



Title	Decentralized Reinforcement Learning with Risk Aversion in Multi-Agent Systems
Author(s)	Ishikawa, Daichi; Ichino, Taisei; Hayashi, Naoki et al.
Citation	Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics. 2026, p. 6506-6511
Version Type	AM
URL	https://hdl.handle.net/11094/104542
rights	©2025 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

The University of Osaka

Decentralized Reinforcement Learning with Risk Aversion in Multi-Agent Systems

Daichi Ishikawa, Taisei Ichino, Naoki Hayashi, and Masahiro Inuiguchi

Abstract—This study addresses risk-averse distributed reinforcement learning in multi-agent systems, where agents collaboratively optimize their policies with a risk index in the objective function. Unlike conventional reinforcement learning, which maximizes the expected cumulative reward, the proposed risk-averse reinforcement learning incorporates Conditional Value at Risk (CVaR) to account for rare but significant adverse events. We propose a risk-averse distributed reinforcement learning algorithm based on the policy gradient method, which enables agents to collaboratively update their policies by sharing information through a communication network. Through a theoretical analysis, we show that the proposed algorithm ensures agreement among agents on their estimated policies. Moreover, we show that the consensus value converges in a neighborhood of a locally optimal solution, which ensures that each agent’s learned policy remains aligned with risk-aware optimization criteria.

I. INTRODUCTION

Distributed optimization has gained significant attention across various engineering fields such as transportation networks, power grids, and sensor networks [1]–[6]. Recently, the principles of distributed optimization have been extended to reinforcement learning [7]–[9]. In distributed reinforcement learning, multiple agents interact with the environment in parallel to improve learning efficiency. By having each agent experience different environmental conditions and action choices, it is possible to aggregate a variety of data that cannot be obtained by a single agent.

The goal of traditional reinforcement learning is to maximize the expected cumulative reward [10]. Although this approach effectively optimizes average performance, it implicitly assumes that occasional adverse outcomes are acceptable. However, this assumption cannot be allowed in critical applications where rare events are unacceptable regardless of how well the system performs in typical scenarios. For example, in the control of autonomous vehicles, minimizing accident risk as well as improving average driving efficiency is an important issue [11]. In such cases, it is necessary to learn policies to achieve stable driving while avoiding unsafe control input.

To address these challenges, risk-averse reinforcement learning has gained attention as an important extension of the conventional risk-neutral framework. The Value at Risk (VaR) and Conditional Value at Risk (CVaR) are commonly

used as measures to evaluate risk [12]–[15]. The VaR is an indicator that shows the maximum value at which the loss does not exceed the threshold at a certain confidence level. On the other hand, the CVaR represents the average value of losses that exceed VaR and can be used to evaluate the expected loss in the worst-case scenario. Therefore, incorporating these indicators into the objective function enables risk-sensitive decision making.

This study investigates risk-averse reinforcement learning over a multi-agent network by incorporating CVaR. The proposed method with CVaR is suitable for scenarios where avoiding extreme losses is crucial. However, incorporating CVaR in reinforcement learning presents significant challenges due to the lack of time consistency. To overcome this difficulty, we apply an approach that transforms risk-averse reinforcement learning into an equivalent risk-neutral reinforcement learning formulation [14]. This approach enables the application of existing optimization techniques while preserving risk sensitivity. We propose a distributed policy gradient algorithm based on consensus-based algorithms to ensure efficient learning in multi-agent systems. This method allows agents to collaboratively optimize risk-aware policies while efficiently exchanging information across the network.

Numerous distributed methods have been proposed for safe reinforcement learning [16]–[18]. Most of these methods formulate the problem as a minimax optimization problem with safety constraints. While these approaches provide robustness to a certain extent, they do not explicitly account for rare but critical events that can significantly impact system performance. In contrast, the proposed risk-averse approach with CVaR can handle such extreme risks. As part of CVaR-based distributed reinforcement learning, Qiu et al. introduced a cooperative reinforcement learning framework that incorporates risk-sensitive policies [19]. Their method effectively integrates risk awareness into multi-agent learning, but it does not take into account the impact of information sharing among agents, which is crucial in distributed settings where communication plays a key role in coordination and learning efficiency. In this study, we apply distributed optimization methods [20], which allows agents to exchange policy updates and improve learning performance through consensus-based mechanisms. Another relevant approach is the QD-learning algorithm with CVaR, which addresses decision-making problems in unknown environments using a value-based learning framework [21]. While this method effectively incorporates risk-awareness, it relies on value-function approximation, which can be computationally intensive and may not scale efficiently in large-scale multi-agent

This work partially supported by JSPS KAKENHI Grant Number JP21K04121.

The authors are with the Graduate School of Engineering Science, Osaka University, Toyonaka, Osaka 560-8531, Japan (e-mail: {d.ishikawa, Ichino}@inulab.sys.es.osaka-u.ac.jp, {n.hayashi, inuiguti}@sys.es.osaka-u.ac.jp). (Corresponding author: Daichi Ishikawa)

systems. In contrast, this study focuses on a policy-based approach to distributed risk-averse reinforcement learning, which enables direct policy optimization across networked agents. By integrating risk-awareness with policy gradient methods, the proposed approach provides a scalable and flexible framework for multi-agent reinforcement learning in environments with uncertainty.

This paper is organized as follows. In Section II, we address the problem formulation of risk-averse reinforcement learning for multi-agent systems. In Section III, we propose a distributed learning algorithm for risk-averse reinforcement learning. The analysis of the proposed algorithm is shown in Section IV. A numerical example is presented in Section V. Finally, concluding remarks are given in Section VI.

II. PROBLEM FORMULATION

For a set $\mathcal{S}$, $|\mathcal{S}|$ represents the cardinality of $\mathcal{S}$. $\mathbf{e}_i$ represents the canonical basis where the i -th element is 1 and other elements are 0. $\mathbf{1}$ represents the vector with an appropriate dimension whose elements are all 1. The notation $[A]_{ij}$ is used to represent the (i, j) -th element of the matrix A . Similarly, $[a]_i$ represents the i -th element of the vector a . A sequence of vectors $(a_{i,1}, a_{i,2}, \dots, a_{i,k})$ is represented as $a_{i,[1,k]}$. The positive part function is defined as $[x]_+ = \max\{x, 0\}$ for $x \in \mathbb{R}$.

We consider a multi-agent system where the information exchange among agents is represented by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, 2, \dots, N\}$ is the set of agents and $\mathcal{E}$ is the set of the communication links.

Assumption 1: The graph $\mathcal{G}$ is strongly connected.

Each agent $i \in \mathcal{V}$ is modeled as a Markov Decision Process (MDP) $M_i = (\mathcal{S}, \mathcal{A}, C_i, \Pr, \gamma_i, \rho)$, where $\mathcal{S}$ and $\mathcal{A}$ are the finite state and action spaces of agent i . $C_i(s_i, a_i) \in [0, 1]$ represents the loss that depends on state $s_i \in \mathcal{S}$ and action $a_i \in \mathcal{A}$. $\Pr(\cdot | s_i, a_i)$ is the transition probability that defines the probability of transitioning to any next state when agent i is in state $s_i \in \mathcal{S}$ and takes action $a_i \in \mathcal{A}$. $\gamma_i \in (0, 1)$ is the discount factor that adjusts the weights given to losses that occur further in the future. ρ is the probability distribution that the initial state follows.

A stationary Markov policy $\pi_i^\theta : \mathcal{S} \rightarrow \Delta(\mathcal{A})$ is parameterized by θ and defines a mapping from each state $s_i \in \mathcal{S}$ to a probability distribution over the action space. For simplicity of notation, we denote the policy as π_i . The notation $\Delta(\mathcal{A})$ represents the probability simplex, which ensures that the policy satisfies the fundamental properties of a valid probability distribution: (i) $0 \leq \pi_i(a_i | s_i) \leq 1$ for all $a_i \in \mathcal{A}$ and $i \in \mathcal{V}$, and (ii) $\sum_{a_i \in \mathcal{A}} \pi_i(a_i | s_i) = 1$ for all $s_i \in \mathcal{S}$ and $i \in \mathcal{V}$.

Given a policy π_i , each agent follows a stochastic process over trajectories $\{(s_{i,k}, a_{i,k}, C_i(s_{i,k}, a_{i,k}))\}_{k=1}^\infty$, where $s_{i,k}$ represents the state of agent i at time $k \in \{1, 2, \dots\}$, $a_{i,k}$ is the action taken at that time, and $C_i(s_{i,k}, a_{i,k})$ is the cost incurred. For simplicity of notation, we rewrite $C_i(s_{i,k}, a_{i,k})$ as $c_{i,k}$. The trajectory begins with an initial state $s_{i,1}$ drawn from the given initial state distribution ρ . At each subsequent time step k , the action $a_{i,k}$ is sampled according to the

policy, that is, $a_{i,k} \sim \pi_i(\cdot | s_{i,k})$. The next state $s_{i,k+1}$ is then determined by the transition probability function $\Pr(\cdot | s_{i,k}, a_{i,k})$.

The Q-value function $Q_i^{\pi_i} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ shows the performance by the expected total cost when starting from a state-action pair (s_i, a_i) and subsequently following policy π_i . The Q-value function is defined as

$$Q_i^{\pi_i}(s_i, a_i) = \mathbb{E} \left[\sum_{t=1}^{\infty} \gamma_i^{t-1} C_i(s_{i,t}, a_{i,t}) \mid \pi_i, s_{i,1} = s_i, a_{i,1} = a_i \right]. \quad (1)$$

The risk indicator value used in this study is given by the following upper α_i -tail CVaR [14]:

$$\rho_{i,k}^{s_{i,[1,k-1]}}(c_{i,k}) = \text{CVaR}_{\alpha_i}[c_{i,k} | s_{i,[1,k-1]}], \quad (2)$$

where

$$\text{CVaR}_{\alpha_i}[c_{i,k} | s_{i,[1,k-1]}] = \min_{\eta_{i,k} \in \mathbb{R}} \left\{ \eta_{i,k} + \frac{1}{\alpha_i} \mathbb{E} \left[[c_{i,k} - \eta_{i,k}]_+ \mid s_{i,[1,k-1]} \right] \right\}, \quad (3)$$

and $\eta_{i,k} \in \mathcal{H} = [0, 1]$ is an auxiliary variable. The minimum of the right-hand side is achieved when the auxiliary variable $\eta_{i,k}$ is set to the VaR at level α_i , that is, $\eta_{i,k}^* = \text{VaR}_{\alpha_i}[c_{i,k} | s_{i,[1,k-1]}] := \inf\{v \mid F_i(v) \geq 1 - \alpha_i\}$, where F_i is a cumulative distribution function. Thus, the CVaR can be interpreted as the expected loss in the worst α_i -fraction of scenarios, that is, $\text{CVaR}_{\alpha_i}[c_{i,k} | s_{i,[1,k-1]}] = \mathbb{E}[c_{i,k} | c_{i,k} > \text{VaR}_{\alpha_i}[c_{i,k} | s_{i,[1,k-1]}]]$. The parameter α_i controls the level of risk aversion, that is, smaller values of α_i indicate a greater focus on extreme events. This flexibility allows CVaR to adapt to varying levels of risk tolerance by managing tail risk in uncertain environments.

In this paper, we consider transforming risk-averse reinforcement learning with expected conditional risk measures into an equivalent formulation that allows it to be treated as a risk-neutral reinforcement learning problem [14]. This transformation enables the application of standard techniques for reinforcement learning while preserving the underlying risk-sensitive decision-making framework. To achieve this, we introduce the following modified loss:

$$\bar{C}_{i,1}(s_{i,1}, a_{i,1}, \eta_{i,2}) = C_i(s_{i,1}, a_{i,1}) + \gamma_i \eta_{i,2}, \quad (4)$$

$$\bar{C}_{i,k}(s_{i,k}, \eta_{i,k}, a_{i,k}, \eta_{i,k+1}) = \frac{1}{\alpha_i} [C_i(s_{i,k}, a_{i,k}) - \eta_{i,k}]_+ + \gamma_i \eta_{i,k+1}, \quad k \geq 2. \quad (5)$$

As shown in [14], if we use (4) and (5) as the new reward functions and define the action at time step k as the pair $(a_{i,k}, \eta_{i,k+1})$, then the original risk-averse reinforcement learning problem can be equivalently formulated as a standard risk-neutral reinforcement learning problem. Moreover, it has been shown that under this transformation, there exists an optimal policy that is deterministic and stationary in the transformed problem setting [14]. This result implies that optimal decision-making in risk-averse reinforcement

learning can be effectively realized using a policy structure similar to that in conventional risk-neutral settings.

Based on this insight, we consider a class of policies $\pi_i = (\pi_{1,i}, \pi_{2,i}) \in \Delta(\mathcal{A} \times \mathcal{H})^{|\mathcal{S}|(1+|\mathcal{H}|)}$. The policy for the first time step is given by $\pi_{1,i}(a_{i,1}, \eta_{i,2} \mid s_{i,1})$, which determines the initial action $a_{i,1}$ and auxiliary variable $\eta_{i,2}$ based on the initial state $s_{i,1}$. For subsequent time steps $k \geq 2$, the policy follows $\pi_{2,i}(a_{i,k}, \eta_{i,k+1} \mid s_{i,k}, \eta_{i,k})$, where the choice of action $a_{i,k}$ and auxiliary variable $\eta_{i,k+1}$ depends on the current state $s_{i,k}$ and the auxiliary variable $\eta_{i,k}$. In this paper, we consider the following softmax policy with the parameters $\theta_{i,1} \in \mathbb{R}^{|\mathcal{A}||\mathcal{H}||\mathcal{S}|}$ and $\theta_{i,2} \in \mathbb{R}^{|\mathcal{A}||\mathcal{H}|^2|\mathcal{S}|}$:

$$\begin{aligned} \pi_{1,i}(a_{i,1}, \eta_{i,2} \mid s_{i,1}) &= \frac{\exp(\theta_{i,1}(s_{i,1}, a_{i,1}, \eta_{i,2}))}{\sum_{a'_{i,1}, \eta'_{i,2}} \exp(\theta_{i,1}(s_{i,1}, a'_{i,1}, \eta'_{i,2}))}, \\ \pi_{2,i}(a_{i,k}, \eta_{i,k+1} \mid s_{i,k}, \eta_{i,k}) &= \frac{\exp(\theta_{i,2}(s_{i,k}, \eta_{i,k}, a_{i,k}, \eta_{i,k+1}))}{\sum_{a'_{i,k}, \eta'_{i,k+1}} \exp(\theta_{i,2}(s_{i,k}, \eta_{i,k}, a'_{i,k}, \eta'_{i,k+1}))}. \end{aligned}$$

The goal of the agents is to estimate the optimal parameters of the following problem:

$$\underset{\theta}{\text{minimize}} \quad L(\theta) = \sum_{i=1}^N J_i^\pi(\rho), \quad (6)$$

where $\theta = [\theta_1^\top, \theta_2^\top]^\top \in \mathbb{R}^{|\mathcal{A}||\mathcal{H}||\mathcal{S}|(1+|\mathcal{H}|)}$ and

$$\begin{aligned} J_i^\pi(\rho) &= \mathbb{E}_{s_{i,1} \sim \rho} [J_i^\pi(s_{i,1})] \\ &= \mathbb{E}_{s_{i,1} \sim \rho} \mathbb{E}_{(a_{i,1}, \eta_{i,2}) \sim \pi_{1,i}(\cdot, \cdot \mid s_{i,1})} [Q_i^{\pi_{2,i}}(s_{i,1}, a_{i,1}, \eta_{i,2})]. \end{aligned} \quad (7)$$

The Q-value is given by

$$\begin{aligned} Q_i^{\pi_{2,i}}(s_{i,1}, a_{i,1}, \eta_{i,2}) &= C_i(s_{i,1}, a_{i,1}) + \gamma_i \eta_{i,2} \\ &\quad + \gamma_i \mathbb{E}_{s_{i,2}}^{s_{i,1}, a_{i,1}} \mathbb{E}_{(a_{i,2}, \eta_{i,3}) \sim \pi_{2,i}(\cdot, \cdot \mid s_{i,2}, \eta_{i,2})} \left[\left\{ \frac{1}{\alpha_i} [C_i(s_{i,2}, a_{i,2}) - \eta_{i,2}]_+ + \gamma_i \eta_{i,3} \right. \right. \\ &\quad \left. \left. + \gamma_i \mathbb{E}_{s_{i,3}}^{s_{i,2}, a_{i,2}} \mathbb{E}_{(a_{i,3}, \eta_{i,4}) \sim \pi_{2,i}(\cdot, \cdot \mid s_{i,3}, \eta_{i,3})} \left[\left\{ \frac{1}{\alpha_i} [C_i(s_{i,3}, a_{i,3}) - \eta_{i,3}]_+ + \gamma_i \eta_{i,4} + \dots \right\} \right] \right\} \right], \end{aligned} \quad (8)$$

where $\mathbb{E}_{s_{i,k}}^{s_{i,k-1}, a_{i,k-1}} = \mathbb{E}[\cdot \mid s_{i,k-1}, a_{i,k-1}]$.

We also define the value function for $t \geq 2$ as follows:

$$\hat{J}_i^{\pi_{2,i}}(s_{i,k}, \eta_{i,k}) = \mathbb{E}_{(a_{i,k}, \eta_{i,k+1}) \sim \pi_{2,i}(\cdot, \cdot \mid s_{i,k}, \eta_{i,k})} [\hat{Q}_i^{\pi_{2,i}}(s_{i,k}, \eta_{i,k}, a_{i,k}, \eta_{i,k+1})], \quad (9)$$

where

$$\begin{aligned} \hat{Q}_i^{\pi_{2,i}}(s_{i,k}, \eta_{i,k}, a_{i,k}, \eta_{i,k+1}) &= \frac{1}{\alpha_i} [C_i(s_{i,k}, a_{i,k}) - \eta_{i,k}]_+ \\ &\quad + \gamma_i \eta_{i,k+1} + \gamma_i \mathbb{E}_{s_{i,k+1}}^{s_{i,k}, a_{i,k}} [\hat{J}_i^{\pi_{2,i}}(s_{i,k+1}, \eta_{i,k+1})]. \end{aligned}$$

Algorithm 1 Distributed Risk-Averse Reinforcement Learning Algorithm

- 1: Initialize $\theta_{1,i,1}(s_{i,1}, a_{i,1}, \eta_{i,1})$, $\theta_{2,i,1}(s_{i,1}, \eta_{i,1}, a_{i,1}, \eta'_{i,1})$, and $z_{i,1} = \mathbf{e}_i$ for all $s_{i,1} \in \mathcal{S}$, $a_{i,1} \in \mathcal{A}$, $\eta_{i,1}, \eta'_{i,1} \in \mathcal{H}$, and $i \in \mathcal{V}$. Set $k = 1$ and the horizon period T .
- 2: Set the softmax policy:

$$\pi_{1,i}^{\theta_{1,i,1}}(a_{i,1}, \eta_{i,2} \mid s_{i,1}) = \frac{\exp(\theta_{1,i,1}(s_{i,1}, a_{i,1}, \eta_{i,2}))}{\sum_{a'_{i,1}, \eta'_{i,2}} \exp(\theta_{1,i,1}(s_{i,1}, a'_{i,1}, \eta'_{i,2}))}, \quad (10)$$

$$\pi_{2,i}^{\theta_{2,i,k}}(a_{i,k}, \eta_{i,k+1} \mid s_{i,k}, \eta_{i,k}) = \frac{\exp(\theta_{2,i,k}(s_{i,k}, \eta_{i,k}, a_{i,k}, \eta_{i,k+1}))}{\sum_{a'_{i,k}, \eta'_{i,k+1}} \exp(\theta_{2,i,k}(s_{i,k}, \eta_{i,k}, a'_{i,k}, \eta'_{i,k+1}))}. \quad (11)$$

- 3: **while** $k \leq K$ **do**
- 4: **Generate** a trajectory $(s_{i,1}, a_{i,1}, \eta_{i,2}, c_{i,1}, s_{i,2}, \dots, s_{i,T-1}, a_{i,T-1}, \eta_{i,T}, c_{i,T-1}, s_{i,T})$ on policy $\pi_i = (\pi_{1,i}, \pi_{2,i})$ and modify the costs as follows:

$$\bar{c}_{i,1} = c_{i,1} + \gamma_i \eta_{i,2}, \quad (12)$$

$$\bar{c}_{i,k} = \frac{1}{\alpha_i} [c_{i,k} - \eta_{i,k}]_+ + \gamma_i \eta_{i,k+1}, \quad k \geq 2. \quad (13)$$

- 5: **Update** the estimations as follows:

$$\theta_{1,i,k+1} = \sum_{j=1}^N p_{ij} \theta_{1,j,k} - \zeta \frac{\nabla_{\theta_1} J_i^\pi(\rho)}{[z_{i,k}]_i}, \quad (14)$$

$$\theta_{2,i,k+1} = \sum_{j=1}^N p_{ij} \theta_{2,j,k} - \zeta \frac{\nabla_{\theta_2} J_i^\pi(\rho)}{[z_{i,k}]_i}, \quad (15)$$

$$z_{i,k+1} = \sum_{j=1}^N p_{ij} z_{j,k}. \quad (16)$$

- 6: **Set** $k := k + 1$.

- 7: **end while**
-

III. DISTRIBUTED RISK-AVERSE REINFORCEMENT LEARNING

In this section, we propose a distributed risk-averse algorithm for the multi-agent reinforcement learning problem (6). Let $\theta_{1,i,k}$ and $\theta_{2,i,k}$ be the estimations for the optimal parameters θ_1^* and θ_2^* at iteration k , respectively. The proposed algorithm for agent i is summarized in Algorithm 1.

After the initialization of parameters, agent i sets the softmax policies by (10) and (11). Within each iteration of the main loop, each agent first generates a trajectory by following its current policy. Then, each agent modifies the immediate costs by (12) and (13) to incorporate risk sensitivity into the standard reinforcement learning framework. Once a trajectory is generated, each agent updates its parameters by (14) and (15) that combines a consensus term and a gradient descent term. The gradient terms adjust parameters in the

direction that reduces expected costs and are given by

$$\begin{aligned} \nabla_{\theta_1} J_i^\pi(\rho) &= \mathbb{E}_{s_{i,1} \sim \rho} \mathbb{E}_{(a_{i,1}, \eta_{i,2}) \sim \pi_{1,i}(\cdot | s_{i,1})} \\ &\quad [\nabla_{\theta_1} \log \pi_{1,i}(a_{i,1}, \eta_{i,2} | s_{i,1}) Q_i^{\pi_{2,i}}(s_{i,1}, a_{i,1}, \eta_{i,2})], \end{aligned} \quad (17)$$

$$\begin{aligned} \nabla_{\theta_2} J_i^\pi(\rho) &= \frac{\gamma_i}{1 - \gamma_i} \mathbb{E}_{(s_{i,k}, \eta_{i,k}) \sim d_{\rho_i}^\pi} \mathbb{E}_{(a_{i,k}, \eta_{i,k+1}) \sim \pi_{2,i}(\cdot | s_{i,k}, \eta_{i,k})} \\ &\quad [\nabla_{\theta_2} \log \pi_{2,i}(a_{i,k}, \eta_{i,k+1} | s_{i,k}, \eta_{i,k}) \\ &\quad \hat{Q}_i^{\pi_{2,i}}(s_{i,k}, \eta_{i,k}, a_{i,k}, \eta_{i,k+1})], \end{aligned} \quad (18)$$

where d_ρ^π is the state visitation distribution, $\rho_i^\pi(s, \eta) = \sum_{s_{i,1}} \rho(s_{i,1}) \Pr^{\pi_{1,i}}(s_{i,2} = s, \eta_{i,2} = \eta | s_{i,1})$, and $\Pr^\pi(\tau | s)$ is the probability of observing a trajectory τ starting from state s and following a policy π . The gradient term is scaled by a learning rate ζ and modified by $z_{i,k}$. It is known that, for the directed communication network, the convergence point without modifying the subgradient is deviated by the normalized left eigenvector of the weight matrix $P = [p_{ij}] \in \mathbb{R}^{N \times N}$ corresponding to the eigenvalue 1 [22]. The modification by $z_{i,k}$ is incorporated into the algorithm to compensate for potential imbalances in the information flow. This variable $z_{i,k}$ is updated in (16) to estimate the normalized left eigenvector of the weight matrix P .

In this paper, we make the following assumption on the edge weight.

Assumption 2: For each agent $i \in \mathcal{V}$, there exists a positive constant $C_p \in (0, 1)$ such that

$$p_{ij} \begin{cases} \geq C_p, & j \in \mathcal{N}_i, \\ = 0, & j \notin \mathcal{N}_i, \end{cases},$$

and $p_{ii} = 1 - \sum_{j \in \mathcal{N}_i} p_{ij} > C_p$ for all $i \in \mathcal{V}$, where $\mathcal{N}_i = \{j \in \mathcal{V} | (j, i) \in \mathcal{E}\}$ is the neighbor set of agent i .

IV. CONVERGENCE ANALYSIS

Since the cost $c_{i,k}$ and the auxiliary variable $\eta_{i,k}$ are bounded, it can be shown that the gradients are also bounded [14], that is, there exist positive constants D_1 and D_2 such that $\|\nabla_{\theta_1} J_i^\pi(\rho)\| \leq D_1$ and $\|\nabla_{\theta_2} J_i^\pi(\rho)\| \leq D_2$ for all $i \in \mathcal{V}$. Moreover, it can be shown that the objective function L is smooth.

Lemma 1: The cost function $L(\theta)$ is σ -smooth, where $\sigma = \max_{i \in \mathcal{V}} \{N((1/\alpha_i) + 1) + (8/(1 - \gamma_i)^3)b_i\}$ and $b_i = (1/\alpha_i) + \gamma_i$.

Lemma 1 can be proven in the same way as Lemma 3.9 in [14].

From Assumption 2, the weight matrix P is row-stochastic. For the weight matrix P , we have the following results:

Lemma 2 ([22]): Under Assumptions 1 and 2, there exists a positive normalized vector $w = [w_1, w_2, \dots, w_N]^\top \in \mathbb{R}^N$ such that

- (a) $w^\top \mathbf{1} = 1$,
- (b) $w^\top = w^\top P$,

- (c) $|[P^k]_{ij} - w_j| \leq Br^k, \quad \forall i, j \in \mathcal{V}, k \geq 1$,
- (d) $|[z_{i,k}]_i - w_i| \leq Br^k, \quad \forall i \in \mathcal{V}, k \geq 1$,

where $B > 0$, $r \in (|\lambda_2(P)|, 1)$, and $\lambda_2(P)$ is the second largest eigenvalue of P in modulus. Moreover, there exists $C_z > 1$ such that $(1/C_z) \leq [z_{i,k}]_i \leq 1$ for $i \in \mathcal{V}$ and $k \geq 1$.

Let $\bar{\theta}_{1,k} = \sum_{i=1}^N w_i \theta_{1,i,k}$ and $\bar{\theta}_{2,k} = \sum_{i=1}^N w_i \theta_{2,i,k}$ be the average estimations of the parameters. The following lemma shows that the consensus errors of the estimations are bounded.

Lemma 3: Under Assumptions 1 and 2, by Algorithm 1, for all $i \in \mathcal{V}$, we have

$$\begin{aligned} \|\theta_{1,i,k+1} - \bar{\theta}_{1,k+1}\| &\leq E_{1,k+1}, \quad k \geq 1, \\ \|\theta_{2,i,k+1} - \bar{\theta}_{2,k+1}\| &\leq E_{2,k+1}, \quad k \geq 1, \\ \|\theta_{1,i,1} - \bar{\theta}_{1,1}\| &\leq E_{1,1}, \\ \|\theta_{2,i,1} - \bar{\theta}_{2,1}\| &\leq E_{2,1}, \end{aligned}$$

where $E_{1,k+1} = 2NBr^k \max_{j \in \mathcal{V}} \{\|\theta_{1,j,1}\|\} + 2D_1C_z\zeta + 2ND_1C_zB\zeta r(1 - r^k)/(1 - r)$ ($k \geq 1$), $E_{2,k+1} = 2NBr^k \max_{j \in \mathcal{V}} \{\|\theta_{2,j,1}\|\} + 2D_2C_z\zeta + 2ND_2C_zB\zeta r(1 - r^k)/(1 - r)$ ($k \geq 1$), $E_{1,1} = 2NBr \max_{j \in \mathcal{V}} \{\|\theta_{1,j,1}\|\} + 2D_1C_z\zeta$, and $E_{2,1} = 2NBr \max_{j \in \mathcal{V}} \{\|\theta_{2,j,1}\|\} + 2D_2C_z\zeta$.

Lemma 3 can be shown in the same way as Lemma 3 of [23].

The following theorem shows that the consensus values of the parameters approach a local optimum.

Theorem 1: Under Assumptions 1 and 2, by Algorithm 1, we have

$$\frac{1}{K} \sum_{k=1}^K \|\nabla_{\theta_1} L(\bar{\theta}_{1,k})\|^2 \leq \frac{1}{K} F_1 + F'_1 \zeta, \quad (19)$$

$$\frac{1}{K} \sum_{k=1}^K \|\nabla_{\theta_2} L(\bar{\theta}_{2,k})\|^2 \leq \frac{1}{K} F_2 + F'_2 \zeta, \quad (20)$$

where F_1 , F_2 , F'_1 , and F'_2 are positive constants.

V. NUMERICAL EXAMPLE

We consider a stochastic 4×4 Cliffwalk environment [14]. In this environment, each agent incurs a loss of 1 for every action taken, which represents the energy or cost associated with the movement. However, if an agent enters a cliff cell located at (3, 1) or (3, 2), it suffers a significantly higher loss of 5 as penalty for falling off the cliff. Following such an event, the agent is returned to the starting point. The environment also introduces slippery cells adjacent to the cliff (2, 1) and (2, 2). These cells have a 10% probability of causing the agent to slip into the cliff to reflect the scenarios where risky areas may indirectly lead to adverse outcomes. The agents are tasked with collaboratively identifying an optimal route that minimizes the expected cumulative loss while avoiding risky areas. In this experiment, we consider three agents each of which starts from different initial positions: (3, 0), (2, 2), and (1, 2). The communication network among the agents is shown in Figs. 2.

To evaluate the agents' performance, we set the discount factor to $\gamma = 0.9999$, which assigns a high weight to future losses to encourage the agents to prioritize long-term

Cost=1 (0,0)	Cost=1 (0,1)	Cost=1 (0,2)	Cost=1 (0,3)
Cost=1 (1,0)	Cost=1 (1,1)	Cost=1 (1,2)	Cost=1 (1,3)
Cost=1 (2,0)	Cost=1 (2,1)	Cost=1 (2,2)	Cost=1 (2,3)
Cost=1 (3,0) Start	Cost=5 (3,1) Cliff	Cost=5 (3,2) Cliff	Cost=0 (3,3) Goal

Fig. 1. Cliffwalk environment

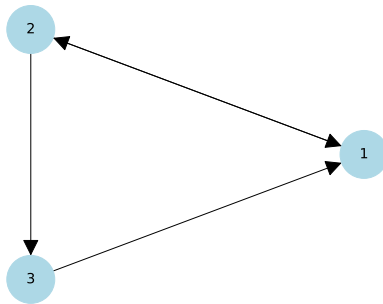


Fig. 2. Communication network

performance. The confidence level for the CVaR is set to $\alpha = 0.05$ to emphasize risk aversion by focusing on the worst 5% of outcomes. The step size for policy updates is $\zeta = 0.0008$.

At each learning step from 0 to 16000, the total loss incurred by the agent starting at the position (3, 0) is shown in Figs. 3 and 4, which corresponds to the learning dynamics under both the risk-neutral and proposed risk-averse methods, respectively. These figures show the gradual reduction in total loss as the learning progresses. The observed decrease in loss indicates that the agent is successfully adapting its strategy to minimize cumulative costs. In the case of the risk-neutral method, the agent focuses on maximizing average performance without significant consideration of rare but severe events. Conversely, the risk-averse method aims to balance performance optimization with risk mitigation, and hence, the agent avoids extreme losses even at the cost of slightly higher cumulative loss. These results show the effectiveness of the proposed learning framework in guiding the agent toward optimal and risk-sensitive decision-making.

Fig. 5 shows the costs incurred by the agent under both the risk-neutral and risk-averse methods during the learning steps from 15980 to 16000. The results show that, in the case of the risk-neutral method, the policy converges to one with a cost of 4, whereas for the risk-averse method, the policy converges to one with a cost of 6. This difference in cost reflects the routes chosen by the agent under the two methods. In the risk-neutral setting, the agent selects

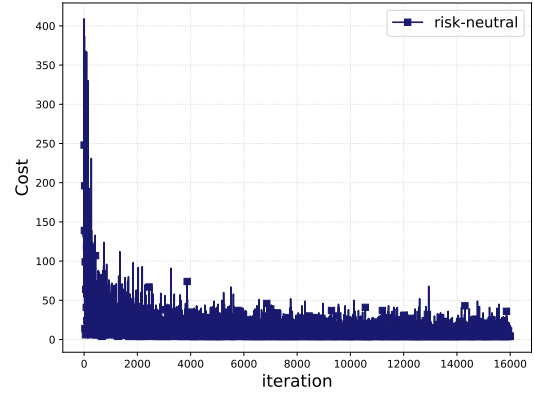


Fig. 3. Cost for risk-neutral method.

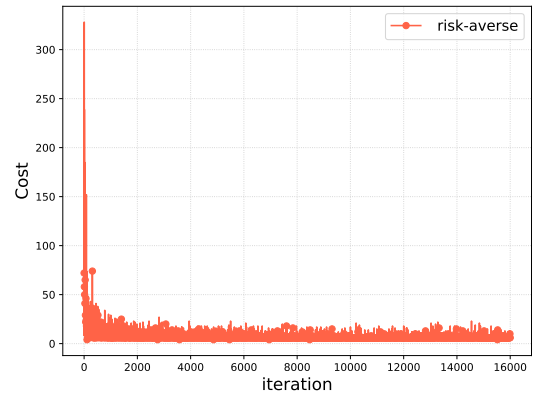


Fig. 4. Cost for risk-averse method.

the route $(3,0) \rightarrow (2,0) \rightarrow (2,1) \rightarrow (2,2) \rightarrow (2,3) \rightarrow (3,3)$, which minimizes the total cost but involves the risk of passing through (2,1) and (2,2), both slippery cells of falling into the cliff. This path prioritizes efficiency over safety, as the risk-neutral method focuses on minimizing the expected cumulative cost without accounting for rare but severe events.

In contrast, the risk-averse method leads the agent to adopt a safer route, $(3,0) \rightarrow (2,0) \rightarrow (1,0) \rightarrow (1,1) \rightarrow (1,2) \rightarrow (1,3) \rightarrow (2,3) \rightarrow (3,3)$. This path avoids the slippery cliff-adjacent cells and ensures the agent does not encounter the high penalty associated with falling into the cliff. Although this route results in a higher total cost of 6 compared to the risk-neutral approach, it effectively mitigates the risk of extreme losses, which reflects the risk sensitivity in the decision-making process.

These results illustrate the trade-off between efficiency and safety in the two methods. The risk-neutral approach emphasizes cost minimization at the expense of potential exposure to rare adverse events, while the proposed risk-averse method prioritizes safety, even if it entails a modest increase in total cost. This comparison shows the effectiveness of the

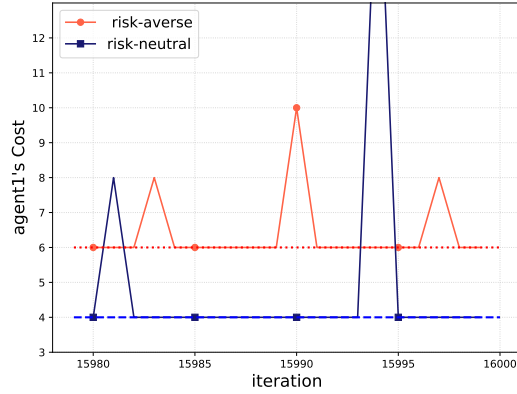


Fig. 5. Comparison of the Costs for risk-averse and risk-neutral

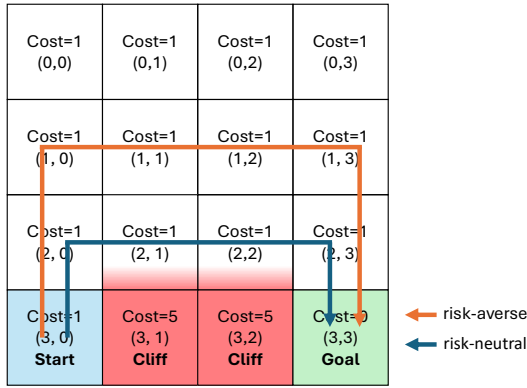


Fig. 6. Comparison of the policies for risk-averse and risk-neutral

proposed risk-averse framework in navigating environments where avoiding extreme risks is crucial.

VI. CONCLUSION

In this study, we have presented risk-averse distributed reinforcement learning in multi-agent systems. By incorporating CVaR into the distributed policy gradient method, we have developed an approach that enables multiple agents to collaboratively learn an optimal policy that effectively balances performance optimization with risk mitigation. We have shown that the policy generated by the algorithm converges to a locally optimal solution. The numerical experiments have demonstrated that, compared to traditional risk-neutral methods, the proposed algorithm achieves more stable learning to avoid rare but adverse events. Potential extensions include investigating adaptive risk sensitivity based on environmental conditions by incorporating heterogeneous risk preferences among agents.

REFERENCES

- [1] A. Nedić and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.
- [2] G. Qu and N. Li, "Harnessing smoothness to accelerate distributed optimization," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 3, pp. 1245–1260, 2018.

- [3] Y. Kajiyama, N. Hayashi, and S. Takai, "Linear convergence of consensus-based quantized optimization for smooth and strongly convex cost functions," *IEEE Transactions on Automatic Control*, vol. 66, no. 3, pp. 1254–1261, 2021.
- [4] S. Cheng, S. Liang, Y. Fan, and Y. Hong, "Distributed gradient tracking for unbalanced optimization with different constraint sets," *IEEE Transactions on Automatic Control*, vol. 68, no. 6, pp. 3633–3640, 2023.
- [5] H. Sakuma, N. Hayashi, and S. Takai, "Distributed primal-dual method on unbalanced digraphs with row stochasticity," *International Journal of Control*, vol. 97, no. 6, pp. 1377–1388, 2024.
- [6] J. Zhang, S. Liu, and L. Xie, "Distributed convex optimization over nonlinear networks under set constraints," *IEEE Transactions on Automatic Control*, pp. 1–8, 2025. [Online]. Available: 10.1109/TAC.2025.3535578
- [7] T. T. Nguyen, N. D. Nguyen, and S. Nahavandi, "Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications," *IEEE Transactions on Cybernetics*, vol. 50, no. 9, pp. 3826–3839, 2020.
- [8] Y. Zhang, M. Qiu, and H. Gao, "Communication-efficient stochastic gradient descent ascent with momentum algorithms," *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pp. 4602–4610, 2023.
- [9] Z. Ning and L. Xie, "A survey on multi-agent reinforcement learning and its application," *Journal of Automation and Intelligence*, vol. 3, no. 2, pp. 73–91, 2024.
- [10] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 2018.
- [11] S. Gu, L. Yang, Y. Du, G. Chen, F. Walter, J. Wang, and A. Knoll, "A review of safe reinforcement learning: Methods, theories, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 12, pp. 11 216–11 235, 2024.
- [12] Y. Chow, A. Tamar, S. Mannor, and M. Pavone, "Risk-sensitive and robust decision-making: A cvar optimization approach," *Proceedings of the 29th International Conference on Neural Information Processing Systems*, pp. 1522–1530, 2015.
- [13] R. T. Rockafellar and S. Uryasev, "Optimization of conditional value-at-risk," *Journal of Risk*, vol. 2, no. 3, pp. 21–42, 2000.
- [14] X. Yu and L. Ying, "On the global convergence of risk-averse policy gradient methods with expected conditional risk measures," *Proceedings of the 40th International Conference on Machine Learning*, vol. 202, pp. 40 425–40 451, 2023.
- [15] Q. Zhang, S. Leng, X. Ma, Q. Liu, X. Wang, B. Liang, Y. Liu, and J. Yang, "CVaR-constrained policy optimization for safe reinforcement learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 36, no. 1, pp. 830–841, 2025.
- [16] S. Lu, K. Zhang, T. Chen, T. Başar, and L. Horesh, "Decentralized policy gradient descent ascent for safe multi-agent reinforcement learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 10, pp. 8767–8775, 2021.
- [17] L. Zhang, L. Li, W. Wei, H. Song, Y. Yang, and J. Liang, "Scalable constrained policy optimization for safe multi-agent reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 37, pp. 138 698–138 730, 2025.
- [18] H. Du, F. Gou, and Y. Cai, "Scalable safe multi-agent reinforcement learning for multi-agent system," *arXiv preprint arXiv:2501.13727*, 2025.
- [19] W. Qiu, X. Wang, R. Yu, R. Wang, X. He, B. An, S. Obraztsova, and Z. Rabinovich, "RMIX: Learning risk-sensitive policies for cooperative reinforcement learning agents," *Advances in Neural Information Processing Systems*, vol. 34, pp. 23 049–23 062, 2021.
- [20] A. Nedić, A. Olshevsky, and M. G. Rabbat, "Network topology and communication-computation tradeoffs in decentralized optimization," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 953–976, 2018.
- [21] A. Al Maruf, L. Niu, B. Ramasubramanian, A. Clark, and R. Poovendran, "Risk-aware distributed multi-agent reinforcement learning," *Proceedings of the 2024 American Control Conference*, pp. 4012–4019, 2024.
- [22] V. S. Mai and E. H. Abed, "Distributed optimization over directed graphs with row stochasticity and constraint regularity," *Automatica*, vol. 102, pp. 94–104, 2019.
- [23] A. Nedić, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, 2010.