Learning Stationary Correlated Equilibria in Constrained General-Sum Stochastic Games

Vesal Hakami and Mehdi Dehghan, *Member, IEEE*

***Abstract*—We study constrained general-sum stochastic games with unknown Markovian dynamics. A distributed constrained no-regret Q-learning scheme (CNRQ) is presented to guarantee convergence to the set of stationary correlated equilibria of the game. Prior art addresses the unconstrained case only, is structured with nested control loops, and has no convergence result. CNRQ is cast as a single-loop three-timescale asynchronous stochastic approximation algorithm with set-valued update increments. A rigorous convergence analysis with differential inclusion arguments is given which draws on recent extensions of the theory of stochastic approximation to the case of asynchronous recursive inclusions with set-valued mean fields. Numerical results are given for the exemplary application of CNRQ to decentralized resource control in heterogeneous wireless networks.**

***Index Terms*—Asynchronous stochastic approximation, constrained stochastic game, correlated equilibrium (CE), multiagent systems, no-regret learning, Q-learning.**

I. Introduction

STOCHASTIC games [1] are very broad framework, generalizing both Markov decision processes (MDPs) and repeated games. In particular, stochastic games are extensions of MDPs to the multiagent case, and of repeated games to the multistate case. A stochastic game is played in a sequence of stages. At the beginning of each stage, the game is in a certain state. The agents select their actions, and each agent receives a reward that depends on both current state and action profile of all the agents. The game then transitions to a new state with a certain probability which, by Markov property, depends only on the previous state and the actions chosen by all agents. This process recurs at the new state, and the interaction goes on for a finite or infinite number of stages. Similarly to the case with MDPs, each agent participating in a stochastic game aims to maximize an expected cumulative reward measure often calculated as either average reward per stage or total discounted reward. However, the solution concept differs from the case of MDPs in that the agents should settle instead for competitive optimality which corresponds to some notion of strategic equilibrium. The most common notions of equilibria are Nash [2] and correlated equilibria [3]. A Nash equilibrium (NE) is a vector of independent strategies, each of which is a probability distribution over actions, in which each agent's strategy is optimal given the strategies of the other agents. Correlated equilibrium (CE) is more general than NE in that it allows for dependencies among agents' strategies: a CE is a probability distribution over the agents' joint actions such that if a joint action is drawn from this distribution (presumably by a trusted third party), and each agent is told separately its own component, then it has no incentive to choose a different action, because, assuming that all others also obey, the suggested action is the best in expectation.

Stochastic games are particularly appealing since they capture both strategic and stochastic aspects of a real-world scenario. Stochastic games with constraints [4] are even more interesting as they can also account for multiple objectives or for bounds on consumption of resources. In constrained stochastic games, the agents incur an additional cost at each stage which, similarly to the instantaneous reward, is a function of both current state and current action profile of the agents. The equilibrium policy should then be feasible under the agents' individual average/discounted constraints.

A. Literature Review

Computational methods for equilibria in stochastic games have been actively pursued over the past decades. The majority of the schemes work in an offline fashion, i.e., for the case where Markovian dynamics (transition probabilities) are known *a priori*. Under this assumption, an extensive account on solution methods for stochastic games with special structures and with various reward criteria is given in [5]. Constrained stochastic games are typically approached via mathematical programming. See [4] and [6] for treatments of constrained games with jointly-controlled and independently-controlled state processes, respectively.

However, when Markovian dynamics are unknown, one may instead resort to learning-theoretic online solutions. It is within this perspective that stochastic games are also proposed as the standard framework for multiagent reinforcement learning (MARL) [7]. Given the complexity of the strategy space in stochastic games, the solution concept sought in MARL algorithms is typically expressed in terms of stationary policies. The stationarity of a policy implies that it depends on the history of the game only through the current state. Traditionally,

Manuscript received January 25, 2015; revised June 12, 2015; accepted June 26, 2015. Date of publication August 7, 2015; date of current version June 14, 2016. This paper was recommended by Associate Editor B. Chaib-draa.

The authors are with the Department of Computer Engineering and Information Technology, Amirkabir University of Technology, Tehran 15914, Iran (e-mail: vhakami@aut.ac.ir; dehghan@aut.ac.ir).

This paper has supplementary downloadable multimedia material available at http://ieeexplore.ieee.org provided by the authors. This includes appendices of the paper, including Appendix A: Proofs of Lemmas 1 to 4 from Section IV.B, Appendix B: Lemma B.1, and Appendix C: Proof of Theorem 3. This material is 8.36 KB in size.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2015.2453165

the MARL literature's prime interest has been directed toward learning stationary NE. Such equilibria have been shown to exist for both discounted and average reward stochastic games, with an extra ergodicity assumption on the transition structure of the latter [5]. These existence results also carry over to the constrained case, albeit with an additional strong Slater feasibility condition [4]. MARL algorithms for the computation of stationary NE in infinite-horizon general-sum stochastic games are primarily proposed for the unconstrained case. Depending on their informational assumptions, these algorithms can be divided into two broad classes: 1) joint action learning (JAL) and 2) independent action learning (IAL).

JAL algorithms constitute the early research on learning equilibria in stochastic games. These algorithms learn in the joint action space, and require that agents observe the actions and possibly the rewards of the other agents. Prominent examples are Nash-Q for discounted games [8], its variant, Nash-R [9] for average reward games, and FF-Q [10]. These are all multiagent extensions of the celebrated Q-learning scheme of the MDP literature [11], with the distinction that they maintain Q-values for all joint actions at a given state. The main drawback of the current instances of JAL algorithms is that they all require repeated calls to an NE solver during the learning process, and this solver needs that the agents' Q-values be public information. Also, the convergence results are limited to a restricted class of games (e.g., common interest or zero-sum); finally, multiagent Q-learning algorithms can generally take a long time to converge, and some recent studies advocate the use of heuristics [12], model-based learning [13], or leveraging prior knowledge [14] to speed up their convergence. In IAL algorithms, on the other hand, the agents only rely on their own past received rewards without knowing the actions or rewards of the other agents. IAL algorithms can thus operate in more informationally opaque scenarios, and unlike the case with JAL algorithms, their memory footprint is not exponential in the number of agents. A pioneer IAL algorithm is win or learn fast policy hill climbing [15] which is only empirically evaluated and its convergence is not theoretically analyzed. The Markov game interconnected learning automata algorithm in [16] is based on an interconnected learning automata abstraction and is only provably convergent in average reward games with pure NE policies. More recently, an online stochastic game sub-problem algorithm [17] has been proposed which is proved to be generally convergent to stationary NE policies in discounted games.

In this paper, we depart from the NE-centric mainstream of MARL research, and instead address the problem of learning stationary CE in stochastic games [18]. The importance of CE arises from the fact that unlike NE, the concept of CE permits coordination between agents, and CE that are not NE can achieve higher rewards than NE, by avoiding positive probability mass on less desirable outcomes [3]. Within the context of normal-form games, the most efficient procedure for learning CE is the no-regret algorithm [19], [20]. No-regret learning essentially requires that agents depart from their current play with probabilities that are proportional to measures of regret for not having used other strategies in the past. It is shown in [19] that once all the players' regrets approach zero, the joint empirical frequency of play converges with probability one to the set of CE of the game. A key property of no-regret learning is that it is an uncoupled update rule [21]; i.e., each agent only needs to know its own reward function and to monitor the actions taken by the others to adjust its play probabilities.

When it comes to stochastic games, however, the literature on learning CE is very thin. The existence of stationary CE is implied by the existence of stationary NE in general-sum stochastic games. A direct proof is also given in [18] using a fixed point argument. As for the algorithms, CE-Q [22] and QnR [23] are the only MARL algorithms we know of that address the problem of learning stationary CE. Both algorithms belong to the JAL family, and use Q-learning to estimate the joint action values for each state of the game. Similarly to Nash-Q, CE-Q relies on an equilibrium solver with access to all agents' Q-tables to update the CE policy in each iteration. Given the structural simplicity of CE with respect to NE (convex polytope versus fixed points), each call to a CE solver requires solving a set of linear inequalities, as opposed to an NE solver which has to deal with a nonlinear program. Yet again, the convergence results for CE-Q are limited to zero-sum and common-interest games only. QnR [23], on the other hand, is a fully decentralized algorithm. Realizing that a no-regret algorithm can serve as a natural learning backdrop for the agents to reach CE, QnR eliminates the calls to a bulky equilibrium solver by interfacing Q-learning with no-regret-learning in a nested loop configuration. In the outer loop, the agents update their Q-values based on the empirical frequency of play that arises from the no-regret algorithm in the inner loop. The inner loop is in itself a virtual game governed by no-regret updates to which Q-values from the outer loop are fed as the agents' rewards. Hence, each iteration of the outer loop should essentially await the asymptotic convergence of the inner loop to zero-regret play probabilities. QnR's main advantage is that it works without the luxury of a CE solver, and thus the agents' Q-tables remain private. In fact, owing to the uncoupledness of the no-regret algorithm, all an agent needs to observe is its opponents' play at each stage. Hence, QnR itself can also be regarded as an uncoupled learning rule. The convergence of the algorithm, however, has not been analyzed in [23]. Moreover, it is challenging in practice to synchronize the agents for a virtual game in between two actual plays. Finally, the QnR's nested loop configuration also makes it difficult to extend the algorithm to a constrained stochastic game setup.

B. Contributions and Outline

In this paper, we take the first step toward revitalizing interest in CE-centric MARL research by revamping QnR in two ways: 1) removing its virtual game interlude and 2) extending it to also handle constrained games. In particular, we make the following contributions.

1) Realizing that no-regret and Q-learning are both variants of stochastic approximation algorithms [20], [24], we exploit the multi-timescale extension of the theory of stochastic approximation to operate QnR's inner and outer loops concurrently with two different step-size

schedules. More specifically, we recast QnR as a single-loop algorithm with no-regret learning moving on an effectively faster timescale than Q-learning. This way, we remove the virtual game interlude, while still preserving QnR's main spirit: no-regret learning sees current Q-values as quasi-static, while Q-learning sees the estimated CE policy as essentially equilibrated.

2) QnR's recast as a stochastic approximation also makes it readily extensible to constrained setups. To show this, we first exploit the methodology in [18] to view the dynamics of the constrained game through the prism of a single agent. This is done by having each agent assume all the others adhere to the policy of an imaginary correlation device so that the environment reduces to a constrained MDP (CMDP) in its eyes. Using standard Lagrange duality [25] and the one-shot deviation principle of MDPs [26], we argue how the realization of CE in stochastic game amounts to simultaneous primal maximization in all agents' CMDPs. With this understanding, we may view the coupled iterates on joint policy and Q-values as primal ascent in individual agents' CMDPs which should then be augmented by a dual descent in Lagrange multiplier (LM) space. With QnR's recast as a stochastic approximation, this augmentation can be done as easily as running stochastic sub-gradient descent on a slower third timescale. We refer to the overall algorithm as constrained no-regret Q-learning (CNRQ).
3) Given the set-valued update increments of no-regret learning and the asynchronous nature of Q-learning iterations, CNRQ would essentially correspond to a three-timescale asynchronous stochastic approximation with set-valued update increments. We give rigorous convergence results with differential inclusion arguments which draw on recent extensions of the theory of stochastic approximation to the case of asynchronous recursive inclusions with set-valued mean fields. The proof framework is due to Perkins and Leslie [27] who come up with conditions under which the asynchronicity of the process can be incorporated into the mean field to yield convergence results similar to those of an equivalent synchronous process. We verify that CNRQ in fact satisfies these conditions and thus its asymptotic analysis can be facilitated via the arguments in [27].
4) Finally, we present an example constrained stochastic game setup from the wireless networking domain. We use this example as a test bed to evaluate CNRQ's performance and convergence behavior.

The outline of the rest of this paper is as follows. In Section II, we express the formalism of constrained stochastic games, with emphasis on both individual agent-level and system-wide control problems. In Section III, we present the machinery for learning stationary CE. To this end, we remark on the connection of both Q- and no-regret learning with stochastic approximation, and highlight the main idea in QnR-learning, which paves way for the description of our CNRQ algorithm. In Section IV, we establish CNRQ's convergence. Finally, in Section V, we present numerical results for application of CNRQ-learning to an exemplary case from wireless networks. This paper concludes in Section VI.

II. Constrained General-Sum Stochastic Game

In this section, we begin with some notation and terminology that are associated with the definition of a constrained general-sum stochastic game. We then continue by formalizing the decision problem faced by each individual agent in Section II-A, and the social-level control problem in Section II-B which leads to the definition of a stationary CE. Finally, in Section II-C, we give an example embodiment of the game specification which serves both as a motivation for our algorithm in Section III and as a test bench to present numerical experiments in Section V.

A discrete-time, constrained stochastic game is denoted by a quintuple $\Gamma = \langle \mathcal{K}, \mathbf{A}, \mathcal{S}, (u_k(.))_{k\in\mathcal{K}}, (c_k(.))_{k\in\mathcal{K}}\rangle$ and can be defined as follows (see [4] for similar specifications).

1) *Agents:* The agents participating in the game are indexed by the set $\mathcal{K} = \{1, 2, \ldots, K\}$, in which $K = |\mathcal{K}|$ (i.e., the cardinality of the set $\mathcal{K}$).
2) *Actions:* We use $a_k^n \in A_k$ to denote the control action of the kth agent at time $n = 0, 1, 2, \ldots$. Let $\mathbf{a}^n = (a_1^n, \ldots, a_K^n) \in \mathbf{A}$ denote the composition of the actions from all the agents at time n, where $\mathbf{A}$ is their joint action space. Also, denote by $\mathbf{a}_{-k}^n = (a_{\acute{k}}^n)_{\acute{k}\in\mathcal{K},\acute{k}\neq k}$ the action profile of agent k's opponents at time n.
3) *States:* The stochastic system state is modeled as a discrete time Markov decision chain. We use the random variable $s^n \in \mathcal{S} = \{1, 2, \ldots, S\}$ to indicate the system state at time n. We denote by $\mathcal{P}_{s\mathbf{a}\acute{s}}$ the transition probability between states s and $\acute{s}$ under the joint action $\mathbf{a} \in \mathbf{A}$.
4) *Instantaneous Utilities:* The utility u_k^n accrued by each agent k at time n can generally be expressed by a function $u_k : \mathcal{S} \times \mathbf{A} \to e$ of both system state s^n and action profile $(a_k^n, \mathbf{a}_{-k}^n)$. e denotes a compact interval in $\mathbb{R}$.
5) *Instantaneous Constraints:* The immediate cost c_k^n incurred by each agent k at time n is specified by a function $c_k : \mathcal{S} \times \mathbf{A} \to d$ of both system state s^n and action profile $(a_k^n, \mathbf{a}_{-k}^n)$. d denotes a compact interval in $\mathbb{R}$. We specify later that the costs c_k^n are involved in a long-term discounted constraint to be satisfied by the kth agent.
6) *Stationary Randomized Joint Policies:* Since we are interested in the set of CE of the game, it is easier to abstractly assume that there is a referee (or a correlation device in game-theoretic parlance) which issues recommendations to the agents at each stage of the game. Let $\boldsymbol{\pi}_s(.)$ be the policy used by the referee to sample joint plays at state s. $\boldsymbol{\pi}_s(.)$ is defined to be stationary in that it is a randomization over the joint action space $\mathbf{A}$ given only the current state s and is independent of the history of the game. Each entry $\boldsymbol{\pi}_s(a_k, \mathbf{a}_{-k})$ represents the joint probability of taking action $a_k \in A_k$ by agent k and action profile $\mathbf{a}_{-k} \in \mathbf{A}_{-k}$ by others at state s. We denote the entire set of the referee's joint policies over all states by $\Pi = (\Delta(\mathbf{A}))^{|\mathcal{S}|}$; i.e., $\boldsymbol{\pi}_s(.) \in \Delta(\mathbf{A})$. The nth stage of the game Γ unfolds as follows: all agents and the

referee observe the system state s^n; based on its policy $\boldsymbol{\pi}_{s^n}(.)$, the referee recommends an action $a_k^{\text{ref},n}$ to each agent k. Given its recommendation, each k chooses an action a_k^n, and the joint action $\mathbf{a}^n$ is played. All agents accrue payoffs $u_k(s^n, \mathbf{a}^n)$, and incur cost $c_k(s^n, \mathbf{a}^n)$. The play proceeds to stage $(n+1)$ where s^{n+1} is determined randomly by $\mathcal{P}_{s^n\mathbf{a}^n s^{n+1}}$.

A. Individual Agent's Control Problem

Assume all other agents but k play according to the referee's policy π. Knowing $\boldsymbol{\pi}$ and given its recommended play a_k^{ref} at state $s \in \mathcal{S}$, the agent k can form *a posteriori* belief about the joint opponents' play $\mathbf{a}_{-k}$

$$\boldsymbol{\pi}_s\Big(\mathbf{a}_{-k} \;\Big|\; a_k^{\text{ref}}\Big) = \frac{\boldsymbol{\pi}_s\big(\mathbf{a}_{-k},\ a_k^{\text{ref}}\big)}{\sum_{\mathbf{b}_{-k}\in\mathbf{A}_{-k}} \boldsymbol{\pi}_s\big(\mathbf{b}_{-k},\ a_k^{\text{ref}}\big)}. \tag{1}$$

Hence, from the point of view of the kth agent, the environment reduces to a CMDP. Similarly to [18], in this MDP, we may break down the nth stage of the play (from $n = 1, 2, \ldots$ onward) as: agent k first observes the actions $\mathbf{a}_{-k}^{n-1}$ taken by its opponents in the previous round of Γ, perceives the payoff u_k^{n-1} it has accrued during the $(n-1)$st stage together with its cost constraint c_k^{n-1}. It then observes the current state s^n, receives its advice $a_k^{\text{ref},n}$ from the referee, and chooses an action a_k^n. We denote this CMDP by $\mathrm{M}_k = \langle \breve{A}_k, \breve{\mathcal{S}}_k, \breve{u}_k(.), \breve{c}_k(.)\rangle$ as follows.

1) *Actions:* $\breve{A}_k = A_k$.
2) *States:* We include in state $\breve{s}_k$ of agent k from the stochastic game Γ, the previous actions of the other agents $\mathbf{a}_{-k}$, the current state s, and the referee's advice a_k^{ref}; i.e., $\breve{\mathcal{S}}_k = \{(\mathbf{a}_{-k}, s, a_k^{\text{ref}}) \in \mathbf{A}_{-k}\times\mathcal{S}\times A_k \mid \boldsymbol{\pi}_s(a_k^{\text{ref}}) > 0\}$. The transition probabilities associated with this new state definition can be calculated as follows. Let $\breve{s}_k = (\mathbf{a}_{-k}, s, a_k^{\text{ref}})$ and $\breve{s}_k' = (\acute{\mathbf{a}}_{-k}, \acute{s}, \acute{a}_k^{\text{ref}})$. We have
$$\breve{\mathcal{P}}_{\breve{s}_k a_k \breve{s}_k'} = \boldsymbol{\pi}_s\Big(\acute{\mathbf{a}}_{-k} \;\Big|\; a_k^{\text{ref}}\Big)\cdot \mathcal{P}_{s(a_k,\acute{\mathbf{a}}_{-k})\acute{s}} \cdot \boldsymbol{\pi}_{\acute{s}}\Big(\acute{a}_k^{\text{ref}}\Big). \tag{2}$$
3) *Utility:* $\breve{u}_k(\breve{s}_k, a_k, \breve{s}_k') = u_k(s, (a_k, \acute{\mathbf{a}}_{-k}))$.
4) *Constraint:* $\breve{c}_k(\breve{s}_k, a_k, \breve{s}_k') = c_k(s, (a_k, \acute{\mathbf{a}}_{-k}))$.

Let $\breve{\pi}_{k,\breve{s}_k}(.),\ \forall \breve{s}_k \in \breve{\mathcal{S}}_k$ denote agent k's stationary policy, and consider a discount factor $\rho \in [0, 1]$. Then, k's discounted utility conditioned on initial state $\breve{s}_k \in \breve{\mathcal{S}}_k$ is defined as

$$\breve{\mathcal{U}}_{k,\breve{s}_k}(\breve{\pi}_k) \overset{\text{def}}{=} \mathbb{E}\left[(1-\rho)\sum_{n=1}^{\infty}\rho^{n-1}\breve{u}_k\Big(\breve{s}_k^n, a_k^n, \breve{s}_k^{n+1}\Big)\Big|\breve{s}_k^1 = \breve{s}_k\right] \tag{3}$$

where the normalization factor $(1-\rho)$ ensures that the range of $\breve{\mathcal{U}}_k$ falls in the compact set $e^{|\breve{\mathcal{S}}_k|} \subset \mathbb{R}^{|\breve{\mathcal{S}}_k|}$. Now, the control problem faced by the kth agent can be expressed as follows:

$$\max_{\breve{\pi}_k}\ \breve{\mathcal{U}}_{k,\breve{s}_k}(\breve{\pi}_k),\quad \forall \breve{s}_k \in \breve{\mathcal{S}}_k$$

s.t. the discounted cost constraint : $\breve{\mathcal{C}}_{k,\breve{s}_k}(\breve{\pi}_k)$

$$\overset{\text{def}}{=} \mathbb{E}\left[(1-\rho)\sum_{n=1}^{\infty}\rho^{n-1}\breve{c}_k\Big(\breve{s}_k^n, a_k^n, \breve{s}_k^{n+1}\Big)\Big|\breve{s}_k^1 = \breve{s}_k\right] \le \overline{\mathcal{D}}_k. \tag{4}$$

The constrained problem in (4) can be converted into an unconstrained form using standard Lagrangian approach [25], [28]. Let $\lambda_k \ge 0$ be a real number, called the LM. For agent k, define the instantaneous Lagrangian $\breve{\ell}_k : \mathbb{R}^+ \times \breve{\mathcal{S}}_k \times A_k \times \breve{\mathcal{S}}_k \to c$, where c is a compact interval whose boundaries can be specified from e, d, and by ensuring that λ_k is within an interval, say $[0, \mathrm{MAX}] \subset \mathbb{R}^+$. The function $\breve{\ell}_k$ is defined as

$$\breve{\ell}_k\Big(\lambda_k, \breve{s}_k, a_k, \acute{\breve{s}}_k\Big) \overset{\text{def}}{=} \breve{u}_k\Big(\breve{s}_k, a_k, \acute{\breve{s}}_k\Big) - \lambda_k\Big(\breve{c}_k\Big(\breve{s}_k, a_k, \acute{\breve{s}}_k\Big) - \overline{\mathcal{D}}_k\Big). \tag{5}$$

For $\forall \breve{s}_k \in \breve{\mathcal{S}}_k$, the expected total discounted Lagrangian associated with (5) is as follows:

$$\begin{aligned}\breve{\mathcal{L}}_{k,\breve{s}_k}^{\lambda_k}(\breve{\pi}_k) &\overset{\text{def}}{=} \breve{\mathcal{U}}_{k,\breve{s}_k}(\breve{\pi}_k) - \lambda_k\Big[\breve{\mathcal{C}}_{k,\breve{s}_k}(\breve{\pi}_k) - \overline{\mathcal{D}}_k\Big] \\ &= \mathbb{E}\left[(1-\rho)\sum_{n=1}^{\infty}\rho^{n-1}\breve{\ell}_k\Big(\lambda_k, \breve{s}_k^n, a_k^n, \breve{s}_k^{n+1}\Big)\Big|\breve{s}_k^1 = \breve{s}_k\right].\end{aligned} \tag{6}$$

The unconstrained counterpart to (4) is to determine the optimal pair $(\breve{\pi}_k^*, \lambda_k^*)$ such that the following saddle point optimality condition holds for $\forall \breve{s}_k \in \breve{\mathcal{S}}_k$ [28]:

$$\breve{\mathcal{L}}_{k,\breve{s}_k}^{\lambda_k^*}(\breve{\pi}_k) \le \breve{\mathcal{L}}_{k,\breve{s}_k}^{\lambda_k^*}(\breve{\pi}_k^*) \le \breve{\mathcal{L}}_{k,\breve{s}_k}^{\lambda_k}(\breve{\pi}_k^*). \tag{7}$$

With (7) satisfied, $\breve{\mathcal{L}}_{k,\breve{s}_k}^{\lambda_k^*}(\breve{\pi}_k^*)$ is the optimal value of the problem (4), and it can be computed as [28]

$$\breve{\mathcal{L}}_{k,\breve{s}_k}^{\lambda_k^*}(\breve{\pi}_k^*) = \min_{\lambda_k\ge 0}\max_{\breve{\pi}_k}\breve{\mathcal{L}}_{k,\breve{s}_k}^{\lambda_k}(\breve{\pi}_k),\ \ \forall \breve{s}_k \in \breve{\mathcal{S}}_k. \tag{8}$$

However, in the setup described by Γ, the maximization in (8) is solved concurrently by all agents, which undermines our simplifying single-agent abstraction. Next, we introduce a system-wide objective, which, when realized, amounts to $\breve{\mathcal{L}}_{k,\breve{s}_k}^{\lambda_k}(\breve{\pi}_k)$ being maximized simultaneously for all $k \in \mathcal{K}$.

B. System-Wide Objective: Stationary Correlated Equilibria

Before giving a formal definition of stationary CE, we first express the long-term discounted Lagrangian of agent k under the assumption that all agents (including k) follow the recommendations from a given referee's policy $\boldsymbol{\pi}$. Let $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_K]^T$ be a fixed vector of LMs for $\forall k \in \mathcal{K}$. Similarly to Π, define $\Pi^{\boldsymbol{\lambda}}$ as the set of all stationary joint policies for the unconstrained version of the game Γ with λ_k-parameterized individual Lagrangian utilities (denoted by $\Gamma^{\boldsymbol{\lambda}}$ for easier reference). We have for $\forall s \in \mathcal{S}$

$$\mathcal{L}_{k,s}^{\lambda_k}(\boldsymbol{\pi}) = \mathbb{E}\left[(1-\rho)\sum_{n=0}^{\infty}\rho^n\ell_k\big(\lambda_k, s^n, \mathbf{a}^{\mathbf{n}}\big)\big|s^0 = s\right] \tag{9}$$

where $\ell_k(\lambda_k, s, \mathbf{a}) \overset{\text{def}}{=} u_k(s, \mathbf{a}) - \lambda_k(c_k(s, \mathbf{a}) - \overline{\mathcal{D}}_k)$. It is well-known that $\mathcal{L}_{k,s}^{\lambda_k}$ has the following standard dynamic programming expansion (also known as Bellman equations):

$$\mathcal{L}_{k,s}^{\lambda_k}\Big(Q_k^{\lambda_k}, \boldsymbol{\pi}\Big) = \sum_{\mathbf{a}\in\mathcal{A}}\boldsymbol{\pi}_s(\mathbf{a})\cdot Q_{k,(s,\mathbf{a})}^{\lambda_k},\qquad \forall s \in \mathcal{S} \tag{10}$$

where $\mathcal{L}_{k,s}^{\lambda_k}$ is defined with an abuse of notation by making its dependence on $Q_k^{\lambda_k}$ explicit. $Q_k^{\lambda_k} : c^{|\mathcal{S}|} \to c^{|\mathcal{S}\times\mathbf{A}|}$ is a $|\mathcal{S}\times\mathbf{A}|$-dimensional λ_k-parameterized mapping whose $(s, \mathbf{a})$th component evaluated at $\mathcal{L}_k^{\lambda_k}$ is defined as

$$Q_{k,(s,\mathbf{a})}^{\lambda_k}\left(\mathcal{L}_k^{\lambda_k}\right) = (1-\rho).\mathbb{E}[\ell_k(\lambda_k, s, \mathbf{a})] + \rho \sum_{\acute{s}\in\mathcal{S}} \mathcal{P}_{s\mathbf{a}\acute{s}} \mathcal{L}_{k,\acute{s}}^{\lambda_k}\left(Q_k^{\lambda_k}, \boldsymbol{\pi}\right). \quad (11)$$

Clearly, $Q_k^{\lambda_k}$ is an affine function of $\mathcal{L}_k^{\lambda_k}$, and the value function $\mathcal{L}_k^{\lambda_k} : c^{|\mathcal{S}\times\mathbf{A}|} \times \Pi^{\boldsymbol{\lambda}} \to c^{|\mathcal{S}|}$ is a bilinear function of the policy $\boldsymbol{\pi}$ and action value function $Q_k^{\lambda_k}$. The dependence of $\mathcal{L}_k^{\lambda_k}$ on the policy as well as the interdependence of $\mathcal{L}_k^{\lambda_k}$ and $Q_k^{\lambda_k}$ is made explicit only on few occasions for emphasis. This dependence is otherwise suppressed to simplify notation. Now, we are ready to define $\Gamma^{\boldsymbol{\lambda}}$'s set of stationary CE.

Definition 1: The set $\mathcal{C}_{\text{ce}}^{\boldsymbol{\lambda}} \subset \Pi^{\boldsymbol{\lambda}}$ is called the set of stationary CE of the stochastic game $\Gamma^{\boldsymbol{\lambda}}$ if under each $\boldsymbol{\pi}^{\text{ce}} \in \mathcal{C}_{\text{ce}}^{\boldsymbol{\lambda}}$, it holds that for each agent k, for $\forall s \in \mathcal{S}$, for $\forall a_k^{\text{ref}} \in A_k$ with $\boldsymbol{\pi}_s^{\text{ce}}(a_k^{\text{ref}}) > 0$, and any alternative action $\acute{a}_k \in A_k$

$$\sum_{\mathbf{a}_{-k}\in\,\mathbf{A}_{-k}} \boldsymbol{\pi}_s^{\text{ce}}\left(\mathbf{a}_{-k} \,\middle|\, a_k^{\text{ref}}\right) \cdot Q_{k,\left(s,\left(a_k^{\text{ref}},\mathbf{a}_{-k}\right)\right)}^{\lambda_k} \geq \sum_{\mathbf{a}_{-k}\in\,\mathbf{A}_{-k}} \boldsymbol{\pi}_s^{\text{ce}}\left(\mathbf{a}_{-k} \,\middle|\, a_k^{\text{ref}}\right) \cdot Q_{k,\left(s,\left(\acute{a}_k,\mathbf{a}_{-k}\right)\right)}^{\lambda_k}. \quad (12)$$

The inequality in (12) can be better understood if we intuitively consider $\Gamma^{\boldsymbol{\lambda}}$ as a set of auxiliary normal-form games indexed by $s \in \mathcal{S}$ and with payoffs $Q_{k,(s,\mathbf{a})}^{\lambda_k}$ (see [1], [5]). By playing joint action $\mathbf{a}$ in the sth auxiliary game, agent k's payoff is the sum of its instantaneous payoff and the payoff it expects to gain from the next state onward, assuming joint policy $\boldsymbol{\pi}$. Now, $\boldsymbol{\pi}^{\text{ce}} \in \mathcal{C}_{\text{ce}}^{\boldsymbol{\lambda}}$ if and only if it is simultaneously a CE for all auxiliary games $s \in \mathcal{S}$; i.e., if the referee draws its actions from $\boldsymbol{\pi}^{\text{ce}}$, k realizes that every recommendation a_k^{ref} it receives in each game $s \in \mathcal{S}$ is a best response to the estimated play of the other agents (assuming they all follow their recommendations). Now, we relate this collective notion with the agent-level objectives through the following theorem.

Theorem 1: For $\forall k \in \mathcal{K}$, $\forall \check{s}_k \in \check{\mathcal{S}}_k$, it holds that: $\check{\mathcal{L}}_{k,\check{s}_k}^{\lambda_k}(\check{\pi}_k^*) = \mathcal{L}_{k,s}^{\lambda_k}(Q_k^{\lambda_k}, \boldsymbol{\pi}^{\text{ce}})$.

Proof: As argued in [18, Th. 7], if all other agents but k play according to the referee's policy $\boldsymbol{\pi}^{\text{ce}}$, then from the point of view of the kth agent, the environment reduces to MDP M_k, defined in Section II-A. By construction in [18], based on the one-shot deviation principle for MDPs [26], the referee's policy $\boldsymbol{\pi}^{\text{ce}}$ is a CE in the stochastic game if and only if its implementation in M_k is an optimal policy simultaneously for all $k \in \mathcal{K}$. It then follows that the expected discounted Lagrangian of all agents under the CE policy $\boldsymbol{\pi}^{\text{ce}}$ is equal to the expected discounted Lagrangian of the corresponding optimal policy in their MDPs. ■

Now define Lagrange dual function $\mathcal{G}_k(\lambda_k) \overset{\text{def}}{=} \mathcal{L}_k^{\lambda_k}(Q_k^{\lambda_k}, \boldsymbol{\pi}^{\text{ce}})$ as the solution of the primal problem $\max_{\check{\pi}_k} \check{\mathcal{L}}_{k,\check{s}_k}^{\lambda_k}(\check{\pi}_k)$ for $\forall \check{s}_k \in \check{\mathcal{S}}_k$ in (8). The optimal λ_k^* can then be obtained by conducting dual descent on $\mathcal{G}_{k,s}(\lambda_k)$ for $\forall s \in \mathcal{S}$. In Section III, we present a distributed learning procedure to compute $\boldsymbol{\pi}^{\text{ce}}$ together with the optimal λ_k^* for $\forall k \in \mathcal{K}$.

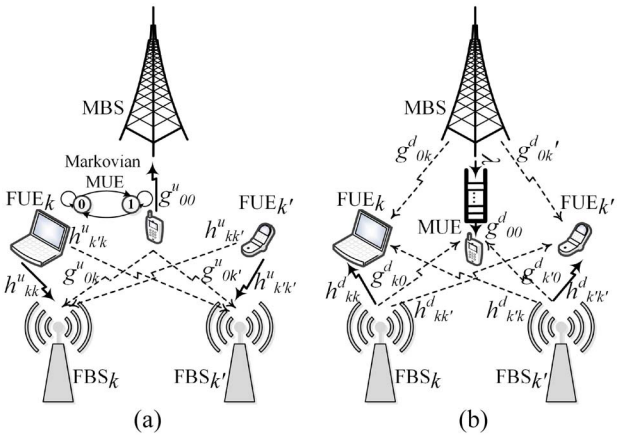

Fig. 1. Example constrained stochastic game setup. Decentralized resource control in two-tier small-cell networks. (a) Uplink. (b) Downlink.

C. Illustrative Example

Before delving into the technicalities of learning a stationary CE, we give an illustrative example as typical real-world problems that can be modeled by the generic game described above. This example is a simplified yet an illustrative scenario from the domain of wireless networks which also provides a test bench to demonstrate both convergence behavior as well as efficacy of the algorithm discussed in Section III. The setting we consider is the resource control problem in hierarchical small-cell networks, more generally known as heterogeneous networks (HetNets) [29]. HetNets are wireless deployments where small cells (e.g., femto-cells) with lower signal power are positioned within the coverage area of a macro-cell primarily to multiply the capacity of this area. Traffic steering and load balancing are key aspects of HetNets as femto-cells can be installed in hotspots to offload much of the traffic from the macro layer. As envisioned in [30], HetNets will be integral to wireless deployments in near future. However, the coexistence of macro and small cell elements is not without ramifications. As argued in [30], HetNet topologies call for a new approach to run networks that is more complex, that requires a higher level of automation and more sophisticated resource control. Our examples in Sections II-C1 and II-C2 entail uplink and downlink HetNet communication scenarios, respectively.

1) Spectrum Access Control in Uplink Communications: Consider a two-tier CDMA femto-cell network [see Fig. 1(a)]. It is assumed that the system consists of a single macro-cell base-station (MBS) receiving data from macro-user equipments (MUEs) in a region. Within this region, there are also K co-channel femto-cells deployed by home or office users on the same frequency band (with bandwidth W) as the macro-cell. In each femto-cell, there is one femto base-station (FBS) receiving data from a number of femto user equipments (FUEs). For simplicity, only one active FUE is assumed in each cell. Let $h_{\acute{k}k}^u$ denote the gain of the link between FUE k and FBS $\acute{k}$. Also, N_o denotes the noise

power on all channels. Each FUE seeks to maximize its own transmission rate which, by Shannon–Hartley's theorem (see [31]), depends on its received signal-to-interference plus noise ratio (SINR). We assume each FUE only gets to decide how aggressively (high/low in terms of power) it should transmit its signal; i.e., $a_k^{u,n} \in A_k = \{\text{Low}, \text{High}\}$. Due to the shared nature of the wireless channel, each FUE's perceived SINR depends not only on its own action but also on the actions of other FUEs. An FUE transmitting at a high power level, though may increase its own SINR, will interfere with the transmissions of the other FUEs, prompting them in turn to adopt a more aggressive behavior. Such a situation is undesirable since FUEs usually operate on limited batteries which require judicious consumption. In fact, the immediate cost $c_k^{u,n}$ incurred by each FUE k from choosing action $a_k^{u,n}$ is its consumed power, i.e., $c_k^{u,n} = a_k^{u,n}$, with the restriction that the average power consumption over time should not exceed a prespecified constraint $\bar{a}_k^u$. Moreover, given the two-tier structure of our setup, the activity of the MUEs is yet another source of interference, causing FUEs' signals to be further attenuated at their FBSs. Let g_{0k}^u denote the channel gain between MUE and FBS k. MUE's interference activity over the shared channel is typically modeled as a time-homogenous discrete time Markov chain (see [32]). We use the binary random variable $s^n \in \mathcal{S} = \{0, 1\}$ to indicate the macro activity at time n; i.e., $s^n = 1$ if the channel is occupied, in which case the interference power sensed at kth FBS would be: $g_{0k}^u \cdot a_0$, where a_0 denotes the MUE's transmit power, and g_{0k}^u denotes the gain of the link between MUE and FBS k. Also, $s^n = 0$, if the channel is idle. Hence, the uplink spectrum access control problem gives rise to a setting which is both strategic and stochastic. It is strategic since the FUEs' objectives are coupled due to mutual interference, and it is stochastic because FUEs' decisions have to be made under the effect of MUEs' Markovian dynamics. In this scenario, the utility $u_k^{u,n}$ accrued by each FUE k at time n is its instantaneous Shannon rate

$$u_k^{u,n} = u_k^u\left(s^n, a_k^{u,n}, \mathbf{a}_{-k}^{u,n}\right) = W \cdot \log_2\left[1 + \frac{a_k^{u,n}.h_{kk}^u}{No + \mathbb{I}_{\{s^n\}}.g_{0k}^u.a_0^u + \sum_{\acute{k}\in\mathcal{K},\acute{k}\neq k} a_{\acute{k}}^{u,n}.h_{\acute{k}k}^u}\right]. \tag{13}$$

Each FUE seeks a policy which maximizes its long-run rate utility subject to its power constraint. At the collective, social-level, it is desired to coordinate FUEs' decisions by striking a CE-based consensus. Our algorithm in Section III-B3 allows FUEs to reach this consensus based only on their instantaneous rate and power consumption as feedbacks.

2) Power Control in Downlink Communications: In the same topology, consider the reverse scenario of downlink transmissions from MBS to its MUE and from FBSs down to their associated FUEs [Fig. 1(b)]. We assume that MBS transmits at a constant power a_0^d, while each FBS chooses its power $a_k^{d,n}$ from a finite set of power levels. Let $h_{k\acute{k}}^d$ denote the gain of the link between FBS k and FUE $\acute{k}$; likewise, $\{g_{0k}^d\}_{k\in\mathcal{K}}$ (resp. $\{g_{k0}^d\}_{k\in\mathcal{K}}$) denotes MBS-FUE (resp., FBS-MUE) channel gains. Consistent with the common characterization of femto entities as best effort users, the traffic in FBS is assumed to be backlogged, while it is bursty and stochastic in MBS. Let $\mathcal{A}^n$ be the random number of packets arrived in the nth timeslot to MBS's buffer whose capacity is capped by N_B packets. The process $\{\mathcal{A}^n\}_{n\in\mathbb{N}}$ is assumed to be i.i.d. with general distribution $\mathbb{P}\{\mathcal{A}\}$ and mean $\mathbb{E}[\mathcal{A}]$. By Shannon's law, MBS's achievable bit rate is given below

$$r_0^n = W \cdot \log_2\left[1 + \frac{a_0^d \cdot g_{00}^d}{No + \sum_{k\in\mathcal{K}} a_k^{d,n} \cdot g_{k0}^d}\right]. \tag{14}$$

Accordingly, the evolution of the system state (i.e., the buffer length in MBS) can be described as follows:

$$b_0^{n+1} = \min\left(\left(b_0^n - \frac{\tau.r_0^n}{L}\right)^+ + \mathcal{A}^n, N_B\right) \tag{15}$$

where τ denotes the timeslot duration, L is the packet length in bits, and $(.)^+$ stands for $\max(., 0)$. In this game, FBS agents are interested in maximizing their expected physical throughput (16) with the restriction that their interference to the macro layer be low enough so that the expected length of MBS's buffer remains below a certain threshold $\bar{b}_0$

$$u_k^{d,n} = u_k^d\left(s^n, a_k^{d,n}, \mathbf{a}_{-k}^{d,n}\right) = W \cdot \log_2\left[1 + \frac{a_k^{d,n} \cdot h_{kk}^d}{No + g_{0k}^d \cdot a_0^d + \sum_{\acute{k}\in\mathcal{K},\acute{k}\neq k} a_{\acute{k}}^{d,n} \cdot h_{\acute{k}k}^d}\right]. \tag{16}$$

Again, it is desired that FBSs learn a stationary CE behavior by only receiving instantaneous feedbacks on their own rate $u_k^{d,n}$ and on MBS's buffer occupancy state b_0^n.

III. Learning Stationary Correlated Equilibria

As with the case of MDPs, the fundamental update procedure for learning a policy can be derived from operationalizing Bellman equations in (10) and (11). For now, consider an unconstrained game, and imagine a centralized entity iteratively running the update equations below, for all $k \in \mathcal{K}$, for all $s \in \mathcal{S}$, and for all $\mathbf{a} \in \mathbf{A}(s)$

$$\hat{V}_{k,s}^{n+1} \coloneqq \sum_{\mathbf{a}\in\mathbf{A}(s)} \widehat{\boldsymbol{\pi}}_s^n(\mathbf{a}) \hat{Q}_{k,(s,\mathbf{a})}^n \tag{17}$$

$$\hat{Q}_{k,(s,\mathbf{a})}^{n+1} \coloneqq (1-\gamma).u_k(s,\mathbf{a}) + \gamma. \sum_{\acute{s}\in\mathcal{S}} \mathcal{P}_{s\mathbf{a}\acute{s}} \hat{V}_{k,\acute{s}}^{n+1} \tag{18}$$

$$\widehat{\boldsymbol{\pi}}_s^{n+1} \in \Pi^{\text{ce}}\left(\left\{\hat{Q}_{k,(s,.)}^{n+1}\right\}_{k\in\mathcal{K}}\right) \tag{19}$$

where Π^{ce} returns the set of all policies satisfying Definition 1. The first step to make this procedure more practical is to do without assuming that the matrix $\mathcal{P}$ of transition probabilities is known *a priori*. This is of particular interest in applications where the statistical knowledge regarding the processes underlying the system evolution is not available beforehand. In the context of our example from Section II-C, this corresponds to FUEs having to reach a consensus with no knowledge of the MUE's stochastic occupancy behavior, or FBSs adjusting their power levels without knowing the statistics of the

TABLE I
SKETCH OF THE CE-Q ALGORITHM [22]

Loop:
1. Execute action a_k in state s;
2. Observe joint agents' play $\boldsymbol{a}_{-k}$;
3. Observe own instantaneous reward $r_k(s, a_k, \boldsymbol{a}_{-k})$ as well as the reward $r_{\acute{k}}(.)$ for each agent $\acute{k} \in \mathcal{K}\backslash\{k\}$;
4. Observe next state $\acute{s}$;
5. Choose a CE policy $\boldsymbol{\pi}^{CE}_{\acute{s}} \in \Pi^{ce}\left(\left\{\hat{Q}_{k,(\acute{s},.)}\right\}_{k\in\mathcal{K}}\right)$ via solving the system of linear inequalities associated with CE;
6. for each agent $k \in \mathcal{K}$ **do**
6.1. Update estimate for the value of the next state:
$$\hat{V}_{k,\acute{s}} := \sum_{\boldsymbol{b}\in\boldsymbol{A}} \boldsymbol{\pi}^{CE}_{\acute{s}}(\boldsymbol{b})\hat{Q}^{n}_{k,(\acute{s},\boldsymbol{b})};$$
6.2. Update estimate for Q-value of current state-action pair:
$$\hat{Q}_{k,(s,\boldsymbol{a})} := \hat{Q}_{k,(s,\boldsymbol{a})} + \alpha\left[(1-\rho)\, r_k(s,\boldsymbol{a}) + \rho\hat{V}_{k,\acute{s}} - \hat{Q}_{k,(s,\boldsymbol{a})}\right];$$
7. Choose action $\acute{a}_k$ (off-policy) and set $a_k := \acute{a}_k$;
8. Update $s := \acute{s}$; Go to 1;

TABLE II
SKETCH OF THE QNR ALGORITHM [23]

Outer Loop (on-policy Q-learning):
1. Execute action a_k and observe joint agents' play $\boldsymbol{a}_{-k}$;
2. Update joint empirical frequency of play $\hat{\boldsymbol{\pi}}_s(\boldsymbol{a})$;
3. Observe instantaneous reward $r_k(s, a_k, \boldsymbol{a}_{-k})$ and next state $\acute{s}$;
4. Update estimate for the value of the next state:
$$\hat{V}_{k,\acute{s}} := \sum_{\boldsymbol{a}\in\boldsymbol{A}} \hat{\boldsymbol{\pi}}_{\acute{s}}(\boldsymbol{a}).\hat{Q}_{k,(\acute{s},\boldsymbol{a})};$$
$$\hat{U}_k(s,a_k) := (1-\rho)r_k(s,\boldsymbol{a}) + \rho\hat{V}_{k,\acute{s}};$$
5. Update estimate for Q-value of current state-action pair:
$$\hat{Q}_{k,(s,\boldsymbol{a})} := \hat{Q}_{k,(s,\boldsymbol{a})} + \alpha\left[\hat{U}_k(s,a_k) - \hat{Q}_{k,(s,\boldsymbol{a})}\right];$$
6. Inner Loop (no-regret learning):
for $m = 0,1,2,\ldots,M-1$ **do**
6.1. Use (20) to update regret matrix $R^{m+1}_{k,\langle .,.\rangle,s}$ for state s with $\hat{U}_k(s,.)$ as payoffs;
6.2. Update the action transition probability matrix $T^{m+1}_{k,s}$;
6.3. Use (21) to compute play probabilities $p^{m+1}_{k,s}$;
6.4. Update average distribution of play:
$$\bar{p}^{m+1}_{k,s} := \frac{1}{m+1}\sum_{\tau=1}^{m+1} p^{\tau}_{k,s};$$
7. Choose $\acute{a}_k$ by an ε-soft version of $\bar{p}^{M}_{k,\acute{s}}(\acute{a}_k)$, and set $a_k := \acute{a}_k$;
8. Update $s := \acute{s}$; Go to 1;

packet arrival process in MBS (e.g., $\mathbb{E}[\mathcal{A}]$). The standard way to tackle the case of unknown $\mathcal{P}$ is to adopt an asynchronous variant of (18) with a decaying step size, better known as Q-learning. With this modification, the learning task would proceed by simulating a joint action, actually observing the next state, and run (18) for one state-action pair per learning iteration (see Table I for a procedure of this spirit).

The second step toward practice is decentralization so that each agent runs its local version of the above process. A naive decentralization, however, is subject to possible miscoordination in the equilibrium selection step in (19). We discuss the known remedies for this issue as we review the existing CE learning processes in the sequel. Another issue concerns the extent of knowledge an agent is assumed to have about its opponents. In fact, one can distinguish between coupled and uncoupled equilibrium learning processes [21]. In coupled learning, agents must know the utilities of their opponents [e.g., the rates in (13) and (16)]; uncoupled learning, however, is more practical as it proceeds without that luxury.

In this section, we first review the existing ideas for learning stationary CE. We begin by the coupled algorithm of CE-Q [22], and highlight its shortcomings. Then, we turn to uncoupled QnR learning [23], which prepares the ground for presentation of our CNRQ algorithm, a provably convergent, constrained, and single-loop recast of QnR.

A. Existing Procedures for Learning CE in Stochastic Games

CE-Q [22] and QnR [23] are the only algorithms we know of that address the problem of learning stationary CE. At each iteration, both CE-Q and QnR use Q-learning to update the Q-values based on the estimated CE policy for the next state. However, when it comes to update the CE policy itself, CE-Q and QnR differ significantly. In what follows, we briefly discuss the idea utilized in each algorithm for estimating CE and highlight their shortcomings.

1) CE-Q Learning: Much in the same way as the basic update rules in (17)–(19), in CE-Q [22], the estimate for the CE policy is obtained by solving the system of linear inequalities corresponding to the definition of CE (step 5 in Table I). Therefore, each agent k is assumed to observe the rewards of all others and to maintain a model of their Q tables. This requirement makes CE-Q a coupled learning procedure. For ease of reference, we call this version of CE-Q as semi-distributed. Also, as argued in [22], in the presence of multiple equilibria, semi-distributed CE-Q is subject to mis-coordination in the equilibrium selection step. This problem has been alleviated in [22] by introducing some equilibrium selection mechanisms. For instance, a utilitarian selector chooses an equilibrium which maximizes the sum of all agents' Q-values; however, except in very special games (e.g., zero-sum), CE-Q in general needs that the play be centralized. We refer to this version of CE-Q as centralized CE-Q learning.

2) QnR Learning: The QnR algorithm in [23] eliminates the need for calls to an equilibrium solver. Instead, each agent relies on a no-regret learning algorithm to independently generate its own policy. Agents play according to their own policies, and compute their respective value functions based on the joint empirical distribution of play. This approach is theoretically sound since in the context of normal-form games, no-regret algorithms converge in empirical frequency to CE [19]. As we also rely on a no-regret procedure for our algorithm in Section III-B, we first briefly review the idea of no-regret learning, and then present QnR's pseudo-code in Table II.

Consider a normal form game with payoff functions $(r_k(.))_{k\in\mathcal{K}}$. In no-regret learning [19], the agents reinforce the actions they regret not having played enough in the past. In particular, each agent k has a regret matrix $R_{k,\langle i,j\rangle}$ which maintains, for every pair of actions $i, j \in A_k$, the difference in average payoff if k had taken action j in the past every time

it took action i; that is

$$R^n_{k,\langle i,j\rangle} = \frac{1}{n}\sum_{\tau=1}^{n}\left[r_k\left(j, \mathbf{a}^\tau_{-k}\right) - r_k\left(i, \mathbf{a}^\tau_{-k}\right)\right]\cdot \mathbb{I}_{\{a^\tau_k = i\}}. \quad (20)$$

In $(n+1)$st round, given $a^n_k = i$, agent k transitions to action $a^{n+1}_k = j$ with a probability $T^{n+1}_k(a^{n+1}_k = j | a^n_k = i)$ proportional to $R^n_{k,\langle i,j\rangle}$, and sticks to i with $1 - \sum_{j\in A_k, j\neq i} T^{n+1}_k(a^{n+1}_k = j | a^n_k = i)$. In fact, the play probabilities p^{n+1}_k for the next stage $(n+1)$ are obtained by solving the following balance equations:

$$p^{n+1}_k(i)\sum_{j\neq i} T^{n+1}_{k,\langle i,j\rangle} = \sum_{j\neq i} p^{n+1}_k(j).T^{n+1}_{k,\langle j,i\rangle}. \quad (21)$$

The learning proceeds by exploring choices and transitioning to actions which are conceived better according to the regret measure. Naturally, an agent's objective is to select a sequence of actions which guarantees to it no regret in the long run, no matter what the other agents do. Let $z^n(\mathbf{a})$ be the number of times the joint action profile $\mathbf{a}$ is actually played in the first n periods, divided by n. In fact, $z^n(\mathbf{a})$ denotes the empirical distribution of play and is a probability distribution over $\mathbf{A}$. The no-regret learning of [19] has the property that when all agents' regret matrices approach to the nonpositive orthant $\mathbb{R}^{|A_k\times A_k|}_-$, z^n converges to the set of CE. Now, in QnR [23], in order to utilize the idea of no-regret learning in the context of stochastic games, each agent k runs two nested control loops: 1) Q-learning as the outer loop and 2) multiple copies of the no-regret algorithm (one per state) as the inner loop (see Table II). At each outer loop iteration, the sth copy of the no-regret algorithm of the inner loop starts afresh, fed by the current estimate of Q-values as k's payoff function. The inner loop iterates until the agent's regret matrix $R^m_{k,s}$ converges to the nonpositive orthant. Once inner loop converges, the outer loop begins its next iteration knowing that the joint empirical frequency of play for state s corresponds to a CE of the game in s. QnR learning goes on until all Q-table entries converge.

QnR's advantage is that it works without requiring an equilibrium solver, and that the agents need not know their opponents' rewards to update their play strategies. Hence, QnR falls into the category of boundedly rational uncoupled learning dynamics [21]. QnR's main disadvantage, however, is its nested loop structure. This not only makes it difficult to conduct a theoretical convergence analysis, but has some practical implications too. First, the virtual play in the inner loop, apart from being an interlude in the actual game, would require that the agents agree on a second iteration index during the learning process. The other drawback is with the extension of this paradigm to handle constrained problems, which leads to a third control loop and even more complications.

B. Proposed Algorithm

In this section, we present a stochastic approximation-based re-expression of QnR which handles constrained games and, more importantly, is amenable to convergence analysis. As discussed in Section III-A2, QnR relies on the joint operation of no-regret and Q-learning working together in a nested loop configuration. Since both of these algorithms can be expressed in the form of a typical stochastic approximation [24], [33], [34], we first very briefly remark on some general forms of stochastic approximation algorithms in Section III-B1, and then highlight the connection of no-regret- and Q-learning with relevant notions from the theory of stochastic approximation in Section III-B2. Finally, in Section III-B3, we give our version of things, referred to as CNRQ-learning.

1) Some General Forms of Stochastic Approximation Algorithms: Let $\mathcal{J} = \{1, \ldots, |\mathcal{J}|\}$. A general stochastic recursive process has the following structure [34]:

$$x^{n+1} - x^n = \kappa(n)\cdot f\left(x^n, v^n\right) \quad (22)$$

where $x^n \in \mathbb{R}^{|\mathcal{J}|}, f(x^n, v^n) : \mathbb{R}^{|\mathcal{J}|}\times\mathbb{R}^{|\mathcal{J}|} \to \mathbb{R}^{|\mathcal{J}|}$, $v_n \in \mathbb{R}^{|\mathcal{J}|}$ is a random noise, and $\{\kappa(n)\}_{n\in\mathbb{N}}$ is a sequence of small, usually decreasing step-sizes. It is common to capture the noise effect as an additive term, by introducing: $F(x^n) = \mathbb{E}_v[f(x^n, v^n)]$, and $\mathcal{V}^{n+1} = f(x^n, v^n) - \mathbb{E}_v[f(x^n, v^n)]$, where $F(x^n)$ is referred to as the mean field, and $\{\mathcal{V}^n\}_{n\in\mathbb{N}}$ is, by construction, a martingale difference sequence. In cases where the mean-field F is a set-valued map (correspondence), we refer to the iteration above as a stochastic approximation with set-valued update increments or more concisely as a stochastic recursive inclusion [33]

$$x^{n+1} - x^n \in \kappa(n)\cdot\left[F\left(x^n\right) + \mathcal{V}^{n+1}\right].$$

Finally, let $2^{\mathcal{J}}$ be the power set of $\mathcal{J}$. If we denote by $\overline{J}^n \in 2^{\mathcal{J}}$ the components of the iterates $\{x^n\}_{n\in\mathbb{N}}$ updated at iteration n, we may use a counter $\chi^n(j) = \sum_{i=1}^n \mathbb{I}_{\{j\in\overline{J}^i\}}$ to record how many times each component of $\{x^n\}_{n\in\mathbb{N}}$ have been updated until n. The following process, then, is called an asynchronous stochastic approximation [33] since it is no longer the case that all components of x^n get updated simultaneously at time n:

$$x^{n+1}_j - x^n_j = \kappa\left(\chi^n(j)\right)\cdot\mathbb{I}_{\{j\in\overline{J}^n\}}\left[F_j\left(x^n\right) + \mathcal{V}^{n+1}_j\right].$$

2) Q-Learning and No-Regret Learning as Stochastic Approximations: Fix $\boldsymbol{\pi}\in\Pi$ as a stationary randomized policy over the joint action space $\mathbf{A}$. The function $Q_{k,(s,\mathbf{a})} \overset{\text{def}}{=} (1-\rho)\cdot\mathbb{E}[r_k(s,\mathbf{a})] + \rho\sum_{\acute{s}\in\mathcal{S}}\mathcal{P}_{s\mathbf{a}\acute{s}}V_{k,\acute{s}}(\hat{Q}_k, \boldsymbol{\pi})$ is the expected long-term value of taking action $\mathbf{a}$ in state s, and following $\boldsymbol{\pi}$ thereafter. To learn this value without having to know about $\mathcal{P}$, one can use the Q-learning algorithm [11]. Due to [24], the exact form of the Q-learning update equation (e.g., step 5 in Table II) is given below

$$\hat{Q}^{n+1}_{k,(s,\mathbf{a})} - \hat{Q}^n_{k,(s,\mathbf{a})} \coloneqq \kappa\left(\chi^n(s,\mathbf{a})\right)\cdot\mathbb{I}_{\{(s,\mathbf{a})=(s^n,\mathbf{a}^n)\}} \times\left[F_{(s,\mathbf{a})}\left(\hat{Q}^n_k, \boldsymbol{\pi}\right) + \mathcal{V}^{n+1}_{k,(s,\mathbf{a})}\right] \quad (23)$$

with mean-field $F_{(s,\mathbf{a})}(\hat{Q}^n_k) = Q_{k,(s,\mathbf{a})} - \hat{Q}^n_{k,(s,\mathbf{a})}$, noise $\mathcal{V}^{n+1}_{(s,\mathbf{a})} = (1-\rho)\cdot r_k(s,\mathbf{a}) + \rho.V^n_{k,s^{n+1}} - Q_{k,(s,\mathbf{a})}$, and $V^n_{k,s} = \sum_{\mathbf{a}}\pi_s(\mathbf{a})\cdot\hat{Q}^n_{k,(s,\mathbf{a})}$. Of particular note in (23) is the use of asynchronous counters $\chi^n(s,\mathbf{a}) \overset{\text{def}}{=} \sum_{i=1}^n \mathbb{I}_{\{(s^i,\mathbf{a}^i)=(s,\mathbf{a})\}}$. Such counters are needed as the time to visit each $(s,\mathbf{a})$ is random, and we might not have complete control over which component is

to be updated next. Hence, by structure, Q-learning is an asynchronous stochastic approximation.

As for no-regret learning, it is noted in [20] that (20) can be cast as a moving average with step size $\kappa(n) = 1/n$

$$R_k^{n+1} - R_k^n = \kappa(n) \cdot \Big(\big[r_k\big(j, \mathbf{a}_{-k}^n\big) - r_k\big(i, \mathbf{a}_{-k}^n\big) \big] \cdot \mathbb{I}_{\{a_k^n = i\}} - R_k^n \Big). \tag{24}$$

Also, it has been shown in [20] that the above equation has the following correspondence $F(R_k^n)$ as its mean field:

$$F\big(R_k^n\big) = C_k\big(p_k^n \times \Delta(\mathbf{A}_{-k})\big) - R_k^n \tag{25}$$

and, for $x \in \Delta(\mathbf{A})$, $C_k(x)$ is a $|A_k| \times |A_k|$ matrix with entries

$$C_{k,\langle i,j\rangle}(x) \stackrel{\text{def}}{=} \sum_{\mathbf{a} \in \mathbf{A}: a_k = i} x(\mathbf{a}) \cdot \big[r_k(j, \mathbf{a}_{-k}) - r_k(i, \mathbf{a}_{-k}) \big]. \tag{26}$$

Hence, the regret update procedure can be rewritten as a stochastic recursive inclusion of the form

$$R_k^{n+1} - R_k^n \in \kappa(n) \cdot \Big(F\big(R_k^n\big) + \mathcal{V}_k^{n+1} \Big) \tag{27}$$

where the random noise term $\mathcal{V}_k^{n+1}$ is given below

$$\mathcal{V}_k^{n+1} \in \big[r_k^n\big(j, \mathbf{a}_{-k}^n\big) - r_k^n\big(i, \mathbf{a}_{-k}^n\big) \big] \cdot \mathbb{I}_{\{a_k^n = i\}} - C_k\big(p_k^n \times \Delta(\mathbf{A}_{-k})\big). \tag{28}$$

3) CNRQ Learning: In Section III-B2, we remarked on the fact that both no-regret- and Q-learning are, by structure, special cases of stochastic approximation algorithms. In this section, we resort to the multi-timescale extension of standard stochastic approximation theory [34] to recast the nested loop structure of the QnR algorithm as a single-loop two-timescale stochastic approximation. The idea is to have the Q-learning and no-regret iterations proceed simultaneously with different step-size schedules so that Q-table entries get updated on a slower effective timescale compared to the regret-matrix updates. Multi-timescale arguments of stochastic approximation [34] then guarantee that no-regret iterations see Q-learning as quasi-static while the latter sees the former as nearly equilibrated, thus mimicking the QnR's nested loop configuration. Following the same methodology, we introduce an even slower third timescale for updating the LMs associated with the game's constraints. More specifically, we leverage on Theorem 1 and the saddle point property in (7) to cast the algorithm as a primal-dual scheme; i.e., given a fixed λ_k for each agent k, "primal" maximization reduces to computing, in a distributed fashion, a CE behavior $\boldsymbol{\pi}^{\text{ce}}$ (see Definition 1) of the stochastic game $\Gamma^{\boldsymbol{\lambda}}$. Also, given that $\mathcal{G}_k(\lambda_k) = \mathcal{L}_k^{\lambda_k}(Q_k^{\lambda_k}, \boldsymbol{\pi}^{\text{ce}})$, the correct multiplier λ_k^* can be learned by stochastic gradient descent in the "dual" space, performed on the slowest timescale, so that it sees the primal maximization as having essentially equilibrated. We refer to the overall algorithm as CNRQ-learning. Given the set-valued update increments of no-regret learning and the asynchronous nature of the Q-learning iterations, CNRQ would essentially correspond to a three-timescale asynchronous stochastic recursive inclusion. We save the formalization of these ideas for Section IV, where we give a detailed convergence analysis. Here, we mainly establish notation and discuss the algorithm's workflow. Let the learning rates $\{\alpha(n)\}_{n\in\mathbb{N}}$, $\{\beta(n)\}_{n\in\mathbb{N}}$, and $\{\gamma(n)\}_{n\in\mathbb{N}}$ satisfy the following:

$$\begin{aligned} \sum_n \alpha(n) = \sum_n \beta(n) = \sum_n \gamma(n) \; &= \; \infty \\ \sum_n \Big(\alpha(n)^2 + \beta(n)^2 + \gamma(n)^2 \Big) \; &< \; \infty \\ \frac{\alpha(n)}{\gamma(n)}, \frac{\beta(n)}{\alpha(n)} \to 0 \quad &\text{as} \quad n \to \infty. \end{aligned} \tag{A1}$$

Also, for $\forall s \in \mathcal{S}$ and $\forall \mathbf{a} \in \mathbf{A}$, let $\phi^n(s)$ and $\upsilon^n(s, \mathbf{a})$ be two asynchronous counters: $\phi^n(s) := \sum_{i=1}^n \mathbb{I}_{\{s^i = s\}}$ and $\upsilon^n(s, \mathbf{a}) := \sum_{i=1}^n \mathbb{I}_{\{(s^i, \mathbf{a}^i) = (s, \mathbf{a})\}}$. We organize the CNR$Q$'s workflow into nine steps, as listed in Table III.

1) In step 0, each agent initializes the empirical frequency of joint play $\boldsymbol{\pi}_s^0(.)$, LM λ_k^0, Q-table $\hat{Q}_{k,(s,.)}^0$, and state-dependent regret matrix $R_{k,\langle .,. \rangle,s}^0$. It then samples its action a_k^0 from a uniform distribution.
2) In step 1, according to the observed joint opponents' play $\mathbf{a}_{-k}^n$, agent k updates the empirical distribution

TABLE III
CNRQ

0) Initialization: $\forall s \in \mathcal{S}$, set: $\boldsymbol{\pi}_s^0(.) = 0$; $\hat{Q}_{k,(s,.)}^0 = 0$; $\lambda_k^0 = 0$; $R_{k,\langle .,. \rangle,s}^0 = 0$; $a_k^0 \sim 1/|A_k|$;

for $n = 0,1,2,...$ repeat the following steps:

1) Observe the opponents' play $\boldsymbol{a}_{-k}^n$ and update the empirical frequency of $\forall \boldsymbol{a} \in \boldsymbol{A}$ for $\forall s \in \mathcal{S}$:
$\boldsymbol{\pi}_s^{n+1}(\boldsymbol{a}) := \boldsymbol{\pi}_s^n(\boldsymbol{a}) + \gamma\big(\phi^n(s)\big).\mathbb{I}_{\{s=s^n\}}.\big(e_{\boldsymbol{a}^n} - \boldsymbol{\pi}_s^n(\boldsymbol{a})\big)$;
// $e_{\boldsymbol{a}}$ is the unit vector in $\Delta(\boldsymbol{A})$ w.r.t. $\boldsymbol{a} \in \boldsymbol{A}$

2) Observe utility $u_k(s^n, \boldsymbol{a}^n)$, cost $c_k(s^n, \boldsymbol{a}^n)$, calculate Lagrangian $\ell_k(\lambda_k^n, s^n, \boldsymbol{a}^n)$, observe next state s^{n+1}.

3) Calculate the long-term Lagrangian:
$$\mathcal{L}_{k,s^{n+1}}^{\lambda_k} := \sum_{\hat{\boldsymbol{a}} \in \mathcal{A}} \boldsymbol{\pi}_{s^{n+1}}^{n+1}(\hat{\boldsymbol{a}}).\hat{Q}_{k,\left(s^{n+1},\hat{\boldsymbol{a}}\right)}^n;$$

4) Update Lagrangian Q-table for $\forall s \in \mathcal{S}, \forall \boldsymbol{a} \in \boldsymbol{A}$ via Q-learning:
$\hat{Q}_{k,(s,\boldsymbol{a})}^{n+1} := \hat{Q}_{k,(s,\boldsymbol{a})}^n + \alpha\big(\upsilon^n(s,\boldsymbol{a})\big).\mathbb{I}_{\{(s,\boldsymbol{a})=(s^n,\boldsymbol{a}^n)\}}.$
$\Big[(1-\rho).\ell_k(\lambda_k^n, s, \boldsymbol{a}) + \rho.\mathcal{L}_{k,s^{n+1}}^{\lambda_k} - \hat{Q}_{k,(s,\boldsymbol{a})}^n\Big]$;

5) Update Lagrange multiplier via stochastic (sub)gradient descent:
$\lambda_k^{n+1} := [\lambda_k^n + \beta(n).(c_k(s^n, \boldsymbol{a}^n) - \overline{\mathcal{D}}_k)]^+$;
// $[.]^+$ denotes projection onto $[0, MAX]$.

6) Update the $|A_k| \times |A_k|$ state-dependent regret matrix R_k^n for $\forall i, j \in A_k$ and $\forall s \in \mathcal{S}$:
$R_{k,\langle i,j\rangle,s}^{n+1} := R_{k,\langle i,j\rangle,s}^n + \gamma\big(\phi^n(s)\big).\mathbb{I}_{\{s=s^n\}}.$
$\Big[\big(\hat{Q}_{k,(s,j,\boldsymbol{a}_{-k}^n)}^{n+1} - \hat{Q}_{k,(s,i,\boldsymbol{a}_{-k}^n)}^{n+1}\big).\mathbb{I}_{\{a_k^n = i\}} - R_{k,\langle i,j\rangle,s}^n\Big]$;

7) Update action transition probability for $\forall s \in \mathcal{S}$ and $\forall j \in A_k$:
$$T_{k,s}^{n+1}\big(a_k^{n+1} = j \big| a_k^n = i\big) := \begin{cases} \Upsilon\big(R_{k,\langle i,j\rangle,s}^{n+1}\big)/\mu, & j \neq i \\ 1 - \sum_{l \in A_k, l \neq j} T_{k,s}^{n+1}\big(a_k^{n+1} = l \big| a_k^n = i\big), & j = i \end{cases};$$
where, μ is the inertia constant and is large enough to ensure that $T_{k,s}^{n+1}\big(a_k^{n+1} = i \big| a_k^n = i\big) > 0$ for $\forall i \in A_k$.

8) Action selection: Choose action a_k^{n+1} in state s^{n+1} according to the following distribution:
$p_{k,s^{n+1}}^{n+1} := (1-\varepsilon).\hat{p}_{k,s^{n+1}}^{n+1} + \frac{\varepsilon}{|A_k|}.\mathbf{1}_{|A_k|}$,
where, $\hat{p}_k^{n+1}$ is an invariant measure for T_k^{n+1}, $\mathbf{1}_{|A_k|}$ is a $|A_k| \times 1$ vector of 1's, and $0 < \varepsilon \ll 1$ is a small tremble.

$\boldsymbol{\pi}_s^n(.)$ on the fastest timescale. It would be convenient to express $\boldsymbol{\pi}_s^n(.)$ in closed-form as below

$$\boldsymbol{\pi}_s^n(\mathbf{a}) = \sum_{\eta \le n} \gamma\left(\phi^{\eta-1}(s)\right)\left[\prod_{\zeta=\eta}^{n-1}\left(1-\gamma\left(\phi^{\zeta}(s)\right)\right)\right] e_{\mathbf{a}^\eta}. \tag{29}$$

3) In step 2, agent k calculates its instantaneous Lagrangian $\ell_k(\lambda_k^n, s^n, \mathbf{a}_{-k}^n)$ for its played action a_k^n, and observes the next state of the system s^{n+1}.
4) Steps 3 and 4 update the Q-table $\hat{Q}_k^n$ using Q-learning on the moderate timescale. This step unfolds as follows: agent k first computes its Lagrangian value function $\mathcal{L}_{k,s^{n+1}}^{\lambda_k}$ for the next state s^{n+1} based on the empirical frequency of play $\boldsymbol{\pi}^{n+1}$ and current estimate $\hat{Q}_k^n$. It then updates its Q-table using both its instantaneous Lagrangian ℓ_k and its long-term Lagrangian $\mathcal{L}_k^{\lambda_k}$.
5) In step 5, LM λ_k^n is updated based on the perceived cost $c_k(s^n, \mathbf{a}^n)$ and using stochastic (sub-)gradient descent on the slowest timescale.
6) Step 6 is devoted to the state-dependent regret matrix update. The regret matrix $R_{k,\langle i,j\rangle,s}^n$ is conditional on k's current play $a_k^n = i$, and is calculated as the Q-value differential between i and every alternative action j. Similarly to step 1, this update equation runs on the fastest timescale. To make more explicit the dependency of R_k^n on both $\hat{Q}_k^n$ and $\boldsymbol{\pi}^n$, one may use (29) to rewrite $R_{k,\langle i,j\rangle,s}^n$ for $\forall i,j \in A_k$ as follows:

$$\begin{aligned} R_{k,\langle i,j\rangle,s}^n &= \sum_{\eta \le n: a_k^\eta = i} \gamma\left(\phi^{\eta-1}(s)\right).\left[\prod_{\zeta=\eta}^{n-1}\left(1-\gamma\left(\phi^{\zeta}(s)\right)\right)\right] \\ &\quad \times \left(\hat{Q}^{\eta}_{k,\left(s,j,\mathbf{a}_{-k}^n\right)} - \hat{Q}^{\eta}_{k,\left(s,i,\mathbf{a}_{-k}^n\right)}\right) \\ &= \sum_{\mathbf{a}\in\mathbf{A}: a_k = i} \boldsymbol{\pi}_s^n(\mathbf{a}).\left(\hat{Q}^{n}_{k,\left(s,j,\mathbf{a}_{-k}^n\right)} - \hat{Q}^{n}_{k,\left(s,i,\mathbf{a}_{-k}^n\right)}\right). \end{aligned} \tag{30}$$

7) Step 7 uses the updated regret-values to compute the action transition probabilities $T_{k,s}^{n+1}(a_k^{n+1} = j | a_k^n = i)$ from the current action i to every alternative action j. T_k^{n+1} is proportional to the positive part of the regret measure; i.e., $\max(R_{k,s}^{n+1}, 0)$. However, to ensure the smoothness of these transitions, we use a function $\Upsilon(.)$ as a smooth version of $\max(.,0)$, defined as: $\Upsilon(x) \overset{\text{def}}{=} \begin{cases} x, & x > 0 \\ 0, & x < 0 \end{cases}$ for any $\delta > 0$ and $x \notin \delta-$neighborhood(0).
8) Finally, in step 8, the action for the next stage $(n+1)$ is sampled from $p_{k,s^{n+1}}^{n+1}$ which is an ε-soft version of the regret-based strategy $\hat{p}_{k,s^{n+1}}^{n+1}$ with ε being the exploration factor. $\hat{p}_k^{n+1}$ is an invariant measure for the stochastic transition matrix T_k^{n+1}. Therefore, p_k^{n+1} can be viewed as the invariant measure for the ε-trembled version of T_k^{n+1} denoted by $\tilde{T}_k^{n+1}$, and can be obtained by solving the following balance equations for $\forall s \in \mathcal{S}, \forall a \in A_k$:

$$p_{k,s}^{n+1}(a) \sum_{\acute{a}\in A_k - \{a\}} \tilde{T}_{k,\langle a,\acute{a}\rangle,s}^{n+1} = \sum_{\acute{a}\in A_k - \{a\}} p_{k,s}^{n+1}(\acute{a}).\tilde{T}_{k,\langle \acute{a},a\rangle,s}^{n+1} \tag{31}$$

and

$$\tilde{T}_{k,\langle i,j\rangle,s}^{n+1} := (1-\varepsilon)\frac{\Upsilon\left(R_{k,\langle i,j\rangle,s}^{n+1}\right)}{\mu} + \frac{\varepsilon}{|A_k|}, \quad \forall i,j \in A_k. \tag{32}$$

IV. Convergence Analysis

CNRQ-learning is essentially a three-timescale asynchronous stochastic recursive inclusion. To establish CNRQ's convergence, we exploit the recent results by Perkins and Leslie [27] which facilitate the asymptotic analysis in cases such as ours where the update patterns involved are both asynchronous and set-valued. The proof framework we use is called asynchronous stochastic approximation with differential inclusions. The results given in [27] already account for two-timescale setups as well. Also, since in general, the ideas underlying the multi-timescale arguments carry over when the number of timescales is more than two [34], using the two-timescale analysis in [27], we first analyze the coupled recursions of no-regret- and Q-learning by freezing $\lambda_k^n \approx \lambda_k$; in fact, CNR$Q$'s iterates can be interpreted as a primal-dual scheme, with $(\hat{Q}_k^n, \boldsymbol{\pi}^n)$ getting updated by primal iterations and λ_k^n by dual iterations. Now, in view of $\beta(n) = o(\alpha(n))$, the dual minimization is carried out at a slower timescale so that it sees the primal maximization as equilibrated while the latter sees the former as quasi-static. The analysis of the pair $(\hat{Q}_k^n, \boldsymbol{\pi}^n)$ can be conducted by invoking the results of [27, Sec. 4]. Once the almost sure convergence of the primal iterates is established, we have $d(\boldsymbol{\pi}^n, \mathcal{C}_{\text{ce}}^{\boldsymbol{\lambda}}) \to 0$, $\hat{Q}_k^n - Q_k^{\lambda_k} \overset{n\uparrow\infty}{\longrightarrow} 0$, and thus $\mathcal{L}_k^{\lambda_k}(\hat{Q}_k^n, \boldsymbol{\pi}^n) \to \mathcal{G}_k(\lambda_k)$. Then, using results from constrained reinforcement learning (see [35]), we can prove that the dual iterates λ_k^n also converge to λ_k^*. We organize our convergence analysis into three parts: first, we extract the mean-field and noise components associated with the primal iterates $(\hat{Q}_k^n, \boldsymbol{\pi}^n)$ in Section IV-A. Next, in Section IV-B, we verify the conditions which should be satisfied by these components so that the results in [27] become applicable to our case. Finally, in Section IV-C, we come up with differential inclusion arguments to establish the convergence of CNRQ along the lines of [27, Sec. 4].

A. Identifying the Mean-Field and Noise Components

According to Table III, the estimates $\{\boldsymbol{\pi}^n\}_{n\in\mathbb{N}}$ and $\{\hat{Q}_k^n\}_{n\in\mathbb{N}}$ are given iteratively by the coupled process

$$\begin{aligned} \hat{Q}_{k,(s,\mathbf{a})}^{n+1} - \hat{Q}_{k,(s,\mathbf{a})}^{n} &= \alpha\left(\upsilon^n(s,\mathbf{a})\right) \cdot \mathbb{I}_{\{(s,\mathbf{a})=(s^n,\mathbf{a}^n)\}} \\ &\quad \times \left[F_{(s,\mathbf{a})}\left(\hat{Q}_k^n, \boldsymbol{\pi}^n\right) + V_{k,(s,\mathbf{a})}^{n+1}\right] \end{aligned} \tag{33}$$

$$\boldsymbol{\pi}_s^{n+1} - \boldsymbol{\pi}_s^n \in \gamma\left(\phi^n(s)\right) \cdot \mathbb{I}_{\{s=s^n\}} \cdot \left[G_s\left(\hat{Q}_k^n, \boldsymbol{\pi}^n\right) + U_{k,s}^{n+1}\right] \tag{34}$$

and

$$F_{(s,\mathbf{a})}\Big(\hat{Q}_k^n, \boldsymbol{\pi}^n\Big) = \mathrm{H}_{(s,\mathbf{a})}^{\lambda_k}\Big(\hat{Q}_k^n, \boldsymbol{\pi}^n\Big) - \hat{Q}_{k,(s,\mathbf{a})}^n \tag{35}$$

$$G_s\Big(\hat{Q}_k^n, \boldsymbol{\pi}^n\Big) = \Psi_s\Big(\hat{Q}_k^n, \boldsymbol{\pi}^n\Big) - \boldsymbol{\pi}_s^n. \tag{36}$$

The mapping $\mathrm{H}_{(s,\mathbf{a})}^{\lambda_k}(.,.)$ in (35) is defined for general $\hat{Q}_k \in c^{|\mathcal{S}\times\mathbf{A}|}$ and $\boldsymbol{\pi} \in \Pi^{\boldsymbol{\lambda}}$ as

$$\mathrm{H}_{(s,\mathbf{a})}^{\lambda_k}\Big(\hat{Q}_k, \boldsymbol{\pi}\Big) = (1-\rho)\cdot\mathbb{E}[\ell_k(\lambda_k, s, \mathbf{a})] + \rho\sum_{\acute{s}\in\mathcal{S}}\mathcal{P}_{s\mathbf{a}\acute{s}}\mathcal{L}_{k,\acute{s}}^{\lambda_k}\Big(\hat{Q}_k, \boldsymbol{\pi}\Big). \tag{37}$$

Ψ_s in (36) is a correspondence evaluated at some $\Big(\hat{Q}_k, \boldsymbol{\pi}\Big)$ as

$$\Psi_s\Big(\hat{Q}_k, \boldsymbol{\pi}\Big) \ \Delta q\ \big\{p_{k,s}\times\Delta(\mathbf{A}_{-k})\big|p_{k,s}\ \text{satisfies (31)}\big\}. \tag{38}$$

See that in view of (32) and (30), Ψ_s depends on both $\hat{Q}_k$ and $\boldsymbol{\pi}$.

To specify the stochastic components $\{V_k^n\}_{n\in\mathbb{N}}$, $\{U_k^n\}_{n\in\mathbb{N}}$, let $\overline{H} = \{((s,\mathbf{a}),s); s\in\mathcal{S}, a\in\mathbf{A}\}$ with $\overline{H}^n \in \overline{H}$ being the updated component across $\overline{H}$ at iteration n; also, let $z^n = (\hat{Q}_k^n, \boldsymbol{\pi}^n)$. Define $\mathcal{F}_n$ as the σ-algebra containing all the information up until the end of the nth iteration; i.e., for $\forall m \le n, s\in\mathcal{S}, (s,\mathbf{a})\in\mathcal{S}\times\mathbf{A}$, $\mathcal{F}_n \overset{\text{def}}{=} \sigma(\{\overline{H}^m\}_m, \{z^m\}_m, \{\upsilon^m(s,\mathbf{a})\}_{(s,\mathbf{a}),m}, \{\phi^m(s)\}_{s,m})$; Then, $\{V_k^n\}_{n\in\mathbb{N}}$ and $\{U_k^n\}_{n\in\mathbb{N}}$ are, by construction, $\mathcal{F}_n$-adapted martingale difference processes defined on $\mathbb{R}^{|\mathcal{S}\times\mathbf{A}|}$ and $\mathbb{R}^{|\mathcal{S}|}$ resp. as follows:

$$V_{k,(s,\mathbf{a})}^{n+1} = (1-\rho)\cdot\ell_k\big(\lambda_k^n, s, \mathbf{a}\big) + \rho\cdot\mathcal{L}_{k,s^{n+1}}^{\lambda_k} - \mathrm{H}_{(s,\mathbf{a})}^{\lambda_k}\Big(\hat{Q}_k^n, \boldsymbol{\pi}^n\Big) \tag{39}$$

$$U_{k,s}^{n+1} \in e_{\mathbf{a}^n} - \Psi_s\Big(\hat{Q}_k^n, \boldsymbol{\pi}^n\Big). \tag{40}$$

B. Verifying Technical Assumptions

In this section, we verify the conditions required by [27] on the mean-field and noise components of the $(\hat{Q}_k^n, \boldsymbol{\pi}^n)$ iterates. We do this by presenting a sequence of lemmas (1–4) corresponding resp. to [27, Assumptions (B1), (B3)–(B5)]. Assumption (B2) in [27] is already satisfied by our Assumption (A1) on step-sizes in Section III-B3. Please refer to Appendix A in the supplementary materials of this paper for proofs of Lemmas 1–4.

Lemma 1: For compact sets, $C \subset \mathbb{R}^{|\mathcal{S}\times\mathbf{A}|}$, $D \subset \mathbb{R}^{|\mathcal{S}|}$, $\hat{Q}_k^n \in C$, $\boldsymbol{\pi}^n \in D$ for all k and n.

Lemma 2: The following hold:

1) $G(.,.)$: $c^{|\mathcal{S}\times\mathbf{A}|}\times\Pi^{\boldsymbol{\lambda}} \to \Pi^{\boldsymbol{\lambda}}$ is a Marchaud map [36]; that is: a) the graph and domain of G are nonempty and closed; b) the values $G(\hat{Q}_k, \boldsymbol{\pi})$ are convex; and c) the growth of G is linear.
2) $F(.,.)$: $c^{|\mathcal{S}\times\mathbf{A}|}\times\Pi^{\boldsymbol{\lambda}} \to c^{|\mathcal{S}\times\mathbf{A}|}$ is upper semi-continuous, and for all $\boldsymbol{\pi}\in\Pi^{\boldsymbol{\lambda}}, F(.,\boldsymbol{\pi}) : c^{|\mathcal{S}\times\mathbf{A}|} \to c^{|\mathcal{S}\times\mathbf{A}|}$ is a Marchaud map.

Lemma 3: Consider $\mathcal{H}_n, \mathcal{H}_{n+1} \in \overline{H}$, then:

1) $\mathbb{P}(\overline{H}_{n+1} = \mathcal{H}_{n+1} \mid \mathcal{F}_n) = \mathcal{Q}_{(\mathcal{H}_n,\mathcal{H}_{n+1})}(z) \overset{\text{def}}{=} \mathbb{P}(\overline{H}_{n+1} = \mathcal{H}_{n+1} \mid \overline{H}_n = \mathcal{H}_n, z_n = z)$;
2) for all $z \in c^{|\mathcal{S}\times\mathbf{A}|}\times\Pi^{\boldsymbol{\lambda}}$, the transition probabilities $\mathcal{Q}_{(\mathcal{H}_n,\mathcal{H}_{n+1})}(z)$ form aperiodic, irreducible Markov chains over $\overline{H}$ and for all $s\in\mathcal{S}$ and $(s,\mathbf{a})\in\mathcal{S}\times\mathbf{A}$ there exists $\mathcal{H}, \acute{\mathcal{H}} \in \overline{H}$, such that $s\in\mathcal{H}$ and $(s,\mathbf{a})\in\acute{\mathcal{H}}$;
3) the map $z \longmapsto \mathcal{Q}_{(\mathcal{H}_n,\mathcal{H}_{n+1})}(z)$ is Lipschitz continuous.

In effect, Lemma 3 verifies that asymptotically every state of the game Γ will be visited a minimum proportion of time, say $\tau > 0$. Also, the ε-trembled action transition probabilities in (32) ensures that every joint action will be selected with a nonzero probability; hence, every state-action pair is used a minimum proportion of time, say $\acute{\tau} > 0$.

Lemma 4: Given any norm $\|.\|$ on $\mathbb{R}^{|\mathcal{S}\times\mathbf{A}|}$ and on $\mathbb{R}^{|\mathcal{S}|}$, there exists constants A, B, C, and D such that

$$\mathbb{E}\Big[\Big(V_{k,(s,\mathbf{a})}^n\Big)^2\Big|\mathcal{F}_n\Big] < A + B\Big\|\hat{Q}_{k,(s,\mathbf{a})}^n\Big\|^2$$

and

$$\mathbb{E}\Big[\big(U_{k,s}^n\big)^2\Big|\mathcal{F}_n\Big] < C + D\big\|\boldsymbol{\pi}_s^n\big\|^2 \quad \forall s, \mathbf{a}.$$

C. Differential Inclusion Arguments

In this section, we proceed to characterize the limiting behavior of CNRQ using differential inclusion arguments from [27]. Methodologically, the arguments in [27] are based on the well-established ordinary differential equation (ODE) approach [33] which treats the stochastic approximation (22) as a noisy discretization of an autonomous ODE with $F(x)$ as its mean-field. More specifically, under appropriate conditions on the step-sizes, mean-field and noise components of (22), it follows that the continuous-time linear interpolation of x_n asymptotically tracks the stable fixed points of the dynamical system $\dot{x} = F(x)$. Hence, the limit sets of (22) will coincide with the set of stable fixed points of its associated ODE, and one can study instead the stability of the deterministic system $\dot{x} = F(x)$ to establish the convergence of the random sequence $\{x^n\}_{n\in\mathbb{N}}$. The results in [27] extend the ODE method to the case of asynchronous stochastic approximation with set-valued mean-fields. Within this perspective, our next lemma (Lemma 5) characterizes the limiting behavior of the fast stochastic recursion in (34). Before stating the lemma, we briefly hint on the main theoretical result in [27] which considerably facilitates our analysis in this paper. Our overview here is merely to convey the key idea in [27], and an avid reader is encouraged to consult [27] for more thorough exposition.

In case we were dealing with a synchronous updating pattern in CNRQ, standard arguments (see [33]) would suggest that we may analyze the convergence behavior of the discrete-time iterates $\{\boldsymbol{\pi}^n\}_{n\in\mathbb{N}}$ by studying the limit sets of an associated ordinary differential inclusion (ODI) with correspondence $G(.,.)$ as its mean field

$$\frac{d\boldsymbol{\pi}^t}{dt} \in G\Big(\hat{Q}_k, \boldsymbol{\pi}^t\Big)$$

where $\{\hat{Q}_k^n\}_{n\in\mathbb{N}}$ iterates on the slow timescale are freezed at $\hat{Q}_k$ by standard multi-timescale results [34]. However, unlike synchronous stochastic approximation where the steps sizes $\alpha(n)$ are deterministic, CNRQ features random and time-varying

step sizes of the form $\gamma(\phi^n(s)) \cdot \mathbb{I}_{\{s=s^n\}}$. The conventional way to deal with such asynchronicity does not readily extend to set-valued mean fields; also, even if $G(.,.)$ were single-valued, the standard procedure would be to study the limit sets of a nonautonomous ODE of the form [33]

$$\frac{d\boldsymbol{\pi}^t}{dt} = M(t) \cdot G\Big(\hat{Q}_k, \boldsymbol{\pi}^t\Big) \tag{41}$$

where $M(.)$ is a matrix-valued measurable process such that $M(t)$ for each t is a diagonal matrix with non-negative diagonal entries, reflecting the relative instantaneous rates with which the different components of $\boldsymbol{\pi}$ get updated. The existing theory does not explicitly define the scaling matrix $M(.)$ and it is further assumed that in the limit all the components of $\boldsymbol{\pi}$ are updated in an equally spaced manner and some "specific" minimum proportion of the iterations. To work around the difficulties in studying (41), the approach in [27] shows that under the conditions stated in Lemmas 1–4, the diagonal elements of $M(t)$ lie almost surely in the closed set $[\tau, 1]$, for some $\tau > 0$. It then combines the set $[\tau, 1]$ with the mean field $G(.,.)$ to form a set-valued mean-field. More specifically, let $\Omega^{\tau}_{|\mathcal{S}|}$ be the $|\mathcal{S}| \times |\mathcal{S}|$ diagonal matrix of the form: $\Omega^{\tau}_{|\mathcal{S}|} \coloneqq \{\text{diag}(\xi_1, \ldots, \xi_{|\mathcal{S}|}); \xi_s \in [\tau, 1], \forall s \in \mathcal{S}\}$. It is shown in [27] that the limit set of the asynchronous iterates $\{\boldsymbol{\pi}^n\}_{n\in\mathbb{N}}$ can be characterized by the asymptotic analysis of the ODI below

$$\frac{d\boldsymbol{\pi}^t}{dt} \in \Omega^{\tau}_{|\mathcal{S}|} \cdot G\Big(\hat{Q}_k, \boldsymbol{\pi}^t\Big).$$

This procedure pays off in two ways: 1) the analysis can be done by studying an "autonomous" rather than a "non-autonomous" system and 2) it extends the previous theory to also capture the behavior of asynchronous updates with "set-valued" mean fields. Moreover, as argued in [27], it only suffices to verify that τ is positive; i.e., to ensure that all the components of the iterates get updated some minimum proportion of time. The key advantage lies in that the exact value of τ does not need to be known, as the analysis will be conducted for every $\tau > 0$. Now, recall from Lemma 3 that every $s \in \mathcal{S}$ in CNRQ is, in fact, selected some minimum proportion of time, $\tau > 0$. Armed with this understanding, we are now prepared to state Lemma 5 which corresponds to [27, Assumption (B6)]. Define the correspondence $\Pi^{\text{ce}}(.) : c^{|\mathcal{S}\times\mathbf{A}|} \mapsto \Pi^{\boldsymbol{\lambda}}$ such that for all $\hat{Q}_k \in c^{|\mathcal{S}\times\mathbf{A}|}$, one has $\boldsymbol{\pi} \in \Pi^{\boldsymbol{\lambda}}$ is in $\Pi^{\text{ce}}(\hat{Q}_k)$ if and only if it satisfies Definition 1 for CE policies. $\Pi^{\text{ce}}(.)$ is an upper semi-continuous set-valued map (see [18, Lemma 16]), such that for all $\hat{Q}_k \in c^{|\mathcal{S}\times\mathbf{A}|}$, $\Pi^{\text{ce}}(\hat{Q}_k)$ is compact, convex, and nonempty (see [18, Lemma 16]).

Lemma 5: For all $\hat{Q}_k \in c^{|\mathcal{S}\times\mathbf{A}|}$:

1) the differential inclusion

$$\dot{\boldsymbol{\pi}}^{\mathbf{t}}_{\mathbf{s}} = \frac{d\boldsymbol{\pi}^{\mathbf{t}}_{\mathbf{s}}}{dt} \in \Omega^{\tau}.G_s\Big(\hat{Q}_k, \boldsymbol{\pi}^{\mathbf{t}}_{\mathbf{s}}\Big), \qquad \text{for all } s \in \mathcal{S} \tag{42}$$

is globally attracted by $\Pi^{\text{ce}}(\hat{Q}_k)$;

2) $F(\hat{Q}_k, \Pi^{\text{ce}}(\hat{Q}_k))$ is a convex map.

Proof: Following [19] and [20], the correspondence $\Pi^{\text{ce}}(\hat{Q}_k)$ coincides with the set of no-regret policies:

$$\Big\{\boldsymbol{\pi} \in \Pi^{\boldsymbol{\lambda}} : \mathcal{R}_{k,\langle a,\acute{a}\rangle,s}\Big(\hat{Q}_k, \boldsymbol{\pi}\Big) \le 0, \forall k \in \mathcal{K}, s \in \mathcal{S}, a, \acute{a} \in A_k\Big\} \tag{43}$$

where for general $\hat{Q}_k \in c^{|\mathcal{S}\times\mathbf{A}|}$ and $\boldsymbol{\pi} \in \Pi^{\boldsymbol{\lambda}}$, $\mathcal{R}_{k,\langle a,\acute{a}\rangle,s}$ is defined as (for $\forall k \in \mathcal{K}, \forall s \in \mathcal{S}, \forall a, \acute{a} \in A_k$)

$$\mathcal{R}_{k,\langle a,\acute{a}\rangle,s}\Big(\hat{Q}_k, \boldsymbol{\pi}\Big) \overset{\text{def}}{=} \sum_{\mathbf{a}\in\mathbf{A}:a_k=a} \boldsymbol{\pi}_s(\mathbf{a}).\Big[\hat{Q}_{k,(s,\acute{a},\mathbf{a}_{-k})} - \hat{Q}_{k,(s,a,\mathbf{a}_{-k})}\Big]. \tag{44}$$

Equation (43) implies that the solutions to (42) steer the state-dependent regret matrix $R_{k,s}$ toward the closed negative orthant $\mathbb{R}^{|A_k\times A_k|}_{-}$, denoted for short by Θ. The analysis will be more direct if we consider the equivalent dynamics in the regret space; i.e., to show that for the solutions to (45) below

$$\dot{R}^t_{k,s} = \frac{dR^t_{k,s}}{dt} \in \Omega^{\tau}.\Big[\mathcal{R}_{k,s}\Big(\hat{Q}_k, \Psi_s\Big(\hat{Q}_k, \boldsymbol{\pi}^t\Big)\Big) - R^t_{k,s}\Big] \tag{45}$$

we have that

$$R^t_{k,\langle a,\acute{a}\rangle,s} \to \Theta \quad \text{as} \quad t \to \infty, \quad \forall k \in \mathcal{K}, \forall s \in \mathcal{S}, \forall a, \acute{a} \in A_k.$$

Following [27, Theorem 5.2], we now produce a Lyapunov function for (45) to show that Θ (resp. Π^{ce}) is a global attractor for (45) [resp. (25)]. Define

$$\mathcal{L}(R_k) = \frac{1}{2} \sum_{s\in\mathcal{S},a,\acute{a}\in A_k} \Big[\Upsilon\Big(R_{k,\langle a,\acute{a}\rangle,s}\Big)\Big]^2.$$

Clearly, $\mathcal{L} \ge 0$, $\mathcal{L}(\Theta^{|\mathcal{S}|}) = 0$, and $\nabla\mathcal{L}(R_k) = \Upsilon(R_k)$. To show that $\mathcal{L}$ is a Lyapunov function for the ODI (45), we need to verify for any fixed $\omega_s \in \Omega^{\tau}$ and any $\acute{\boldsymbol{\pi}} \in \Psi_s(\hat{Q}_k, \boldsymbol{\pi}^t)$

$$\begin{aligned} &\prec \nabla\mathcal{L}(R_k), \dot{R}^t_k \succ \\ &\quad = \sum_{s\in\mathcal{S}} \prec \Upsilon\big(R_{k,s}\big), \omega_s \cdot \Big[\mathcal{R}_{k,s}\Big(\hat{Q}_k, \acute{\boldsymbol{\pi}}\Big) - R_{k,s}\Big] \succ \ < 0 \\ &\qquad \text{for } \forall R_{k,s} \in \mathbb{R}^{|A_k\times A_k|}\backslash\Theta \end{aligned} \tag{46}$$

where $\prec .,. \succ$ denotes the Frobenius inner product. It can be shown (see Lemma B.1 in the supplementary materials of this paper) that $\prec \Upsilon(R_{k,s}), \mathcal{R}_{k,s}(\hat{Q}_k, \acute{\boldsymbol{\pi}}) \succ = 0$ for all $s \in \mathcal{S}$; hence, (46) reduces to: $-\sum_{s\in\mathcal{S}} \prec \Upsilon(R_{k,s}), \omega_s.R_{k,s} \succ$ which is clearly less than 0. This concludes part (a).

As for part (b), the convexity of $F(\hat{Q}_k, \Pi^{\text{ce}}(\hat{Q}_k))$ is immediate given that the function H^{λ_k} in the definition of F is an affine function of $\mathcal{L}^{\lambda_k}_k$, and for any fixed $\hat{Q}_k \in c^{|\mathcal{S}\times\mathbf{A}|}$, $\mathcal{L}^{\lambda_k}_k$ reduces to a linear function of $\boldsymbol{\pi} \in \Pi^{\boldsymbol{\lambda}}$. ■

Now remember from Lemma 3 that every state-action pair is used a minimum proportion of time, $\acute{\tau} > 0$. Define $\Omega^{\acute{\tau}}_{|\mathbf{A}|}$ to be the $|\mathbf{A}| \times |\mathbf{A}|$ diagonal matrix of the form: $\Omega^{\acute{\tau}}_{|\mathbf{A}|} \coloneqq \{\zeta_1, \ldots, \zeta_{|\mathbf{A}|}; \zeta_{\mathbf{a}} \in [\acute{\tau}, 1], \forall \mathbf{a} \in \mathbf{A}\}$. In light of [27, Th. 4.7], under assumption (A1) and with Lemmas 1–5 holding, the linear interpolation of the iterative process in (33) is an asymptotic pseudo-trajectory to the differential inclusion

$$\frac{d\hat{Q}^t_{k,s}}{dt} \in \Omega^{\acute{\tau}}_{|\mathbf{A}|} \cdot F_s\Big(\hat{Q}^t_k, \Pi_{\text{ce}}\Big), \ \forall s \in \mathcal{S} \tag{47}$$

where F_s stands for the $|\mathbf{A}|$-vector of the $F_{(s,\mathbf{a})}$ terms; i.e., for any $\boldsymbol{\pi}^{\text{ce}} \in \Pi^{\text{ce}}$

$$F_s\Big(\hat{Q}^t_k, \boldsymbol{\pi}^{\text{ce}}\Big) = \text{H}^{\lambda_k}_s\Big(\hat{Q}^t_k, \boldsymbol{\pi}^{\text{ce}}\Big) - \hat{Q}^t_{k,s}, \qquad \text{for all } s \in \mathcal{S}$$

and $\mathrm{H}_s^{\lambda_k}(\hat{Q}_k^t, \boldsymbol{\pi}^{\mathrm{ce}})$ is the $|\mathbf{A}|$-vector of $\mathrm{H}_{(s,\mathbf{a})}^{\lambda_k}(\hat{Q}_k^t, \boldsymbol{\pi}^{\mathrm{ce}})$ terms, defined in (37).

The next lemma establishes the convergence of the $\{\hat{Q}_k^n\}_{n\in\mathbb{N}}$ iterates on the moderate time-scale.

Lemma 6: $Q_{k,s}^{\lambda_k}$ [defined in (11)] is the unique global attractor of the differential inclusion (47).

Proof: Clearly, $\mathrm{H}^{\lambda_k}(Q_k^{\lambda_k}, \boldsymbol{\pi}) = Q_k^{\lambda_k}$. Also, for any fixed $\boldsymbol{\pi} \in \Pi^{\boldsymbol{\lambda}}$, $\mathrm{H}_s^{\lambda_k}(\hat{Q}_k, \boldsymbol{\pi})$ is a contraction mapping with respect to sup norm $\|.\|_\infty$ (see [24]); i.e., $\forall \hat{Q}_k, \hat{\acute{Q}}_k \in c^{|\mathcal{S}\times\mathbf{A}|}$

$$\left\| \mathrm{H}_s^{\lambda_k}\left(\hat{Q}_k, \boldsymbol{\pi}\right) - \mathrm{H}_s^{\lambda_k}\left(\hat{\acute{Q}}_k, \boldsymbol{\pi}\right) \right\|_\infty \leq \rho \cdot \left\| \hat{Q}_{k,s} - \hat{\acute{Q}}_{k,s} \right\|_\infty$$

which means that $Q_{k,s}^{\lambda_k}$ is its unique fixed point. Now, for any fixed $\omega \in \Omega_{|\mathbf{A}|}^{\tau}$, we have

$$\omega \cdot \left(\mathrm{H}_s^{\lambda_k}\left(\hat{Q}_k, \boldsymbol{\pi}\right) - \hat{Q}_{k,s} \right) = \mathrm{H}_s^{\lambda_k,\omega}\left(\hat{Q}_k, \boldsymbol{\pi}\right) - \hat{Q}_{k,s}$$

where $\mathrm{H}_s^{\lambda_k,\omega}(\cdot) \stackrel{\mathrm{def}}{=} (\mathbf{I} - \omega) \cdot \hat{Q}_{k,s} + \omega \cdot \mathrm{H}_s^{\lambda_k}(\hat{Q}_k, \boldsymbol{\pi})$. Since ω's diagonal elements are bounded by 1, it holds that: $\forall \hat{Q}_k, \hat{\acute{Q}}_k \in c^{|\mathcal{S}\times\mathbf{A}|}$

$$\left\| \mathrm{H}_s^{\lambda_k,\omega}\left(\hat{Q}_k, \boldsymbol{\pi}\right) - \mathrm{H}_s^{\lambda_k,\omega}\left(\hat{\acute{Q}}_k, \boldsymbol{\pi}\right) \right\|_\infty \leq \overline{\rho} \cdot \left\| \hat{Q}_{k,s} - \hat{\acute{Q}}_{k,s} \right\|_\infty$$

where $\overline{\rho} \stackrel{\mathrm{def}}{=} 1 - \zeta^*(1-\rho) \in (0,1)$, and $\zeta^* = \max_i \zeta_i$. Thus, $\mathrm{H}_s^{\lambda_k,\omega}(\hat{Q}_k, \boldsymbol{\pi})$ is also a contraction mapping, and $Q_{k,s}^{\lambda_k}$ is its unique fixed point; i.e., $\mathrm{H}^{\lambda_k,\omega}(Q_k^{\lambda_k}, \boldsymbol{\pi}) = Q_k^{\lambda_k}$. From this, it follows that $\{\hat{Q}_k^n\}_{n\in\mathbb{N}}$ converge to true values $Q_k^{\lambda_k}$ for any policy $\boldsymbol{\pi} \in \Pi^{\boldsymbol{\lambda}}$, and in particular for CE policies in Π^{ce}. ■

Theorem 2: The coupled process $(\hat{Q}_k^n, \boldsymbol{\pi}^n)$ from (33) and (34) converges to the limit $(Q_k^{\lambda_k}, \Pi^{\mathrm{ce}}(Q_k^{\lambda_k}))$ where $\boldsymbol{\pi}^{\mathrm{ce}} \in \Pi^{\mathrm{ce}}(Q_k^{\lambda_k})$ is a stationary CE policy for the stochastic game $\Gamma^{\boldsymbol{\lambda}}$ with λ_k-parameterized individual Lagrangian utilities and $Q_k^{\lambda_k}$ is the associated Lagrangian state-action value function.

Proof: Immediate by Lemma 6 and [27, Corollary 4.8]. ■

Theorem 3: As $n \to \infty$, for all $k \in \mathcal{K}$, $\lambda_k^n \to \lambda_k^*$ and $(\hat{Q}_k^n, \boldsymbol{\pi}^n, \lambda_k^n) \to (Q_k^{\lambda_k^*}, \Pi^{\mathrm{ce}}(Q_k^{\lambda_k^*}), \lambda_k^*)$.

Proof: Please see Appendix C in supplementary materials. ■

V. Numerical Results

In this section, we follow up on the example scenarios from Section II-C, and give numerical results on CNRQ-learning algorithm. We conduct experiments for a two-tier network with four femto-cells and a single macro-cell. We investigate CNRQ's convergence and also compare its social welfare with that computed from both centralized and semi-distributed variants of the CE-Q learning algorithm both implemented with a utilitarian equilibrium selection mechanism [22]. In order to apply CE-Q to our constrained game example, we have adopted the Lagrangian approach similarly to CNRQ and have augmented CE-Q with LM iterations that run on a slower timescale with respect to Q-value iterations. Since centralized CE-Q is convergent to stationary CE, this would result in proper handling of the constraints in the game; however,

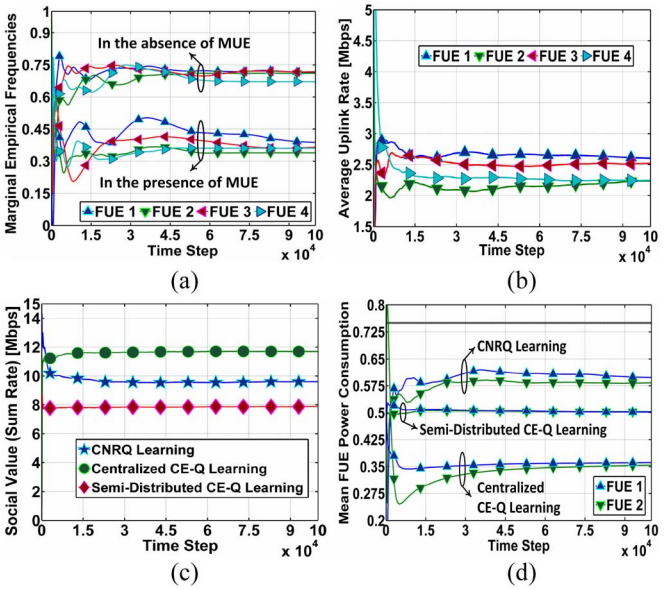

Fig. 2. (a) FUEs' marginal frequency of transmissions (uplink). (b) Individual FUE's average rate utilities. (c) Average uplink social welfare. (d) Convergence of individual FUE's average power consumption.

semi-distributed CE-Q is susceptible to mis-coordination, and convergence to CE is not guaranteed in general. This is also corroborated by our experiments in the downlink scenario in that semi-distributed CE-Q violates the constraint on MBSs buffer length.

First, consider the uplink HetNet setup from Section II-C1. The simulation parameters are listed in Table IV. Fig. 2(a) and (b) exhibits the convergence behavior of CNRQ in this scenario. In Fig. 2(a), we plot the marginal empirical frequency of high power transmissions (action $a_k^u = 1$) by FUEs for both cases of MUE's occupancy state. In Fig. 2(b), the progression of the average individual rate utility achieved by all FUEs is depicted. In Fig. 2(c), we compare CNRQ's social welfare (measured in terms of the sum of FUE's rate utilities) with that obtained from both semi-distributed and centralized versions of the CE-Q algorithm. As can be seen, CNRQ outperforms semi-distributed CE-Q, but its social welfare is upper bounded by centralized CE-Q. We show in Fig. 2(d) the average power consumption by FUEs. To keep the figure from being cluttered, the results are shown only for FUEs 1 and 2. The imposed average power constraint (0.75 mW in Table IV) is respected asymptotically by all three algorithms.

To experiment with the downlink setup, we use the simulation parameters listed in the left half of Table IV. Fig. 3(a) shows the convergence of marginal empirical frequency of play for action $a_k^d = 10$ [mW] by each FBS when the system state (i.e., MBS buffer length) is $b_0 = 5$. In Fig. 3(b) and (c), we compare CNRQ's social welfare and constraint satisfaction with the other two baselines. As evidenced, semi-distributed CE-Q, despite achieving a slightly higher sum rate, has violated the constraint on MBS buffer length by a relatively large margin. We also study the impact of the MBS's Poisson traffic arrival rate on the downlink social welfare and on the constraint on MBS buffer length. To this end, the traffic intensity

TABLE IV
SIMULATION PARAMETERS

Downlink		Uplink	
Parameter	**Value**	**Parameter**	**Value**
MBS transmit power a_0^d	500 mW	noise power No	10^{-7} mW
FBS transmit power $a_k^d, k = 1 \dots 4$	{0,10,100} mW	MUE transmit power a_0^u	5 mW
MBS-FUE channel gains $g_{0k}^d, k = 1 \dots 4$	(0.003, 0.005, 0.008, 0.002)	FUE transmit power $a_k^u, k = 1 \dots 4$	$\{'low' \stackrel{\text{def}}{=} 0, 'high' \stackrel{\text{def}}{=} 1\}$ mW
FBS-MUE channel gains $g_{k0}^d, k = 1 \dots 4$	(0.055, 0.051, 0.035, 0.012)	MUE-FBS channel gains g_{0k}^u, $k = 1 \dots 4$	(0.038, 0.082, 0.071, 0.086)
FBS-FUE channel gains h_{kk}^d, $k, \acute{k} = 1 \dots 4$	$\begin{pmatrix} 0.68 & 0.09 & 0.03 & 0.04 \\ 0.07 & 0.82 & 0.04 & 0.04 \\ 0.01 & 0.04 & 0.16 & 0.03 \\ 0.03 & 0.08 & 0.01 & 0.29 \end{pmatrix}$	FUE-FBS channel gains h_{kk}^u, $k, \acute{k} = 1 \dots 4$	$\begin{pmatrix} 0.44 & 0.10 & 0.02 & 0.10 \\ 0.07 & 0.23 & 0.03 & 0.06 \\ 0.10 & 0.10 & 0.25 & 0.10 \\ 0.10 & 0.05 & 0.09 & 0.24 \end{pmatrix}$
MBS-to-MUE Poisson traffic rate $\mathbb{E}[\mathcal{A}]$	5.5 pkt/msec	bandwidth W	1 MHz
packet size L	256 bytes	mean FUE power constraint $\bar{a}_k^u, k = 1 \dots 4$	0.75 mW
mean MBS buffer length constraint $\bar{b}_0$	10 pkt		

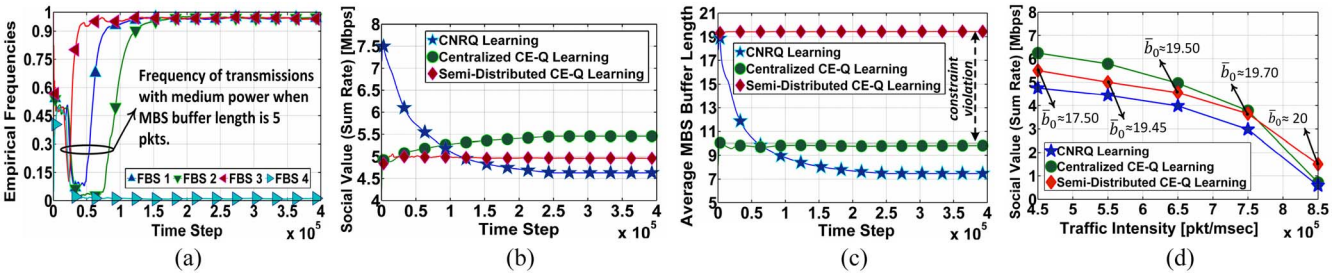

Fig. 3. (a) FBSs' marginal frequency of transmissions (downlink). (b) Average downlink social welfare. (c) Convergence of MBS's average buffer length. (d) Downlink social welfare versus MBS traffic intensity (note the violation of constraint in semi-distributed CE-Q learning).

is varied from 4.5 to 8.5 pkt/ms. MBS buffer length constraint is consistently respected by both CNRQ and centralized CE-Q. However, as shown in Fig. 3(d), despite its high social welfare, semi-distributed CE-Q has consistently violated the constraint on buffer length.

VI. CONCLUSION

We presented a CNRQ algorithm for the online computation of stationary CE in constrained general-sum stochastic games. CNRQ builds on previous ideas which involve two control loops consisting of Q-learning (outer-loop) for estimating action value functions and no-regret-learning (inner-loop) for estimating a CE policy. We employed the technique of timescale separation from stochastic approximation to allow for a single-loop concurrent execution of Q-learning (on the slower timescale) and no-regret-learning (on the faster timescale), which eliminates the backstage virtual plays as required by prior art in inner-loop iterations. Moreover, by regarding distributed CE estimation as simultaneous primal maximization across all agents, we extended the algorithm for constrained setups as well. Thanks to our stochastic approximation-based expression of the learning process, the constrained extension comes as easily as introducing a slower third timescale to the operation of the algorithm for conducting dual descent in LM space. Overall, CNRQ has been cast as a three-timescale asynchronous stochastic approximation with set-valued update increments. Unlike prior art which lacks a rigorous convergence analysis, we analyzed the asymptotic behavior of CNRQ using differential inclusion arguments. Our analysis draws on recent extensions of the theory of stochastic approximation to the case of asynchronous recursive inclusions with set-valued mean fields. We also applied CNRQ-learning to an exemplary case of emerging wireless HetNet deployments.

REFERENCES

[1] L. S. Shapley, "Stochastic games," *Proc. Natl. Acad. Sci.*, vol. 39, no. 10, pp. 1095–1100, Oct. 1953.
[2] J. F. Nash, "Non-cooperative games," *Ann. Math.*, vol. 54, no. 2, pp. 286–295, Sep. 1951.
[3] R. Aumann, "Subjectivity and correlation in randomized strategies," *J. Math. Econ.*, vol. 1, no. 1, pp. 67–96, Mar. 1974.
[4] E. Altman and A. Shwartz, "Constrained Markov games: Nash equilibria," in *Advances in Dynamic Games and Applications* (Annals of the International Society of Dynamic Games), vol. 5, V. Gaitsgory, J. Filar, and K. Mizukami, Eds. Boston, MA, USA: Birkhäuser, 2000, pp. 213–221.
[5] J. Filar and K. Vrieze, *Competitive Markov Decision Processes*. New York, NY, USA: Springer, 1997.
[6] V. V. Singh and N. Hemachandra, "A characterization of stationary Nash equilibria of constrained stochastic games with independent state processes," *Oper. Res. Lett.*, vol. 42, no. 1, pp. 48–52, Jan. 2014.
[7] M. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, New Brunswick, NJ, USA, 1994, pp. 157–163.
[8] J. Hu and M. Wellman, "Nash Q-learning for general-sum stochastic games," *J. Mach. Learn. Res.*, vol. 4, pp. 1039–1069, Nov. 2003.
[9] J. Li, "Learning average reward irreducible stochastic games: Analysis and applications," Ph.D. dissertation, Dept. Ind. Manage. Syst. Eng., Univ. South Florida, Tampa, FL, USA, 2003.
[10] M. Littman, "Friend-or-Foe Q-learning in general-sum Markov games," in *Proc. Int. Conf. Mach. Learn.*, San Francisco, CA, USA, 2001, pp. 322–328.
[11] C. J. C. H. Watkins and P. Dayan, "Technical note: Q-learning," *J. Mach. Learn.*, vol. 8, nos. 3–4, pp. 279–292, May 1992.

[12] R. A. C. Bianchi, M. F. Martins, C. H. C. Ribeiro, and A. H. R. Costa, "Heuristically-accelerated multiagent reinforcement learning," *IEEE Trans. Cybern.*, vol. 44, no. 2, pp. 252–265, Feb. 2014.
[13] K.-S. Hwang, W.-C. Jiang, and Y.-J. Chen, "Model learning and knowledge sharing for a multiagent system with Dyna-Q learning," *IEEE Trans. Cybern.*, vol. 45, no. 5, pp. 964–976, May 2015.
[14] M. L. Koga, V. Freire, and A. H. R. Costa, "Stochastic abstract policies: Generalizing knowledge to improve reinforcement learning," *IEEE Trans. Cybern.*, vol. 45, no. 1, pp. 77–88, Jan. 2015.
[15] M. Bowling and M. Veloso, "Rational and convergent learning in stochastic games," in *Proc. Int. Joint Conf. AI*, vol. 2. San Francisco, CA, USA, 2001, pp. 1021–1026.
[16] P. Vrancx, K. Verbeeck, and A. Nowe, "Decentralized learning in Markov games," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 4, pp. 976–981, Aug. 2008.
[17] H. L. Prasad, L. A. Prashanth, and S. Bhatnagar, "Two-timescale algorithms for learning Nash equilibria in general-sum stochastic games," in *Proc. 14th AAMAS*, Richland, SC, USA, 2015, pp. 1371–1379.
[18] A. Greenwald and M. Zinkevich, "A direct proof of the existence of correlated equilibrium policies in general-sum Markov games," Dept. Comput. Sci., Brown Univ., Providence, RI, USA, Tech. Rep. CS-05-07, Jul. 2005.
[19] S. Hart and A. Mas-Colell, "A simple adaptive procedure leading to correlated equilibrium," *Econometrica*, vol. 68, no. 5, pp. 1127–1150, Sep. 2000.
[20] M. Benaïm, J. Hofbauer, and S. Sorin, "Stochastic approximations and differential inclusions, part II: Applications," *Math. Oper. Res.*, vol. 31, no. 4, pp. 673–695, Nov. 2006.
[21] H. P. Young, "Strategic learning and its limits," in *Arne Ryde Memorial Lectures Series*, New York, NY, USA: Oxford Univ. Press, 2004.
[22] A. Greenwald, M. Zinkevich, and P. Kaelbling, "Correlated Q-learning," in *Proc. Int. Conf. Mach. Learn.*, Washington, DC, USA, 2003, pp. 242–249.
[23] D. Gondek, A. Greenwald, and K. Hall. (Jul. 23, 2015). *QnR-Learning in Markov Games*. [Online]. Available: http://cs.brown.edu/people/amy/papers/qnr.ps.gz
[24] J. N. Tsitsiklis, "Asynchronous stochastic approximation and Q-learning," *J. Mach. Learn.*, vol. 16, no. 3, pp. 185–202, Sep. 1994.
[25] D. P. Bertsekas, *Nonlinear Programming*. Belmont, MA, USA: Athena Scientific, 1999.
[26] D. Blackwell, "Discounted dynamic programming," *Ann. Math. Statist.*, vol. 36, no. 1, pp. 226–235, 1965.
[27] S. Perkins and D. S. Leslie, "Asynchronous stochastic approximation with differential inclusions," *Stoch. Syst.*, vol. 2, no. 2, pp. 409–446, 2012.
[28] E. Altman, *Constrained Markov Decision Processes*. London, U.K.: Chapman and Hall, 1999.
[29] J. Hoadley and P. Maveddat, "Enabling small cell deployment with HetNet," *IEEE Wireless Commun.*, vol. 19, no. 2, pp. 4–5, Apr. 2012.
[30] A. Ghosh *et al.*, "Heterogeneous cellular networks: From theory to practice," *IEEE Commun. Mag.*, vol. 50, no. 6, pp. 54–64, Jun. 2012.
[31] A. Goldsmith, *Wireless Communications*. Cambridge, U.K.: Cambridge Univ. Press, 2005.
[32] S. Sardellitti, A. Carfagna, and S. Barbarossa, "Optimal resource allocation in fetmocell networks based on Markov modeling of interferers' activity," *EURASIP J. Wireless Commun.*, vol. 371, pp. 1–20, Dec. 2012.
[33] V. S. Borkar, *Stochastic Approximation: A Dynamical Systems Viewpoint*. Cambridge, U.K.: Cambridge Univ. Press, 2008.
[34] S. Bhatnagar, H. L. Prasad, and L. A. Prashanth, *Stochastic Recursive Algorithms for Optimization: Simultaneous Perturbation Methods*. London, U.K.: Springer, 2013.
[35] V. S. Borkar, "An actor-critic algorithm for constrained Markov decision processes," *Syst. Control Lett.*, vol. 54, no. 3, pp. 207–213, Mar. 2005.
[36] J. P. Aubin, *Dynamic Economic Theory: A Viability Approach*, vol. 174. Berlin, Germany: Springer, 1997.

Vesal Hakami received the bachelor's degree (with distinction) in computer engineering and the M.S. degree in information technology from the Amirkabir University of Technology, Tehran, Iran, in 2004 and 2008, respectively, where he is currently pursuing his Ph.D. in information technology.

In his Ph.D. thesis, he studies the applications of stochastic control theory and game-theoretic learning to wireless networks.

Mehdi Dehghan (M'10) received the B.Sc. degree in computer engineering from the Iran University of Science and Technology, Tehran, Iran, in 1992, and the M.Sc. degree in computer engineering and the Ph.D. degree in electrical engineering from the Amirkabir University of Technology, Tehran, in 1995 and 2001, respectively.

He is currently an Associate Professor with the AUT, where he is the Director of Wireless Networks Laboratory with the Department of Computer Engineering and Information Technology. His current research interests include multimedia networking, distributed systems, and wireless networks.