Risk Averse Reinforcement Learning for Mixed Multi-Agent Environments Supplementary Material

0.1 Model Architecture

Figure 3 captures the details of our algorithm framework.

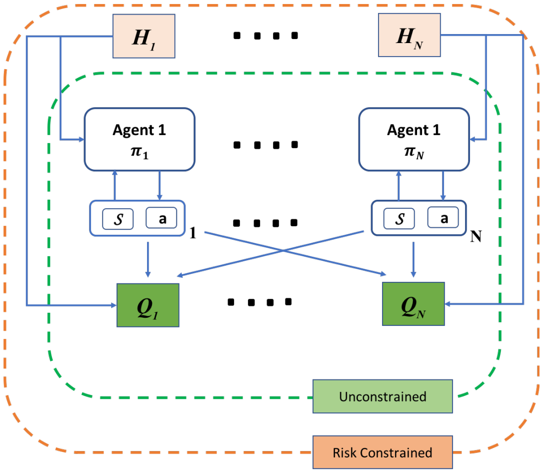

Figure 1: Overview of our risk constrained multi-agent decentralized actor, centralized critic system.

1 Convergence Analysis: Detailed Proof

In this section, we will show the convergence analysis of the general multi-agent algorithms for the mixed multi-agent environments. We show that, if the algorithm is developed in such a way that it satisfies certain assumptions, then our convergence results holds. We now list the assumption under which the convergence of the algorithm is shown:

- **Assumption 1:** The underlying Markov chain for any agent i determining the dynamics $\{X_n^{\theta_i}\}$, under stationary parameterized random policy π^{θ_i}, is irreducible.
- **Assumption 2:** The policy $\pi^{\theta_i}(s,a)$ is continuously differentiable in θ_i and $\nabla_{\theta_i}\pi^{\theta_i}(s,a)$ is a Lipschitz continuous function in θ_i, $\forall i$ and $\forall(s,a)$.
- **Assumption 3:** There exists a policy π^{θ_i} such that $H_{\alpha_i}(R_i^{\theta_i}(s_i^0),\nu_i)) < \beta_i, \forall i$.
- **Assumption 4:** The step sizes a^n, b^n, c^n for all $n \geq 0$ satisfy the standard stochastic approximation algorithm conditions:

$$\begin{aligned} &\Sigma_n a_n = \Sigma_n b_n = \Sigma_n c_n = \infty \\ &\Sigma_n(a_n)^2, \Sigma_n(b_n)^2, \Sigma_n(c_n)^2 < \infty \\ &a_n = o(b_n), b = o(c_n). \end{aligned} \tag{1}$$

- **Assumption 5:** The unbiased estimate of the state value function Q^π. One can get unbiased estimate of the Q^π under full state representation. Also, for linear function approximation one can design algorithms that converge to approximate Q^π.

Remark 1: One can estimate the centralized action value function Q^π in multi-agent case by maintaining the "centralized critic".

Under the Assumptions $1-5$, for a given Lagrange multiplier λ_i the policy updates for each agent in the Algorithm **??** converges almost surely to a locally optimal policy, i.e $\theta_i^n \rightarrow \theta_i^*$ satisfying the respective risk sensitive measure. The following are the main steps of the proof:

- Show that each parameter update in the multi-time scale stochastic approximation algorithms converge to a stationary point.
- Need to show that the iterates are stable. Construct the Lyapunov function.
- Use the saddle point theorem to argue that the stationary point satisfies the risk sensitive constraints

Step 1: (ν convergence)

The ν_i update for agent i converges on the faster time scale compared to θ_i and λ_i. The ν_i-update can be written as by fixing the (θ_i, λ_i):

$$\nu_i^{n+1} = \nu_i^n - c^n(\lambda_i - \frac{\lambda_i}{(1-\alpha_i)N}\Sigma_{j=1}^N \mathbb{1}\{R^{\theta_i}(s_i^0) \geq \nu_n\}) \tag{2}$$

We need to show that the iterates converge to the set which depends on λ_i and θ_i. The corresponding ODE is as follows:

$$\dot{\nu} = \bar{\Gamma}_\nu(-\partial_\nu L(\nu,\theta,\lambda)) \tag{3}$$

where $\bar{\Gamma}_\nu$ is defined as follows: And let f be any continuous and bounded function

$$\bar{\Gamma}_\nu(f(\nu^n)) = \lim_{0<\eta\rightarrow 0}\frac{\Gamma(\nu+\eta f(\nu^n)) - \nu^n}{\eta} \tag{4}$$

Note that for any $\nu^n \in C_\nu$ then $\bar{\Gamma}_\nu(f(\nu^n)) = f(\nu^n)$, however Γ acts as a projection operator for $\nu^n \notin C_\nu$. Here set C_ν is the one in which ν_i takes the values.

$$\dot{\nu} = \bar{\Gamma}_\nu(-\partial_\nu L(\nu, \theta, \lambda)) \tag{5}$$

One can analyze the convergence properties of ν from the Lemma 1, Chapter 6 of [Borkar, 2009]. For verifying whether algorithm satisfies the conditions of the Lemma, one can see Theorem 7 of [Chow *et al.*, 2018]. From this one can argue that the $\{\nu_i^n\}$ sequence converges to the fixed point of the ODE 3, where

$$\nu^* \in C_\nu := \{\nu \in [\nu_{Min}, \nu_{Max}] \mid \bar{\Gamma}_\nu[-\partial_\nu L(\nu, \theta, \lambda)] = 0\} \tag{6}$$

Now, following Lyapunov function can be used to prove the stability:

$$L_{\theta,\lambda}(\nu) = L(\nu, \theta, \lambda) - L(\nu^*, \theta, \lambda) \tag{7}$$

Here, ν^* is a minimum point. One can conclude that the trajectory of 3 converges to ν^*.

Step 2: (**θ convergence**) The ν_i update for agent i converges on the faster time scale compared to θ_i and λ_i. In order to write the θ_i update, we can assume that the ν_i is converged to $\nu^*(\theta_i)$. Since the λ_i is on slower time scale compared to θ_i, one can assume that λ_i as a fixed quantity. The θ_i-update can be written as :

$$\theta_i^{n+1} = \theta_i^n - c^n(\nabla_{\theta_i}(L(\nu_i, \theta_i, \lambda_i))) \tag{8}$$

We need to show that the iterates converge to the set which depends on λ_i. One can refer [Chow *et al.*, 2018] for similar convergence results. The corresponding ODE is as follows:

$$\dot{\theta} = \bar{\Gamma}_\Theta(-\nabla_\theta L(\nu, \theta, \lambda)) \tag{9}$$

Where $\bar{\Gamma}$ is defined as follows: And let f be any continuous and bounded function

$$\bar{\Gamma}_\Theta(f(\theta^n)) = \lim_{0<\eta\to 0} \frac{\Gamma(\theta + \eta f(\theta^n)) - \theta^n}{\eta} \tag{10}$$

Note that for any $\theta^n \in \Theta$ then $\bar{\Gamma}(f(\theta^n)) = f(\theta^n)$, however Γ acts as a projection operator for $\theta^n \notin \Theta$. Here set Θ is the one in which θ_i takes the values.

From Assumption 5, the gradient points to the descent direction from the policy gradient theorem. This assumption is made possible because of the "centralized critic" architecture where critic observes the state and actions of other agents, and can estimate the Q^π. However, this assumption is valid only for full state information settings or with linear function approximation. Our algorithm does not satisfy this assumption but there are several algorithms for which this assumption is valid (However, empirically our algorithm is shown to perform well for multi-agent settings [Lowe *et al.*, 2017]). Hence, one can see that the θ_i update is the stochastic approximation of the ODE 9. The following function can be used as Lyapunov function for any given λ:

$$L_\lambda(\theta) = L(\nu^*(\theta), \theta, \lambda) - L(\nu^*(\theta^*), \theta^*, \lambda) \tag{11}$$

Here, θ^* is a minima. One can use the similar arguments from [Chow *et al.*, 2018] to prove that the $\{\theta_i^n\}$ converges to θ^*. And $\{\theta_k, \nu_k\}$ converges to a local minimum of $L(\nu, \theta, \lambda)$,i.e, (θ^*, ν^*) for any fixed λ.

Step 3: (**λ convergence**) The λ updates are on the slower time scale. While analyzing the λ update one can use the converged $\theta^*(\lambda)$ and $\nu^*(\lambda)$. The λ update can be written as follows:

$$\lambda_i^{n+1} = \max(0, \lambda_i^n + c^n(\nabla_\lambda L(\nu, \theta, \lambda)) \tag{12}$$

The corresponding ODE is as follows:

$$\dot{\lambda} = \bar{\Gamma}_\lambda(-\nabla_\lambda L(\nu, \theta, \lambda)) \tag{13}$$

Where $\bar{\Gamma}_\lambda$ is defined as follows: And let f be any continuous and bounded function

$$\bar{\Gamma}_\lambda(f(\lambda^n)) = \lim_{0<\eta\to 0} \frac{\Gamma(\lambda + \eta f(\lambda^n)) - \lambda^n}{\eta} \tag{14}$$

Note that for any $\lambda^n \in C_\lambda$ then $\bar{\Gamma}_\lambda(f(\lambda^n)) = f(\lambda^n)$, however Γ acts as a projection operator for $\lambda^n \notin C_\lambda$. Here set C_λ is the one in which λ_i takes the values. Consider the Lyapunov function:

$$L_\lambda(\lambda) = -L(\theta^*(\lambda), \nu^*(\lambda), \lambda) + L(\theta^*(\lambda^*), \nu^*(\lambda^*), \lambda^*) \tag{15}$$

where λ^* is a local minimum point. One can use the similar arguments of [Chow *et al.*, 2018] to prove that the sequence $\{\lambda_i^n\}$ converges to $\lambda^* \in C_\lambda$.

Now one can use the similar arguments from [Chow *et al.*, 2018] to argue that the $\theta_i^*, \nu_i^*, \lambda_i^*$ is a saddle point of the Lagrangian **??** for agent i. By saddle point theorem, θ_i^* is the optimal solution that satisfy the $CVaR$ constraints of the agent i.

2 Description of Scenarios

Keep Away: This environment consists of M good agents whose target is to reach a target landmark T from a set of landmarks L. These agents pick the target landmark T at the start of the game and learn to cooperate between themselves to confuse the N adversarial agents from reaching T. Naturally, the goal for the adversarial agents is to identify the target T and keep the good agents away from it. But the adversaries are not aware of the correct target T and they cooperate to detect the target from the movements of good agents. The reward for both the good agents and adversaries is based on its distance to the landmark T and additionally for adversaries its distance to the good agents are included.

Physical Deception: This environment consists of L landmarks, M good agents and N adversaries. Similar to the previous task, target landmark T is chosen at the start of the game by the good agents. The adversaries learn the target from the movements of the good agents. Again, all agents observe the position of the landmarks and other agents. Good agents are rewarded if one of them reach the target and are penalized if the adversary gets closer to the target. Therefore, the agents learn to cooperate to spread out and cover landmarks to deceive the adversary away from the target.

Predator Prey: This environment consists of N predators who must learn to cooperate together to catch M preys. The preys have faster movement but are impeded with the L large obstacles in the environment. The agents can observe the relative positions, velocities of the agents and the positions of the landmarks. There can be two reward strategies in this game:

- **Shared reward :** where if any of the agent catch the prey, the reward is shared among all the cooperative agents otherwise it is none for all agents.
- **Individual reward :** where only the agent who catch the prey gets the complete reward and others none.

2.1 Hyperparameter Details

The policy parameters for the actor and critic are updated at two different timescales, the critic being at a faster timescale (with learning rate of 0.01) and the actor at a lower timescale (learning rate 0.005). Further, while the lagrangian parameter needs to be updated at a much lower timescale (learning rate 0.001), the cVaR parameter can be updated at a faster timescale (learning rate 0.01).

References

[Borkar, 2009] Vivek S Borkar. *Stochastic approximation: a dynamical systems viewpoint*, volume 48. Springer, 2009.

[Chow *et al.*, 2018] Yinlam Chow, Mohammad Ghavamzadeh, Lucas Janson, and Marco Pavone. Risk-constrained reinforcement learning with percentile risk criteria. *Journal of Machine Learning Research*, 18(167):1–51, 2018.

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

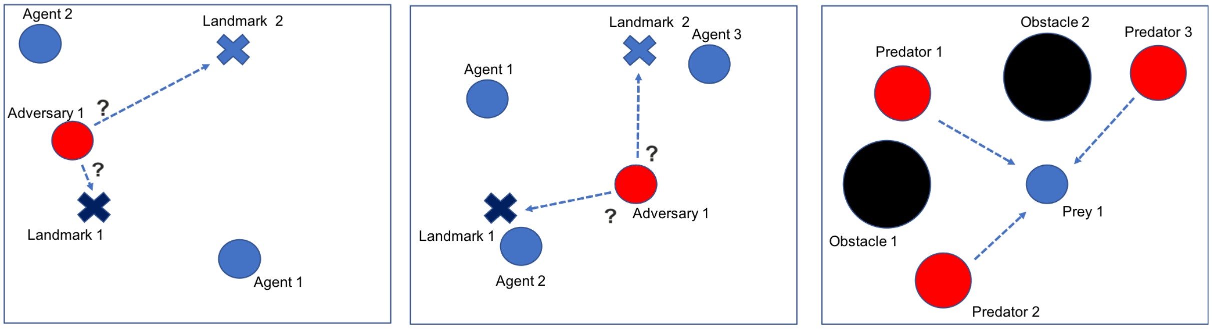

Figure 2: **Illustration of scenarios:** Overview of our risk constrained multi-agent decentralized actor, centralized critic system.

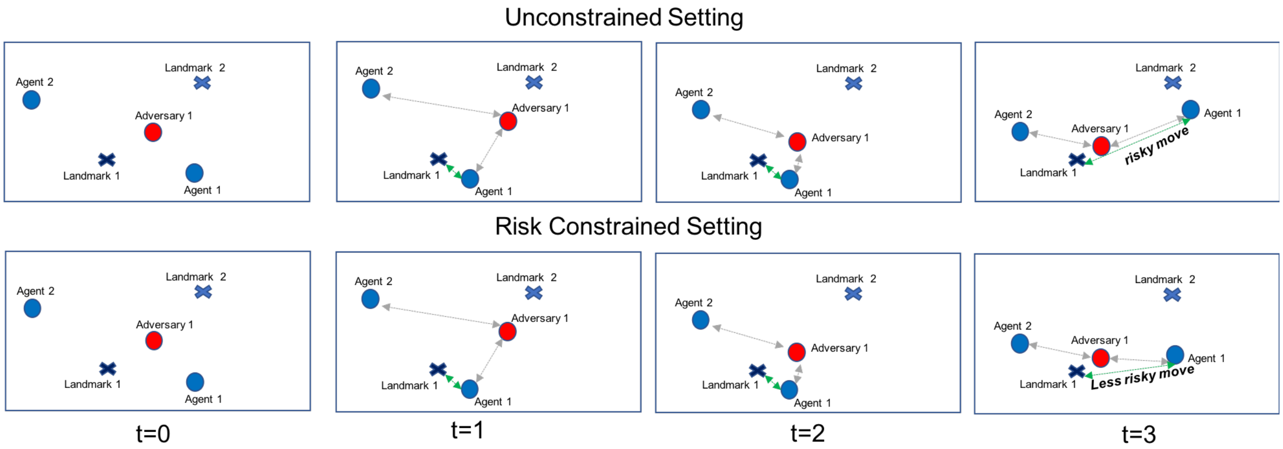

Figure 3: **Sequences of events for the two settings 1)Unconstrained 2)Risk Constrained for the keep away task for 4 continuous time steps**. Time steps t=0-2 are common to both the settings which shows the observation by agents and movement towards landmark. The good agents movements are rewarded based on its close distance to the target Landmark 1. The adversary learns the target Landmark based on the movements of agents and is rewarded based on its minimal distance to the target landmark 1 and its distance to the other agents. At time step t=3, the first row shows the high variance in policy of the agent near the landmark. This is understandable by the giant leap by the agent to move away from the adversary towards the other landmark to fool the adversary. But this is a risky move, since it might take longer sequence of bad rewards before reaching the target. The second row shows the constrained setting wherein the agent near the landmark is restricted to shorter movements leading to smaller variance even when the adversary is near by thereby, reduces the risk of bad rewards