Distributed reinforcement learning in multi-agent networks

Soummya Kar*, José M. F. Moura* and H. Vincent Poor†

*Department of ECE, Carnegie Mellon University, Pittsburgh, PA 15213, {soummyak, moura}@ece.cmu.edu
†Department of EE, Princeton University, Princeton, NJ 08544, poor@princeton.edu

***Abstract*—Distributed reinforcement learning algorithms for collaborative multi-agent Markov decision processes (MDPs) are presented and analyzed. The networked setup consists of a collection of agents (learners) which respond differently (depending on their instantaneous one-stage random costs) to a global controlled state and the control actions of a remote controller. With the objective of jointly learning the optimal stationary control policy (in the absence of global state transition and local agent cost statistics) that minimizes network-averaged infinite horizon discounted cost, the paper presents distributed variants of Q-learning of the *consensus + innovations* type in which each agent sequentially refines its learning parameters by locally processing its instantaneous payoff data and the information received from neighboring agents. Under broad conditions on the multi-agent decision model and *mean* connectivity of the inter-agent communication network, the proposed distributed algorithms are shown to achieve *optimal learning* asymptotically, i.e., almost surely (a.s.) each network agent is shown to learn the value function and the optimal stationary control policy of the collaborative MDP asymptotically. Further, convergence rate estimates for the proposed class of distributed learning algorithms are obtained.**

***Index Terms*—Multi-agent stochastic control, distributed Q-learning, reinforcement learning, collaborative network processing, consensus + innovations, distributed stochastic approximation.**

I. Introduction

This paper considers multi-agent decision-making in dynamic and uncertain environments, with a network of *agents*[1] and a controlled global state process or signal (a finite state Markov chain with controlled transitions). The actions of a remote controller and the resulting controlled state influence the statistical distribution of the random instantaneous costs incurred at the agents. These Markov decision processes (MDPs) pertain to collaborative welfare; the agent network is interested in obtaining the optimal stationary control strategy that minimizes the network-averaged infinite horizon discounted cost. As an illustration, the multi-agent setup can be a thermostatically controlled *smart* building where the global state represents environmental dynamics affecting the spatial temperature distribution and the agents correspond to sensors distributed throughout the building. The objective of the building thermostatic controller may be to minimize the average of the squared deviations of the measured temperatures at the sensing locations from a desired reference value. In a second example, now drawn from the financial markets, the global signal may relate to the dynamic market interest rate affecting, for example, the investment patterns of the social agents, with the economic policies (actions) of the regulator (controller) aiming to sustain overall economic growth. Our formulation transcends these examples to include many practical scenarios, ranging from large-scale load control for efficient demand-side management in energy networks [1] to collaborative decision-making in multi-agent robotic networks [2]; see also the survey articles [3], [4] for related and other variants of MDPs in multi-agent settings.

Reinforcement learning, of which Q-learning [5], [6] is an instance, is a practical methodology for MDPs lacking prior information on the problem statistics, including the transition behavior of the controlled state process and, as in our multi-agent setting, the statistical distributions of the agents' instantaneous costs (varying with the agent). Rather than relying on exact problem statistics, Q-learning reformulates the Bellman equation to generate sequential (stochastic) approximations of the value function using instantiations of state-action trajectories that may correspond to online real-time data obtained while implementing the control, e.g., [6], in which case the resulting Q-learning methods are, in fact, instances of direct adaptive control [7], or, may correspond to training data obtained through simulated state-action responses, see [8] for various exploration methods. Direct application of classical reinforcement learning techniques to our proposed multi-agent setting with possibly geographically distributed agents would require a centralized computing architecture with access to the instantaneous one-stage costs of all the agents at all times (see Section II). Since the instantaneous one-stage costs are only observed locally at the agents, this, in turn, would require each network agent to forward its one-stage cost to the remote central location at all times, not feasible due to limited energy resources at the agents and a bit-budgeted communication medium. To cope with these difficulties, a fully distributed variant of Q-learning, the $\mathcal{QD}$-learning, was proposed and analyzed in [9] in which optimal learning of the MDP value function and the associated optimal stationary control policy is achieved at each network agent through local computation and peer-to-peer information sharing (cooperation) over a preassigned *sparse* possibly time-varying communication network. The distributed learning framework of [9] is very general and caters to both online adaptive control based and simulation based scenarios. In this paper, we study in detail a simulation based instance of $\mathcal{QD}$-learning, in which learning data (state transition and cost instantiations) at each stage is simulated by independently and identically generating state-action pairs and observing the one-step system response. Such independent and identically distributed (i.i.d.) sampling of state-action pairs for generating learning data is common in the (centralized) Q-learning literature [10], and the temporal convergence rate of the corresponding (centralized) instantiation of Q-learning is well known. As a main result of this paper, we show that, in this i.i.d. sampling scenario, the simulation-based instance of

The work of Kar was partially supported by NSF grant 1306128. The work of Moura was partially supported by NSF grant 1011903 and by AFOSR grant FA95501010291. The work of Poor was partially supported by NSF grant DMS-1118605.

[1] *Agent* is a generic term, its scope depends on the application.

978-1-4673-3146-3/13/$31.00 ©2013IEEE

our distributed $\mathcal{QD}$-learning procedure is order-optimal, i.e., as far as the time-order of convergence is concerned, it is as good as the corresponding simulation-based instance of the centralized Q-learning procedure. Furthermore, we show that the local Q-value estimates generated by our distributed algorithm are asymptotically normal, the asymptotic covariance (same for all the network agents) being a function of the state-action sampling distribution and the true model statistics.

Finally, we note that, $\mathcal{QD}$-learning as described here does not address two issues that may be relevant in applications: partial state observation and decentralized actuation. Specifically, we assume that each network agent observes perfectly the global state, and, in contrast to setups with local decentralized agent actuations, we assume that the control actions are generated by a remote (global) controller and are perfectly known at the agents[2].

The rest of the paper is organized as follows. Spectral graph theory notation is reviewed next. The multi-agent learning setup is described in Section II. Section III presents the proposed distributed reinforcement algorithm and derives its convergence, the main results of the paper. Finally, Section IV concludes the paper and discusses future research avenues.

Spectral graph theory: The inter-agent communication network is an *undirected* simple connected graph $G = (V, E)$, with $V = [1 \cdots N]$ and E denoting the set of agents (nodes) and communication links. The neighborhood of node n is

$$\Omega_n = \{l \in V \,|\, (n, l) \in E\} \tag{1}$$

Node n has degree $d_n = |\Omega_n|$. The structure of the graph is described by the $N \times N$ adjacency matrix, $A = A^\top = [A_{nl}]$, $A_{nl} = 1$, if $(n, l) \in E$, $A_{nl} = 0$, otherwise. Let $D = \text{diag}\,(d_1 \cdots d_N)$. The graph Laplacian $L = D - A$ is positive definite, with eigenvalues ordered as $0 = \lambda_1(L) \leq \lambda_2(L) \leq \cdots \leq \lambda_N(L)$. The eigenvector of L corresponding to $\lambda_1(L)$ is $(1/\sqrt{N})\mathbf{1}_N$. The multiplicity of its zero eigenvalue equals the number of connected components of the network; for a connected graph, $\lambda_2(L) > 0$. This second eigenvalue is the algebraic connectivity or the Fiedler value of the network.

II. System Model

Let $\{\mathbf{x}_t\}$ be a controlled Markov chain taking values in a finite state space $\mathcal{X} = [1, \cdots, M]$, and $\mathcal{U}$ be the finite set of control actions $\mathbf{u}$. Assume[3] the state transition is governed by

$$\mathbb{P}\left(\mathbf{x}_{t+1} = j \,|\, \mathbf{x}_t = i, \mathbf{u}_t = u\right) = p^u_{i,j},\ \forall i, j \in \mathcal{X}, u \in \mathcal{U} \tag{2}$$

where $\sum_{j \in \mathcal{X}} p^u_{i,j} = 1$ for all $i \in \mathcal{X}$.

There are N agents, agent n incurring a random one-stage cost[4] $c_n(i, u)$ whenever control u is applied at state i. For a stationary control policy π, i.e., where $\{\mathbf{u}_t\}$ satisfies $\mathbf{u}_t = \pi(\mathbf{x}_t)$ for some $\pi : \mathcal{X} \mapsto \mathcal{U}$, the state process $\{\mathbf{x}^\pi_t\}$ (the superscript π indicates the dependence on the control policy π) evolves as a homogenous Markov chain with[5]

$$\mathbb{P}\left(\mathbf{x}^\pi_{t+1} = j \Big| \mathbf{x}^\pi_t = i\right) = p^{\pi(i)}_{i,j}. \tag{3}$$

For a stationary policy π and initial state i of the process $\{\mathbf{x}^\pi_t\}$, the infinite horizon discounted cost is given by

$$V^\pi_i = \limsup_{T \to \infty} \mathbb{E}\left[\frac{1}{N}\sum_{n=1}^{N}\sum_{t=0}^{T}\gamma^t c_n\left(\mathbf{x}^\pi_t, \pi(\mathbf{x}^\pi_t)\right)\Big| \mathbf{x}^\pi_0 = i\right], \tag{4}$$

where $0 < \gamma < 1$ is the discounting factor. The cost V^π_i is a global (centralized), as it involves the one-stage costs of all the agents. Our Markov decision problem (MDP) evaluates the optimal infinite horizon discounted cost

$$V^*_i = \inf_\pi V^\pi_i \tag{5}$$

and the associated stationary policy π^*, if it exists.

Let $\mathbf{V}^* = [V^*_1, \cdots, V^*_M]^T$. Denote by $\mathcal{T} : \mathbb{R}^M \mapsto \mathbb{R}^M$ the (centralized) dynamic programming operator with

$$\mathcal{T}_i(\mathbf{V}) = \min_{u \in \mathcal{U}}\left\{\frac{1}{N}\sum_{n=1}^{N}\mathbb{E}\left[c_n(i, u)\right] + \gamma\sum_{j \in \mathcal{X}} p^u_{i,j} V_j\right\}, \tag{6}$$

where $\mathcal{T}_i(\cdot)$ is the i-th component functional of $\mathcal{T}(\cdot)$, such that, for $\mathbf{V} \in \mathbb{R}^M$, $\mathcal{T}(\mathbf{V}) = [\mathcal{T}_1(\mathbf{V}), \cdots, \mathcal{T}_M(\mathbf{V})]^T$. The Bellman equation [11] asserts that $\mathbf{V}^*$ is a fixed point of $\mathcal{T}(\cdot)$, i.e., $\mathcal{T}(\mathbf{V}^*) = \mathbf{V}^*$. Further, for strictly less than one discounting factors γ, the dynamic programming operator $\mathcal{T}(\cdot)$ is a strict contraction, [11], thus implying the value function $\mathbf{V}^*$ to be its unique fixed point. As such, starting with an arbitrary initial approximation $\mathbf{V}_0 \in \mathbb{R}^M$, one obtains a sequence of iterates $\{\mathbf{V}_t\}$ of $\mathcal{T}(\cdot)$, with $\mathbf{V}_t = \mathcal{T}^t(\mathbf{V}_0)$, such that, $\mathbf{V}_t \to \mathbf{V}^*$ as $t \to \infty$. The above iterative construction forms the basis of classical policy iteration methods for evaluating the desired value function V^* (and hence the corresponding optimal policy $\pi^*(\cdot)$), at least when $\gamma < 1$. However, in doing so, i.e., in constructing successive iterates of $\mathcal{T}(\cdot)$, the value iteration techniques assume that the problem statistics (the expected one-stage costs and the state transition probabilities $p^u_{i,j}$) are perfectly known apriori.

Q-learning: Reinforcement learning methods are motivated by scenarios lacking information about the problem statistics. Based on a reformulation of the Bellman equation, $\mathcal{T}(\mathbf{V}^*) = \mathbf{V}^*$, Q-learning methods generate sequential (stochastic) approximations of the value function[6] using instantiations of state-action trajectories, as opposed to relying on exact problem statistics. The state-action trajectory instantiations for value function learning may correspond to online real-time data obtained while implementing the control, in which case the resulting Q-learning methods are instances of direct adaptive control [7], or correspond to offline training data obtained through simulated state-action responses. For purpose of analysis, the former subsumes the latter, as trajectories obtained in the process of real-time control implementation

[2]The assumption that each network agent has access to the control actions of the remote controller may be relevant, for example, in financial or social network applications, where the market or network entities are typically informed about the policies of the global welfare organization. Even when such direct observability is not possible, the control information might be disseminated through network-wide broadcasts by the remote controller, given that, being a global entity, it may have sufficient energy resources and that the action broadcasts are finite-bit (due to the finiteness of the action space).

[3]The letters i and j will be reserved mostly to denote a generic element of the state space $\mathcal{X}$, whereas, u will denote a generic element of the control space $\mathcal{U}$. The state and control stochastic processes are bold symbols, $\{\mathbf{x}_t\}$ and $\{\mathbf{u}_t\}$ respectively, although they assume a finite number of values only.

[4]The instantaneous costs $c_n(\cdot)$ depend only on the current state of the process and the control applied, but not on the successor state as is the case with some control problems. The latter may often be reduced to the former (i.e., current state and control dependence only) by proper state augmentation.

[5]Note that, in general, the set of actions $\mathcal{U}$ is state-dependent, which can be accommodated in our formulation by redefining $\mathcal{U}$ to be the union of all state-dependent action sets and modifying the one-stage costs appropriately.

[6]Instead of generating successive approximations of the state-value function V^*_i, $i \in \mathcal{X}$, Q-learning methods generate approximations of the state-action value functions $Q^*_{i,u}$, $(i, u) \in \mathcal{X} \times \mathcal{U}$, (often known as the Q-matrices or factors) from which the desired value functions are recovered.

incur temporal statistical dependencies due to memory in the sequential control selection task. While the Q-learning techniques discussed above are appealing as they relax the requirement of prior system model knowledge, for our multi-agent setting, they rely on a centralized architecture that requires the instantaneous agent one-stage costs $c_n(\mathbf{x}_t, \mathbf{u}_t)$ (for each network agent n) to be available at a centralized computing resource at all times t with a view to obtaining an approximation of the sum of expectations in (6). Since, the instantaneous one-stage costs may only be observed at the agents, this, in turn, requires each network agent to transmit its one-stage cost to the remote central location at all times, which may not be feasible due to limited energy resources at the agents and a bit-budgeted communication medium. This motivates us to consider a fully distributed alternative, in which the agents autonomously engage in the learning process through collaborative local communication and computation.

III. A Distributed Q-learning Algorithm and its Convergence

In the simulation based $\mathcal{QD}$-learning scenario we consider in this paper, we assume that at each (simulation) time instant t, a state-action pair $(\mathbf{x}_t, \mathbf{u}_t)$ is generated independently and identically (over time t) according to a prespecified sampling distribution η_d on the state-action space $\mathcal{X} \times \mathcal{U}$. Once the pair $(\mathbf{x}_t, \mathbf{u}_t)$ is realized, the (one-step) system transition $\acute{\mathbf{x}}_t$ is observed and, in addition, each agent n obtains its local (random) one-stage payoff $c_n(\mathbf{x}_t, \mathbf{u}_t)$. In $\mathcal{QD}$-learning, based on the observed (simulated) system transition behavior $(\acute{\mathbf{x}}_t, \mathbf{x}_t, \mathbf{u}_t)$ and the instantaneous local payoffs $c_n(\mathbf{x}_t, \mathbf{u}_t)$ realized, each network agent n updates its learning parameters (to be formalized in Section III-A) with a view to refining its estimates of the value function and the optimal policy (4)-(5). Further, in addition to the locally generated or observed data, each agent n also incorporates the information sent to it by its communication neighbors (see Section III-A) in the estimate update process.

Before proceeding to a description of the distributed $\mathcal{QD}$-learning algorithm, we formalize the assumptions on the state-action sampling process and the inter-agent communication in the following.

(M.1): *The (one-step) state transition $\acute{\mathbf{x}}_t$ in response to a selected state-action pair $(\mathbf{x}_t, \mathbf{u}_t)$ is consistent with the true (but unknown) controlled Markov chain model at all times t*, i.e.,

$$\mathbb{P}\left(\acute{\mathbf{x}}_t = j \mid \mathbf{x}_t = i, \mathbf{u}_t = u\right) = p^u_{i,j}. \tag{7}$$

(M.2): *The state-action sampling distribution η_d assigns positive probability to each pair in $\mathcal{X} \times \mathcal{U}$, i.e., for all $(i, u) \in \mathcal{X} \times \mathcal{U}$ we have*

$$\mathbb{P}\left((\mathbf{x}_t, \mathbf{u}_t) = (i, u)\right) = \eta_d(i, u) > 0. \tag{8}$$

Note that assumption **(M.2)** implies, in particular, that all state-action pairs are simulated infinitely often as $t \to \infty$.

(M.3): *The one-stage random costs possess super-quadratic moments, i.e., there exists a constant $\varepsilon_1 > 0$ (could be arbitrarily small) such that*

$$\mathbb{E}\left[c_n^{2+\varepsilon_1}(i, u)\right] < \infty, \ \ \forall n, i, u. \tag{9}$$

(M.4): *The sequence $\{L_t\}$ of Laplacian matrices modeling the time-varying inter-agent communication network is independent and identically distributed, and connected in the mean, i.e., $\lambda_2(\overline{L}) > 0$, where $\overline{L} = \mathbb{E}[L_t]$ is the mean Laplacian.*

A. Distributed $\mathcal{QD}$-learning Algorithm

In $\mathcal{QD}$-learning, each network agent n updates a $\mathbb{R}^{|\mathcal{X}\times\mathcal{U}|}$-valued sequence $\{\mathbf{Q}^n_t\}$ (approximations of the so-called Q matrices) with components $Q^n_{i,u}(t)$ for every possible state-action pair (i, u). With this, the sequence $\{Q^n_{i,u}(t)\}$ at each agent n for each pair (i, u) evolves in a collaborative distributed fashion as follows:

$$Q^n_{i,u}(t+1) = Q^n_{i,u}(t) - \beta_{i,u}(t) \sum_{l \in \Omega_n(t)} \left(Q^n_{i,u}(t) - Q^l_{i,u}(t)\right)$$
$$+ \alpha_{i,u}(t) \left(c_n(\mathbf{x}_t, \mathbf{u}_t) + \gamma \min_{v \in \mathcal{U}} Q^n_{\acute{\mathbf{x}}_t, v}(t) - Q^n_{i,u}(t)\right). \tag{10}$$

The weight sequences $\{\beta_{i,u}(t)\}$ and $\{\alpha_{i,u}(t)\}$ are stochastic processes for each pair (i, u) and given by:

$$\beta_{i,u}(t) = \begin{cases} \frac{b}{(k+1)^{\tau_2}} & \text{if } t = T_{i,u}(k) \text{ for some } k \geq 0 \\ 0 & \text{otherwise,} \end{cases} \tag{11}$$

$$\alpha_{i,u}(t) = \begin{cases} \frac{a}{(k+1)^{\tau_1}} & \text{if } t = T_{i,u}(k) \text{ for some } k \geq 0 \\ 0 & \text{otherwise,} \end{cases} \tag{12}$$

a and b being positive constants, where $T_{i,u}(k)$ denotes the $(k+1)$-th sampling instant of the state-action pair (i, u). As reflected by the weight sequences (11)-(12), at each agent n, the component $Q^n_{i,u}(t)$ is updated at an instant[7] t *iff* the current state-action pair $(\mathbf{x}_t, \mathbf{u}_t)$ corresponds to (i, u); otherwise stays constant.

In addition to the processes $\{\mathbf{Q}^n_t\}$, each agent n updates an $\mathbb{R}^{|\mathcal{X}|}$-valued process $\{\mathbf{V}^n_t\}$, that serves as an approximation of the desired value function $\mathbf{V}^*$. The i-th component of $\mathbf{V}^n_t$, $V^n_i(t)$, is successively refined as

$$V^n_i(t) = \min_{u \in \mathcal{U}} Q^n_{i,u}(t), \ \ i = 1, \cdots, M. \tag{13}$$

We remark that the algorithm $\mathcal{QD}$ incurs no more computation at each agent than its centralized counterpart and being recursive is quite efficient in terms of memory and storage requirements. (Note that the $\mathbf{Q}^n_t$'s, the $\mathbf{V}^n_t$'s as well as the messages received from neighboring agents need not be stored over time as the update depends only on the current values of these quantities.)

The update rule (10) is in *consensus + innovations* form, [12]; it is the interplay between an agreement or consensus potential (agent collaboration) and a local innovation potential that incorporates newly obtained intelligence (local sensing of the instantaneous cost). The convergence of the resulting algorithm may only be achieved by intricately trading off these potentials, which, in turn, imposes further restrictions on the algorithm weight sequences as follows:

(M.5): *Persistence: The constants τ_1 and τ_2 in* (11)-(12) *satisfy $\tau_1 = 1$ and $0 < \tau_2 < 1/2 - 1/(2+\varepsilon_1)$, with ε_1 in* (9). Assumptions **(M.2)** and **(M.5)** guarantee that the excitations from the consensus and innovation potentials are persistent, i.e., the (stochastic) sequences $\{\alpha_{i,u}(t)\}$ and $\{\beta_{i,u}(t)\}$ sum to ∞, for each state-action pair (i, u). They further guarantee that the innovation weight sequences are square summable, i.e., $\sum_{t>0} \alpha^2_{i,u}(t) < \infty$ a.s., and that the consensus potential dominates the innovation potential eventually, i.e., $\beta_{i,u}(t)/\alpha_{i,u}(t) \to \infty$ a.s. as $t \to \infty$ for each pair (i, u).

[7]The expression *updated at an instant* t refers to the transition $Q^n_{i,u}(t)$ to $Q^n_{i,u}(t+1)$, an event that occurs after the one-stage cost $c_n(\mathbf{x}_t, \mathbf{u}_t)$ has been incurred and the successor state $\acute{\mathbf{x}}_t$ reached. In terms of implementation, such an update may be realized at the end of the time slot t.

B. Main Result

Our main result concerns the convergence of the $\mathcal{QD}$-learning procedure under assumptions **(M.1)**-**(M.5)**. It is stated as follows:

Theorem 1 Let $\{\mathbf{Q}_t^n\}$ and $\{\mathbf{V}_t^n\}$ be the successive iterates obtained at agent n by the distributed algorithm (10)-(13). Then, under **(M.1)**-**(M.5)**, there exists $\mathbf{Q}^* \in \mathbb{R}^{|\mathcal{X}\times\mathcal{U}|}$, such that, for each network agent n and all $\tau \in [0, 1/2)$, we have

$$\mathbb{P}\left(\lim_{t\to\infty}(t+1)^\tau\|\mathbf{Q}_t^n - \mathbf{Q}^*\| = 0\right) = 1, \tag{14}$$

and

$$\sqrt{t+1}\,(\mathbf{Q}_t^n - \mathbf{Q}^*) \Longrightarrow \mathcal{N}(\mathbf{0}, J), \tag{15}$$

where J is a matrix of appropriate dimensions and, in particular, independent of n; $\mathcal{N}(\cdot,\cdot)$ and $\Longrightarrow$ denote the Gaussian distribution and weak convergence, respectively.

Further, for each $i \in \mathcal{X}$, we have

$$\min_{u\in\mathcal{U}} Q_{i,u}^* = V_i^*, \tag{16}$$

and, hence, in particular, $(t+1)^\tau\|\mathbf{V}_t^n - \mathbf{V}^*\| \to 0$ as $t \to \infty$ a.s. for each n and all $\tau \in [0, 1/2)$, where $\mathbf{V}^*$ denotes the value function (5).

The proof of Theorem 1 is omitted due to space limitations. Note that, the consistency, i.e., $\mathbf{Q}_t^n \to \mathbf{Q}^*$ as $t \to \infty$ a.s. is an immediate consequence of Theorem 3.1 in [9] (which establishes consistency of generic $\mathcal{QD}$-learning procedures) by noting that the i.i.d. sampling assumption **(M.2)** implies that all state-action pairs are generated (simulated) infinitely often a.s. The additional properties concerning order of convergence (14) and asymptotic normality (15) follow from general properties of distributed stochastic approximation algorithms obtained in [12], which are applicable due to the special structure imposed by the i.i.d. state-action sampling scheme.

We note that the pathwise convergence established in (14) is order-optimal, in that, for centralized Q-learning in the i.i.d. sampling scenario, there exists in general no $\tau \geq 1/2$ such that the pathwise convergence rate of the corresponding (centralized) Q-values is $o(t^{-\tau})$ (see [10]). Another interesting thing to note is that the asymptotic covariance (J in (15)) is same for all the agents; however, J depends on the particular state-action sampling strategy in place. However, to optimize over sampling strategies in order to *minimize*[8] J (which is another indicator of convergence rate, the smaller the better), one needs to consider adaptive sampling strategies since, in general, such a minimizer will depend on the true model parameters (transition probabilities and cost distributions) which are not known in advance (see also Section IV for related discussion).

IV. Conclusion

The paper has investigated a distributed multi-agent reinforcement learning setup in a networked environment, in which the agents (for instance, temperature sensors in smart thermostatically controlled building applications, or, more generally, autonomous entities in social computing and decision making applications) respond differently to a global environmental signal or trend. Our setup is collaborative and non-competitive, with the overall network objective being global welfare, i.e., specifically, the network is interested in learning and evaluating the optimal stationary control strategy that minimizes the network-average infinite horizon discounted one-stage costs. Based on the generic $\mathcal{QD}$-learning framework developed in [9] for obtaining distributed algorithmic solutions for such collaborative networked MDPs, we have provided a distributed version of simulation based Q learning in which state-action pairs are assumed to be generated (and system behavior simulated) by i.i.d. sampling of the state-action space. We have shown that the convergence of our distributed learning approach is order-optimal, i.e., as far as the time-order of convergence is concerned, it is as good as the optimal centralized Q-learning procedure. Furthermore, the local Q-value estimates are shown to be asymptotically normal, the asymptotic covariance being a function of the state-action sampling distribution and the true model statistics. Future research would consider studying the impact of the state-action sampling distribution on the asymptotic covariance (a measure of the second-order convergence rate, the smaller the better) of the Q-value estimates. In general, the optimal sampling distribution that minimizes such asymptotic covariance is a function of the true model and cost statistics; since these parameters are not known in advance, adaptive sampling strategies need to be considered for the proposed performance optimization. Two other practically motivating and challenging future research topics concern the partial state information case, in which the global state process may not be perfectly observable at the local agent level, and the distributed actuation case, in which, instead of a remote controller acting on the global signal, the agents are themselves responsible for local actuations.

[8] Note, minimizing here is to be interpreted in the sense of the partial order induced by positive semidefiniteness of matrices.

References

[1] D. Callaway and I. Hiskens, "Achieving controllability of electric loads," *Proceedings of the IEEE*, vol. 99, no. 1, pp. 184 – 199, Jan. 2011.

[2] M. M. Veloso, P. Stone, K. Han, and S. Achim, "CMUnited: A team of robotic soccer agents collaborating in an adversarial environment," in *H. Kitano, editor, RoboCup-97: The First Robot World Cup Soccer Games and Conferences*. Springer Verlag, 1997, pp. 242–256.

[3] Y. Shoham, R. Powers, and T. Grenager, "Multi-agent reinforcement learning: a critical survey," May 2003, computer Science Dept., Stanford University, Stanford, CA. [Online]: http://ece.ut.ac.ir/classpages/F85/ControlOfStochasticSystems/res/Multi_Agent_Reinforcement_Learning.pdf.

[4] L. Busoniu, R. Babuska, and B. Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, vol. 38, no. 2, pp. 156–172, March 2008.

[5] C. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, pp. 279–292, 1992.

[6] J. Tsitsiklis, "Asynchronous stochastic approximation and Q-learning," *Machine Learning*, vol. 16, pp. 185–202, 1994.

[7] R. Sutton, A. Barto, and R. Williams, "Reinforcement learning is direct adaptive control," *IEEE Control Systems Magazine*, pp. 19 – 22, April 1992.

[8] A. Barto, S. Bradtke, and S. Singh, "Real-time learning and control using asynchronous dynamic programming," *Artificial Intelligence*, 1995.

[9] S. Kar, J. M. F. Moura, and H. V. Poor, "QD-learning: a collaborative distributed strategy for multi-agent reinforcement learning through consensus + innovations," *IEEE Transactions on Signal Processing*, vol. 61, no. 7, pp. 1848–1862, April 2013.

[10] C. Szepesvari, "The asymptotic convergence-rate of Q-learning," in *Advances in Neural Information Processing Systems*, M. Jordan, M. Kearns, and S. Solla, Eds., 1998, vol. 10, p. 1064 1070.

[11] D. Bertsekas, *Dynamic Programming and Stochastic Control*. New York, NY: Academic Press, Inc., 1976.

[12] S. Kar, J. M. F. Moura, and K. Ramanan, "Distributed parameter estimation in sensor networks: nonlinear observation models and imperfect communication," *IEEE Transactions on Information Theory*, vol. 58, no. 6, pp. 3575 – 3605, June 2012.