Decentralized Policy Gradient Descent Ascent for Safe Multi-Agent Reinforcement Learning

Songtao Lu[1], Kaiqing Zhang[2], Tianyi Chen[3], Tamer Başar[2], Lior Horesh[1]
[1]IBM Thomas J. Watson Research Center, Yorktown Heights, New York 10598, USA
[2]University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA
[3]Rensselaer Polytechnic Institute, Troy, New York 12144, USA
songtao@ibm.com, kzhang66@illinois.edu, chent18@rpi.edu, basar1@illinois.edu, lhoresh@us.ibm.com

Abstract

This paper deals with distributed reinforcement learning problems with safety constraints. In particular, we consider that a team of agents cooperate in a shared environment, where each agent has its individual reward function and safety constraints that involve all agents' joint actions. As such, the agents aim to maximize the team-average long-term return, subject to all the safety constraints. More intriguingly, no central controller is assumed to coordinate the agents, and both the rewards and constraints are only known to each agent locally/privately. Instead, the agents are connected by a peer-to-peer communication network to share information with their neighbors. In this work, we first formulate this problem as a *distributed constrained Markov decision process* (D-CMDP) with networked agents. Then, we propose a decentralized policy gradient (PG) method, *Safe Dec-PG*, to perform policy optimization based on this D-CMDP model over a network. Convergence guarantees, together with numerical results, showcase the superiority of the proposed algorithm. To the best of our knowledge, this is the first decentralized PG algorithm that accounts for the *coupled safety* constraints with a quantifiable convergence rate in multi-agent reinforcement learning. Finally, we emphasize that our algorithm is also novel in solving a class of decentralized stochastic nonconvex-concave minimax optimization problems, where both the algorithm design and corresponding theoretical analysis are of independent interest.

Introduction

Reinforcement learning (RL) has achieved tremendous success in many sequential decision-making problems in (Mnih et al. 2015; Sutton and Barto 2018), such as operations research, optimal control, bounded rationality, machine learning, etc., where an agent explores the interactions with an environment so that it is able to maximize a cumulative reward through this learning process. Beyond applying the classical RL techniques in control systems, physical constraints or safety considerations will also be the key components of determining the performance of an RL system. Especially, this is more important in multi-agent RL (MARL) that models the sequential decision-making of multiple agents in a shared environment, while each agent's objective and the system evolution are both affected by the decisions made by all agents (Nguyen et al. 2014).

Background of Multi-Agent RL

The studies of MARL can be traced back to Q-learning in (Claus and Boutilier 1998) and (Wolpert, Wheeler, and Tumer 1999), with applications to network routing (Boyan and Littman 1994) and power network control (Schneider et al. 1999). However, all the algorithms involved in these works are heuristic without performance guarantees. Recent empirical results of deep multi-agent collaborative RL algorithms can also be found in (Gupta, Egorov, and Kochenderfer 2017; Lowe et al. 2017; Omidshafiei et al. 2017). One of the earliest distributed RL algorithm with convergence guarantees was reported in (Lauer and Riedmiller 2000), which is tailored to the tabular multi-agent *Markov decision process* (MDP) setting, and another one (Nguyen et al. 2014). Then, a distributed Q-learning algorithm was developed with being provably able to learn the desired value function and the optimal stationary control policy at each network agent through a consensus network, where each agent can only communicate with their neighbors (Kar, Moura, and Poor 2013). In the same setup, fully decentralized actor-critic algorithms with function approximation were developed in (Zhang et al. 2018) to handle large or even continuous state-action spaces. However, the convergence in (Zhang et al. 2018) was again established in an asymptotic sense. For a fixed policy, decentralized policy evaluation (value function approximations) approaches for MARL have been studied in (Wai et al. 2018; Doan, Maguluri, and Romberg 2019; Qu et al. 2019). (Please see also the recent surveys (Zhang, Yang, and Başar 2019; Lee et al. 2020) and references therein.)

Related Work

Decentralized and distributed algorithms with quantifiable convergence rate guarantees in the optimization community have been developed for many decades (Nedic, Ozdaglar, and Parrilo 2010) in various scenarios, including (strongly) convex and non-convex cases. Recent advances in distributed non-convex optimization show that decentralized stochastic gradient descent or tracking (DSGD/DSGT) is able to train neural networks much faster than the centralized algorithms in terms of running time numerically (Lian et al. 2017; Lu et al. 2019). Also, it has been indicated in theory that there is a linear speed-up of performing decentralized optimization compared with the centralized one in terms of the number of nodes (Lian et al. 2017; Tang et al. 2018; Lu and Wu 2020).

Algorithm	Rate	Decentralized	Implementation
PGSMD (Rafique et al. 2018)	$\mathcal{O}\left(\epsilon^{-6}\right)$	✗	double-loop
GDA (Lin, Jin, and Jordan 2020)	$\mathcal{O}\left(\epsilon^{-8}\right)$	✗	single-loop
Safe Dec-PG (this work)	$\mathcal{O}\left(\epsilon^{-4}\right)$	✓	single-loop

Table 1: A comparison of stochastic non-convex concave minmax algorithms with convergence to the first-order game-stationary points (FOSPs).

Moreover, in practice, the data would be collected through the sensors over a network, so the distributed learning becomes one of the most powerful signal, data, and information processing tools. (Please see a survey (Chang et al. 2020) of recent distributed non-convex optimization algorithms and their applications.) However, the safe RL problem is not only maximizing rewards but also takes practical issues into account or introduces some prior knowledge of the model in advance, where there would be multiple cumulative long-term reward functions incorporated as the constraints (Paternain et al. 2019a; Wachi and Sui 2020). Unfortunately, none of the existing works deal with the safety constraints that are also non-convex, no need to mention their distributed implementation over a network.

By the primal-dual optimization framework, the safe RL problem can be formulated as a min-max saddle point form by the method of Lagrange multipliers or dualizing the constraints (Boyd and Vandenberghe 2004). However, different from the classical supervised learning, e.g., support vector machine and least squares regression, the policy in RL is mostly parametrized by a (deep) neural network so that the cumulative reward functions are non-convex. Hence, the duality gap in this case is not zero in general, which makes the optimization process much more difficult than the traditional convex-concave min-max problem even in the centralized setting. Interestingly, some recent exciting results illustrate that the duality gap in safe RL problems could be zero (Paternain et al. 2019b) by assuming some oracle that can find the global optimal solution of the Lagrangian with respect to policy. It is inspiring that safe RL might be solved efficiently to high-quality solutions by the non-convex min-max solvers.

During the last few years, solving non-convex min-max saddle-point problems has gained huge popularity and indicated significant power of optimizing the interest of parameters in many machine learning and/or artificial intelligence problems, including adversarial learning, robust neural nets or generative adversarial nets (GANs) training, fair resource allocation (Razaviyayn et al. 2020). The main idea of designing these algorithms is to perform gradient descent and ascent with respect to the objective functions, such as gradient descent ascent (GDA) algorithm (Lin, Jin, and Jordan 2020), multi-GDA (Nouiehed et al. 2019), proximally guided stochastic mirror descent method (PGSMD) (Rafique et al. 2018), and hybrid block successive approximation (HiBSA) (Lu et al. 2020). The difference between GDA and multi-GDA is that the latter performs multiple steps of gradient ascent updates instead of one. Among these algorithms, HiBSA achieves the fastest convergence rate with only a single loop update rule to optimization variables for the deterministic non-convex case. However, there is no theoretical guarantee that HiBSA is amenable to handle the stochasticity of the samples in the non-convex (strongly) concave min-max problems. Further, all these algorithms are centralized, so it is not clear whether they can be used for a multi-agent system. Recently, there are some interesting works regarding the distributed training for a class of GANs (Liu et al. 2020a,b), where the problem is formulated as a decentralized non-convex saddle-point problem. But both of them require that the objective function satisfy the Minty variational inequality (MVI), otherwise, these methods cannot converge to an ϵ-first-order stationary point (FOSP) of the considered problem even the number of iterations is infinite. While in RL/MARL there is no evidence which can indicate that discounted cumulative reward function satisfies MVI again due to the nonconvexity of the loss function when the policy at each node is parametrized by a neural net.

Main Contributions

In this work, by leveraging the min-max saddle-point formulation, we propose the first safe decentralized policy gradient (PG) descent and ascent algorithm, i.e., *Safe Dec-PG*, which is able to deal with a class of multi-agent safe RL problems over a graph. Importantly, we provide theoretical results that quantify the convergence rate of *Safe Dec-PG* to an ϵ-first-order stationary points (FOSP) of the considered non-convex min-max problem in the order of $1/\epsilon^4$ (or equivalently the optimality gap is shrinking in the order of $1/\sqrt{N}$, where N denotes the total number of iterations). When the graph is fully connected in the sense that there is no consensus error (each agent can know all the other agents' policy at each iteration), *Safe Dec-PG* will reduce to a centralized algorithm. Even in this case, the obtained convergence rate is still the state-of-the-art result to the best of our knowledge. A more detailed comparison between proposed *Safe Dec-PG* and other existing stochastic non-convex concave min-max algorithm in the *centralized setting* is shown in Table **??**. The main advantages of *Safe Dec-PG* are highlighted as follows:

- (Simplicity) The structure of implementing the algorithm is single-loop, where the parameters that need to be tuned are only the stepsizes in the minimization and maximization subproblems.
- (Theoretical Guarantees) It is theoretically provable that *Safe Dec-PG* is able to find an ϵ-FOSP of the formulated non-convex min-max problem within $\mathcal{O}(1/\epsilon^4)$ number of iterations, matching the standard convergence rate of *centralized* stochastic gradient descent (SGD) and decentralized SGD to ϵ-FOSPs in non-convex scenarios.
- (Applicability) *Safe Dec-PG* is also a general optimiza-

tion problem solver, which can be applied for dealing with many non-convex min-max problems rather than the RL/MARL problems, and it could be implemented in either a decentralized way over a network or on a single machine.

Multiple numerical results showcase the superiority of the algorithms applied in the problems of safe decentralized RL compared with the classic decentralized methods without safety considerations. Due to the page limitation, proofs of all the lemmas, the main theorem and additional numerical results are included in the supplemental materials.

Safe MARL with Decentralized Agents

In this section, we introduce the background and formulation of the safe MARL problem with decentralized agents.

Multi-Agent Constrained Markov Decision Process (M-CMDP)

Consider a team of n agents operating in a common environment, denoted by $\mathcal{N} = [n]$. No central controller exists to either make the decisions or collect any information for the agents. Agents are instead allowed to communicate with each other over a communication network $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, with $\mathcal{E}$ being the set of communication links that connect the agents. Such a decentralized model with networked agents finds broad applications in distributed cooperative control problems (Fax and Murray 2004; Corke, Peterson, and Rus 2005; Dall'Anese, Zhu, and Giannakis 2013), and has been advocated as one of the most popular paradigms in decentralized MARL (Zhang et al. 2018; Wai et al. 2018; Doan, Maguluri, and Romberg 2019; Qu et al. 2019; Zhang, Yang, and Başar 2019; Lee et al. 2020). More importantly, each agent has some safety constraints, in the forms of bounds on some long term cost, that involve the joint policy of all agents. We formally introduce the following model of *networked multi-agent constrained MDP (M-CMDP)* to characterize this setting.

Definition 1 (*Networked Multi-agent CMDP (M-CMDP)*)**.** *A networked multi-agent CMDP is described by a tuple $(\mathcal{S}, \{\mathcal{A}_i\}_{i\in\mathcal{N}}, P, \{R_i\}_{i\in\mathcal{N}}, \mathcal{G}, \{C_i\}_{i\in\mathcal{N}}, \gamma)$ where $\mathcal{S}$ is the state space shared by all the agents, $\mathcal{A}_i$ is the action space of agent i, and $\mathcal{G}$ is a communication network (a well-connected graph). Let $\mathcal{A} = \prod_{i=1}^n \mathcal{A}_i$ be the joint action space of all agents; then, $R_i : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ and $C_i : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ are the local rewards and cost functions of agent i, and $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the state transition probability of the MDP. $\gamma \in (0, 1)$ denotes the discount factor. The states $\boldsymbol{s}$ and actions $\boldsymbol{a}$ are globally observable, while the rewards and costs are observed locally/privately at each agent.*

The networked M-CMDP proceeds as follows. At time t, each agent i chooses its own action $\boldsymbol{a}_i^t$ given $\boldsymbol{s}^t$, according to its local policy $\pi_i : \mathcal{S} \to \Delta(\mathcal{A}_i)$, which is usually parametrized as $\pi_{\mathbf{w}_i}$ by some parameter $\mathbf{w}_i \in \Theta_i$ with dimension d_i. The networked agents try to learn a joint policy $\pi_{\mathbf{w}_i} : \mathcal{S} \to \Delta(\mathcal{A})$ given by $\pi_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{a}) = \prod_{i\in\mathcal{N}} \pi_{\mathbf{w}_i}(\boldsymbol{s}, \boldsymbol{a}_i)$ with $\boldsymbol{\theta} = [\mathbf{w}_1^\top \ldots \mathbf{w}_n^\top]^\top \in \mathbb{R}^d$, where $d = \sum_{i=1}^n d_i$ denotes the whole problem dimension. As a team, the objective of all agents is to collaboratively maximize the *globally average* return over the network (equivalently to minimize the opposite of it), dictated by $\overline{R}(\boldsymbol{s}, \boldsymbol{a}) = n^{-1} \cdot \sum_{i\in\mathcal{N}} R_i(\boldsymbol{s}, \boldsymbol{a})$, with only its *local* observations of the rewards, subject to some safety constraints dictated by $C_i(\boldsymbol{s}, \boldsymbol{a})$. At each node, there would be multiple safety constraints. These rewards describe different objectives that the agent is required to achieve, such as remaining with a region of the state space, or not running out of memory/battery. Here, we assume that each agent is associated with m cost functions, so $C_i(\boldsymbol{s}, \boldsymbol{a})$ is a mapping from $\mathcal{S} \times \mathcal{A}$ to $\mathbb{R}^m$. Specifically, the team aims to find the joint policy $\pi_{\boldsymbol{\theta}}$ that

$$\min_{\boldsymbol{\theta}\in\Theta} \; J_0^R(\boldsymbol{\theta}) \triangleq \mathbb{E}\left(-\frac{1}{n}\sum_{t\geq 0}\gamma^t \sum_{i\in\mathcal{N}} R_i(\boldsymbol{s}^t, \boldsymbol{a}^t)\,\middle|\, \boldsymbol{s}^0, \pi_{\boldsymbol{\theta}}\right) \tag{1a}$$

$$\text{s.t.} \;\; J_i^C(\boldsymbol{\theta}) \triangleq \mathbb{E}\left(\sum_{t\geq 0}\gamma^t C_i(\boldsymbol{s}^t, \boldsymbol{a}^t)\,\middle|\, \boldsymbol{s}^0, \pi_{\boldsymbol{\theta}}\right) \geq \mathbf{c}_i, \forall i \in \mathcal{N} \tag{1b}$$

where $\Theta = \prod_{i=1}^N \Theta_i$ is the joint policy parameter space, $J_0^R(\boldsymbol{\theta})$ corresponds to the negative team-average discounted long-term return, $J_i^C(\boldsymbol{\theta}) : \mathbb{R}^d \to \mathbb{R}^m$ denotes the long-term costs of agent i, $\mathbf{c}_i \in \mathbb{R}^m, \forall i$ are the lower-bounds of $J_i^C(\boldsymbol{\theta}), \forall i$ that impose the safety constraints, and $\mathbb{E}$ is taken over all randomness including the policy and the underlying Markov chain. Each agent i only has access to its own reward and cost R_i and C_i, and the desired bound $\mathbf{c}_i$. Note that our ensuing results can be straightforwardly generalized to the setting where each agent has different number of costs, at the expense of unnecessarily complicated notations. In general, the long-term return $J_0^R(\boldsymbol{\theta})$ is *non-convex* with respect to the policy parameter $\boldsymbol{\theta}$ (Zhang et al. 2020; Liu et al. 2019; Agarwal et al. 2020), so do the constraint functions $J_i^C(\boldsymbol{\theta}), \forall i$, which makes the problem challenging to solve using the first-order PG methods.

Primal-Dual for Safe M-CMDP

Viewing the team as a single agent, the problem above falls into the regime of the standard constrained MDP (Altman 1999), which has been widely studied in single-agent safe RL. Nonetheless, in a decentralized paradigm, standard RL algorithms for solving CMDP are not applicable, as they require the instantaneous access to the team-average reward and all cost functions $\{C_i\}_{i\in\mathcal{N}}$ (Borkar 2005; Prashanth and Ghavamzadeh 2016; Achiam et al. 2017; Yu et al. 2019; Paternain et al. 2019b). Instead, we re-formulate the problem as a decentralized non-convex optimization problem with non-convex constraints, in order to develop decentralized policy optimization algorithms. In particular, letting $J_i^R(\boldsymbol{\theta}) \triangleq \mathbb{E}(-\sum_{t\geq 0}\gamma^t R_i(\boldsymbol{s}^t, \boldsymbol{a}^t) \,|\, \boldsymbol{s}^0, \pi_{\boldsymbol{\theta}})$, we have the networked M-CMDP as

$$\min_{\{\boldsymbol{\theta}_i\in\Theta\}} \;\; \frac{1}{n}\sum_{i\in\mathcal{N}} J_i^R(\boldsymbol{\theta}_i) \tag{2}$$
$$\text{s.t.} \quad \boldsymbol{\theta}_i = \boldsymbol{\theta}_j \quad j \in \mathcal{N}_i, \quad \mathbf{c}_i - J_i^C(\boldsymbol{\theta}_i) \leq 0, \;\; \forall i \in \mathcal{N},$$

where $\mathcal{N}_i \subseteq \mathcal{N}$ denotes the set of the neighboring agents of agent i over the network, and $\boldsymbol{\theta}_i$ is the local copy of the policy parameter $\boldsymbol{\theta}$ (i.e., the concatenation of all the agents' parameters). By the Lagrangian method (Boyd and Vandenberghe 2004), the problem (2) can be written as

$$\min_{\{\boldsymbol{\theta}_i\in\Theta\}} \max_{\boldsymbol{\lambda}\geq 0} \quad \mathcal{L}(\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_n,\boldsymbol{\lambda}_1,\ldots,\boldsymbol{\lambda}_n) \tag{3a}$$

$$\text{s.t.} \quad \boldsymbol{\theta}_i=\boldsymbol{\theta}_j \quad j\in\mathcal{N}_i, \ \forall i, \tag{3b}$$

where

$$\mathcal{L}(\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_n,\boldsymbol{\lambda}_1,\ldots,\boldsymbol{\lambda}_n) \triangleq \frac{1}{n}\sum_{i\in\mathcal{N}} J_i^R(\boldsymbol{\theta}_i)+\langle g_i(\boldsymbol{\theta}_i),\boldsymbol{\lambda}_i\rangle, \tag{4}$$

$g_i(\boldsymbol{\theta}_i)\triangleq \mathbf{c}_i - J_i^C(\boldsymbol{\theta}_i)$, and $\boldsymbol{\lambda}_1,\ldots,\boldsymbol{\lambda}_n$ denote the dual variables.

Main Challenges of Solving Safe Decentralized RL

To this end, the multi-agent safe RL problem has been formulated as (3). Unfortunately, there is no existing work that is able to solve this problem to its FOSPs with any theoretical guarantees. The main difficulties here are four-fold as follows:

- There are two types of constraints in this problem: one is the consensus equality constraint and the other one is the long term cumulative reward related inequality constraint.
- The constraints and loss functions are both in an expected discounted cumulative reward form and possibly non-convex, while most of the classical non-convex algorithms, e.g., neural nets training, are designed for the case where only the loss functions are non-convex.
- The problem is stochastic in nature and the PG estimate is biased instead of unbiased due to the finite-horizon approximation, so we need extra efforts to quantify how biased estimates affect the convergence results.
- From a min-max saddle-point perspective, the minimization problem is non-convex and the maximization problem is concave (linear), while there would be also a consensus error coupled with both minimization and maximization optimization variables. Disentangling this error from the minimization and maximization processes will result in a significant different theorem proving technique compared with the existing theoretical works.

Therefore, solving this family of stochastic non-convex problems over a graph is much more challenging than the classical ones, e.g., centralized min-max saddle-point problems, decentralized consensus problems, stochastic non-convex problems, and so on. Next, we will propose the new gradient tracking based single loop primal dual algorithm to deal with this M-CMDP problem.

Safe M-CMDP Algorithm

First, we introduce the safe policy gradient used in *Safe Dec-PG* as the following.

Safe Policy Gradient

The search for an optimal policy can thus be performed by applying the gradient descent-type iterative methods to the parametrized optimization problem (3). The gradient of each agent's cumulative loss $J_i^R(\boldsymbol{\theta}_i)$ in (3) can be written as (Baxter and Bartlett 2001)

$$\nabla_{\boldsymbol{\theta}_i} J_i^R(\boldsymbol{\theta}_i)=\mathbb{E}\left[\sum_{t=0}^{\infty}\left(\sum_{\tau=0}^{t}\nabla\log\pi_i(\boldsymbol{a}_i^\tau|\boldsymbol{s}^\tau;\boldsymbol{\theta}_i)\right)\gamma^t R_i(\boldsymbol{s}^t,\boldsymbol{a}^t)\right]$$

where $\{\boldsymbol{a}^t,\boldsymbol{s}^t\}$ are obtained from each trajectory under the joint policy (parametrized by $\{\boldsymbol{\theta}_i,\forall i\}$). When the MDP model is unknown, the stochastic estimate of PG is often used, that is

$$\widehat{\nabla}_{\boldsymbol{\theta}_i} J_i^R(\boldsymbol{\theta}_i)=\sum_{t=0}^{\infty}\left(\sum_{\tau=0}^{t}\nabla\log\pi_i(\boldsymbol{a}_i^\tau|\boldsymbol{s}^\tau;\boldsymbol{\theta}_i)\right)\gamma^t R_i(\boldsymbol{s}^t,\boldsymbol{a}^t),$$

which was proposed in (Baxter and Bartlett 2001) and called the gradient of a partially observable MDP (abbreviated as G(PO)MDP PG). The G(PO)MDP gradient is an unbiased estimator of the PG (Papini et al. 2018; Xu, Gao, and Gu 2020).

Likewise, the stochastic PG estimate of each agent's $J_i^C(\boldsymbol{\theta}_i)$ in (3) can be written as

$$\widehat{\nabla}_{\boldsymbol{\theta}_i} J_i^C(\boldsymbol{\theta}_i)=\sum_{t=0}^{\infty}\left(\sum_{\tau=0}^{t}\nabla\log\pi_i(\boldsymbol{a}_i^\tau|\boldsymbol{s}^\tau;\boldsymbol{\theta}_i)\right)\gamma^t C_i(\boldsymbol{s}^t,\boldsymbol{a}^t).$$

Let $f_i(\boldsymbol{\theta}_i,\boldsymbol{\lambda}_i)\triangleq J_i^R(\boldsymbol{\theta}_i)+\langle\mathbf{c}_i-J_i^C(\boldsymbol{\theta}_i),\boldsymbol{\lambda}_i\rangle,\forall i$ for notational simplicity. Then, the policy gradients with respect to primal variables are

$$\widehat{\nabla}_{\boldsymbol{\theta}_i} f_i(\boldsymbol{\theta}_i,\boldsymbol{\lambda}_i)=\widehat{\nabla}_{\boldsymbol{\theta}_i} J_i^R(\boldsymbol{\theta}_i)-\langle\widehat{\nabla}_{\boldsymbol{\theta}_i} J_i^C(\boldsymbol{\theta}_i),\boldsymbol{\lambda}_i\rangle,\forall i \tag{5}$$

and the policy gradients with respect to dual variables are

$$\widehat{\nabla}_{\boldsymbol{\lambda}_i} f_i(\boldsymbol{\theta}_i,\boldsymbol{\lambda}_i)=\mathbf{c}_i-\widehat{J}_i^C(\boldsymbol{\theta}_i),\forall i \tag{6}$$

where $\widehat{J}_i^C(\boldsymbol{\theta}_i)\triangleq\sum_{t=0}^{\infty}\gamma^t C_i(\boldsymbol{s}^t,\boldsymbol{a}^t|\boldsymbol{s}^0,\pi_{\boldsymbol{\theta}})$. Note that the stochastic gradients in (5) and (6) use only one trajectory of the Markov chain, which may incur large variance. Akin to mini-batch in SGD, a natural solution is to average over K trajectories to obtain the policy gradient with respect to the primal variables denoted as $\widehat{\nabla}_{\boldsymbol{\theta}_i}^K f_i(\boldsymbol{\theta}_i,\boldsymbol{\lambda}_i),\forall i$, and with respect to the dual variables denoted as $\widehat{\nabla}_{\boldsymbol{\lambda}_i}^K f_i(\boldsymbol{\theta}_i,\boldsymbol{\lambda}_i),\forall i$. In simulations, sampling an infinite trajectory may not be tractable, and a finite-horizon approximation of the PGs (5) and (6) is usually used (Chen et al. 2018), which are denoted as $\widehat{\nabla}_{\boldsymbol{\theta}_i}^{T,K} f_i(\boldsymbol{\theta}_i,\boldsymbol{\lambda}_i)$ and $\widehat{\nabla}_{\boldsymbol{\lambda}_i}^{T,K} f_i(\boldsymbol{\theta}_i,\boldsymbol{\lambda}_i)$. Also, we can have a set of globally observable states and actions denoted by $\{\boldsymbol{a}_k^\tau,\boldsymbol{s}_k^\tau\}$, where k denotes the index of trajectories and τ denotes the index of time. Consequently, the stochastic estimate of PG with K trajectories (samples) and a finite-horizon truncation of length T can be expressed as

$$\widehat{\nabla}_{\boldsymbol{\theta}_i}^{T,K} f_i(\boldsymbol{\theta}_i,\boldsymbol{\lambda}_i)=\widehat{\nabla}_{\boldsymbol{\theta}_i}^{T,K} J_i^R(\boldsymbol{\theta}_i)-\langle\widehat{\nabla}_{\boldsymbol{\theta}_i}^{T,K} J_i^C(\boldsymbol{\theta}_i),\boldsymbol{\lambda}_i\rangle, \tag{7a}$$

$$\widehat{\nabla}_{\boldsymbol{\lambda}_i}^{T,K} f_i(\boldsymbol{\theta}_i,\boldsymbol{\lambda}_i)=\mathbf{c}_i-(\widehat{J}_i^C)^{T,K}(\boldsymbol{\theta}_i)\triangleq\widehat{g}_i(\boldsymbol{\theta}_i) \tag{7b}$$

where

$$\widehat{\nabla}_{\boldsymbol{\theta}_i}^{T,K} J_i^R(\boldsymbol{\theta}_i)\triangleq$$

$$\frac{1}{K}\sum_{k=1}^{K}\sum_{t=0}^{T}\left(\sum_{\tau=0}^{t}\nabla\log\pi(\boldsymbol{a}_{i,k}^\tau|\boldsymbol{s}_k^\tau;\boldsymbol{\theta}_i)\right)\gamma^t R_i(\boldsymbol{s}_k^t,\boldsymbol{a}_k^t), \tag{8}$$

$$\widehat{\nabla}^{T,K}_{\boldsymbol{\theta}_i} J_i^C(\boldsymbol{\theta}_i) \triangleq$$
$$\frac{1}{K}\sum_{k=1}^{K}\sum_{t=0}^{T}\left(\sum_{\tau=0}^{t} \nabla \log \pi(\boldsymbol{a}^{\tau}_{i,k}|\boldsymbol{s}^{\tau}_{k};\boldsymbol{\theta}_i)\right)\gamma^t C_i(\boldsymbol{s}^t_k,\boldsymbol{a}^t_k), \quad (9)$$

and $(\widehat{J}_i^C)^{T,K}(\boldsymbol{\theta}_i) \triangleq K^{-1}\sum_{k=1}^{K}\sum_{t=0}^{T}\gamma^t C_i(\boldsymbol{s}^t_k,\boldsymbol{a}^t_k)$. Note that the finite length horizontal truncation will make the stochastic estimate PG become biased.

Safe Dec-PG: Safe Decentralized Policy Gradient

After getting the PG estimates, *Safe Dec-PG* algorithm we proposed is given below. For notational simplicity, in the following we assume the problem dimension is 1. We first update the parameters of the parametrized policy at each node by

$$\boldsymbol{\theta}_i^{r+1} = \sum_{j\in\mathcal{N}_i} \mathbf{W}_{ij}\boldsymbol{\theta}_j^r - \beta^r\boldsymbol{\vartheta}_i^r, \quad (10)$$

where r denotes the index of the iterations, β^r is the stepsize of PG descent, $\boldsymbol{\vartheta}_i^r$ is an auxiliary (tracking) variable (which will be introduced with more details later in (11)), and $\mathbf{W}_{ij}$ is a weight matrix that characterizes the relations among the nodes over graph $\mathcal{G}$.

Next, we provide detailed descriptions about $\mathbf{W}$ and $\boldsymbol{\vartheta}$: 1) The weight matrix is double stochastic (i.e., the graph is well-connected.), which is defined as follows: if there exists a link between node i and node j, then $\mathbf{W}_{ij} > 0$, otherwise $\mathbf{W}_{ij} = 0$, and $\mathbf{W}$ satisfies $\mathbf{W1} = \mathbf{1}$ and $\mathbf{1}^\top\mathbf{W} = \mathbf{1}^\top$. There are many ways of designing the weight matrix based on the connectivity of the graph. The standard ones include Metropolis-Hasting weight, maximum-degree weight, Laplacian weight (Xiao and Boyd 2004; Boyd, Diaconis, and Xiao 2004); 2) due to the partial observability of each agent, the variable $\boldsymbol{\vartheta}_i^r$ here is proposed for approximating the full PG of the network (i.e., $n^{-1}\sum_{i=1}^{n}\widehat{\nabla}_{\boldsymbol{\theta}_i} f_i(\boldsymbol{\theta}_i,\boldsymbol{\lambda}_i)$), and is updated locally as

$$\boldsymbol{\vartheta}_i^{r+1} = \sum_{j\in\mathcal{N}_i}\mathbf{W}_{ij}\boldsymbol{\vartheta}_j^r + \widehat{\nabla}^{T,K}_{\boldsymbol{\theta}_i} f_i(\boldsymbol{\theta}_i^{r+1},\boldsymbol{\lambda}_i^r) - \widehat{\nabla}^{T,K}_{\boldsymbol{\theta}_i} f_i(\boldsymbol{\theta}_i^{r},\boldsymbol{\lambda}_i^r), \forall i \quad (11)$$

with $\boldsymbol{\vartheta}_i^0 \triangleq \mathbf{0}, \forall i$. This update rule is similar to the (stochastic) gradient tracking technique proposed for both classical consensus based (deterministic or stochastic) distributed optimization problems (Di Lorenzo and Scutari 2016; Sun, Daneshmand, and Scutari 2019). But here since we also have dual variable updates, at each time the evaluated gradient is also dependent on $\boldsymbol{\lambda}_i^r$, so it is not clear whether the tracked full PG by $\boldsymbol{\vartheta}_i^r$ is still accurate enough so that the resulting sequence can converge to the stationary points of problem (3). In our performance analysis section, we will show the conditions that can ensure the convergence of *Safe Dec-PG* in solving problem (3).

Algorithm 1 Safe Dec-PG

```
Input: $\boldsymbol{\theta}_i^0, \boldsymbol{\vartheta}_i^0 = \boldsymbol{\lambda}_i^0 = \mathbf{0}, \forall i$
for $r = 1, \ldots$ do
    for Each agent $i$ do
        Update $\boldsymbol{\theta}_i^{r+1}$ by (10)
        Perform rollout to get $\widehat{\nabla}^{T,K}_{\boldsymbol{\theta}_i} f_i(\boldsymbol{\theta}_i^r, \boldsymbol{\lambda}_i^r)$
        Update $\boldsymbol{\vartheta}_i^{r+1}$ by (11)
        Calculate $(\widehat{J}_i^C)^{T,K}(\boldsymbol{\theta}_i^{r+1})$
        Update $\boldsymbol{\lambda}_i^{r+1}$ by (13)
    end for
end for
```

In this work, instead of performing a vanilla dual update, we propose to add a (quadratic) perturbation term (a.k.a. smoothing technique) to the maximization procedure as follows:

$$\boldsymbol{\lambda}_i^{r+1} = \arg\max_{\boldsymbol{\lambda}_i\geq 0}\left\langle \widehat{\nabla}^{T,K}_{\boldsymbol{\lambda}_i} f_i(\boldsymbol{\theta}_i^{r+1},\boldsymbol{\lambda}_i^r), \boldsymbol{\lambda}_i - \boldsymbol{\lambda}_i^r\right\rangle - \frac{1}{2\rho}\|\boldsymbol{\lambda}_i - \boldsymbol{\lambda}_i^r\|^2 - \frac{\gamma^r}{2}\|\boldsymbol{\lambda}_i\|^2, \forall i \quad (12)$$

where $\rho > 0$ is the stepsize of PG ascent in updating $\boldsymbol{\lambda}_i^r$, γ^r (to be defined later) is a diminishing parameter. The perturbation term $\gamma^r/2\|\boldsymbol{\lambda}_i\|^2$ plays one of the most key roles of ensuring the convergence of *Safe Dec-PG*. It adds some (desired) curvature to this subproblem (12) in such a way it is possible to quantify the maximum ascent of our constructed potential function (a Lyapunov-like function that will be used to measure the progress of the proposed algorithm) after the update of $\boldsymbol{\lambda}_i^r$. Then, this parameter gradually reduces the problem curvature to resemble the original subproblem such that the obtained solution is the FOSP of problem (3) rather a deviated one. Note that (12) can also be easily implemented locally by

$$\boldsymbol{\lambda}_i^{r+1} = \mathcal{P}_\Lambda\left((1-\rho\gamma^r)\boldsymbol{\lambda}_i^r + \rho\widehat{\nabla}^{T,K}_{\boldsymbol{\lambda}_i} f_i(\boldsymbol{\theta}_i^{r+1},\boldsymbol{\lambda}_i^r)\right), \forall i \quad (13)$$

where $\mathcal{P}_\Lambda$ denotes the projection operator, and $\Lambda = \{\boldsymbol{\lambda}_i|\boldsymbol{\lambda}_i \geq 0\}, \forall i$ stands for the feasible set.

It can be seen that one of the major advantages of *Safe Dec-PG* is regarding its simplicity of updating rules for all the parameters: 1) a single loop algorithm; 2) each variable can be only updated locally through exchanging the parameters over the communication channel. From the following convergence analysis, we will show that when some mild conditions hold, *Safe Dec-PG* is guaranteed to find the FOSPs of problem (3) by controlling the stepsizes used in the minimization and maximization procedures properly.

Performance Analysis of *Safe Dec-PG*

Before showing our theoretical results, we first give the standard assumptions as follows.

Assumptions

To begin with, we assume that $f_i, g_i, \forall i$ satisfy a Lipschitz continuous condition. To be more specific, we have

Assumption 1. *Assume functions $\nabla f_i(\boldsymbol{\theta}_i,\boldsymbol{\lambda}_i), \forall i$ have L-Lipschitz continuity with respect to $\boldsymbol{\theta}_i, \forall i$ and functions $g_i(\boldsymbol{\theta}_i), \forall i$ have L'-Lipschitz continuity with respect to $\boldsymbol{\theta}_i, \forall i$.*

Next, we assume the connectivity of the graph, which specifies the topology of the communication channel so that the consensus step can be performed in a decentralized way.

Assumption 2. *Assume the network is well-connected (a.k.a. strongly-connected), i.e.,* $\mathbf{W}$ *is a double stochastic matrix. Also* $\underline{\lambda}_{\max}(\mathbf{W}) \triangleq \eta < 1$, *where* $\underline{\lambda}_{\max}(\mathbf{W})$ *denotes the second largest eigenvalue of the weight matrix* $\mathbf{W}$.

Assumption 3. *We assume that the rewards in both objective and constraints are upper bounded by G, i.e.,* $\max\{R_i(\boldsymbol{s}^t, \boldsymbol{a}^t), C_i(\boldsymbol{s}^t, \boldsymbol{a}^t), \forall i\} \leq G$, *and the true PG is upper bounded by* G', *i.e.,* $\|\nabla \log \pi_i(\boldsymbol{a}_i^\tau | \boldsymbol{s}^\tau; \boldsymbol{\theta}_i)\| \leq G', \forall i, \tau$.

The first part of Assumption 3 requires the boundedness of the instantaneous reward, which makes sense in practice since the physical systems commonly output finite magnitudes of responses. The second part requires the partial derivatives of the log function of the policies, i.e., $\|\nabla \log \pi_i(\boldsymbol{a}_i^\tau | \boldsymbol{s}^\tau; \boldsymbol{\theta}_i)\|$ to be bounded, which can be satisfied by e.g., parametrized Gaussian policies.

Assumption 4. *Assume that the Slater condition is satisfied and the size of* Λ *is upper bounded by* σ_λ, *i.e.,* $\Lambda = \{\boldsymbol{\lambda}_i | \boldsymbol{\lambda}_i \geq 0, \|\boldsymbol{\lambda}_i\| \leq \sigma_\lambda\}, \forall i$.

Convergence Rate

Since functions J_i^R and $J_i^C, \forall i$ are possibly non-convex, finding the global optimal solution for this min-max problem is NP-hard in general (Nouiehed, Lee, and Razaviyayn 2018). It is of interest to obtain the FOSPs of problem (1). First, we define the optimality gap as

$$\mathcal{G}(\{\boldsymbol{\theta}_i, \boldsymbol{\lambda}_i, \forall i\}) = \left\| \frac{1}{n} \sum_i^n \nabla f_i(\boldsymbol{\theta}_i, \boldsymbol{\lambda}_i) \right\|$$

$$\frac{1}{n} \sum_{i=1}^n \|\boldsymbol{\lambda}_i - \mathcal{P}_\Lambda[\boldsymbol{\lambda}_i + g_i(\boldsymbol{\theta}_i)]\| + \frac{1}{n} \sum_{i=1}^n \|\boldsymbol{\theta}_i - \bar{\boldsymbol{\theta}}\|, \quad (14)$$

where the first and second terms of the right hand side of (14) are the standard optimality gap of non-convex min/max problems while the third term is the consensus violation gap that characterizes the difference among the weights over the network, where $\bar{\boldsymbol{\theta}} \triangleq n^{-1} \sum_{i=1}^n \boldsymbol{\theta}_i$.

Definition 2. *If a point* $(\{\boldsymbol{\theta}_i^*, \boldsymbol{\lambda}_i^*, \forall i\})$ *satisfies* $\|\mathcal{G}(\{\boldsymbol{\theta}_i^*, \boldsymbol{\lambda}_i^*, \forall i\})\| \leq \epsilon$, *then we call this point as an* ϵ*-approximate first-order stationary points of* (3), *abbreviated as* ϵ*-FOSP.*

Remark 1. Note that points $(\{\boldsymbol{\theta}_i^*, \boldsymbol{\lambda}_i^*, \forall i\})$ satisfying condition $\mathcal{G}(\{\boldsymbol{\theta}_i^*, \boldsymbol{\lambda}_i^*, \forall i\}) = 0$ is also known as "quasi-Nash equilibrium" points (Pang and Scutari 2011) or "first-order Nash equilibrium" points (Nouiehed et al. 2019).

The convergence results of *Safe Dec-PG* are given below.

Theorem 1. *Suppose Assumption 1 to Assumption 4 hold and the iterates* $\{\boldsymbol{\theta}_i^r, \boldsymbol{\vartheta}_i^r, \boldsymbol{\lambda}_i^r, \forall i\}$ *are generated by Safe Dec-PG. If the total number of iterations of the algorithm is* N *and*

$$T \sim \Omega(\log(N)), \quad \gamma^r \sim \mathcal{O}\left(\frac{1}{\sqrt{r}}\right), \quad \beta^r \sim \mathcal{O}\left(\frac{1}{\sqrt{r}}\right), \quad (15)$$

then we have

$$\mathbb{E}[\mathcal{G}^2(\{\boldsymbol{\theta}_i^{r'}, \boldsymbol{\lambda}_i^{r'}, \forall i\})] \leq \mathcal{O}\left(\frac{\log(N)}{\sqrt{N}}\right) + \mathcal{O}(\sigma_g^2(T, K)) \quad (16)$$

where constant $\sigma_g^2(T, K)$ *denotes the variance of PG estimate with respect to function* $g(\cdot)$, *and* r' *is picked randomly from* $1, \ldots, N$.

Theorem 1 says that *Safe Dec-PG* is able to find the solution of (1) at a rate of at least $\mathcal{O}(\log(N)/N^{1/2})$ to a neighborhood of the ϵ-FOSP of this problem, where the radius of this ball is determined by the number of trajectories and length of the horizon approximation. The number of trajectories is or the longer the length is, the smaller the radius will be.

Corollary 1. *Suppose Assumption 1 to Assumption 4 hold and the iterates* $\{\boldsymbol{\theta}_i^r, \boldsymbol{\vartheta}_i^r, \boldsymbol{\lambda}_i^r, \forall i\}$ *are generated by Safe Dec-PG. When* T, γ^r, β^r *satisfy* (15) *and* $K \sim \mathcal{O}(\sqrt{N})$, *then we have*

$$\mathbb{E}[\mathcal{G}^2(\{\boldsymbol{\theta}_i^{r'}, \boldsymbol{\lambda}_i^{r'}, \forall i\})] \leq \mathcal{O}\left(\frac{\log(N)}{\sqrt{N}}\right) \quad (17)$$

where the total number of iterations of the algorithm is N, *and* r' *is picked randomly from* $1, \ldots, N$.

Note that the proposed *Safe Dec-PG* is not only applicable to constrained MDP problems, but also amenable to solve a wide class of stochastic non-convex concave min-max optimization problems.

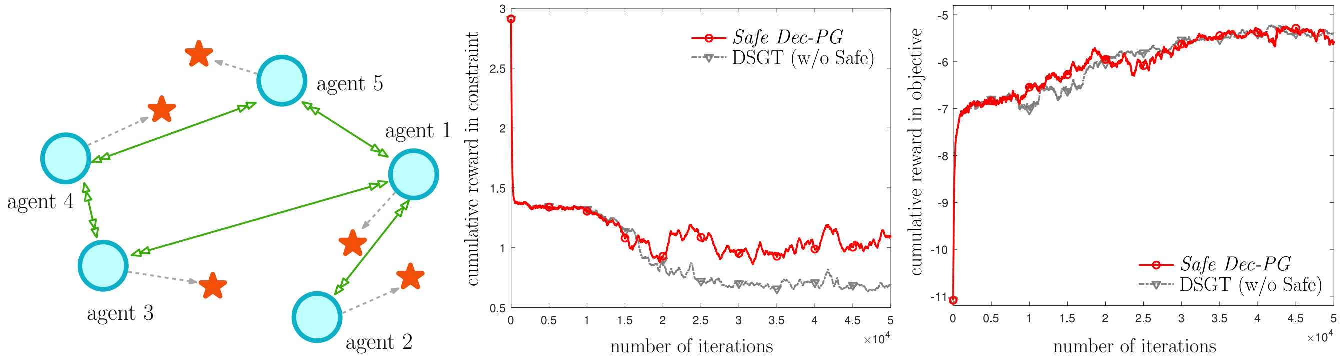
Remark 2. To the best of our knowledge, our results are new in both RL and optimization communities.

- When T is infinitely large, i.e., $\epsilon_f(T) = \epsilon_g(T) = 0$, *Safe Dec-PG* is reduced to a decentralized stochastic non-convex min-max optimization algorithm. In this regime, *Safe Dec-PG* also provides the state-of-the-art convergence rate to a neighborhood of FOSPs.
- When K and T are both infinitely large, i.e., $\epsilon_f(T) = \epsilon_g(T) = \sigma_f^2(T, K) = \sigma_g^2(T, K) = 0$, *Safe Dec* is reduced to a deterministic decentralized non-convex min-max algorithm. The convergence rate of *Safe Dec-PG* is still $\mathcal{O}(\log(N)/N^{1/2})$ but with guarantees to the ϵ-FOSPs, matching the convergence rate of HiBSA in the centralized case.

Remark 3. The number of nodes, n, is not shown up in the numerator of the convergence rate result, indicating that the achievable rate in (17) will not be slowed down by increasing the number of agents and the radius of the neighborhood in (16) will not be magnified as well.

Numerical Results

Problem setting To show the performance of safe decentralized RL, we test our algorithm on the environment of the Cooperative Navigation task in (Lowe et al. 2017), which is built on the popular OpenAI Gym paradigm (Brockman et al. 2016). The experiments were run on the NVIDIA Tesla V100 GPU with 32GB memory. In the first experiment, we have $n = 5$ agents aiming at finding their own landmarks, and all agents are connected by a well-connected graph as shown in Figure 1(a), where every agent can only exchange their parameters $\boldsymbol{\theta}_i$ with its neighbors through the communication



(a) Network structure (b) Constrained reward (averaged) (c) Objective reward

Figure 1: (a) Diagram of a decentralized safe RL system, where the green line denotes the communication graph $\mathcal{G}$, the red star represents the landmark, the blue circle stands for the agents; (b) long-term cumulative reward of the constraints v.s. the number of iterations; (c) long-term cumulative reward of the objective functions v.s. the number of iterations. The initial stepsizes of *Safe Dec-PG* and DSGT are both 0.1 and $c_i = 0.8, \forall i$.

channel (denoted by the green lines). Furthermore, each agent has 5 action options: stay, left, right, up, and down. We assume the states and actions of all the agents to be globally observable. The goal of the teamed agents is to find an optimal policy such that the long term discounted cumulative reward averaged over the network is maximized under a minimum number of collisions with other agents in a long term perspective.

Environment Different from the existing simulation environment, we create a new one based on the cooperative navigation task, where we set the agent and landmark as pairs and require that each agent only targets its own corresponding landmark. The rewards considered in the objective function include two parts: i) the first one is based on the distance between the location of the node to its desired landmark, which is a monotonically decreasing function of the distance, (i.e., the smaller the distance, the higher the reward will be); ii) the second one is determined by the minimum distance between two agents. If the distance between two agents is lower than a threshold, then we consider that a collision happens, and both of the agents will be penalized by a large negative reward value, i.e., -1. Finally, the reward at each agent is further scaled by different positive coefficients, representing the heterogeneity, e.g., priority levels, of different agents. The rewards considered in the constraints of (3) are monotonically increasing functions of the minimum distance between two agents, i.e., the closer the two agents are, the lower the reward will be. Here, since only the minimum distance is taken into account at each node, so $m = 1$.

Parameters The policy at each agent is parametrized by a neural network, where there are two hidden layers with 30 neurons in the first layer and 10 neurons in the second. The states of each agent include its position and velocity. Thus, the dimension of the input layer is 20, and the output layer is 5. The discounting factor γ in the cumulative loss is 0.99 in all the tests, and for each episode, the length of the horizon approximation of PG is $T = 20$. Also, we run $K = 10$ Monte Carlo trials independently to compute the approximate PG at each iteration.

In this section, we only show the results of comparing *Safe Dec-PG* and DSGT without safety considerations in Figure 1(b) and Figure 1(c), and additional results with more problem settings, e.g., larger networks, are included in the supplemental materials. From Figure 1(b), it can be observed that the averaged network constrained rewards obtained by *Safe Dec-PG* are much higher than the ones achieved by DSGT and *Safe Dec-PG* converges faster than DSGT as well. From the statistic perspective, this long term cumulative rewards in the constraints could be interpreted as some prior knowledge accounted in MDP. From Figure 1(c), we can see that the rewards in objective function achieved by both *Safe Dec* and DSGT are similar, implying that the added constraints would not affect the loss of the objective rewards.

Concluding Remarks

In this work, we have proposed the first algorithm of being able to solve multi-agent CMDP problems, where the cumulative rewards in both loss function and constraints are included. By leveraging the primal-dual optimization framework, the proposed *Safe Dec-PG* is to maximize the averaged network long term cumulative rewards and take the safety related constraints as well. Theoretically, we provide the first convergence rate guarantees of the decentralized stochastic gradient descent ascent method to an ϵ-FOSP of a class of non-convex min-max problems at a rate of $\mathcal{O}(1/\epsilon^4)$. Numerical results show that the obtained constraint rewards by *Safe Dec-PG* are indeed much higher than the case where the safety consideration is not incorporated without loss of both convergence rate and final objective rewards.

Acknowledgement

The research of K.Z. and T.B. was supported in part by the US Army Research Laboratory (ARL) Cooperative Agreement W911NF-17-2-0196, in part by the Office of Naval Research (ONR) MURI Grant N00014-16-1-2710, and in part by the Air Force Office of Scientific Research (AFOSR) Grant FA9550-19-1-0353. The work of T. Chen was supported by the RPI-IBM Artificial Intelligence Research Collaboration (AIRC).

Ethical Impact

Our main contributions are regarding the theoretical results for solving a non-convex min-max optimization problem over a graph/network. The algorithm design and convergence analysis are both new. Although *Safe Dec* is developed in this paper for dealing with a constrained Markov decision process related problem, it can be also applied to solve other min-max optimization problems. Our theoretical analysis includes multiple new theorem proving techniques that would be used for performing convergence analysis for other algorithms. This works would be beneficial for students, scientists and professors who are conducting research in the areas of reinforcement learning, optimization, data science, finance, etc. We haven't found any negative impact of this work on both ethical aspects and future societal consequences.

References

Achiam, J.; Held, D.; Tamar, A.; and Abbeel, P. 2017. Constrained policy optimization. In *Proc. of International Conference on Machine Learning*, 22–31.

Agarwal, A.; Kakade, S. M.; Lee, J. D.; and Mahajan, G. 2020. Optimality and approximation with policy gradient methods in Markov decision processes. In *Proc. of .Conference on Learning Theory*, 64–66.

Altman, E. 1999. *Constrained Markov Decision Processes*, volume 7. CRC Press.

Baxter, J.; and Bartlett, P. L. 2001. Infinite-horizon policy-gradient estimation. *J. Artificial Intelligence Res.* 15: 319–350.

Borkar, V. S. 2005. An actor-critic algorithm for constrained Markov decision processes. *Systems & control letters* 54(3): 207–213.

Boyan, J. A.; and Littman, M. L. 1994. Packet routing in dynamically changing networks: A reinforcement learning approach. In *Proc. Advances in Neural Information Processing System*, 671–678. Denver, CO.

Boyd, S.; Diaconis, P.; and Xiao, L. 2004. Fastest mixing Markov chain on a graph. *SIAM Review* 46(4): 667–689.

Boyd, S.; and Vandenberghe, L. 2004. *Convex optimization*. Cambridge University Press.

Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; and Zaremba, W. 2016. OpenAI gym. *arXiv preprint arXiv:1606.01540* .

Chang, T.-H.; Hong, M.; Wai, H.-T.; Zhang, X.; and Lu, S. 2020. Distributed learning in the non-convex world: From batch to streaming data, and beyond. *IEEE Signal Processing Magazine* 37(3): 26–38.

Chen, T.; Zhang, K.; Giannakis, G. B.; and Başar, T. 2018. Communication-efficient distributed reinforcement learning. *arXiv preprint arXiv:1812.03239* .

Claus, C.; and Boutilier, C. 1998. The dynamics of reinforcement learning in cooperative multiagent systems. In *Proc. of the Assoc. for the Advanc. of Artificial Intell.*, 746–752. Orlando, FL.

Corke, P.; Peterson, R.; and Rus, D. 2005. Networked robots: Flying robot navigation using a sensor net. *Robotics Research* 234–243.

Dall'Anese, E.; Zhu, H.; and Giannakis, G. B. 2013. Distributed optimal power flow for smart microgrids. *IEEE Transactions on Smart Grid* 4(3): 1464–1475.

Di Lorenzo, P.; and Scutari, G. 2016. Next: In-network nonconvex optimization. *IEEE Transactions on Signal and Information Processing over Networks* 2(2): 120–136.

Doan, T.; Maguluri, S.; and Romberg, J. 2019. Finite-time analysis of distributed TD (0) with linear function approximation on multi-agent reinforcement learning. In *Proc. of International Conference on Machine Learning*, 1626–1635.

Fax, J. A.; and Murray, R. M. 2004. Information flow and cooperative control of vehicle formations. *IEEE Transactions on Automatic Control* 49(9): 1465–1476.

Gupta, J. K.; Egorov, M.; and Kochenderfer, M. 2017. Cooperative multi-agent control using deep reinforcement learning. In *Intl. Conf. Auto. Agents and Multi-agent Systems*, 66–83.

Kar, S.; Moura, J. M.; and Poor, H. V. 2013. QD-Learning: A collaborative distributed strategy for multi-agent reinforcement learning through consensus + innovations. *IEEE Trans. Sig. Proc.* 61(7): 1848–1862.

Lauer, M.; and Riedmiller, M. 2000. An algorithm for distributed reinforcement learning in cooperative multi-agent systems. In *Proc. Intl. Conf. Machine Learning*. Stanford, CA.

Lee, D.; He, N.; Kamalaruban, P.; and Cevher, V. 2020. Optimization for reinforcement learning: From single agent to cooperative agents. *IEEE Signal Processing Magazine* 37(3): 123–135.

Lian, X.; Zhang, C.; Zhang, H.; Hsieh, C.-J.; Zhang, W.; and Liu, J. 2017. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In *Proc. of Advances in Neural Information Processing Systems*, 5330–5340.

Lin, T.; Jin, C.; and Jordan, M. 2020. On gradient descent ascent for nonconvex-concave minimax problems. In *International Conference on Machine Learning*, 6083–6093. PMLR.

Liu, B.; Cai, Q.; Yang, Z.; and Wang, Z. 2019. Neural Trust Region/Proximal Policy Optimization Attains Globally Optimal Policy. In *Proc. of Advances in Neural Information Processing Systems*, 10564–10575.

Liu, M.; Zhang, W.; Mroueh, Y.; Cui, X.; Ross, J.; Yang, T.; and Das, P. 2020a. A decentralized parallel algorithm for training generative adversarial nets. In *Proc. of Advances in Neural Information Processing Systems*.

Liu, W.; Mokhtari, A.; Ozdaglar, A.; Pattathil, S.; Shen, Z.; and Zheng, N. 2020b. A decentralized proximal point-type method for saddle point problems. *arXiv:1910.14380* .

Lowe, R.; Wu, Y.; Tamar, A.; Harb, J.; Abbeel, P.; and Mordatch, I. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Proc. Advances in Neural Information Processing System*. Long beach, CA.

Lu, S.; Tsaknakis, I.; Hong, M.; and Chen, Y. 2020. Hybrid block successive approximation for one-sided non-convex min-max problems: algorithms and applications. *IEEE Transactions on Signal Processing* 68: 3676–3691.

Lu, S.; and Wu, C. W. 2020. Decentralized stochastic nonconvex optimization over weakly connected time-varying digraphs. In *Proc. of International Conference on Acoustics, Speech and Signal Processing*, 5770–5774.

Lu, S.; Zhang, X.; Sun, H.; and Hong, M. 2019. GNSD: a gradient-tracking based nonconvex stochastic algorithm for decentralized optimization. In *Proc. of IEEE Data Science Workshop*, 315–321.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518(7540): 529.

Nedic, A.; Ozdaglar, A.; and Parrilo, P. A. 2010. Constrained consensus and optimization in multi-agent networks. *IEEE Transactions on Automatic Control* 55(4): 922–938.

Nguyen, D. T.; Yeoh, W.; Lau, H. C.; Zilberstein, S.; and Zhang, C. 2014. Decentralized multi-agent reinforcement learning in average-reward dynamic DCOPs. In *Proc. of Proc. of the Assoc. for the Advanc. of Artificial Intell.*

Nouiehed, M.; Lee, J. D.; and Razaviyayn, M. 2018. Convergence to second-order stationarity for constrained non-convex optimization. *arXiv preprint arXiv:1810.02024* .

Nouiehed, M.; Sanjabi, M.; Huang, T.; Lee, J. D.; and Razaviyayn, M. 2019. Solving a class of non-convex min-max games using iterative first order methods. In *Proc. of Advances in Neural Information Processing Systems*, 14905–14916.

Omidshafiei, S.; Pazis, J.; Amato, C.; How, J. P.; and Vian, J. 2017. Deep decentralized multi-task multi-agent reinforcement learning under partial observability. In *Proc. Intl. Conf. Machine Learning*, 2681–2690. Sydney, Australia.

Pang, J.-S.; and Scutari, G. 2011. Nonconvex games with side constraints. *SIAM Journal on Optimization* 21(4): 1491–1522.

Papini, M.; Binaghi, D.; Canonaco, G.; Pirotta, M.; and Restelli, M. 2018. Stochastic Variance-Reduced Policy Gradient. In *Proc. of International Conference on Machine Learning*, 4026–4035.

Paternain, S.; Calvo-Fullana, M.; Chamon, L. F.; and Ribeiro, A. 2019a. Safe policies for reinforcement learning via primal-dual methods. *arXiv preprint arXiv:1911.09101* .

Paternain, S.; Chamon, L.; Calvo-Fullana, M.; and Ribeiro, A. 2019b. Constrained reinforcement learning has zero duality gap. In *Proc. of Advances in Neural Information Processing Systems*, 7553–7563.

Prashanth, L.; and Ghavamzadeh, M. 2016. Variance-constrained actor-critic algorithms for discounted and average reward MDPs. *Machine Learning* 105(3): 367–417.

Qu, C.; Mannor, S.; Xu, H.; Qi, Y.; Song, L.; and Xiong, J. 2019. Value propagation for decentralized networked deep multi-agent reinforcement learning. In *Proc. of Advances in Neural Information Processing Systems*, 1182–1191.

Rafique, H.; Liu, M.; Lin, Q.; and Yang, T. 2018. Non-convex min-max optimization: Provable algorithms and applications in machine learning. *arXiv preprint arXiv:1810.02060* .

Razaviyayn, M.; Huang, T.; Lu, S.; Nouiehed, M.; Sanjabi, M.; and Hong, M. 2020. Nonconvex min-max optimization: Applications, challenges, and recent theoretical advances. *IEEE Signal Processing Magazine* 37(5): 55–66.

Schneider, J.; Wong, W.-K.; Moore, A.; and Riedmiller, M. 1999. Distributed value functions. In *Proc. Intl. Conf. Machine Learning*, 371–378. Bled, Slovenia.

Sun, Y.; Daneshmand, A.; and Scutari, G. 2019. Convergence rate of distributed optimization algorithms based on gradient tracking. *arXiv preprint arXiv:1905.02637* .

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement Learning: An Introduction*.

Tang, H.; Lian, X.; Yan, M.; Zhang, C.; and Liu, J. 2018. D^2: Decentralized training over decentralized data. In *Proc. of International Conference on Machine Learning*, 4848–4856.

Wachi, A.; and Sui, Y. 2020. Safe reinforcement learning in constrained Markov decision processes. In *Proc. of International Conference on Machine Learning*.

Wai, H.-T.; Yang, Z.; Wang, Z.; and Hong, M. 2018. Multi-agent reinforcement learning via double averaging primal-dual optimization. In *Proc. of Advances in Neural Information Processing Systems*, 9649–9660.

Wolpert, D. H.; Wheeler, K. R.; and Tumer, K. 1999. General principles of learning-based multi-agent systems. In *Proc. of the Annual Conf. on Autonomous Agents*, 77–83. Seattle, WA.

Xiao, L.; and Boyd, S. 2004. Fast linear iterations for distributed averaging. *Systems & Control Letters* 53(1): 65–78.

Xu, P.; Gao, F.; and Gu, Q. 2020. Sample efficient policy gradient methods with recursive variance reduction. In *Proc. of International Conference on Learning Representations*.

Yu, M.; Yang, Z.; Kolar, M.; and Wang, Z. 2019. Convergent policy optimization for safe reinforcement learning. In *Proc. of Advances in Neural Information Processing Systems*, 3121–3133.

Zhang, K.; Koppel, A.; Zhu, H.; and Başar, T. 2020. Global convergence of policy gradient methods to (almost) locally optimal policies. *SIAM Journal on Control and Optimization* 58(6): 3586–3612.

Zhang, K.; Yang, Z.; and Başar, T. 2019. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *arXiv preprint arXiv:1911.10635* .

Zhang, K.; Yang, Z.; Liu, H.; Zhang, T.; and Başar, T. 2018. Fully decentralized multi-agent reinforcement learning with networked agents. In *Proc. of International Conference on Machine Learning*, 5872–5881.