SCIENCE CHINA
Information Sciences

• LETTER •

December 2025, Vol. 68, Iss. 12, 229203:1–229203:2
https://doi.org/10.1007/s11432-024-4578-2

Multi-agent robust policy evaluation for reinforcement learning via primal-dual online time-averaging

Gang CHEN[1*], Changli PU[1], Yaoyao ZHOU[1], Xiumin LI[1] & Huimiao CHEN[2]

[1]*College of Automation, Chongqing University, Chongqing 400044, China*
[2]*Tsinghua Laboratory of Brain and Intelligence, Tsinghua University, Beijing 100084, China*

Received 16 May 2024/Revised 26 August 2024/Accepted 14 July 2025/Published online 23 October 2025

Citation Chen G, Pu C L, Zhou Y Y, et al. Multi-agent robust policy evaluation for reinforcement learning via primal-dual online time-averaging. Sci China Inf Sci, 2025, 68(12): 229203, https://doi.org/10.1007/s11432-024-4578-2

Reinforcement learning aims to find the best sequence of actions that will generate the optimal return. Since some complex tasks require the collaborative cooperation of multiple agents, multi-agent reinforcement learning (MARL) has attracted much more attention in recent years.

Motivated by the real applications, two classes of MARL are investigated in this study. In the first kind of MARL, we consider a group of agents that carry out the parallel computing to learn the value function of a given joint policy. For the second kind of MARL, we partition the state space into many subspaces and each agent implements the distributed exploration in each subspace by a given policy. Inspired by [1, 2], we mainly focus on the distributed policy evaluation problem of MARL.

Most of the MARL studies are feasible under the framework of undirected or fixed topologies [1,3]. Considering the real scenarios, the directed and time-varying communication networks are more suitable. For example, due to the actual privacy consideration, an agent may broadcast its local information to another agent, but the receiver may be unwilling to transmit the local information back to the previous agent. In this study, we investigate MARL on a sequence of time-varying directed and jointly connected communication topologies. Motivated by the Laplacian averaging [4], we combine the MARL with primal-dual running-time averaging in the process of policy evaluation.

The main contributions are twofold. First, we establish a robust distributed policy evaluation algorithm with primal-dual online time-averaging. The online time averaging scheme has the filtering capability and thus can reduce the impact of noise. Second, the more general communication structures with time-varying directed and jointly connected topologies are considered in this work. Our algorithm is feasible in the relaxed settings and thus our results are more general as compared with the existing undirected graph-based algorithms [1, 3]. In addition, our analysis method is different from the existing studies [1–3]. In fact, considering the primal-dual optimization and online time-averaging, the analyses become more challenging. Motivated by the time-average analyses [4], we shed some light on its application in the online learning.

Problem description. Let $G = (V, E)$ describe the communication graph, where $V = \{1, 2, \ldots, N\}$ denotes agents and $E \subseteq V \times V$ represents the communication edges. Let $N_i = \{j \in V \,|\, (j, i) \in E\}$ denote the in-neighbor set. The weights are represented by the adjacency matrix $\mathcal{A} = [a_{ij}]$, where the element a_{ij} denotes the weight of the edge (i, j). The laplacian matrix $\mathcal{L}$ is defined as $\mathcal{L} = \mathcal{D} - \mathcal{A}$, where $\mathcal{D} = [d_i]$ is the diagonal in-degree matrix with the elements $d_i = \sum_{j=1}^{N} a_{ij}$. A multi-agent Markov decision process (MDP) can be defined as a quintuple $(S, \{A_i\}_{i=1}^N, P_{\boldsymbol{sa}}, \{R_i\}_{i=1}^N, \gamma)$, where S is the state space; A_i is the action space of agent i; $P_{\boldsymbol{sa}}$ denotes the transition probability from state $\boldsymbol{s} \in S$ to any other state $\boldsymbol{s}' \in S$ under a joint action $\boldsymbol{a} = (a_1, \ldots, a_N) \in A_1 \times \cdots \times A_N$; $R_i = R_i(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{s}')$ represents the reward of agent i; $\gamma \in [0, 1)$ is the discount factor. Under a joint policy $\boldsymbol{\pi}$, we can define the global reward as an average of the local rewards, i.e., $R_c^{\boldsymbol{\pi}}(\boldsymbol{s}) = \frac{1}{N} \sum_{i=1}^{N} R_i^{\boldsymbol{\pi}}(\boldsymbol{s})$, where $R_i^{\boldsymbol{\pi}}(\boldsymbol{s}) := \mathbb{E}_{\boldsymbol{a} \sim \boldsymbol{\pi}(\cdot|\boldsymbol{s})}[R_i(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{s}')]$ is the local reward of agent i at state $\boldsymbol{s}$ if it follows the fixed policy $\boldsymbol{\pi}$. Under this fixed policy, we define the transition matrix as $\boldsymbol{P}^{\pi}$, whose elements denote the probability of multi-agent MDP to take an action $\boldsymbol{a}$ at state $\boldsymbol{s}$ and reach the next state $\boldsymbol{s}'$, i.e., $[\boldsymbol{P}^{\boldsymbol{\pi}}]_{\boldsymbol{s},\boldsymbol{s}'} = P(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{\pi}) = \sum_{\boldsymbol{a} \in A} \boldsymbol{\pi}(\boldsymbol{a}|\boldsymbol{s}) \cdot P(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a})$. We assume that this Markov chain is aperiodic and irreducible and there is a stationary distribution of states $\mu^{\boldsymbol{\pi}}(\boldsymbol{s})$ with the policy $\boldsymbol{\pi}$.

The policy evaluation means to learn the value function under a given joint policy $\boldsymbol{\pi}$. The value function can be described as $V^{\boldsymbol{\pi}}(\boldsymbol{s}) := \mathbb{E}[\sum_{p=0}^{\infty} \gamma^p R_c^{\boldsymbol{\pi}}(\boldsymbol{s}_p) | \boldsymbol{s}_0 = \boldsymbol{s}, \boldsymbol{\pi}]$. Then we define $\boldsymbol{V}^{\boldsymbol{\pi}} = \boldsymbol{R}_c^{\boldsymbol{\pi}} + \gamma \boldsymbol{P}^{\boldsymbol{\pi}} \boldsymbol{V}^{\boldsymbol{\pi}}$, where $\boldsymbol{R}_c^{\boldsymbol{\pi}}$ is constructed by stacking up $R_c^{\boldsymbol{\pi}}(\boldsymbol{s})$, $\boldsymbol{P}^{\boldsymbol{\pi}}$ is the transition matrix. Let $V_\theta(\boldsymbol{s}) = \boldsymbol{\phi}^\top(\boldsymbol{s})\theta$ be the approximation of the value function, where $\boldsymbol{\phi}(\boldsymbol{s}) \in \mathbb{R}^d$ is a feature vector such as the feature mapping of neural network and $\theta \in \mathbb{R}^d$ denote the parameters to be estimated. Let $\boldsymbol{V}_\theta$ represent a vector of $V_\theta(\boldsymbol{s})$

* Corresponding author (email: chengang@cqu.edu.cn)

for all states and $\boldsymbol{\Phi} = [\boldsymbol{\phi}(1), \boldsymbol{\phi}(2), \ldots, \boldsymbol{\phi}(\boldsymbol{s})]^\top \in \mathbb{R}^{|S|\times d}$. Thus, the policy evaluation problem is simplified to find a parameter θ that minimizes the mean squared projected Bellman error (MSPBE), i.e., $J(\theta) = \frac{1}{2}\|\boldsymbol{\Pi}_{\boldsymbol{\Phi}}(\boldsymbol{V}_\theta - \gamma \boldsymbol{P}^{\boldsymbol{\pi}}\boldsymbol{V}_\theta - \boldsymbol{R}_c^{\boldsymbol{\pi}})\|_{\boldsymbol{D}}^2 + \frac{1}{2}\rho\|\theta\|^2$, where $\boldsymbol{D} = \mathrm{diag}(\mu^{\boldsymbol{\pi}}(\boldsymbol{s}))$ with the entries $\mu^{\boldsymbol{\pi}}(\boldsymbol{s})$ being the stationary distribution under the policy $\boldsymbol{\pi}$, $\boldsymbol{\Pi}_{\boldsymbol{\Phi}}$ donates the projection matrix onto the linear space $\{\boldsymbol{\Phi}\theta, \theta \in \mathbb{R}^d\}$, $\frac{1}{2}\rho\|\theta\|^2$ is the regularization part with $\rho > 0$. Further, we rewrite MSPBE in a standard weighted least-square form $J(\theta) = \frac{1}{2}\|\boldsymbol{A}\theta - \boldsymbol{b}\|_{\boldsymbol{C}^{-1}}^2 + \frac{1}{2}\rho\|\theta\|^2$ where $\boldsymbol{A} = \mathbb{E}_{\mu^{\boldsymbol{\pi}}}[\boldsymbol{\phi}(\boldsymbol{s})(\boldsymbol{\phi}(\boldsymbol{s}) - \gamma\boldsymbol{\phi}(\boldsymbol{s}'))^\top]$, $\boldsymbol{b} = \mathbb{E}_{\mu^{\boldsymbol{\pi}}}[R_c^{\boldsymbol{\pi}}(\boldsymbol{s})\boldsymbol{\phi}(\boldsymbol{s})]$, $\boldsymbol{C} = \mathbb{E}_{\mu^{\boldsymbol{\pi}}}[\boldsymbol{\phi}(\boldsymbol{s})\boldsymbol{\phi}(\boldsymbol{s})^\top]$.

The policy evaluation problems can be reformulated as a consensus form, i.e., $\min_{\theta_i} \frac{1}{N}\sum_{i=1}^N J_i(\theta_i)$. Further, the primal-dual form is described as

$$\begin{aligned} &\min_{\theta=\mathrm{col}(\theta_i)} \max_{\omega=\mathrm{col}(\omega_i)} J(\theta,\omega) = \frac{1}{N}\sum_{i=1}^N J_i(\theta_i,\omega_i) \\ &\text{subject to } \theta_1 = \cdots = \theta_N, \omega_1 = \cdots = \omega_N \end{aligned} \tag{1}$$

with $J_i(\theta_i,\omega_i) = \omega_i^\top(\hat{\boldsymbol{A}}_i\theta_i - \hat{\boldsymbol{b}}_i) - \frac{1}{2}\omega_i^\top\hat{\boldsymbol{C}}_i\omega_i + \frac{1}{2}\rho\|\theta_i\|^2$ and $\hat{\boldsymbol{A}}_i = \hat{\boldsymbol{A}}$, $\hat{\boldsymbol{C}}_i = \hat{\boldsymbol{C}}$ for the parallel computing based reinforcement learning.

Distributed policy evaluation algorithm. The primal-dual policy evaluation protocol with online time-averaging is presented as follows:

$$\begin{aligned} \boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t - \sigma(L_t \otimes I_d)\boldsymbol{\theta}_t - \eta_t g_{\boldsymbol{\theta}_t}, \\ \boldsymbol{\theta}_{t+1}^{\mathrm{a}} &= \frac{t-1}{t}\boldsymbol{\theta}_t^{\mathrm{a}} + \frac{1}{t}\boldsymbol{\theta}_t, \\ \boldsymbol{\omega}_{t+1} &= \boldsymbol{\omega}_t - \sigma(L_t \otimes I_d)\boldsymbol{\omega}_t + \eta_t g_{\boldsymbol{\omega}_t}, \\ \boldsymbol{\omega}_{t+1}^{\mathrm{a}} &= \frac{t-1}{t}\boldsymbol{\omega}_t^{\mathrm{a}} + \frac{1}{t}\boldsymbol{\omega}_t, \end{aligned} \tag{2}$$

where $\boldsymbol{\theta_t} = \mathrm{col}(\theta_1, \ldots, \theta_N)$, $\boldsymbol{\omega_t} = \mathrm{col}(\omega_1, \ldots, \omega_N)$ are the primal and dual variables, respectively; σ is the consensus stepsize with $0 < \sigma < d_{\max}$ and $d_{\max}$ denoting the maximum of in-degree; $g_{\boldsymbol{\theta}_t} = \nabla_{\boldsymbol{\theta_t}} J(\boldsymbol{\theta_t}, \boldsymbol{\omega_t})$, $g_{\boldsymbol{\omega}_t} = \nabla_{\boldsymbol{\omega_t}} J(\boldsymbol{\theta_t}, \boldsymbol{\omega_t})$; η_t is the learning rate.

Main results. We first show that the consensus constraints in (1) are satisfied for Algorithm (2). Let $\theta_0 := \sup_{s=1,\ldots,t}\|\boldsymbol{\theta}_s\|$, $\omega_0 := \sup_{s=1,\ldots,t}\|\boldsymbol{\omega}_s\|$, $g_{\theta_0} := \sup_{s=1,\ldots,t}\|g_{\boldsymbol{\theta}_s}\|$, $g_{\omega_0} := \sup_{s=1,\ldots,t}\|g_{\boldsymbol{\omega}_s}\|$.

Theorem 1. For Algorithm (2), choosing the learning rate $\eta_t = \frac{\varepsilon}{T^\triangle}$ with $\varepsilon > 0$ and $0.5 < \triangle < 1$ or $\eta_t = \frac{\varepsilon}{\sqrt{t}}$, the consensus constraints on the primal and dual variables in (1) are satisfied as $T \to \infty$.

Then, we provide the convergence analyses of Algorithm (2) under the constant learning rate and the time-varying rate, respectively.

Theorem 2. In Algorithm (2), let T denote a fixed time number of iterations. Setting the learning rate to $\eta_t = \frac{\varepsilon}{T^\triangle}$ with $0.5 < \triangle < 1$ and $\varepsilon < T^\triangle$, Algorithm (2) solves the distributed policy evaluation problem (1) and the time averages of the primal and dual variables converge to the saddle point $(\boldsymbol{\theta}^*, \boldsymbol{\omega}^*)$ with the convergence rate given by

$$\begin{aligned} -\frac{1}{2\varepsilon T^{1-\triangle}}(\alpha_{\boldsymbol{\omega}} + \beta_{\boldsymbol{\theta}}) &\leqslant J(\boldsymbol{\theta}_{T+1}^{\mathrm{a}}, \boldsymbol{\omega}_{T+1}^{\mathrm{a}}) - J(\boldsymbol{\theta}^*, \boldsymbol{\omega}^*) \\ &\leqslant \frac{1}{2\varepsilon T^{1-\triangle}}(\alpha_{\boldsymbol{\theta}} + \beta_{\boldsymbol{\omega}}), \end{aligned} \tag{3}$$

where $\alpha_{\boldsymbol{\theta}} = 4\theta_0^2 + 4N\theta_0 g_{\theta_0} + \frac{2N^2\theta_0\nu g_{\theta_0}}{1-r} + (g_{\theta_0}^2\varepsilon^2 + 4Ng_{\theta_0}^2\varepsilon^2 + \frac{2N^2 g_{\theta_0}^2\nu\varepsilon^2}{1-r})T^{1-2\triangle}$, $\beta_{\boldsymbol{\omega}} = 4\omega_0^2 + 8N\omega_0 g_{\omega_0} + \frac{4N^2 g_{\omega_0}\nu\omega_0}{1-r} + (g_{\omega_0}^2\varepsilon^2 + 8Ng_{\omega_0}^2\varepsilon^2 + \frac{4N^2 g_{\omega_0}^2\nu\varepsilon^2}{1-r})T^{1-2\triangle}$.

Theorem 3. Similar to the setting of Theorem 2, the learning rate is set to $\eta_t = \frac{\varepsilon}{\sqrt{t}}$ with $\varepsilon > 0$. Then, we have

$$\begin{aligned} &-\frac{1}{t}(\alpha'_{\omega_1} + \beta'_{\theta_1}) - \frac{1}{\sqrt{t}}(\alpha'_{\omega_2} + \beta'_{\theta_2}) \leqslant J(\boldsymbol{\theta}_{t+1}^{\mathrm{a}}, \boldsymbol{\omega}_{t+1}^{\mathrm{a}}), \\ &- J(\boldsymbol{\theta}^*, \boldsymbol{\omega}^*) \leqslant \frac{1}{t}(\alpha'_{\theta_1} + \beta'_{\omega_1}) + \frac{1}{\sqrt{t}}(\alpha'_{\theta_2} + \beta'_{\omega_2}) \end{aligned} \tag{4}$$

with $\alpha'_{\theta_1} = \frac{2\theta_0^2}{\varepsilon} + 2N\theta_0 g_{\theta_0} + \frac{N^2\nu\theta_0 g_{\theta_0}}{1-r}$, $\alpha'_{\theta_2} = 4\theta_0^2\varepsilon + g_{\theta_0}^2\varepsilon + 4N\varepsilon g_{\theta_0}^2 + \frac{2N^2 g_{\theta_0}^2\nu\varepsilon}{1-r}$, $\beta'_{\omega_1} = \frac{2\omega_0^2}{\varepsilon} + 4N\omega_0 g_{\omega_0} + \frac{2N^2\nu\omega_0 g_{\omega_0}}{1-r}$, $\beta'_{\omega_2} = 4\omega_0^2\varepsilon + g_{\omega_0}^2\varepsilon + 8N\varepsilon g_{\omega_0}^2 + \frac{4N^2 g_{\omega_0}^2\nu\varepsilon}{1-r}$.

Theorem 2 shows that under constant learning rate $\eta = \frac{\varepsilon}{T^\triangle}$ the convergence rate is $O(\frac{1}{T^{1-\triangle}})$. Theorem 3 shows that under time-varying learning rate $\eta_t = \frac{\varepsilon}{\sqrt{t}}$, the convergence rate is $O(\frac{1}{\sqrt{t}})$.

Experimental studies: case 1 parallel computing. The example of 6 networked Mountain car [5] is considered here, where the state is global, the local reward of each car is set at a random proportion of the reward, and the average is equal to $R_c^\pi(s_p, a_p)$. Let $\gamma = 0.9, d = 16, \rho = 0.2$, $\phi(s) = 2\exp((\|s-c\|^2)/b^2)$. We make a comparison with the inexact ADMM [3]. We set Algorithm (2) to run on the time-varying graphs with car 1 and car 4 are intermittently connected to the network. We choose $\eta = 6/1000^{0.55}$ and $\eta_t = 2/\sqrt{t}$, respectively. Figure 1 shows that our algorithm achieves the same accuracy level (till up to 1% error).

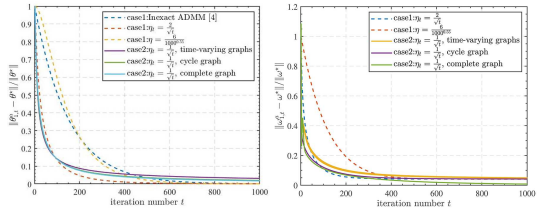

Figure 1 (Color online) Performance of θ and ω.

Experimental studies: case 2 distributed exploring. We partition the 9×6 grid environment into six 3×3 grids and each robot implements the exploring task in a 3×3 grid. Figure 1 shows that the errors can converge to the same level of accuracy even with different communication graphs.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant No. 62073048) and Project for Chongqing's Technological Innovation and Application Development (Grant No. CSTB2023TIAD-GPX0002).

Supporting information Appendix A. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Wai H T, Yang Z, Wang Z, et al. Multi-agent reinforcement learning via double averaging primal-dual optimization. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems, 2018. 9672–9683
2 Sha X, Zhang J, You K, et al. Fully asynchronous policy evaluation in distributed reinforcement learning over networks. Automatica, 2022, 136: 110092
3 Zhao X, Yi P, Li L. Distributed policy evaluation via inexact ADMM in multi-agent reinforcement learning. Control Theor Technol, 2020, 18: 362–378
4 Mateos-Nunez D, Cortes J. Distributed saddle-point subgradient algorithms with Laplacian averaging. IEEE Trans Automat Contr, 2017, 62: 2720–2735
5 Sutton R S, Barto A G. Reinforcement Learning: An Introduction. Cambridge: MIT Press, 2018