithm that combines independent learning dynamics for matrix games with the minimax value iteration. For a noncooperative setting, Moghaddam and Kebriaei [19] combined expected policy gradient with temporal difference learning to find Nash solution under the restriction of monotonicity-convexity assumptions. In a restricted framework of a two-player zero-sum game,/Brown et al. [20] presented a general framework for self-play reinforcement learning and search that provably converges to a NE.

To the best of our knowledge, hereby we propose the first MARL actor–critic method for the NE seeking problem in general-sum Markov games that relies only on communications between agents. Specifically, for the actor step, we employ the stochastic policy gradient (SPG) algorithm and for the critic step, we apply a temporal difference learning [10]. Two general function approximators are used, one to approximate the mapping from state-action pair to long-term costs and the other for updating the policy of the agents. The parameters of the function approximators are updated upon each observation of state transition, players' actions, and the associated costs. The objective is to refine the estimations of long-term value and policy as more and more iterations are performed. In the case of linear function approximators, we prove that under common technical assumptions, the strategies of the agents converge to the first-order NE of the game. In summary, our main contributions are the following.

1) We propose a new noncooperative game framework in which the agents compete in an unknown dynamic environment without having an explicit model of their own cost functions and aim to minimize their long-term discounted costs. This is in sharp contrast to the works of [8] and [21] which deal with an unknown deterministic environment in which players have complete information of their own cost functions and model the players as myopic agents.
2) Using the SPG theorem, a new Nash-seeking algorithm is suggested for general-sum Markov games via TD(λ) actor–critic MARL. To address the MARL algorithms for Nash seeking in noncooperative games, some efforts have been reported in recent years (e.g., [22], [23], and [24]). However, these efforts only cope with zero-sum games or model-based linear environments. Here, on the contrary, the actor–critic agents try to solve the Nash-seeking problem in a dynamic general-sum game with a completely unknown environment.
3) Convergence to the unique NE of the game is proved using the linear function approximation. Here, we address convergence proof for noncooperative settings which profoundly differs from recent cooperative MARL studies that provide convergence proof to an optimal solution, e.g., [25] and [26]. One chief challenge of our analysis is that a Nash-seeking algorithm needs to converge to the intersection of the best responses of the competitive agents [27], instead of converging to a distributed optimal solution of a cooperative reward.

Notation: $\mathbb{R}$ represents the set of real numbers. C^1 is the set of all differentiable functions. For a vector A, $\|A\|$ and $A^\top$ indicate the Euclidean norm and the transpose of A, respectively. $\text{col}(x_1, \ldots, x_N) = [x_1^\top, \ldots, x_N^\top]^\top$ denotes the column augmentation of vectors x_n for $n = 1, \ldots, N$. Vector $[a{:}i{:}b]$ consists of all real numbers between a and b discretized by step-size i. We denote the inner product of two random vectors G and H by $\langle G, H\rangle_\kappa \triangleq G^\top \kappa H$ and the norm on the associated inner product space by $\|.\|_\kappa = \sqrt{\langle ., .\rangle_\kappa}$ where κ is a diagonal matrix consists of steady-state probabilities defined on state space $\mathcal{S}$. We denote the set of $\{Q|\, \|Q\|_\kappa < \infty.\}$ by $\mathcal{L}_\kappa(\mathcal{X})$. The operator $\Pi_\Phi(f)$ is the projection of the function f onto the space Φ.

II. Problem Formulation

In this section, we introduce the decision-making model in a multiagent system and formulate it as a Markov game in an unknown dynamic environment. We also provide the formulation of NE (Nash Q-value and Nash policy) for the proposed Markov game. The formulation is utilized to design a Nash-seeking algorithm in Section III.

A. Decision Making Model

We consider a set of N agents indexed by $n \in \mathcal{N} = \{1, \ldots, N\}$ playing a Markov game. We further consider that all the agents interact with an unknown dynamic stochastic environment characterized by a multiagent Markov decision process (MDP) as follows.

Multiagent MDP: A multiagent MDP is described by a tuple $(\mathcal{S}, \{\mathcal{A}_n\}_{n\in\mathcal{N}}, p, \{R_n\}_{n\in\mathcal{N}})$, where $\mathcal{S}$ and $\mathcal{A}_n$ denote the finite state space and action space of agent n, respectively. Moreover, R_n represents the expected transient cost function of agent n, and $p(s'|s, a)$ denotes the state transition kernel of the MDP given the joint action $a \in \mathcal{A} = \Pi_{j\in\mathcal{N}}\mathcal{A}_j$ and state $s \in \mathcal{S}$. Considering the state $s^t = s$ and joint action $a^t = a$ of the agents at time t, the next state $s^{t+1} = s'$ is sampled from the environment transition kernel $p(s'|s, a)$.

Cost Functions: At each time transition, each agent $n \in \mathcal{N}$ receives a cost $\rho_n^t \in \mathbb{R}$, a random variable whose conditional expectation is affected by its action, the actions of other players, $\mathcal{N}_n$, and also the state of the environment as follows:

$$R_n\big(s^t, a_n^t, a_{-n}^t\big) = \mathbb{E}\big(\rho_n^t \big| s^t, a_n^t, a_{-n}^t\big) \tag{1}$$

where $R_n : \mathcal{S} \times \mathcal{A}_n \times \mathcal{A}_{-n} \to \mathbb{R}$ is the expected transient cost function of agent n. Here, a_n^t is the action of agent n at time slot t, which is selected from the space $\mathcal{A}_n$. Furthermore, $a_{-n}^t = \text{col}(a_1^t, \ldots, a_{n-1}^t, a_{n+1}^t, \ldots, a_N^t)$ is the action of all the agents except agent n, selected from a space $\mathcal{A}_{-n} =$

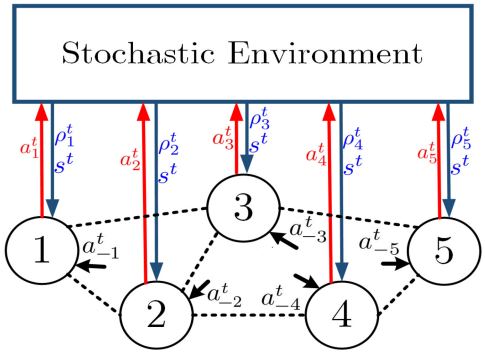

Fig. 1. Information scheme of a multiagent MDP.

$\Pi_{j\neq n}\mathcal{A}_j$. The communication scheme among the agents and environment in time step t is depicted in Fig. 1.

Policies: We let the agents choose actions according to their policies, also called strategies. For each agent n, we define its policy as $\pi_{\theta_n}(a_n^t|s^t)$ parameterized by θ_n, where $\theta_n \in \Theta_n$ for some compact $\Theta_n \subseteq \mathbb{R}^I$ with smooth boundary, i.e., $\partial\Theta_n \in C^1$. The policy $\pi_{\theta_n}(a_n^t|s^t)$ represents the probability density over the action set $\mathcal{A}_n$ given state s^t. Also, we define $\theta = \text{col}(\theta_1, \ldots, \theta_N) \in \Theta$, where $\Theta = \prod_{j\in\mathcal{N}}\Theta_j$. Then, the probability density over the joint action set $\mathcal{A}$ can be written as $\pi_\theta(a|s) = \Pi_{n\in\mathcal{N}}\pi_{\theta_n}(a_n|s)$. We assume π_θ to be stationary, i.e., time-independent.

Value Functions: Starting from initial state $s^0 = s$, the state-value function of agent n under the joint policy π_θ is defined as $J_n^\theta(s) \triangleq \mathbb{E}[\sum_{t=0}^{\infty}(\gamma)^t\rho_n^t|s^0 = s., \pi_\theta]$ where $\gamma \in (0, 1)$ is a discount factor. Then, we define the average cost of player n as follows:

$$J_n(\theta) \triangleq \mathbb{E}_{\bar{\kappa}_\theta}\big[J_n^\theta(s)\big] = \sum_{\mathcal{S}}\bar{\kappa}_\theta J_n^\theta(s) \tag{2}$$

where $\bar{\kappa}_\theta$ is state invariant distribution. Accordingly, starting by actions $a_n^0 = a_n$ and $a_{-n}^0 = a_{-n}$, the action-value function of agent n is written as follows:

$$Q_n^\theta(s, a_n, a_{-n}) \triangleq \mathbb{E}\left[\sum_{t=0}^{\infty}(\gamma)^t\rho_n^t\Big|\, s^0 = s\,, a_n^0 = a_n, a_{-n}^0 = a_{-n}, \pi_\theta\right]. \tag{3}$$

B. Markov Game

Since the actions of the agents affect the cost functions of each other via the term a_{-n}, we refer to this selfish cost minimization problem as the general-sum Game. We formulate our decision-making problem of the agents as the following Markov game:

$$\mathcal{G} = \begin{cases} \text{players: } n \in \mathcal{N} \\ \text{policies: } \pi_{\theta_n} \\ \text{value functions: } J_n^\theta(s) \\ \text{environment dynamics: } p\big(s'\big|s, a\big). \end{cases} \tag{4}$$

Accordingly, we define Nash policy as the best policy that each player can choose in response to other players' policies. Associated with the game $\mathcal{G}$ in (4), let us consider the following notion of equilibrium.

Definition 1: A joint policy π_{θ^*} is said to be ϵ-First-order NE (ϵ-FNE) of the game $\mathcal{G}$ in (4) if for all $n \in \mathcal{N}$, it satisfies $\nabla_{\theta_n}J_n(\theta) \leq \epsilon$ with $J_n(\theta)$ as in (2).

Thus, the compact notation for the joint policy $\pi_{\theta^*} \triangleq \text{col}(\pi_{\theta_1^*}, \ldots, \pi_{\theta_N^*})$ represents the ϵ-FNE policy. At a ϵ-FNE, each agent operates under policy $\pi_{\theta_n^*}$ that approximately satisfies first-order optimality condition in response to others agents' policy $\pi_{\theta_{-n}^*} = \prod_{j\neq n}\pi_{\theta_j^*}$. Considering the ϵ-FNE policy π_θ^*, the ϵ-FNE action-value function $Q_n^*(s, a_n, a_{-n}) \triangleq Q_n^{\theta^*}(s, a_n, a_{-n})$ is achieved when all agents are acting with ϵ-FNE policies from the initial state s onward. Clearly, we have First-order NE for $\epsilon = 0$.

Remark 1: FNE is equivalent to local NE point if the value functions $J_n(\theta)$ satisfy second-order condition of optimality at FNE, i.e., $\nabla_{\theta_n}^2 J_n(\theta^*) > 0$ for all $n \in \mathcal{N}$.

III. Actor–Critic Algorithm

In this section, we aim to design a TD(λ) actor–critic decision-making algorithm for the Markov game which operates under the following conditions: 1) the agents can only use local observations of the actions, the state of the environment and their own transient cost at each execution time and 2) the agents are not aware of the transition function of the environment, nor of the cost functions.

Let us define the set of states-actions as $\mathcal{Y}_n = \{\text{col}(s, a_n, a_{-n})|\ s \in \mathcal{S},\ a_n \in \mathcal{A}_n,\ a_{-n} \in \mathcal{A}_{-n}\}$. Our proposed actor–critic algorithm for the Markov game problem is described in Algorithm 1. In this algorithm, the system is initialized with the initial state $s^0 = s$. Having $y_n^t \in \mathcal{Y}_n$, we show the estimated action-value function and policy of each individual player via two general function approximators, $\tilde{Q}_n^\theta(y_n^t;\, \omega_n)$ and $\pi_{\theta_n}(a_n^t|s^t;\, \theta_n)$ that are parameterized by weight vectors $\omega_n \in \mathbb{R}^K$ and $\theta_n \in \Theta_n$, respectively. We assume that the agents are heterogeneous, i.e., each agent follows a different policy, and therefore acts differently in each underlying state of the environment. In this algorithm, agent n perceives the state of the environment at time step t, and takes the action $a_n^t \in \mathcal{A}_n$ sampled from its current policy $\pi_{\theta_n}(a_n^t|s^t)$. In this way, the agent receives the scalar cost ρ_n^t based on (1) and the environment transits into a new state s^{t+1} which is sampled from kernel $p(.|s^t, a^t.)$. After this transition, the agent observes the action of the others a_{-n}^t as well as the environment's new state, s^{t+1}. Then, the agent performs the next action a_n^{t+1} based on $\pi_{\theta_n}(a_n^{t+1}|s^{t+1})$, receives ρ_n^{t+1}, and observes the other agents' action a_{-n}^{t+1}. Based on this experience, at each transition from t to $t+1$, data $\{s^t, a_n^t, a_{-n}^t, \rho_n^t, s^{t+1}, a_n^{t+1}, a_{-n}^{t+1}\}$ is utilized by agent n to update the parameters of its function approximators (i.e., the parameters ω_n and θ_n).

In the critic step, the agent updates its approximation of action-value function $\tilde{Q}_n^\theta(y_n^t;\, \omega_n)$ via the TD(λ) learning

$$\omega_n^{t+1} = \omega_n^t + \alpha^t\delta_n^t z_n^t \tag{5a}$$

$$z_n^{t+1} = \gamma\lambda z_n^t + \nabla_{\omega_n}\tilde{Q}_n^\theta\big(y_n^t;\, \omega_n^t\big) \tag{5b}$$

where α^t is a time-varying step size and $z_n \in \mathbb{R}^K$ represents Sutton's eligibility trace for agent n [10] which is defined as

Algorithm 1: Actor–Critic method for Markov Games

Input: Initial values of the parameters ω_n, $z_n^0 = 0$, θ_n $\forall n \in \mathcal{N}$; the initial state $s^0 = s$, and $t = 0$;
Each agent $n \in \mathcal{N}$ executes action $a_n^0 \sim \pi_{\theta_n^0}$ and observes neighbors' action a_{-n}^0
repeat
  for *all* $n \in \mathcal{N}$ **do**
    observe $s^{t+1} \sim p(.|s^t, a^t)$ and reward ρ_n^{t+1}
    execute action $a_n^{t+1} \sim \pi_{\theta_n^t}$
    observe neighbors' action a_{-n}^{t+1}
    TD error: $\delta_n^t = \rho_n^t + \gamma \tilde{Q}_n^\theta(s^{t+1}, a_n^{t+1}, a_{-n}^{t+1}; \omega_n) - \tilde{Q}_n^\theta(s^t, a_n^t, a_{-n}^t; \omega_n)$
    critic-step: $\omega_n \leftarrow \omega_n + \alpha^t \delta_n^t z_n$
    $z_n \leftarrow \gamma\lambda z_n + \nabla_{\omega_n} \tilde{Q}_n^\theta(s^t, a_n^t, a_{-n}^t; \omega_n)$
    actor-step:
    $\theta_n \leftarrow \theta_n - \beta^t \tilde{Q}_n^\theta \nabla_{\theta_n} \log \pi_{\theta_n}(a_n^t|s^t)$
  end
  update the iteration counter $t \leftarrow t + 1$
until *Policy Convergence*;
Return $\theta = col(\theta_1, \ldots, \theta_N)$

$z_n^t = \sum_{k=0}^t (\gamma\lambda)^{t-k} \nabla_{\omega_n} \tilde{Q}_n^\theta(y_n^t; \omega_n^t)$. The constant $\lambda \in (0, 1]$ is the trace-decay parameter which determines the tradeoff between estimation bias and performance [10]. Furthermore, $\delta_n^t = \rho_n^t + \gamma \tilde{Q}_n^\theta(y_n^{t+1}; \omega_n^t) - \tilde{Q}_n^\theta(y_n^t; \omega_n^t)$ is the temporal difference error of agent n at time step t which represents the gap between two successive observations of the same value. On the other hand, the actor step resembles the so-called SPG learning, which aims at estimating the best-response policy via a gradient step

$$\theta_n^{t+1} = \theta_n^t - \beta^t \tilde{Q}_n^\theta \nabla_{\theta_n} \log \pi_{\theta_n^t}(a_n^t|s^t) \tag{6}$$

where β^t is a time-varying step size. The aim of the overall algorithm is to execute actor and critic steps at each iteration, so that the actor policy parameter θ_n converges toward the ϵ-FNE as in (1).

IV. Convergence Results

In this section, the convergence of the proposed actor–critic learning algorithm with linear function approximator is studied. To this end, we first theorize the convergence of the critic step while the joint policy π_θ is fixed on this faster time scale. Then, we present convergence of the actor step, i.e., the policy parameter θ^t, on the slower time scale. Finally, based on the well-known two-time-scale theorem [28, Th. 1.1] we conclude the convergence of Algorithm 1. We start by stating the essential assumptions.

Assumption 1: We assume that for any $n \in \mathcal{N}$, $s \in \mathcal{S}$, and $a_n \in \mathcal{A}_n$, the policy $\pi_{\theta_n}(a_n|s; \theta_n) > 0$ for all $\theta_n \in \Theta_n$. Also, π_{θ_n} is continuously differentiable w.r.t θ_n over Θ_n. Furthermore, the Markov chains $\{s^t\}_{t\geq 0}$, and $\{(s^t, a^t)\}_{t\geq 0}$ induced by π_θ are both irreducible and aperiodic, with unique invariant distributions $\bar{\kappa}_\theta$, and $\kappa_\theta = \bar{\kappa}_\theta \pi_\theta(a|s)$, respectively.

Assumption 1 is a standard assumption on MDP and policy functions as in [29].

Assumption 2: The estimated Q-function $\tilde{Q}_n^\theta(y_n^t; \omega_n)$ induced by policy π_θ is assumed to be a linear function approximator as $\tilde{Q}_n^\theta(y_n^t; \omega_n) = \varphi^\top(y_n^t)\omega_n$ where $\omega_n = [\omega_n(1), \ldots, \omega_n(K)]^\top$ are the weight vectors, and $\varphi = [\varphi_1, \ldots, \varphi_K]^\top$ are fixed basis functions defined on the space $\mathcal{Y}_n$. Also, K is the total number of basis functions φ_k. The basis functions $\{\varphi_1, \ldots, \varphi_K\}$ are linearly independent and bounded, i.e., $\varphi_k \in \mathcal{L}_{\kappa_\theta}(\mathcal{Y}_n)$ for all k and $y_n \in \mathcal{Y}_n$.

Based on Assumption 2, the approximation of the Q_n^θ evolves in the functional space $\Phi = \{\varphi^\top(\cdot)\omega_n | \omega_n \in \mathbb{R}^K\} \subseteq \mathcal{L}_{\kappa_\theta}(\mathcal{Y}_n)$, where $\Pi_\Phi Q_n^\theta$ is the natural approximation to Q_n^θ.

Assumption 3: The step sizes α^t and β^t are positive and nonincreasing, and they satisfy $\sum_{t=0}^\infty \alpha^t = \infty$, $\sum_{t=0}^\infty (\alpha^t)^2 < \infty$, $\sum_{t=0}^\infty \beta^t = \infty$, $\sum_{t=0}^\infty (\beta^t)^2 < \infty$. Also, $\lim_{t\to\infty} (\beta^t/\alpha^t) = 0$ is held.

The last condition in Assumption 3 means that the critic is a faster recursion than the actor.

Assumption 4: There exists $\eta > 0$ such that $||\rho_n^t|| \leq \eta$ for all $t \geq 0$ and $n \in \mathcal{N}$.

Assumption 5: For all $n \in \mathcal{N}$ and θ on the boundary of Θ, we have $-\nabla_{\theta_n} J_n(\theta) \notin N_{\Theta_n}(\theta_n)$ where $N_{\Theta_n}(\theta_n)$ is the normal cone of Θ_n at θ_n. ■

As we presume that the set Θ_n has a smooth boundary, it is concluded that the normal cone of Θ_n at any θ_n is a single direction vector. Therefore, Assumption 5 is alleviated to $\nabla_{\theta_n} J_n(\theta) \neq k N_{\Theta_n}(\theta_n)$ for any real number k, which is not a restrictive assumption.

Assumption 6: The policy iterate θ^t in actor step includes a local projection operator $\Gamma : R^{N\times I} \to \Theta$, which projects any θ^t onto the compact set Θ. Also, Θ is large enough to include at least one stable FNE of game $\mathcal{G}$.

Remark 2: For instance, potential games are one of the most widely used examples of multiagent settings with at least one stable FNE [30].

To establish our convergence analysis, we consider the TD(λ) operator [31, Sec. 6.3.3] which is indexed by a parameter $\lambda \in (0, 1)$, and for a $Q^n \in \mathcal{L}_{\kappa_\theta}(\mathcal{Y}_n)$ is described by the operator $T^{(\lambda)} : \mathcal{L}_{\kappa_\theta}(\mathcal{Y}_n) \to \mathcal{L}_{\kappa_\theta}(\mathcal{Y}_n)$, defined as

$$T^{(\lambda)} Q_n(y_n) = (1-\lambda) \sum_{m=0}^\infty \lambda^m \cdot \mathbb{E}\left[\left(\sum_{t=0}^m (\gamma)^t R_n(y_n^t) \right) + (\gamma)^{m+1} Q_n\left(y_n^{m+1}\right) \middle| y_n^0 = y_n \right].$$

Theorem 1 (Critic Step): Under Assumptions 1–6, for any given joint policy π_θ and $n \in \mathcal{N}$, the critic step (5) converges to ω_n^θ which is the unique solution to the following fixed-point problem:

$$\Pi_\Phi T^{(\lambda)}\left(\varphi^\top \omega_n^\theta\right) = \varphi^\top \omega_n^\theta. \tag{7}$$

Furthermore, ω_n^θ satisfies the following bound:

$$\left\| \varphi^\top \omega_n^\theta - Q_n^\theta \right\|_\kappa \leq \frac{1}{1-\xi} \left\| \Pi_\Phi Q_n^\theta - Q_n^\theta \right\|_\kappa \tag{8}$$

where Q_n^θ is the actual Q-function induced by π_θ, and $\xi = [\gamma(1-\lambda)/\sqrt{1-\gamma\lambda}]$.

Considering error bound (8), by selecting compatible basis functions as in [32], the bias term $\|\Pi_\Phi Q_n^\theta - Q_n^\theta\|$ converges to zero, and consequently $\lim_{t\to\infty} \varphi\omega_n^\theta \to Q_n^\theta$. Therefore, for a fixed policy, each agent can obtain an estimation of the corresponding action-value function, even with local information only. This approximation of the action-value function is then adopted in the actor step to approximate the FNE.

In view of the SPG theorem [33, Th. 1], and by using the Q-function defined in (3), the gradient of objective $J_n(\theta) = \mathbb{E}_{\tilde{\kappa}_\theta}[J_n(\theta, s)]$ reads as

$$\nabla_{\theta_n} J_n(\theta) = \mathbb{E}_{\kappa_\theta}\left[\nabla_{\theta_n} \log \pi_{\theta_n} Q_n^\theta(s, a_n, a_{-n})\right]. \tag{9}$$

In the following theorem, we investigate the stochastic a.s. convergence of the actor's parameter θ to the ϵ-FNE. Also, to simplify the notation, hereafter we use $\mathbb{E}_{\kappa_\theta}$ as $\mathbb{E}$.

Theorem 2 (Actor Step): Let Assumptions 1–7 hold. The policy parameter $\{\theta^t\}$ generated by the actor-step in (6) converges almost surly to a ϵ-FNE of the game $\mathcal{G}$ in (4) from any initial condition.

Proof: First by defining the joint gradient vector $\tilde{G}(\theta^t) = \text{col}(\nabla_{\theta_1}\tilde{J}_1(\theta^t), \ldots, \nabla_{\theta_N}\tilde{J}_N(\theta^t))$ with each $\nabla_{\theta_n}\tilde{J}_n(\theta^t) = \nabla_{\theta_n^t}(\log \pi_{\theta_n})\omega_n^t\varphi(s^t, a^t)$ and augmenting actor iterates (6) for all $n \in \mathcal{N}$, we have

$$\theta^{t+1} = \theta^t - \beta^t \tilde{G}(\theta). \tag{10}$$

Now, we consider $\mathcal{F} = \sigma(\theta^\tau, \tau \le t)$ as the σ–field generated by $\{\theta^\tau, \tau \le t\}$. Furthermore, we denote

$$\zeta^{t,1} = G(\theta^t) - \mathbb{E}\big(G(\theta^t)|\mathcal{F}^t\big)$$
$$\zeta^{t,2} = \mathbb{E}\Big[\Big(G(\theta^t) - \tilde{G}(\theta^t)\Big)|\mathcal{F}^t\Big]$$

where $G(\theta) = \text{col}(\nabla_{\theta_1}J_1(\theta), \ldots, \nabla_{\theta_N}J_N(\theta))$. Then, the actor update in (10) with a local projection becomes

$$\theta^{t+1} = \Gamma\Big\{\theta^t - \beta^t\Big[\mathbb{E}\Big(\tilde{G}(\theta^t)|\mathcal{F}^t\Big) + \zeta^{t,1} + \zeta^{t,2}\Big]\Big\}. \tag{11}$$

Note that since the critic converges by Theorem 1, it concludes that $\zeta^{t,2} = o(1)$. Letting $M^t = \sum_{\tau=0}^t \beta^\tau \zeta^{\tau+1,1}$, then $\{M^t\}$ is a martingale sequence. Note that for all $n \in \mathcal{N}$, the sequences $\{\omega_n^t\}$, $\{\varphi_n^t\}$, and $\{z_n^t\}$ are all bounded, and therefore, $\{\zeta^{t,1}\}$ is bounded. Thus, from Assumption 3, we achieve that almost surely

$$\sum_t \mathbb{E}\Big(\Big\|M^{t+1} - M^t\Big\|^2|\mathcal{F}^t\Big) = \sum_{t\ge1}\Big\|\beta^t\zeta^{t,1}\Big\|^2 < \infty. \tag{12}$$

By the Martingale convergence theorem [34, p. 149], the Martingale sequence $\{M^t\}$ converges almost surely. Therefore, we have $\lim_t P(\sup_{i\ge t}\|\sum_{\tau=t}^i \beta^\tau\zeta^{\tau,1}\| \ge \varepsilon) = 0$. Furthermore, by Assumption 1, and implicit function theorem, it can be obtained that $\mathbb{E}(\tilde{G}(\theta^t)|\mathcal{F}^t.)$ is continuous w.r.t θ^t. Hence, we can apply the Kushner–Clark lemma [35, pp. 191-196] which concludes that the update θ converge a.s. to the set of asymptotically stable equilibrium of the following ODE:

$$\dot{\theta} = \hat{\Gamma}\Big[\mathbb{E}(\tilde{G})(\theta)\Big] \tag{13}$$

where

$$\hat{\Gamma}[h(\theta)] = \lim_{0<\upsilon\to0}\{\Gamma(\theta + \upsilon h(\theta)) - \theta\}/\upsilon.$$

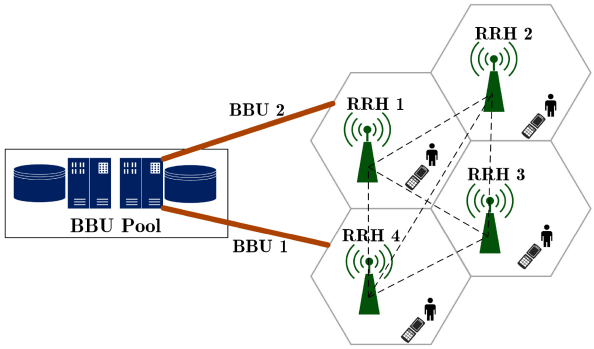

Fig. 2. C-RAN architecture.

Based on Assumption 5, the set of asymptotically stable equilibrium of ODE (13) satisfy $\mathbb{E}(\tilde{G}(\theta)) = 0$ which leads to $\nabla_{\theta_n}J_n(\theta) = 0$ for all $n \in \mathcal{N}$. Therefore, having Assumption 6, it can be concluded that the sequence $\{\theta^t\}$ induced by iterate (11) converges a.s. to a FNE of the game $\mathcal{G}$ in (4).

Remark 3: Since the linear function approximator has nonzero approximation error for Q_n^θ, the convergent point of (6) corresponds to a small neighborhood of local optimum of $J_n(\theta)$, i.e., ϵ-FNE of $\mathcal{G}$. Nevertheless, restricting the linear basis functions to compatible features as in [32] and [36] can lead to convergence to FNE. Furthermore, for strongly convex games as it is argued in [37], there exists a unique asymptotically stable NE and therefore, update (6) converges to the NE of the game.

Finally, having proof of critic and actor steps alongside Assumption 3, the convergence of Algorithm 1 as a two-time scale algorithm is concluded using [28, Th. 1.1].

V. Illustrative Example

In this section, we present an application example for the optimization of Cloud Radio Access Network (C-RAN) to evaluate our algorithm. In [38] this model is studied with complete knowledge about the dynamic model and cost functions, whereas in our scheme, the agents are aware of none of them and try to learn the optimal policy by the proposed actor–critic RL method. In a C-RAN, all conventional tasks of base transceiver stations (BTSs) are fulfilled by baseband units (BBUs) located in a BBU pool. Besides, remote radio heads (RRHs) act as soft relays and interfaces between the BBUs and user equipment by transmitting radio frequency signals to user equipment in downlink and forwarding the baseband uplink signals from the user equipment to the BBUs.

In our application example, we consider 2 BBUs and 4 RRHs (hence, $N = 4$) as it is depicted in Fig. 2. The dynamic difference equation for congestion estimation is defined as $s^{t+1} = As^t + Ba^t + Dw^t$ where $w^t \sim \mathfrak{N}(0, 1)$, and $\mathfrak{N}(0, 1)$ is the white Gaussian noise with zero mean and standard deviation 1. Also, $A = 0.8I_2$, $D = \begin{bmatrix}0.2 & 0.4\end{bmatrix}^\top$, and $B = \begin{bmatrix}0.1 & 0 & 0.2 & 0 & 0.1 & 0 & 0.3 & 0\\ 0 & 0.1 & 0 & 0.2 & 0 & 0.1 & 0 & 0.3\end{bmatrix}$.

Moreover, $s^t \in [0 : 0.1 : 8]$ is the tth stage congestion in BBUs and $a^t = \text{col}(a_1^t, a_2^t, a_3^t, a_4^t)$ where each $a_n^t \in [0 : 0.1 : 4]$ for $n \in \mathcal{N} = \{1, 2, 3, 4\}$ is the strategy of RRHs

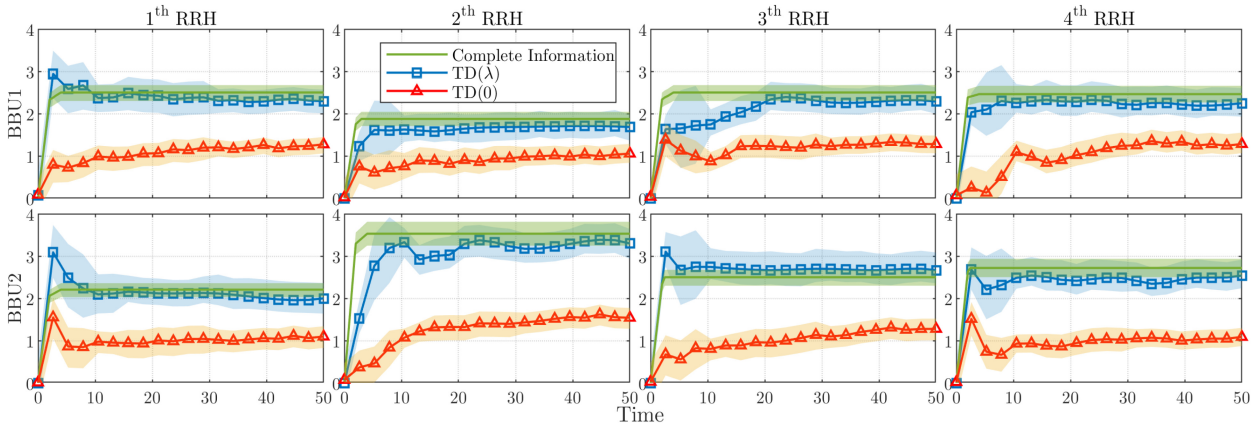

Fig. 3. Mean-variance plot of transmission rates of all RRHs in each BBU.

which is their flow rate transferred to each BBUs. The cost function of agent n can be considered as follows:

$$R_n\left(y_n^t\right) = \left(s^t\right)^\top H_n s^t + \left(s^t\right)^\top E_n a_n^t + \left(pr\left(a_{-n}^t\right)\right)^\top G_n a_n^t$$

where constant matrices H_n, E_n, G_n are as follows.

1) *RRH1:* $H_n = 0.2I_2$, $E_n = 0.35I_2$, and $G_n = 2I_2$.
2) *RRH2:* $H_n = 0.4I_2$, $E_n = 0.65I_2$, and $G_n = 1I_2$.
3) *RRH3:* $H_n = 0.1I_2$, $E_n = 0.35I_2$, and $G_n = 3I_2$.
4) *RRH4:* $H_n = 0.3I_2$, $E_n = 0.45I_2$, and $G_n = 5I_2$.

$pr(a_{-n}^t) = (1/[N-1])\sum_{j=N/n} a_j^t$ determines the price of each BBUs. We let $\gamma = 0.95$ and $\lambda = 0.2$. The action of each agent is sampled from a parameterized Gaussian policy $\mathcal{W}(\mu_{\theta_n}(s), \nu^2)$ where $\mu_{\theta_n}(s) = \psi(s)^\top \theta_n$ is the mean, $\psi(s) \in \mathbb{R}^I$ is the policy basis function and $\nu^2 = 0.1$ is the fixed variance. To satisfy Assumption 3, learning step sizes are selected as $\alpha^t = (1/t)^{0.65}$ and $\beta^t = (1/t)^{0.85}$ for all agents.

To evaluate the stochastic results of the simulations, we run the algorithm for 100 times and depict the mean-variance plot of policies in Fig. 3. In this figure, we depict the policies for two incomplete information algorithms (i.e., TD(λ) and TD(0)) and compare it with the analytical solution obtained under complete information (labeled as Complete Information in Fig. 3). Using the same computer specifications, the convergence time of analytical solution, TD(0) and TD(λ) algorithms are 41.19 s, 101.27 s and 142.66 s, respectively. While TD(λ) is more complex computationally than TD(0) (this is due to the additional process of calculating z_n^t), in return, it outperforms the TD(0) in alleviating estimation error. Also, we compare the performance of TD(λ) and TD(0) algorithms for the first RRH in terms of achieving a better estimation of Nash Q-value function. To this end, we increase the number of basis functions K and plot the error between the actual Nash Q-value and estimated Q-value in Fig. 4. Based on Fig. 4, we can see that increasing the number of basis functions leads to a more accurate approximation until overfitting starts.

VI. Conclusion

In this article, we developed MARL for general-sum Markov games to address the model-free decision-making problem for a set of agents. A TD(λ) actor–critic algorithm is proposed to learn the Nash policy from real system data rather than using a mathematical system model. The proposed actor–critic algorithm iteratively learns each agent's best response to the actions of others. To evaluate our method, a case study on a C-RAN was conducted. The simulation study showed that the proposed method is promising in terms of reducing the estimation error and convergence.

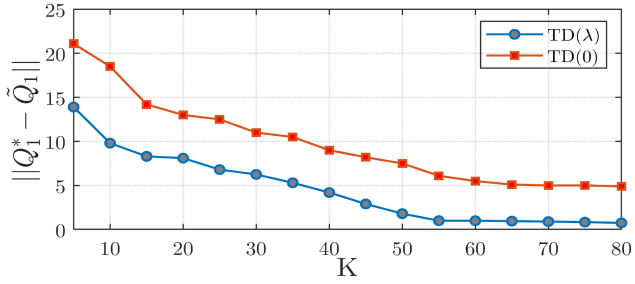

Fig. 4. Estimation error versus number of basis functions.

Acknowledgment

The authors express their sincere thanks to Ali Kahe, a graduate student of ECE at the University of Tehran for his helpful discussions and comments.

References

[1] Z. Deng, "Distributed algorithm design for resource allocation problems of second-order multiagent systems over weight-balanced digraphs," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 6, pp. 3512–3521, Jun. 2021.
[2] S. Tan, Y. Wang, and A. V. Vasilakos, "Distributed population dynamics for searching generalized Nash equilibria of population games with graphical strategy interactions," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 5, pp. 3263–3272, May 2022.
[3] Z. Deng, "Distributed algorithm design for aggregative games of euler-lagrange systems and its application to smart grids," *IEEE Trans. Cybern.*, vol. 52, no. 8, pp. 8315–8325, Aug. 2022.
[4] F. Parise, S. Grammatico, B. Gentile, and J. Lygeros, "Distributed convergence to Nash equilibria in network and average aggregative games," *Automatica*, vol. 117, Jul. 2020, Art. no. 108959.
[5] C. Mu, K. Wang, and C. Sun, "Policy-iteration-based learning for nonlinear player game systems with constrained inputs," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 10, pp. 6488–6502, Oct. 2021.
[6] M. Shokri and H. Kebriaei, "Leader–follower network aggregative game with stochastic agents' communication and activeness," *IEEE Trans. Autom. Control*, vol. 65, no. 12, pp. 5496–5502, Dec. 2020.

[7] T. Başar and G. J. Olsder, *Dynamic Noncooperative Game Theory*, 2nd ed. Philadelphia, PA, USA: SIAM, 1998.
[8] M. Shokri and H. Kebriaei, "Network aggregative game in unknown dynamic environment with myopic agents and delay," *IEEE Trans. Autom. Control*, vol. 67, no. 4, pp. 2033–2038, Apr. 2022.
[9] X. Cai, F. Xiao, and B. Wei, "Resilient Nash equilibrium seeking in multiagent games under false data injection attacks," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 53, no. 1, pp. 275–284, Jan. 2022.
[10] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.
[11] T. Zhu, D. Ye, Z. Cheng, W. Zhou, and S. Y. Philip, "Learning games for defending advanced persistent threats in cyber systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 53, no. 4, pp. 2410–2422, Apr. 2023.
[12] X. You, X. Li, Y. Xu, H. Feng, J. Zhao, and H. Yan, "Toward packet routing with fully distributed multiagent deep reinforcement learning," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 2, pp. 855–868, Feb. 2020.
[13] X. Xin et al., "Online reinforcement learning multiplayer non-zero sum games of continuous-time Markov jump linear systems," *Appl. Math. Comput.*, vol. 412, Jan. 2022, Art. no. 126537.
[14] X. Guo, A. Hu, R. Xu, and J. Zhang, "Learning mean-field games," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–11.
[15] R. Fox, S. M. Mcaleer, W. Overman, and I. Panageas, "Independent natural policy gradient always converges in Markov potential games," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2022, pp. 4414–4425.
[16] D. Ding, C.-Y. Wei, K. Zhang, and M. Jovanovic, "Independent policy gradient for large-scale Markov potential games: Sharper rates, function approximation, and game-agnostic convergence," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 5166–5220.
[17] R. Zhang, J. Mei, B. Dai, D. Schuurmans, and N. Li, "On the global convergence rates of decentralized softmax gradient play in Markov potential games," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, pp. 1923–1935, 2022.
[18] Z. Chen, K. Zhang, E. Mazumdar, A. Ozdaglar, and A. Wierman, "A finite-sample analysis of payoff-based independent learning in zero-sum stochastic games," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, 2024, pp. 1–58.
[19] A. R. Moghaddam and H. Kebriaei, "Expected policy gradient for network aggregative Markov games in continuous space," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Apr. 22, 2024, doi: 10.1109/TNNLS.2024.3387871.
[20] N. Brown, A. Bakhtin, A. Lerer, and Q. Gong, "Combining deep reinforcement learning and search for imperfect-information games," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 17057–17069.
[21] J. Zhang, J. Sun, and C. Zhang, "Stochastic game in linear quadratic gaussian control for wireless networked control systems under DoS attacks," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 2, pp. 902–910, Feb. 2022.
[22] C. Xiong, Q. Ma, J. Guo, and F. L. Lewis, "Data-based optimal synchronization of heterogeneous multiagent systems in graphical games via reinforcement learning," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jul. 18, 2023, doi: 10.1109/TNNLS.2023.3291542.
[23] Y. Zhang, B. Zhao, D. Liu, and S. Zhang, "Event-triggered control of discrete-time zero-sum games via deterministic policy gradient adaptive dynamic programming," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 8, pp. 4823–4835, Aug. 2022.
[24] X. Cai, F. Xiao, B. Wei, M. Yu, and F. Fang, "Nash equilibrium seeking for general linear systems with disturbance rejection," *IEEE Trans. Cybern.*, vol. 53, no. 8, pp. 5240–5249, Aug. 2022.
[25] P. Dai, W. Yu, G. Wen, and S. Baldi, "Distributed reinforcement learning algorithm for dynamic economic dispatch with unknown generation cost functions," *IEEE Trans. Ind. Informat.*, vol. 16, no. 4, pp. 2258–2267, Apr. 2020.
[26] P. Dai, W. Yu, H. Wang, and S. Baldi, "Distributed actor–critic algorithms for multiagent reinforcement learning over directed graphs," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 10, pp. 7210–7221, Oct. 2022.
[27] T. Basar and G. J. Olsder, *Dynamic Noncooperative Game Theory*, vol. 23. Philadelphia, PA, USA: SIAM, 1999,
[28] V. S. Borkar, "Stochastic approximation with two time scales," *Syst. Control Lett.*, vol. 29, no. 5, pp. 291–294, 1997.
[29] V. Konda and J. Tsitsiklis, "Actor–critic algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 12, 1999, pp. 1–7.
[30] M. Ye and G. Hu, "Solving potential games with dynamical constraint," *IEEE Trans. Cybern.*, vol. 46, no. 5, pp. 1156–1164, May 2015.
[31] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Nashua, NH, USA: Athena Sci., 1996.
[32] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 12, 1999, pp. 1–7.
[33] N. Casas, "Deep deterministic policy gradient for urban traffic light control," 2017, *arXiv:1703.09035*.
[34] J. Neveu, *Discrete-Parameter Martingales*. New Amsterdam, The Netherlands: Elsevier, 2006.
[35] H. J. Kushner and D. S. Clark, *Stochastic Approximation Methods for Constrained and Unconstrained Systems*, vol. 26. New York, NY, USA: Springer, 2012.
[36] J. Tsitsiklis and B. Van Roy, "Analysis of temporal-diffference learning with function approximation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 9, 1996, pp. 1–7.
[37] J. B. Rosen, "Existence and uniqueness of equilibrium points for concave N-person games," *Econometrica*, vol. 33, no. 3, pp. 520–534, 1965.
[38] M. Saffar, H. Kebriaei, and D. Niyato, "Pricing and rate optimization of cloud radio access network using robust hierarchical dynamic game," *IEEE Trans. Wireless Commun.*, vol. 16, no. 11, pp. 7404–7418, Nov. 2017.

Alireza Ramezani Moghaddam received the B.Sc. degree in electrical engineering from the Iran University of Science and Technology, Tehran, Iran, in 2013, and the M.Sc. degree in control systems engineering from the University of Tehran, Tehran, in 2017, where he is currently pursuing the Ph.D. degree focusing on reinforcement learning.

His research interests include multiagent reinforcement learning, stochastic optimization, and intelligent control.

Hamed Kebriaei (Senior Member, IEEE) received the Ph.D. degree in electrical engineering and control systems from the University of Tehran, Tehran, Iran, in 2011.

He is an Associate Professor of Control Systems with the School of Electrical and Computer Engineering, University of Tehran. He was a Postdoctoral Researcher with the Università degli Studi del Sannio, Benevento, Italy, from 2014 to 2015. His research interests include optimization and learning in control systems, game theory, distributed optimization, and multiagent reinforcement learning.

Dr. Kebriaei was honored with the Outstanding Reviewer Award from IEEE TRANSACTIONS ON CYBERNETICS in 2022 and the Outstanding Young Researcher Award from the University of Tehran in 2023. He served as a Guest Editor for IEEE CONTROL SYSTEMS LETTERS in 2023. He is also an Associate Editor for EUCA Conference Editorial Board 2025. He is a Technical Committee Member of the IEEE Control Systems Society in Networks and Communication Systems and serves as a Board Member for the Control Systems Chapter, IEEE Iran Section.