Scalable Reinforcement Learning of Localized Policies for Multi-Agent Networked Systems

Guannan Qu GQU@CALTECH.EDU
Adam Wierman ADAMW@CALTECH.EDU
California Institute of Technology, Pasadena, CA 91125, USA

Na Li NALI@SEAS.HARVARD.EDU
Harvard University, Cambridge, MA 02138, USA

Editors: A. Bayen, A. Jadbabaie, G. J. Pappas, P. Parrilo, B. Recht, C. Tomlin, M. Zeilinger

Abstract

We study reinforcement learning (RL) in a setting with a network of agents whose states and actions interact in a local manner where the objective is to find localized policies such that the (discounted) global reward is maximized. A fundamental challenge in this setting is that the state-action space size scales exponentially in the number of agents, rendering the problem intractable for large networks. In this paper, we propose a Scalable Actor Critic (SAC) framework that exploits the network structure and finds a localized policy that is an $O(\rho^{\kappa+1})$-approximation of a stationary point of the objective for some $\rho \in (0, 1)$, with complexity that scales with the local state-action space size of the largest κ-hop neighborhood of the network.

Keywords: Multi-agent reinforcement learning, networked systems, actor-critic methods.

1. Introduction

Having demonstrated impressive performance in a wide array of domains such as game play (Silver et al., 2016; Mnih et al., 2015), robotics (Duan et al., 2016), autonomous driving (Li et al., 2019), Reinforcement Learning (RL) has emerged as a promising tool for decision and control. However, in order to use RL in the context of control of large scale networked systems, such as those in cyber-physical systems, it is necessary to develop scalable RL algorithms for networked systems.

In this paper, we consider a RL problem for a network of n agents, each with state s_i and action a_i, both taking values from finite sets. The agents are associated with an underlying dependence graph $\mathcal{G}$ and interact locally, i.e, the distribution of $s_i(t+1)$ only depends on the current states of the local neighborhood of i as well as the local $a_i(t)$. Further, each agent is associated with stage reward r_i that is a function of s_i, a_i, and the global stage reward is the average of r_i. In this setting, the design goal is to find a decision policy that maximizes the (discounted) global reward. This setting captures a wide range of applications, e.g. epidemics (Mei et al., 2017), social networks (Chakrabarti et al., 2008; Llas et al., 2003), communication networks (Zocca, 2019; Vogels et al., 2003), queueing networks (Papadimitriou and Tsitsiklis, 1999), smart transportation (Zhang and Pavone, 2016), smart building systems (Wu et al., 2016; Zhang et al., 2017).

A fundamental difficulty when applying RL to such networked systems is that, even if individual state and action spaces are small, the entire state profile $(s_1, \ldots, s_n)$ and the action profile $(a_1, \ldots, a_n)$ can take values from a set of size exponentially large in n. This "curse of dimensionality" renders the problem unscalable. For example, most RL algorithms like temporal difference (TD) learning or Q-learning require storage of a Q-function (Bertsekas and Tsitsiklis, 1996) whose

size is the same as the state-action space, which in our problem is exponentially large in n. Such scalability issues have indeed been observed in previous research on variants of the problem we study, e.g. in multi-agent RL (Littman, 1994; Bu et al., 2008) and factored Markov Decision Process (MDP) (Kearns and Koller, 1999; Guestrin et al., 2003). A variety of approaches have been proposed to manage this issue, e.g. the idea of "independent learners" in Claus and Boutilier (1998); or function approximation schemes (Tsitsiklis and Van Roy, 1997). However, such approaches lack rigorous optimality guarantees. In fact, it has been suggested that such MDPs with exponentially large state spaces may be fundamentally intractable, e.g., see Blondel and Tsitsiklis (2000).

In addition to the scalability issue, another challenge is that, even if an optimal policy that maps a global state $(s_1, \ldots, s_n)$ profile to a global action $(a_1, \ldots, a_n)$ can be found, it is usually impractical to implement such a policy for real-world networked systems because of the limited information and communication among agents. For example, in large scale networks, each agent i may only be able to to implement *localized policies*, where its action a_i only depends on its own state s_i. Designing such localized polices with global network performance guarantee can also be challenging, see e.g. Rotkowitz and Lall (2005).

The challenges described above highlight the difficulty of applying RL to control large scale networked systems. However, the network itself provides some structure that can potentially be exploited. The question that motivates this paper is: *Can the network structure be utilized to develop scalable RL algorithms that provably find a (near-)optimal localized policy?*

Contributions. In this work we propose a framework that exploits properties of the network structure to develop RL to learn localized policies for large-scale networked systems in a scalable manner. Specifically, our main result (Theorem 5) shows that our algorithm, Scalable Actor Critic (SAC), finds a localized policy that is a $O(\rho^{\kappa+1})$-approximation of a stationary point of the objective function, with complexity that scales with the local state-action space size of the largest κ-hop neighborhood. To the best of our knowledge, our results are perhaps the first to provide such provable guarantee for scalable RL of localized policies in multi-agent network settings.

The key technique underlying our results is the observation that, when the size of κ-hop neighborhood is bounded, the network structure implies that the Q-function satisfies an *exponential decay property* (Definition 2), which leads to a tractable approximation of the policy gradient. In particular, despite the policy gradient itself being intractable to compute due to the large state-action space size, we introduce a *truncated policy gradient* (see Lemma 4) that can be computed efficiently and can be used in an actor-critic framework which yields an $O(\rho^{\kappa+1})$-approximation. This technique is novel and is a contribution in its own right. It can be used broadly to develop RL in network settings beyond the specific actor-critic algorithm we propose in this paper.

Related Literature. Our problem falls under category of the "succinctly described" MDPs in Blondel and Tsitsiklis (2000, Section 5.2), where the state and/or action space is a product space formed by the individual state and/or action space of multiple agents. As the state/action space is exponentially large, such problems are unscalable in general, even when the problem has structure (Blondel and Tsitsiklis, 2000; Whittle, 1988; Papadimitriou and Tsitsiklis, 1999). Despite this, there is a large literature on RL/MDPs in multi-agent settings under various structural assumptions.

Multi-agent RL dates back to the early work of Littman (1994); Claus and Boutilier (1998); Littman (2001); Hu and Wellman (2003) (see Bu et al. (2008) for a review) and has been actively studied, e.g. Zhang et al. (2018); Kar et al. (2013); Macua et al. (2015); Mathkar and Borkar (2017); Wai et al. (2018), see a more recent review in Zhang et al. (2019). Multi-agent RL encompasses a broad range of settings including competitive agents and Markov games. The case most relevant to ours is the cooperative multi-agent RL where typically, the agents can take their own actions but

Problem	State	Action	Coupling	Representative Literature
Multi-agent RL	global	local	yes	Zhang et al. (2019)
Factored MDP	local	global	local coupling	Guestrin et al. (2003)
Weakly Coupled MDP	local	local	reward only	Meuleau et al. (1998)
Our work	local	local	local coupling	Qu et al. (2019)

Table 1: Comparison of settings in related literature.

they share a common global state and maximize a global reward (Bu et al., 2008). This is contrast to the model we study, in which each agent has its own state and acts upon its own state. Despite the existence of a global state, multi-agent RL still faces scalability issues since the joint-action space is exponentially large. Methods have been proposed to deal with this, including independent learners (Claus and Boutilier, 1998; Matignon et al., 2012), where each agent employs a single-agent RL method. While successful in some cases, the independent learner approach can suffer from instability (Matignon et al., 2012). Alternatively, one can use function approximation schemes to approximate the large Q-table, e.g., linear function approximation (Zhang et al., 2018) or neuro networks (Lowe et al., 2017). Such methods can reduce computation complexity significantly, but it is unclear whether the performance loss caused by the function approximation is small. In contrast, our technique not only reduces computation but also guarantees small performance loss.

Factored MDPs are problems where every agent has its own state and the state transition factorizes in a way similar to our model (Kearns and Koller, 1999; Guestrin et al., 2003; Osband and Van Roy, 2014). However, they differ from the model we consider in that each agent does not have its own action. Instead, there is a global action affecting every agent. Despite the difference, Factored MDPs still suffer from scalability issues. Similar approaches as in the case of Multi-agent RL are used, e.g., Guestrin et al. (2003) proposes a class of "factored" linear function approximators; however, it is unclear whether the loss caused by the approximation is small.

Other Related Work. Our work is also related to *weakly coupled MDPs*, where every agent has its own state and action but their transition is decoupled (Meuleau et al., 1998). Additionally, our model shares some similarity with the epidemic network (Cator and Van Mieghem, 2012; Sahneh et al., 2013; Mei et al., 2017) and Glauber dynamics in physics (Lokhov et al., 2015; Mezard and Montanari, 2009), though our focus is very different from these works. Finally, this work is related to Qu and Li (2019), which assumes the full knowledge of MDP model (not RL) and imposes strong assumptions on the graph. In contrast, our work here does not need knowledge of the MDP and significantly relaxes the network assumptions.

2. Preliminaries

We consider a network of n agents that are associated with an underlying undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N} = \{1, \ldots, n\}$ is the set of agents and $\mathcal{E} \subset \mathcal{N} \times \mathcal{N}$ is the set of edges. Each agent i is associated with state $s_i \in \mathcal{S}_i$, $a_i \in \mathcal{A}_i$ where $\mathcal{S}_i$ and $\mathcal{A}_i$ are finite sets. The global state is denoted as $s = (s_1, \ldots, s_n) \in \mathcal{S} := \mathcal{S}_1 \times \cdots \times \mathcal{S}_n$ and similarly the global action $a = (a_1, \ldots, a_n) \in \mathcal{A} := \mathcal{A}_1 \times \cdots \times \mathcal{A}_n$. At time t, given current state $s(t)$ and action $a(t)$, the next individual state $s_i(t+1)$ is independently generated and is only dependent on neighbors:

$$P(s(t+1)|s(t), a(t)) = \prod_{i=1}^{n} P(s_i(t+1)|s_{N_i}(t), a_i(t)), \tag{1}$$

where notation N_i means the neighborhood of i (including i itself) and s_{N_i} is the states of i's neighbors. In addition, for integer $\kappa \geq 1$, we let N_i^κ denote the κ-hop neighborhood of i, i.e. the nodes whose graph distance to i is less than or equal to κ, including i itself. We also let $f(\kappa) = \sup_i |N_i^\kappa|$.

Each agent is associated with a class of localized policies $\zeta_i^{\theta_i}$ parameterized by θ_i. The localized policy $\zeta_i^{\theta_i}(a_i|s_i)$ is a distribution on the local action a_i conditioned on the local state s_i, and each agent, conditioned on observing $s_i(t)$, takes an action $a_i(t)$ independently drawn from $\zeta_i^{\theta_i}(\cdot|s_i(t))$. We use $\theta = (\theta_1, \ldots, \theta_n)$ to denote the tuple of the localized policies $\zeta_i^{\theta_i}$, and also use $\zeta^\theta(a|s) = \prod_{i=1}^n \zeta_i^{\theta_i}(a_i|s_i)$ to denote the joint policy, which is a product distribution of the localized policies as each agent acts independently.

Further, each agent is associated with a stage reward function $r_i(s_i, a_i)$ that depends on the local state and action, and the global stage reward is $r(s, a) = \frac{1}{n}\sum_{i=1}^n r_i(s_i, a_i)$. The objective is to find localized policy tuple θ such that the discounted global stage reward is maximized, starting from some initial state distribution π_0,

$$\max_\theta J(\theta) := \mathbb{E}_{s\sim\pi_0}\mathbb{E}_{a(t)\sim\zeta^\theta(\cdot|s(t))}\left[\sum_{t=0}^\infty \gamma^t r(s(t), a(t))\middle| s(0) = s\right]. \tag{2}$$

To provide context for what follows, we review a few key concepts in RL. First, fixing a localized policy tuple $\theta = (\theta_1, \ldots, \theta_n)$, the Q-function for this policy θ is:

$$\begin{aligned} Q^\theta(s, a) &= \mathbb{E}_{a(t)\sim\zeta^\theta(\cdot|s(t))}\left[\sum_{t=0}^\infty \gamma^t r(s(t), a(t))\middle| s(0) = s, a(0) = a\right] \\ &= \frac{1}{n}\sum_{i=1}^n \mathbb{E}_{a(t)\sim\zeta^\theta(\cdot|s(t))}\left[\sum_{t=0}^\infty \gamma^t r_i(s_i(t), a_i(t))\middle| s(0) = s, a(0) = a\right] := \frac{1}{n}\sum_{i=1}^n Q_i^\theta(s, a). \end{aligned} \tag{3}$$

In the last step, we have defined $Q_i^\theta(s, a)$ which is the Q function for the individual reward r_i. Both Q^θ and Q_i^θ are exponentially large tables and, therefore, are intractable to compute and store.

Finally, we recall the policy gradient theorem, which is the basis of many algorithmic results in RL. We emphasize that the lemma shows that the gradient of $J(\theta)$ depends on Q^θ and, therefore, is intractable to compute using the form in Lemma 1.

Lemma 1 (Sutton et al. (2000)) *Let π^θ be a distribution on the state space given by $\pi^\theta(s) = (1-\gamma)\sum_{t=0}^\infty \gamma^t \pi_t^\theta(s)$, where π_t^θ is the distribution of $s(t)$ under fixed policy θ when $s(0)$ is drawn from π_0. Then*

$$\nabla J(\theta) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim\pi^\theta, a\sim\zeta^\theta(\cdot|s)} Q^\theta(s, a)\nabla \log \zeta^\theta(a|s). \tag{4}$$

3. Algorithm Design and Results

In this paper we propose an algorithm, Scalable Actor Critic (SAC), which provably finds an $O(\rho^{\kappa+1})$-stationary point of the objective $J(\theta)$ for some $\rho \leq \gamma$,[1] with complexity scaling in the size of the local state-action space of the largest κ-hop neighborhood. We state our main result formally in Theorem 5 after introducing the details of SAC and the key idea underlying its design.

1. In this paper, a ε-stationary point of $J(\theta)$ refers to a θ s.t. $\|\nabla J(\theta)\|^2 \leq \varepsilon$.

3.1. Key Idea: Exponential Decay of Q-function Leads to Efficient Gradient Approximation

Recall that the policy gradient in Lemma 1 is intractable to compute due to the dimension of the Q-function. Our key idea is that exponential decay of the Q function allows efficient approximation of the policy gradient via truncation. To illustrate this, we start with the definition of the exponential decay property. Recall that N_i^κ is the set of κ-hop neighborhood of node i and define $N_{-i}^\kappa = \mathcal{N}/N_i^\kappa$, i.e. the set of agents that are outside of i'th κ-hop neighborhood. We write state s as $(s_{N_i^\kappa}, s_{N_{-i}^\kappa})$, i.e. the states of agents that are in the κ-hop neighborhood of i and outside of κ-hop neighborhood respectively. Similarly, we write a as $(a_{N_i^\kappa}, a_{N_{-i}^\kappa})$. The exponential decay property is then defined as follows.

Definition 2 *The (c, ρ)-exponential decay property holds if, for any localized policy θ, for any $i \in \mathcal{N}$, $s_{N_i^\kappa} \in \mathcal{S}_{N_i^\kappa}$, $s_{N_{-i}^\kappa}, s'_{N_{-i}^\kappa} \in \mathcal{S}_{N_{-i}^\kappa}$, $a_{N_i^\kappa} \in \mathcal{A}_{N_i^\kappa}$, $a_{N_{-i}^\kappa}, a'_{N_{-i}^\kappa} \in \mathcal{A}_{N_{-i}^\kappa}$, Q_i^θ satisfies,*

$$|Q_i^\theta(s_{N_i^\kappa}, s_{N_{-i}^\kappa}, a_{N_i^\kappa}, a_{N_{-i}^\kappa}) - Q_i^\theta(s_{N_i^\kappa}, s'_{N_{-i}^\kappa}, a_{N_i^\kappa}, a'_{N_{-i}^\kappa})| \leq c\rho^{\kappa+1}.$$

It may not be immediately clear when the exponential decay property holds. Lemma 3 highlights that the exponential decay property holds generally, with $\rho = \gamma$. Further, under some mixing time assumptions, the exponential decay property holds with $\rho < \gamma$. For more details on the generality of the exponential decay property, see Appendix A in our online report Qu et al. (2019).

Lemma 3 *If $\forall i$, r_i is upper bounded by $\bar{r}$, then the $(\frac{\bar{r}}{1-\gamma}, \gamma)$-exponential decay property holds.*

The power of the exponential decay property is that it guarantees that the dependence of Q_i^θ on other agents shrinks quickly as the distance between them grows. This motivates us to consider the following class of truncated Q-functions,

$$\hat{Q}_i^\theta(s_{N_i^\kappa}, a_{N_i^\kappa}) = \sum_{s_{N_{-i}^\kappa}, a_{N_{-i}^\kappa}} w_i(s_{N_{-i}^\kappa}, a_{N_{-i}^\kappa}; s_{N_i^\kappa}, a_{N_i^\kappa}) Q_i^\theta(s_{N_i^\kappa}, s_{N_{-i}^\kappa}, a_{N_i^\kappa}, a_{N_{-i}^\kappa}), \tag{5}$$

where $w_i(s_{N_{-i}^\kappa}, a_{N_{-i}^\kappa}; s_{N_i^\kappa}, a_{N_i^\kappa})$ are *any* non-negative weights satisfying

$$\sum_{s_{N_{-i}^\kappa} \in \mathcal{S}_{N_{-i}^\kappa}, a_{N_{-i}^\kappa} \in \mathcal{A}_{N_{-i}^\kappa}} w_i(s_{N_{-i}^\kappa}, a_{N_{-i}^\kappa}; s_{N_i^\kappa}, a_{N_i^\kappa}) = 1, \forall (s_{N_i^\kappa}, a_{N_i^\kappa}) \in \mathcal{S}_{N_i^k} \times \mathcal{A}_{N_i^k}. \tag{6}$$

Finally, our key insight is the following Lemma 4, which says when the exponential decay property holds, the truncated Q-function (5) can be used to accurately approximate the policy gradient. The proof of Lemma 4 is postponed to Appendix B in our online report Qu et al. (2019).

Lemma 4 (Truncated Policy Gradient) *Given i, define the following truncated policy gradient*

$$\hat{h}_i(\theta) = \frac{1}{1-\gamma} \mathbb{E}_{s \sim \pi^\theta, a \sim \zeta^\theta(\cdot|s)} \Big[\frac{1}{n} \sum_{j \in N_i^\kappa} \hat{Q}_j^\theta(s_{N_j^\kappa}, a_{N_j^\kappa}) \Big] \nabla_{\theta_i} \log \zeta_i^{\theta_i}(a_i|s_i), \tag{7}$$

where $\hat{Q}_j^\theta$ can be any truncated Q-function in the form of (5). Then, if (c, ρ)-exponential decay property holds and if $\|\nabla_{\theta_i} \log \zeta_i^{\theta_i}(a_i|s_i)\| \leq L_i, \forall a_i, s_i$, we have $\|\hat{h}_i(\theta) - \nabla_{\theta_i} J(\theta)\| \leq \frac{cL_i}{1-\gamma} \rho^{\kappa+1}$.

The power of this lemma is that the truncated Q function has much smaller dimension than the true Q function, and is thus scalable. However, despite the reduction in dimension, the error of the approximated gradient (7) is small. In the next section, we use this idea to design a scalable algorithm.

3.2. Algorithm Design: Scalable Actor Critic (SAC)

The good properties of the truncated Q-function open many possibilities for algorithm design. For instance, one can first obtain the truncated Q-function in some way (which could be much easier than directly computing the full Q-function) and then do a policy gradient step using the Lemma 4. In this subsection, we propose one particular approach using the actor-critic framework. Our approach, Scalable Actor Critic (SAC), uses temporal difference (TD) learning to obtain the truncated Q-function and then uses policy gradient for policy improvement. Psuedocode of the proposed algorithm is given in Algorithm 1.

Overall structure. The overall structure of SAC is a for-loop from line 1 to line 13. Inside the outer loop, there is an inner loop (line 4 through line 9) that uses temporal difference learning to get the truncated Q-function, which is followed by a policy gradient step that does policy improvement.

The Critic: TD-inner loop. Line 4 through line 9 is the policy evaluation inner loop that obtains the truncated Q function, where line 7 and 8 are the temporal difference update. We note that steps 7 and 8 use the same update equation as TD learning, except that it "pretends" $(s_{N_i^\kappa}, a_{N_i^\kappa})$ is the true state-action pair while the true state-action pair should be (s, a). As will be shown in the theoretic analysis in Appendix C in our online report Qu et al. (2019), such a TD update implicitly gives an estimate of a truncated Q function.

The Actor: Policy Gradient. Steps 10 through 12 define the the actor actions. Here, each agent calculates an estimate of the truncated gradient based on (7), and then conducts a gradient step.

Discussion. Our algorithm serves as an initial concrete demonstration of how to make use of the truncated policy gradient to develop a scalable RL method for networked systems. There are many extensions and other approaches that could be pursued, either within the actor-critic framework or beyond. One immediate extension is to do a warm start, i.e., initialize $\hat{Q}_i^0$ as the final estimate $\hat{Q}_i^T$ in the previous outer-loop. Additionally, one can use the TD-λ variant of TD learning with variance reduction schemes like the advantage function. Further, beyond the actor-critic framework, another direction is to develop Q-learning/SARSA type algorithms based on the truncated Q-functions. These are interesting topics for future work.

Numerical Experiments. Due to space limit, the numerical results are omitted and can be found in our online report (Qu et al., 2019).

3.3. Approximation Bound

In this section we state and discuss the formal approximation guarantee for SAC. Before stating the theorem, we first state the assumptions we use. The first assumption is standard in the RL literature and bounds the reward and state/action space size.

Assumption 1 (Bounded reward and state/action space size) *The reward is upper bounded as* $0 \leq r_i(s_i, a_i) \leq \bar{r}, \forall i, s_i, a_i$. *The individual state and action space size are upper bounded as* $|\mathcal{S}_i| \leq S, |\mathcal{A}_i| \leq A, \forall i$.

Assumption 2 (Exponential Decay) *The* (c, ρ) *exponential decay property holds for some* $\rho \leq \gamma$.

Note that under Assumption 1, Assumption 2 automatically holds with $\rho = \gamma$, cf. Lemma 3. However, we state the exponential decay property as an assumption to account for the more general case that ρ could be strictly less than γ, as detailed in Appendix A in our online report Qu et al. (2019).

Our third assumption can be interpreted as an ergodicity condition which ensures that the state-action pairs are sufficiently visited.

Algorithm 1: SAC: Scalable Actor Critic

Input: $\theta_i(0)$; parameter κ; T, length of each episode; step size parameters h, t_0, η.

for $m = 0, 1, 2, \ldots$ **do**

  Sample initial state $s(0) \sim \pi_0$, each agent i takes action $a_i(0) \sim \zeta_i^{\theta_i(m)}(\cdot|s_i(0))$, receives reward $r_i(0) = r_i(s_i(0), a_i(0))$.

  Initialize $\hat{Q}_i^0 \in \mathbb{R}^{\mathcal{S}_{N_i^\kappa} \times \mathcal{A}_{N_i^\kappa}}$ to be the all zero vector.

  for $t = 1$ **to** T **do**

    Get state $s_i(t)$, take action $a_i(t) \sim \zeta_i^{\theta_i(m)}(\cdot|s_i(t))$, get reward $r_i(t) = r_i(s_i(t), a_i(t))$.

    Update the truncated Q function with step size $\alpha_{t-1} = \frac{h}{t-1+t_0}$,

    $\hat{Q}_i^t(s_{N_i^\kappa}(t-1), a_{N_i^\kappa}(t-1)) =$
    $(1-\alpha_{t-1})\hat{Q}_i^{t-1}(s_{N_i^\kappa}(t-1), a_{N_i^\kappa}(t-1)) + \alpha_{t-1}(r_i(t-1) + \gamma \hat{Q}_i^{t-1}(s_{N_i^\kappa}(t), a_{N_i^\kappa}(t)))$,

    $\hat{Q}_i^t(s_{N_i^\kappa}, a_{N_i^\kappa}) = \hat{Q}_i^{t-1}(s_{N_i^\kappa}, a_{N_i^\kappa})$ for $(s_{N_i^\kappa}, a_{N_i^\kappa}) \neq (s_{N_i^\kappa}(t-1), a_{N_i^\kappa}(t-1))$.

  end

  Each agent i calculates approximated gradient,

  $\hat{g}_i(m) = \sum_{t=0}^{T} \gamma^t \frac{1}{n} \sum_{j \in N_i^\kappa} \hat{Q}_j^T(s_{N_j^\kappa}(t), a_{N_j^\kappa}(t)) \nabla_{\theta_i} \log \zeta_i^{\theta_i(m)}(a_i(t)|s_i(t))$.

  Each agent i conducts gradient step $\theta_i(m+1) = \theta_i(m) + \eta_m \hat{g}_i(m)$ with $\eta_m = \frac{\eta}{\sqrt{m+1}}$.

end

Assumption 3 (Sufficient Local exploration) *There exists positive integer τ and $\sigma \in (0,1)$ s.t. under any fixed policy θ and any initial state-action $(s,a) \in \mathcal{S} \times \mathcal{A}$, $\forall i \in \mathcal{N}, \forall (s'_{N_i^\kappa}, a'_{N_i^\kappa}) \in \mathcal{S}_{N_i^\kappa} \times \mathcal{A}_{N_i^\kappa}$, we have $P((s_{N_i^\kappa}(\tau), a_{N_i^\kappa}(\tau)) = (s'_{N_i^\kappa}, a'_{N_i^\kappa}) | (s(1), a(1)) = (s,a)) \geq \sigma$.*

Assumption 3 requires that every state action pair in the κ-hop neighborhood must be visited with some positive probability after some time. This type of assumption is common for finite time convergence results in RL. For example, in Srikant and Ying (2019), it is assumed that every state-action pair is visited with positive probability in the stationary distribution and the state-action distribution converges to the stationary distribution with some rate. This implies our assumption which is weaker in the sense that we only require local state-action pair $(s_{N_i^\kappa}, a_{N_i^\kappa})$ to be visited as opposed to the full state-action pair (s,a). Having said that, we note Assumption 3 does not consider the exploration-exploitation trade-off, which is a challenging issue even in single agent RL. We leave the study of exploration-exploitation in the multi-agent networked setting as future work.

Finally, we assume boundedness and Lipschitz continuity of the gradients, which is standard in the RL literature.

Assumption 4 (Bounded and Lipschitz continuous gradient) *For any i, a_i, s_i and θ_i, we assume $\|\nabla_{\theta_i} \log \zeta_i^{\theta_i}(a_i|s_i)\| \leq L_i$. As a result, $\|\nabla_\theta \log \zeta^\theta(a|s)\| \leq L = \sqrt{\sum_{i=1}^n L_i^2}$. Further, assume $\nabla J(\theta)$ is L'-Lipschitz continuous in θ.*

Theorem 5 *Under Assumption 1, 2, 3 and 4, for any $\delta \in (0,1)$, $M \geq 3$, suppose the critic step size $\alpha_t = \frac{h}{t+t_0}$ satisfies $h \geq \frac{1}{\sigma} \max(2, \frac{1}{1-\sqrt{\gamma}})$, $t_0 \geq \max(2h, 4\sigma h, \tau)$; and the actor step size satisfies $\eta_m = \frac{\eta}{\sqrt{m+1}}$ with $\eta \leq \frac{1}{4L'}$. Further, if the inner loop length T is large enough s.t.*

$T+1 \geq \log_\gamma \frac{c(1-\gamma)}{\bar{r}} + (\kappa+1)\log_\gamma \rho$ *and*

$$\frac{C_a(\frac{\delta}{2nM}, T)}{\sqrt{T+t_0}} + \frac{C'_a}{T+t_0} \leq \frac{2c\rho^{\kappa+1}}{(1-\gamma)^2}, \tag{8}$$

where $C_a(\delta, T) = \frac{6\bar{\epsilon}}{1-\sqrt{\gamma}}\sqrt{\frac{\tau h}{\sigma}[\log(\frac{2\tau T^2}{\delta}) + f(\kappa)\log SA]}$, $C'_a = \frac{2}{1-\sqrt{\gamma}}\max(\frac{16\bar{\epsilon}h\tau}{\sigma}, \frac{2\bar{r}}{1-\gamma}(\tau+t_0))$, *with* $\bar{\epsilon} = 4\frac{\bar{r}}{1-\gamma} + 2\bar{r}$ *and we recall that* $f(\kappa) = \max_i |N_i^\kappa|$ *is the size of the largest* κ*-neighborhood. Then, with probability at least* $1-\delta$*,*

$$\frac{\sum_{m=0}^{M-1}\eta_m\|\nabla J(\theta(m))\|^2}{\sum_{m=0}^{M-1}\eta_m} \leq \frac{\frac{2\bar{r}}{\eta(1-\gamma)} + \frac{8\bar{r}^2L^2}{(1-\gamma)^4}\sqrt{\log M \log\frac{4}{\delta}} + \frac{96\bar{r}^2L'^2L^2}{(1-\gamma)^4}\eta\log M}{\sqrt{M+1}} + \frac{12L^2c\bar{r}}{(1-\gamma)^5}\rho^{\kappa+1}. \tag{9}$$

The proof of Theorem 5 can be found in Appendix D in our online report Qu et al. (2019). To interpret the result, note that the first term in (9) converges to 0 in the order of $\tilde{O}(\frac{1}{\sqrt{M}})$ and the second term, which we denote as ε_κ, is the bias caused by the truncation of the Q-function and it scales in the order of $O(\rho^{\kappa+1})$. As such, our method SAC will eventually find an $O(\rho^{\kappa+1})$-approximation of a stationary point of the objective function $J(\theta)$, which could be very close to a true stationary point even for small κ as ε_κ decays exponentially in κ.

In terms of complexity, (9) gives that, to reach a $O(\varepsilon_\kappa)$-approximate stationary point, the number of outer-loop iterations required is $M \geq \tilde{\Omega}(\frac{1}{\varepsilon_\kappa{}^2}poly(\bar{r}, L, L', \frac{1}{(1-\gamma)}))$, which scales polynomially with the parameters of the problem. We emphasize that it does not scale exponentially with n. Further, since the left hand side of (8) decays to 0 as T increases in the order of $\tilde{O}(\frac{1}{\sqrt{T}})$ and the right hand side of (8) is in the same order as $O(\varepsilon_\kappa)$, the inner-loop length required is $T \geq \tilde{\Omega}(\frac{1}{\varepsilon_\kappa^2}poly(\tau, \frac{1}{\sigma}, \frac{1}{1-\gamma}, \bar{r}, f(\kappa)))$. Parameters τ and $\frac{1}{\sigma}$ are from Assumption 3 and they scale with the local state-action space size of the largest κ-hop neighborhood. Therefore, the inner-loop length required scale with the size of the local state-action space of the largest κ-neighborhood, which is much smaller than the full state-action space size when the graph is sparse.[2]

4. Conclusion and Discussion

This paper proposes a SAC algorithm that provably finds a close-to-stationary point of $J(\theta)$ in time that scales with the local state-action space size of the largest κ-hop neighbor, which can be much smaller than the full state-action space size when the graph is sparse. This perhaps represents the first scalable RL method for localized control of multi-agent networked systems with such provable guarantee. In addition, the framework underlying SAC, including the truncated Q-function (5) and truncated policy gradient (Lemma 7), is a contribution in its own right and could potentially lead to other scalable RL methods for networked systems, including TD-λ variants and Q-learning/SARSA type methods. Additionally, other future directions include further investigation into the exponential decay property, the trade-off between exploration and exploitation.

Acknowledgments

The research was supported by Resnick Sustainability Institute Fellowship, NSF CAREER 1553407, ONR YIP, AFOSR YIP, the PIMCO Fellowship, and NSF grants AitF-1637598, CNS-1518941.

2. This requirement on T could potentially be further reduced if we do a warm start for the inner-loop, as the Q-estimate from the previous outer-loop should be already a good estimate for the current outer-loop. We leave the finite time analysis of the warm start variant as future work.

References

Dimitri P Bertsekas and John N Tsitsiklis. *Neuro-dynamic programming*, volume 5. Athena Scientific Belmont, MA, 1996.

Vincent D Blondel and John N Tsitsiklis. A survey of computational complexity results in systems and control. *Automatica*, 36(9):1249–1274, 2000.

Lucian Bu, Robert Babu, Bart De Schutter, et al. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(2):156–172, 2008.

Eric Cator and Piet Van Mieghem. Second-order mean-field susceptible-infected-susceptible epidemic threshold. *Physical review E*, 85(5):056111, 2012.

Deepayan Chakrabarti, Yang Wang, Chenxi Wang, Jurij Leskovec, and Christos Faloutsos. Epidemic thresholds in real networks. *ACM Transactions on Information and System Security (TISSEC)*, 10(4):1, 2008.

Caroline Claus and Craig Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. *AAAI/IAAI*, 1998:746–752, 1998.

Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. Benchmarking deep reinforcement learning for continuous control. In *International Conference on Machine Learning*, pages 1329–1338, 2016.

Carlos Guestrin, Daphne Koller, Ronald Parr, and Shobha Venkataraman. Efficient solution algorithms for factored mdps. *Journal of Artificial Intelligence Research*, 19:399–468, 2003.

Junling Hu and Michael P Wellman. Nash q-learning for general-sum stochastic games. *Journal of machine learning research*, 4(Nov):1039–1069, 2003.

Soummya Kar, José MF Moura, and H Vincent Poor. Qd-learning: A collaborative distributed strategy for multi-agent reinforcement learning through consensus + innovations. *IEEE Transactions on Signal Processing*, 61(7):1848–1862, 2013.

Michael Kearns and Daphne Koller. Efficient reinforcement learning in factored mdps. In *IJCAI*, volume 16, pages 740–747, 1999.

Dong Li, Dongbin Zhao, Qichao Zhang, and Yaran Chen. Reinforcement learning and deep learning based lateral control for autonomous driving [application notes]. *IEEE Computational Intelligence Magazine*, 14(2):83–98, 2019.

Michael L Littman. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*, pages 157–163. Elsevier, 1994.

Michael L Littman. Value-function reinforcement learning in markov games. *Cognitive Systems Research*, 2(1):55–66, 2001.

Mateu Llas, Pablo M Gleiser, Juan M López, and Albert Díaz-Guilera. Nonequilibrium phase transition in a model for the propagation of innovations among economic agents. *Physical Review E*, 68(6):066101, 2003.

Andrey Y. Lokhov, Marc Mézard, and Lenka Zdeborová. Dynamic message-passing equations for models with unidirectional dynamics. *Phys. Rev. E*, 91:012811, Jan 2015. doi: 10.1103/PhysRevE.91.012811. URL https://link.aps.org/doi/10.1103/PhysRevE.91.012811.

Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*, pages 6379–6390, 2017.

Sergio Valcarcel Macua, Jianshu Chen, Santiago Zazo, and Ali H Sayed. Distributed policy evaluation under multiple behavior strategies. *IEEE Transactions on Automatic Control*, 60(5):1260–1274, 2015.

Adwaitvedant Mathkar and Vivek S Borkar. Distributed reinforcement learning via gossip. *IEEE Transactions on Automatic Control*, 62(3):1465–1470, 2017.

Laetitia Matignon, Guillaume J Laurent, and Nadine Le Fort-Piat. Independent reinforcement learners in cooperative markov games: a survey regarding coordination problems. *The Knowledge Engineering Review*, 27(1):1–31, 2012.

Wenjun Mei, Shadi Mohagheghi, Sandro Zampieri, and Francesco Bullo. On the dynamics of deterministic epidemic propagation over networks. *Annual Reviews in Control*, 44:116–128, 2017.

Nicolas Meuleau, Milos Hauskrecht, Kee-Eung Kim, Leonid Peshkin, Leslie Pack Kaelbling, Thomas L Dean, and Craig Boutilier. Solving very large weakly coupled markov decision processes. In *AAAI/IAAI*, pages 165–172, 1998.

Marc Mezard and Andrea Montanari. *Information, physics, and computation*. Oxford University Press, 2009.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

Ian Osband and Benjamin Van Roy. Near-optimal reinforcement learning in factored mdps. In *Advances in Neural Information Processing Systems*, pages 604–612, 2014.

Christos H Papadimitriou and John N Tsitsiklis. The complexity of optimal queuing network control. *Mathematics of Operations Research*, 24(2):293–305, 1999.

Guannan Qu and Na Li. Exploiting fast decaying and locality in multi-agent mdp with tree dependence structure. *arXiv preprint arXiv:1909.06900*, 2019.

Guannan Qu, Adam Wierman, and Na Li. Scalable reinforcement learning of localized policies for multi-agent networked systems. *arXiv preprint arXiv:1912.02906*, 2019.

Michael Rotkowitz and Sanjay Lall. A characterization of convex problems in decentralized control. *IEEE transactions on Automatic Control*, 50(12):1984–1996, 2005.

Faryad Darabi Sahneh, Caterina Scoglio, and Piet Van Mieghem. Generalized epidemic mean-field model for spreading processes over multilayer complex networks. *IEEE/ACM Transactions on Networking (TON)*, 21(5):1609–1620, 2013.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

R Srikant and Lei Ying. Finite-time error bounds for linear stochastic approximation and td learning. *arXiv preprint arXiv:1902.00923*, 2019.

Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.

John N Tsitsiklis and Benjamin Van Roy. Analysis of temporal-diffference learning with function approximation. In *Advances in neural information processing systems*, pages 1075–1081, 1997.

Werner Vogels, Robbert van Renesse, and Ken Birman. The power of epidemics: Robust communication for large-scale distributed systems. *SIGCOMM Comput. Commun. Rev.*, 33(1):131–135, January 2003. ISSN 0146-4833. doi: 10.1145/774763.774784. URL http://doi.acm.org/10.1145/774763.774784.

Hoi-To Wai, Zhuoran Yang, Zhaoran Wang, and Mingyi Hong. Multi-agent reinforcement learning via double averaging primal-dual optimization. *arXiv preprint arXiv:1806.00877*, 2018.

Peter Whittle. Restless bandits: Activity allocation in a changing world. *Journal of applied probability*, 25(A):287–298, 1988.

Zijian Wu, Qing-Shan Jia, and Xiaohong Guan. Optimal control of multiroom hvac system: An event-based approach. *IEEE Transactions on Control Systems Technology*, 24(2):662–669, 2016.

Kaiqing Zhang, Zhuoran Yang, Han Liu, Tong Zhang, and Tamer Başar. Fully decentralized multi-agent reinforcement learning with networked agents. *arXiv preprint arXiv:1802.08757*, 2018.

Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *arXiv preprint arXiv:1911.10635*, 2019.

Rick Zhang and Marco Pavone. Control of robotic mobility-on-demand systems: a queueing-theoretical perspective. *The International Journal of Robotics Research*, 35(1-3):186–203, 2016.

Xuan Zhang, Wenbo Shi, Bin Yan, Ali Malkawi, and Na Li. Decentralized and distributed temperature control via hvac systems in energy efficient buildings. *arXiv preprint arXiv:1702.03308*, 2017.

Alessandro Zocca. Temporal starvation in multi-channel csma networks: an analytical framework. *Queueing Systems*, 91(3-4):241–263, 2019.