RL-ARNE: A Reinforcement Learning Algorithm for Computing Average Reward Nash Equilibrium of Nonzero-Sum Stochastic Games

Dinuka Sahabandu, Shana Moothedath, *Member, IEEE,* Joey Allen,
Linda Bushnell, *Fellow, IEEE,* Wenke Lee, *Fellow, IEEE,* and Radha Poovendran, *Fellow, IEEE*

***Abstract*—Stochastic games model the strategic interactions between two or more players that occur in a sequence of stages. In this paper we focus on computing the average reward Nash equilibrium (ARNE) of a nonzero-sum stochastic game when the transition probabilities of the game and reward structure of the players are unknown. We note that the current state-of-the-art reinforcement learning (RL) algorithms that compute the ARNE of nonzero-sum stochastic games requires solving a matrix game corresponding to each state of the game at every iteration of the algorithm, which is PPAD[1]-complete and incurs a memory complexity that is exponential in the number of players. In this paper, we use temporal difference error minimization and stochastic approximation to develop a scalable RL algorithm to compute an ARNE of nonzero-sum stochastic games. We prove the convergence of our algorithm to an ARNE. We evaluate the performance of our algorithm using an attacker-defender game modeled on a real-world ransomware dataset.**

***Index Terms*—Stochastic games, Average reward Nash equilibrium, Reinforcement learning**

I. Introduction

Stochastic games introduced by Shapley generalize Markov decision processes to model the strategic interactions between two or more players that occur in a sequence of stages [1]. Dynamic nature of stochastic games enables the modeling of competitive market scenarios in economics [2], competition within and between species for resources in evolutionary biology [3], resilience of cyber-physical systems in engineering [4], and secure networks under adversarial interventions in computer science [5].

Study of stochastic games is often focused on finding a set of *Nash Equilibrium* (NE) [6] policies for the players such that no player is able to increase their respective payoffs by unilaterally deviating from their NE policies. The payoffs of a stochastic game are usually evaluated under discounted or limiting average payoff [7], [8]. In games with discounted payoff, the future rewards of the players are scaled down by a factor between zero and one, and existence of an NE is always guaranteed [9]. Limiting average payoff on the other hand considers the time-average of the rewards received by the players [8]. The existence of an NE under limiting average payoff criteria for a general stochastic game is an open problem. When an NE exists, value iteration, policy iteration, and linear/nonlinear programming based approaches are proposed in the literature to find an NE [7], [10]. These approaches, however, require the knowledge of transition and the reward structures of the game. Further, these solution approaches are only guaranteed to find an exact NE for special classes of stochastic games, such as zero-sum games, where rewards of the players sum up to zero in all the game states [7].

Multi-agent reinforcement learning (MARL) algorithms have been proposed in the literature to obtain NE policies of stochastic games when the transition probabilities of the game and reward structure of the players are unknown. MARL algorithms can be grouped into three categories based on the objectives of the players [11]. (i) Cooperative games where players coordinate to achieve a common goal. (ii) Competitive games where players compete against each other, and for any set of strategies the sum of the rewards to all players is zero (referred to as zero-sum stochastic games). (iii) Mixed games where each player tries to maximize its individual payoff function and the rewards of the players may not necessarily add up to zero (referred to as nonzero-sum stochastic games). In this paper we focus on MARL algorithms for mixed games.

The authors of [12], [13] introduced a Q-learning algorithm (Nash-R) to learn an NE of average reward stochastic games. Nash-R was *empirically* shown to find an NE of a nonzero-sum game by ensuring the players always use the same NE value for updating their Q-values. However, the convergence guarantee of Nash-R assumes that the game has a unique NE value, which is often not satisfied by most of the games that model real-world applications [7]. Nash-R also requires solving a matrix game corresponding to each state of the game at every iteration of the algorithm which is PPAD[1]-complete [14], [15] and incurs an exponential memory complexity in number of players [12], [13]. Q-learning algorithms proposed in [16], [17] for discounted stochastic games require solving matrix games and similar conditions as in Nash-R for the convergence.

References in [18], [19], [20] developed actor-critic algorithms to enhance the scalability of MARL algorithms for nonzero-sum *discounted* stochastic games. Later [21], [22], [23] introduced efficient MARL algorithms for NE in zero-sum stochastic games, but these are not suited for the nonzero-sum games we explore here. *In this work our goal is to develop a scalable algorithm to compute average reward NE of nonzero-sum stochastic games when the transition structure of the game is unknown.* Though we employ a multi-time scale, TD error minimization method similar to [18], our work diverges by focusing on *average reward* stochastic games and presents distinct convergence proofs, underscoring key differences in approach and analysis. Our contributions are as follows.

- We provide a reinforcement learning algorithm, RL-ARNE, that learns an average reward Nash equilibrium (ARNE) of nonzero-sum stochastic games using TD error minimization.
- We prove the convergence of RL-ARNE algorithm to an ARNE of the game using stochastic approximation.
- We evaluate the performance of RL-ARNE algorithm via an attacker-defender game grounded on ransomware attack data obtained from Refinable Attack INvestigation (RAIN) [24].

Organization of the Paper: Section II presents definitions and existing results. Section III presents an RL algorithm to compute an ARNE of nonzero-sum stochastic games. Section IV provides evaluation of the proposed algorithm using an attacker-defender game grounded on a real-world dataset. Section V presents the conclusions.

D. Sahabandu, L. Bushnell, and R. Poovendran are with the Department of Electrical and Computer Engineering, University of Washington, Seattle, WA 98195, USA. {sdinuka, lb2, rp3}@uw.edu.

S. Moothedath is with the Department of Electrical and Computer Engineering, Iowa State University, IA 50011 USA. mshana@iastate.edu.

J. Allen and W. Lee are with the College of Computing, Georgia Institute of Technology, Atlanta, GA 30332 USA. jallen309@gatech.edu, wenke@cc.gatech.edu.

This work was supported by ONR grant N00014-16-1-2710 P00002, DARPA grant FA8650-15-C-7556, and NSF grant 2229876 and was supported in part by funds provided by DHS, and by IBM. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF or its federal agency and industry partners.

[1]Polynomial Parity Arguments on Directed graphs.

II. Formal Definitions and Existing Results

Stochastic Games: A stochastic game $\mathbb{G}$ is defined as a tuple $< K, \mathbf{S}, \mathcal{A}, \mathbf{P}, r >$, where K denotes the number of players, $\mathbf{S}$ represents the state space, $\mathcal{A} := \mathcal{A}_1 \times \ldots, \times \mathcal{A}_K$ denotes the action space, $\mathbf{P}$ designates the transition probability kernel, and r represents the reward functions. Here $\mathbf{S}$ and $\mathcal{A}$ are finite spaces. Let $\mathcal{A}_k := \cup_{s\in\mathbf{S}} \mathcal{A}_k(s)$ be the action space of each player $k \in \{1,\ldots,K\}$, where $\mathcal{A}_k(s)$ denotes the set of actions allowed for player k at state $s \in \mathbf{S}$. Let $\boldsymbol{\pi}_k$ be the set of stationary policies corresponding to player $k \in \{1,\ldots,K\}$. A policy $\pi_k \in \boldsymbol{\pi}_k$ is said to be a deterministic stationary policy if $\pi_k \in \{0,1\}^{|\mathcal{A}_k|}$ and said to be a stochastic stationary policy if $\pi_k \in [0,1]^{|\mathcal{A}_k|}$. Let $\mathbf{P}(s'|s,a_1,\ldots,a_K)$ be the probability of transitioning from state $s \in \mathbf{S}$ to a state $s' \in \mathbf{S}$ under set of actions $(a_1,\ldots,a_K)$, where $a_k \in \mathcal{A}_k(s)$ denotes the action chosen by player k at the state s. Further let $r_k(s,a_1,\ldots,a_K,s')$ be the reward received by the player k when state of $\mathbb{G}$ transitions from states s to s' under set of actions $(a_1,\ldots,a_K)$.

Average Reward Payoff Structure: Let $\pi = (\pi_1,\ldots,\pi_K)$. Define $\rho_k(s,\pi)$ to be the average reward payoff of player k when the game starts at an arbitrary state $s \in \mathbf{S}$ and the players follow their respective policies π. Let s^t and a_k^t be the state of game at time t and the action of player k at time t, respectively. Then $\rho_k(s,\pi)$ is defined as

$$\rho_k(s,\pi) = \liminf_{T\to\infty} \frac{1}{T+1} \sum_{t=0}^{T} \mathbb{E}_{s,\pi}[r_k(s^t,a_1^t,\ldots,a_K^t)], \tag{1}$$

where the term $\mathbb{E}_{s,\pi}[r_k(s^t,a_1^t,\ldots,a_K^t)]$ denotes the expected reward at time t when the game starts from a state s and the players draw a set of actions $(a_1^t,\ldots,a_K^t)$ at current state s^t based on their respective policies from π. All the players in $\mathbb{G}$ aim to maximize their individual payoff values in Eqn. (1). Let $-k$ be the opponents of a player $k \in \{1,\ldots,K\}$ (i.e., $-k := \{1,\ldots,K\}\backslash k$). Then let $\pi_{-k} := \{\pi_1,\ldots,\pi_K\} \setminus \pi_k$ denotes a set of stationary policies of the opponents of player k. Equilibrium of $\mathbb{G}$ under average reward criteria is given below.

Definition II.1 (ARNE)**.** *A set of stationary policies $\pi^* = (\pi_1^*,\ldots,\pi_K^*)$ forms an ARNE of $\mathbb{G}$ if and only if $\rho_k(s,\pi_k^*,\pi_{-k}^*) \geq \rho_k(s,\pi_k,\pi_{-k}^*)$, for all $s \in \mathbf{S}, \pi_k \in \boldsymbol{\pi}_k$ and $k \in \{1,\ldots,K\}$.*

A policy $\pi^* = (\pi_1^*,\ldots,\pi_K^*)$ is referred to as an ARNE of $\mathbb{G}$. When all the players follow ARNE policy, no player k can increase its payoff by unilaterally deviating from its respective ARNE policy π_k^*.

Unichain Stochastic Games: Let $\mathbf{P}(\pi)$ be the transition probability structure of $\mathbb{G}$ induced by a set of deterministic policies π. Stochastic games that satisfy Assumption II.2-(a) are referred to as *unichain*.

Assumption II.2. *(a) For each deterministic policy set π, induced Markov chain $\mathbf{P}(\pi)$ consists of one recurrent class of states. (b) There exists a state s_0 such that for every deterministic π, s_0 is visited with a nonzero probability within the first m stages for some integer m.*

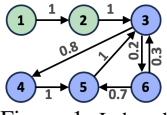


Figure 1: Induced MC with recurrent and transient states.

Assumption II.2-(a) imposes a structural constraint on the Markov chain (MC) induced by *deterministic* stationary policy set. Any $\mathbb{G}$ that also satisfies Assumption II.2-(b) will contain only one recurrent class in $\mathbf{P}(\pi)$ for any given *stochastic* policy set π [25]. MC with a single recurrent class need not necessarily contain all $s \in \mathbf{S}$. There may exist some transient states in $\mathbf{P}(\pi)$. Figure 1 shows an example instance of $\mathbf{P}(\pi)$ induced by π. It distinguishes between the recurrent class (in blue) and transient states (in green), with directional arrows representing possible transitions and probabilities.

Let $\mathbb{R}_l$ and $\mathbb{T}$ denote a set of states in the l^{th} recurrent class of $\mathbf{P}(\pi)$ for $l \in \{1,\ldots,L\}$, and a set of transient states in $\mathbf{P}(\pi)$, respectively, where L denote the number of recurrent classes. Proposition II.3 gives results on the average reward values of the states in each $\mathbb{R}_l$ and $\mathbb{T}$.

Proposition II.3 ([7], Section 3.2)**.** *The following statements are true for any induced MC $\mathbf{P}(\pi)$ of $\mathbb{G}$.*

1) *For $l \in \{1,\ldots,L\}$ and for all $s \in \mathbb{R}_l$, $\rho_k(s,\pi) = \rho_k^l$, where each ρ_k^l denotes a real-valued constant.*
2) *$\rho_k(s,\pi) = \sum_{l=1}^{L} q_l(s)\rho_k^l$, if $s \in \mathbb{T}$, where $q_l(s)$ is the probability of reaching a state in l^{th} recurrent class from s.*

1) in Proposition II.3 implies that the average reward payoff of player k takes the same value ρ_k^l for each state in the l^{th} recurrent class. 2) suggests that the average reward payoff of a transient state is a convex combination of the average payoffs corresponding to L recurrent classes $\rho_k^1,\ldots,\rho_k^L$. Proposition II.3 shows that for any $\mathbb{G}$, the average reward payoffs corresponding to each state solely depends on the average reward payoffs of the recurrent classes in $\mathbf{P}(\pi)$.

ARNE in Unichain Stochastic Games: Existence of an ARNE for nonzero-sum stochastic games is open. However, the existence of ARNE is shown for some special classes of stochastic games [7].

Proposition II.4 ([8], Theorem 2)**.** *There exists an ARNE for a stochastic game that satisfies Assumption II.2.*

Let $\pi_k \in \boldsymbol{\pi}_k$ be expressed as $\pi_k = [\pi_k(s)]_{s\in\mathbf{S}}$, where $\pi_k(s) = [\pi_k(s,a_k)]_{a_k\in\mathcal{A}_k(s)}$. Further let $\bar{a} := (a_1,\ldots,a_K)$ and $a_{-k} := \bar{a}\backslash a_k$. Define $\mathbf{P}(s'|s,a_k,\pi_{-k}) = \sum_{a_{-k}\in\mathcal{A}_{-k}(s)} \mathbf{P}(s'|s,\bar{a})\pi_{-k}(s,a_{-k})$, where $\mathbf{P}(s'|s,\bar{a})$ is the probability of transitioning to a state s' from state s under action set $\bar{a}$. Also let $r_k(s,a_k,\pi_{-k}) = \sum_{s'\in\mathbf{S}} \sum_{a_{-k}\in\mathcal{A}_{-k}(s)} \mathbf{P}(s'|s,\bar{a}) r_k(s,\bar{a},s')\pi_{-k}(s,a_{-k})$, where $r_k(s,\bar{a},s')$ is the reward for player k under action set $\bar{a}$ when a state transitions from s to s'. Let $\Omega_{k,\pi_{-k}}^{s,a_k}$ and $\Delta(\pi)$ be defined as follows:

$$\Omega_{k,\pi_{-k}}^{s,a_k} = \rho_k + v_k(s) - r_k(s,a_k,\pi_{-k}) - \sum_{s'\in\mathbf{S}} \mathbf{P}(s'|s,a_k,\pi_{-k}) v_k(s'), \tag{2}$$

$$\Delta(\pi) \quad = \quad \sum_{k\in\{D,A\}} \sum_{s\in\mathbf{S}} \sum_{a_k\in\mathcal{A}_k(s)} \Omega_{k,\pi_{-k}}^{s,a_k} \pi_k(s,a_k), \tag{3}$$

where $v_k(s)$ is the "value" of the game for player k at state s and ρ_k denotes the average reward value of player k independent of initial state of the game. $\Omega_{k,\pi_{-k}}^{s,a_k}$ denotes the Temporal Difference (TD) error associated with player k at state s when taking action a_k. TD error represents the difference between the predicted future rewards and the actual rewards obtained. TD error is used to update the value function and the policy in RL algorithms. A high TD error indicates that the predictions are significantly different from the actual outcomes, suggesting that the player's model of the environment needs substantial updating. A low TD error suggests that the player's predictions are accurate. The term $\Delta(\pi)$ represents the total TD error, which sums the TD error over all states and actions for all players. Necessary and sufficient conditions for characterizing an ARNE of a stochastic game that satisfies Assumption II.2 is given below.

Proposition II.5 ([8], Theorem 4)**.** *Under Assumption II.2, a set of stochastic policies $\pi = (\pi_1,\ldots,\pi_K)$ forms an ARNE in $\mathbb{G}$ if and only if π satisfies the following for all $s \in \mathbf{S}, a_k \in \mathcal{A}_k(s), k \in \{1,\ldots,K\}$.*

$$\Omega_{k,\pi_{-k}}^{s,a_k} \geq 0, \tag{4a}$$

$$\Delta(\pi) = 0, \tag{4b}$$

$$\sum_{a_k\in\mathcal{A}_k(s)} \pi_k(s,a_k) = 1, \quad \pi_k(s,a_k) \geq 0. \tag{4c}$$

Stochastic Approximation Algorithms: Let $h : \mathscr{R}^{m_z} \to \mathscr{R}^{m_z}$ be a continuous function of a set of parameters $z \in \mathscr{R}^{m_z}$. Then Stochastic Approximation (SA) algorithms solve a set of equations of the form $h(z) = 0$ based on the noisy measurements of $h(z)$.

$$z^{n+1} = z^n + \delta_z^n [h(z^n) + w_z^n], \text{ for } n \geq 0. \tag{5}$$

Here, n denotes the iteration index and z^n denote the estimation of z at n^{th} iteration of the algorithm. The terms w_z^n and δ_z^n represent the zero mean measurement noise associated with z^n and the step-size of the algorithm, respectively. Note that the stationary points of Eqn. (5) coincide with the solutions of $h(z)=0$ when the noise term w_z^n is zero. Convergence analysis of SA algorithms requires investigating their associated Ordinary Differential Equations (ODEs). The ODE form of the SA algorithm in Eqn. (5) is given by $\dot{z}=h(z)$.

Additionally, the following assumptions on step-size δ_z^n are required to guarantee the convergence of an SA algorithm.

Assumption II.6. *δ_z^n satisfies, $\sum_{n=0}^{\infty}\delta_z^n=\infty$ and $\sum_{n=0}^{\infty}(\delta_z^n)^2<\infty$.*

Few examples of δ_z^n that satisfy the conditions given in Assumption II.6 are $\delta_z^n=1/n$ and $\delta_z^n=1/n\log(n)$. A convergence result that holds for a more general class of SA algorithms is given below.

Proposition II.7 ([26], [27])**.** *Consider an SA algorithm in the following form defined over a set of parameters $z\in\mathscr{R}^{m_z}$ and a continuous function $h:\mathscr{R}^{m_z}\to\mathscr{R}^{m_z}$.*

$$z^{n+1}=\Theta(z^n+\delta_z^n[h(z^n)+w_z^n+\kappa^n]),\ \textit{for } n\geq 0, \tag{6}$$

where Θ is a projection operator that projects each z^n iterates onto a compact and convex set $\Lambda\in\mathscr{R}^{m_z}$ and κ^n denotes a bounded random sequence. Let the ODE associated with the iterate in Eqn. (6) *is*

$$\dot{z}=\bar{\Theta}(h(z)), \tag{7}$$

where $\bar{\Theta}(h(z))=\lim_{\eta\to 0}\frac{\Theta(z+\eta h(z))-z}{\eta}$ and $\bar{\Theta}$ denotes a projection operator that restricts the evolution of ODE in Eqn. (7) *to the set Λ. Let the nonempty compact set Z denotes a set of asymptotically stable equilibrium points of Eqn.* (7)*. Then z^n converges almost surely to a point in Z as $n\to\infty$ given the following conditions are satisfied.*

1) *δ_z^n satisfies the conditions in Assumption II.6.*
2) *$\lim_{n\to\infty}\left(\sup_{\bar{n}>n}\left|\sum_{l=n}^{\bar{n}}\delta_z^n w_z^n\right|\right)=0$ almost surely.*
3) *$\lim_{n\to\infty}\kappa^n=0$ almost surely.*

Consider a class of SA algorithms that consist of two interdependent iterates that update on two different time scales (i.e., step-sizes of two iterates are different in the order of magnitude). Let $x\in\mathscr{R}^{m_x}$ and $y\in\mathscr{R}^{m_y}$ and $n\geq 0$. Then the iterates given in the following equations portray a format of such two-time scale SA algorithm.

$$x^{n+1}=x^n+\delta_x^n[f(x^n,y^n)+w_x^n],\ y^{n+1}=y^n+\delta_y^n[g(x^n,y^n)+w_y^n]. \tag{8}$$

The following proposition provides a convergence result related to the aforementioned two-time scale SA algorithm.

Proposition II.8 ([28], Chapter 6)**.** *Consider x^n and y^n iterates given in* (8)*. Then, given the iterates in* (8) *are bounded, $\{(x^t,y^t)\}$ converges to $(\psi(y^*),y^*)$ almost surely under the following conditions.*

(I) $f:\mathscr{R}^{m_x+m_y}\to\mathscr{R}^{m_x}$ and $g:\mathscr{R}^{m_x+m_y}\to\mathscr{R}^{m_y}$ are Lipschitz.
(II) Iterates x^n and y^n are bounded.
(III) Let $\psi:y\to x$. For all $y\in\mathscr{R}^{m_y}$, $\dot{x}=f(x,y)$ has an asymptotically stable critical point $\psi(y)$ such that function ψ is Lipschitz.
(IV) $\dot{y}=g(\psi(y),y)$ has a global asymptotically stable critical point.
(V) Let ξ^n be an increasing σ-field defined by $\xi^n:=\sigma(x^n,\ldots,x^0,y^n,\ldots,y^0,w_x^{n-1},\ldots,w_x^0,w_y^{n-1},\ldots,w_y^0)$. Further let κ_x and κ_y be two positive constants. Then w_x^n and w_y^n are two noise sequences that satisfy, $\mathbb{E}[w_x^n|\xi^n]=0$, $\mathbb{E}[w_y^n|\xi^n]=0$, $\mathbb{E}[\|w_x^n\|^2|\xi^n]\leq\kappa_x(1+\|x^n\|+\|y^n\|)$, and $\mathbb{E}[\|w_y^n\|^2|\xi^n]\leq\kappa_y(1+\|x^n\|+\|y^n\|)$.
(VI) δ_x^n and δ_y^n satisfy Assumption II.6. Additionally, $\limsup_{n\to\infty}\frac{\delta_y^n}{\delta_x^n}=0$.

III. Design and Analysis of RL-ARNE Algorithm

In this section we present a RL algorithm for learning an ARNE of a nonzero-sum stochastic games and analyze its convergence. For brevity we present the algorithm and its convergence results with respect to two players D and A.

A. RL-ARNE: RL Algorithm for Computing ARNE

Algorithm III.1 presents the pseudocode of RL-ARNE, a stochastic approximation-based algorithm with multiple time scales that computes an ARNE of a nonzero-sum stochastic game. The necessary and sufficient condition given in Proposition II.5 is used to find an ARNE policy pair $(\pi_D^\star,\pi_A^\star)$ in Algorithm III.1.

Algorithm III.1 RL-ARNE Algorithm

1: **Input:** State space ($\mathbf{S}$), Rewards (r_D, r_A), Max. iterations ($I>>0$)
2: **Output:** ARNE policies, $(\pi_D^\star,\pi_A^\star)\leftarrow(\boldsymbol{\pi}_D^I,\boldsymbol{\pi}_A^I)$
3: **Initialization:** $n\leftarrow 0$, $v_k^0\leftarrow 0$, $\rho_k^0\leftarrow 0$, $\varepsilon_k^0\leftarrow 0$, $\pi_k^0\leftarrow\boldsymbol{\pi}_k$ for $k\in\{D,A\}$ and $s\leftarrow s_0$.
4: **while** $n\leqslant I$ **do**
5: Draw $a_D=d$ from $\pi_D^n(s)$ and $a_A=a$ from $\pi_A^n(s)$
6: Reveal the next state s' according to $\mathbf{P}$
7: Observe the rewards $r_D(s,d,a,s')$ and $r_A(s,d,a,s')$
8: **for** $k\in\{D,A\}$ **do**
9: $v_k^{n+1}(s)=v_k^n(s)+\delta_v^n[r_k(s,d,a,s')-\rho_k^n+v_k^n(s')-v_k^n(s)]$
10: $\rho_k^{n+1}=\rho_k^n+\delta_\rho^n\left[\frac{n\rho_k^n+r_k(s,d,a,s')}{n+1}-\rho_k^n\right]$
11: $\varepsilon_k^{n+1}(s,a_k)=\varepsilon_k^n(s,a_k)+\delta_\varepsilon^n\big[\sum_{k\in\{D,A\}}(r_k(s,d,a,s')-\rho_k^n+v_k^n(s')-v_k^n(s))-\varepsilon_k^n(s,a_k)\big]$
12: $\pi_k^{n+1}(s,a_k)=\Gamma(\pi_k^n(s,a_k)-\delta_\pi^n\sqrt{\pi_k^n(s,a_k)}\big|r_k(s,d,a,s')-\rho_k^n+v_k^n(s')-v_k^n(s)\big|\mathrm{sgn}(-\varepsilon_k^n(s,a_k)))$
13: **end for**
14: Update the state of the game: $s\leftarrow s'$
15: $n\leftarrow n+1$
16: **end while**

Using SA, iterates in lines 9 and 10 compute the value functions $v_k^n(s)$, at each state $s\in\mathbf{S}$, and average rewards ρ_k^n of D and A corresponding to policy pair (π_D^n,π_A^n), respectively. The iterates, $\varepsilon_k^n(s,a_k)$ in line 11 and $\pi_k^n(s,a_k)$ in line 12, are chosen such that Algorithm III.1 converges to an ARNE of the game. We present below the outline of our approach.

In Theorem III.14 we prove that all the policies (π_D,π_A) such that $\Omega_{k,\pi_{-k}}^{s,a_k}<0$ forms an unstable equilibrium point of the ODE associated with the iterates $\pi_k^n(s,a_k)$. Hence, Algorithm III.1 will not converge to such policies. Consider a policy pair (π_D,π_A) such that $\Omega_{k,\pi_{-k}}^{s,a_k}\geq 0$. Note that, by Eqn. (3), such a policy pair satisfies $\Delta(\pi)\geq 0$. When $\Delta(\pi)>0$, Algorithm III.1 updates the policies of players in a descent direction of $\Delta(\pi)$ to achieve ARNE (i.e., $\Delta(\pi)=0$).

Let the gradient of $\Delta(\pi)$ w.r.t policies π_D and π_A be $\frac{\partial\Delta(\pi)}{\partial\pi}$, where $\pi=(\pi_D,\pi_A)$. Then for each $k\in\{D,A\}$, $s\in\mathbf{S}$, and $a_k\in\mathcal{A}_k(s)$, $\frac{\partial\Delta(\pi)}{\partial\pi_k(s,a_k)}=\sum_{\bar{k}\in\{D,A\}}\Omega_{\bar{k},\pi_{-k}}^{s,a_k}$ represents each component of $\frac{\partial\Delta(\pi)}{\partial\pi}$. The computation of $\frac{\partial\Delta(\pi)}{\partial\pi_k(s,a_k)}$ requires the values of $\mathbf{P}$ which is unknown in our game model. Therefore the iterate $\varepsilon_k^n(s,a_k)$ in line 11 of Algorithm III.1 estimates $\frac{\partial\Delta(\pi)}{\partial\pi_k(s,a_k)}$ using SA. Convergence of $-\varepsilon_k^n(s,a_k)$ to $\frac{\partial\Delta(\pi)}{\partial\pi_k(s,a_k)}$ is proved in Theorem III.9.

Additionally, in line 12 of Algorithm III.1, the map Γ projects the policies to probability simplex defined by condition (4c) in Proposition II.5. Here, $|\cdot|$ denotes the absolute value. The function $\mathrm{sgn}(\chi)$ denotes the continuous version of the standard sign function (e.g., $\mathrm{sgn}(\chi)=\tanh(c\chi)$ for any constant $c>1$). Lemma III.11 shows that the policy iterates in line 12 update in a valid descent direction

of $\Delta(\pi)$ and Theorem III.13 proves the convergence. Theorem III.14 then shows that the converged policies indeed form an ARNE.

The value function iterates in line 9 and the gradient estimate iterates in line 11 of Algorithm III.1 update in a same faster time scale δ_v^n and δ_ε^n, respectively. Policy iterates in line 12 update in a slower time scale δ_π^n. Average reward payoff iterates in line 10 update in an intermediate time scale δ_ρ^n. Hence the step-sizes of the proposed algorithm are chosen such that $\delta_v^n = \delta_\varepsilon^n \gg \delta_\rho^n \gg \delta_\pi^n$. The step-sizes must also satisfy the conditions in Assumption II.6. It is necessary to satisfy these conditions to ensure the convergence. Similar conditions on step sizes have been imposed to prove the convergence of the multi-time scale RL algorithms presented in [18] and [25]. Due to time scale separation, iterations in relatively faster time scales see iterations in relatively slower times scales as quasi-static while the latter sees former as nearly equilibrated (Chapter 6 of [28]).

Remark III.1. *Algorithm III.1 must be trained offline due to the information exchange that is required at line 11 of the algorithm. Here, players are required to exchange the information about their respective temporal difference error estimates, $\tilde{\phi}_k(\rho_k^n, v_k^n) = r_k(s,d,a,s') - \rho_k^n + v_k^n(s') - v_k^n(s)$, as the iterates on each player's gradient estimation includes the term $\sum_{k\in\{D,A\}} \tilde{\phi}_k(\rho_k^n, v_k^n)$. Since the algorithm is trained offline and the policies found at the end of the training only depend on their respective actions, players do not require any information exchange on their respective actions when they execute their learned policies in real-time.*

Proposition III.2. *Let K, A, and S denote the number of players, the maximum cardinality of the action space of any player, and the cardinality of state space. Algorithm III.1 has per iteration computation and memory complexity of $O(KA)$ and $O(KSA)$.*

Proof. RL-ARNE requires $O(K)$ multiplications for the value and average reward updates (lines 9-10 of Algorithm III.1), and $O(KA)$ multiplications for the gradient and policy updates (lines 11-12). Thus, per iteration computation complexity is $O(KA)$. The memory required for the RL-ARNE is $O(KS)$ for the value updates, $O(K)$ for the average reward updates, and $O(KSA)$ for the gradient and policy updates. Thus, memory complexity is $O(KSA)$. □

B. Convergence Proof of the RL-ARNE Algorithm

We rewrite iterations in lines 9-10 as Eqns. (9)-(10) to show the convergence of value and average reward payoff iterates.

$$v_k^{n+1}(s) = v_k^n(s) + \delta_v^n[F(v_k^n,\rho_k^n)(s) - v_k^n(s) + w_v^n]. \quad (9)$$

$$\rho_k^{n+1} = \rho_k^n + \delta_\rho^n[G(\rho_k^n) - \rho_k^n + w_\rho^n]. \quad (10)$$

For brevity we use $\pi(s,d,a) = \pi_D(s,d)\pi_A(s,a)$ and π to denote (π_D,π_A). Let $\mathbf{P}(s'|s,\pi) = \sum_{d\in\mathcal{A}_D(s)}\sum_{a\in\mathcal{A}_A(s)} \pi(s,d,a)\mathbf{P}(s'|s,d,a)$. Two function maps $F(v_k^n)(s)$ and $G(\rho_k^n)$ are defined as

$$F(v_k^n,\rho_k^n)(s) = \sum_{s'\in\mathbf{S}} \mathbf{P}(s'|s,\pi)[r_k(s,d,a,s') - \rho_k^n + v_k^n(s')], \quad (11)$$

$$G(\rho_k^n) = \sum_{s'\in\mathbf{S}} \mathbf{P}(s'|s,\pi)\Big[\frac{n\rho_k^n + r_k(s,d,a,s')}{n+1}\Big]. \quad (12)$$

The zero mean noise parameters w_v^n and w_ρ^n are defined as

$$w_v^n = r_k(s,d,a,s') - \rho_k^n + v_k^n(s') - F(v_k^n,\rho_k^n)(s), \quad (13)$$

$$w_\rho^n = \frac{n\rho_k^n + r_k(s,d,a,s')}{n+1} - G(\rho_k^n). \quad (14)$$

Let $v_k = [v_k(s)]_{s\in\mathbf{S}}$. Then the ODE associated with the iterates given in Eqn. (9) corresponding to all $s\in\mathbf{S}$ and the ODE associated with the iterate in Eqn. (10) are as follows.

$$\dot{v}_k = f(v_k,\rho_k) \text{ and } \dot{\rho}_k = g(\rho_k), \quad (15)$$

where $f:\mathscr{R}^{|\mathbf{S}|}\to\mathscr{R}^{|\mathbf{S}|}$ is such that $f(v_k,\rho_k) = F(v_k,\rho_k) - v_k$, where $F(v_k,\rho_k) = [F(v_k,\rho_k)(s)]_{s\in\mathbf{S}}$ and $g:\mathscr{R}\to\mathscr{R}$ is defined as $g(\rho_k) = G(\rho_k) - \rho_k$. We note that, in Algorithm III.1, value function iterates $(v_k^n(s))$ runs in a relatively faster time scale compared to the average reward iterates (ρ_k^n). As a consequence, $v_k^n(s)$ iterates see ρ_k^n as quasi-static. Hence, for brevity, in the proofs of Lemma III.3, Lemma III.6, and Theorem III.8 we represent $f(v_k,\rho_k)$ and $F(v_k^n,\rho_k^n)(s)$ as $f(v_k)$ and $F(v_k^n)(s)$, respectively.

A set of lemmas that are used to prove the convergence of the iterates in lines 9 and 10 of Algorithm III.1 are given below. Lemma III.3 presents a property of the ODEs in Eqn. (15).

Lemma III.3. *Consider the ODEs $\dot{v}_k = f(v_k,\rho_k)$ and $\dot{\rho}_k = g(\rho_k)$. Then the functions $f(v_k,\rho_k)$ and $g(\rho_k)$ are Lipschitz.*

Proof. First we show $f(v_k)$ is Lipschitz. Consider two distinct value vectors v_k and $\bar{v}_k$. Then,

$$\begin{aligned}\|f(v_k) - f(\bar{v}_k)\|_1 &= \|[F(v_k) - F(\bar{v}_k)] - [v_k - \bar{v}_k]\|_1 \\ &\le \|F(v_k) - F(\bar{v}_k)\|_1 + \|v_k - \bar{v}_k\|_1 \\ &= \sum_{s\in\mathbf{S}} \Big|F(v_k)(s) - F(\bar{v}_k)(s)\Big| + \|v_k - \bar{v}_k\|_1. \end{aligned} \quad (16)$$

$$\begin{aligned}\sum_{s\in\mathbf{S}} \Big|F(v_k)(s) - F(\bar{v}_k)(s)\Big| &= \sum_{s\in\mathbf{S}} \Big|\sum_{s'\in\mathbf{S}} \mathbf{P}(s'|s,\pi)[v_k(s') - \bar{v}_k(s')]\Big| \\ &\le \sum_{s\in\mathbf{S}}\sum_{s'\in\mathbf{S}} \mathbf{P}(s'|s,\pi)\big|v_k(s') - \bar{v}_k(s')\big| \\ \le \sum_{s\in\mathbf{S}}\sum_{s'\in\mathbf{S}} \big|v_k(s') - \bar{v}_k(s')\big| &= \sum_{s\in\mathbf{S}} \|v_k - \bar{v}_k\|_1 = |\mathbf{S}|\,\|v_k - \bar{v}_k\|_1. \end{aligned}$$

The inequalities above follow from the triangle inequality and observing the fact that $\max\{\mathbf{P}(s'|s,\pi)\} = 1$. Then from Eqn. (16),

$$\|f(v_k) - f(\bar{v}_k)\|_1 \le (|\mathbf{S}|+1)\,\|v_k - \bar{v}_k\|_1.$$

Hence $f(v_k)$ is Lipschitz. Next we prove $g(\rho_k)$ is Lipschitz. Let ρ_k and $\bar{\rho}_k$ be two distinct average payoff values. Then,

$$|g(\rho_k) - g(\bar{\rho}_k)| = \Big|\frac{n}{n+1}[\rho_k - \bar{\rho}_k] - [\rho_k - \bar{\rho}_k]\Big| = |\rho_k - \bar{\rho}_k|.$$

Therefore $g(\rho_k)$ is Lipschitz. □

Lemma III.6 shows the map $F(v_k^n) = [F(v_k^n)(s)]_{s\in\mathbf{S}}$ is a pseudo-contraction w.r.t some weighted sup-norm. The definitions of weighted sup-norm and pseudo-contraction are given below.

Definition III.4. *Let $||b||_\varepsilon$ denote the weighted sup-norm of a vector $b\in\mathscr{R}^{m_b}$ w.r.t vector $\varepsilon\in\mathscr{R}^{m_b}$. Then, $||b||_\varepsilon = \max_{q=1,\ldots,n}\frac{|b(q)|}{\varepsilon(q)}$, where $|b(q)|$ represent the absolute value of the q^{th} entry of vector b.*

Definition III.5 (Pseudo contraction)**.** *Let $c,\bar{c}\in\mathscr{R}^{m_c}$. Then a function $\phi:\mathscr{R}^{m_c}\to\mathscr{R}^{m_c}$ is said to be a pseudo contraction w.r.t the vector $\gamma\in\mathscr{R}^{m_c}$ if and only if,*

$$\|\phi(c) - \phi(\bar{c})\|_\gamma \le \eta\,\|c - \bar{c}\|_\gamma, \text{ where } 0\le\eta<1.$$

Lemma III.6. *Consider $F(v_k^n,\rho_k^n)(s)$ defined in Eqn. (11). Then the function map $F(v_k^n,\rho_k^n) = [F(v_k^n,\rho_k^n)(s)]_{s\in\mathbf{S}}$ is a pseudo-contraction w.r.t some weighted sup-norm.*

Proof. Consider two distinct value functions v_k^n and $\bar{v}_k^n$. Then,

$$\begin{aligned}\|F(v_k^n)(s) - F(\bar{v}_k^n)(s)\|_1 &= \|\sum_{s'\in\mathbf{S}} \mathbf{P}(s'|s,\pi)[v_k^n(s') - \bar{v}_k^n(s')]\|_1 \\ = \|\sum_{s'\in\mathbf{S}}\sum_{d\in\mathcal{A}_D(s),\ a\in\mathcal{A}_A(s)} &\pi(s,d,a)\mathbf{P}(s'|s,d,a)[v_k^n(s') - \bar{v}_k^n(s')]\|_1 \\ \le \sum_{d\in\mathcal{A}_D(s),\ a\in\mathcal{A}_A(s)} &\pi(s,d,a)\sum_{s'\in\mathbf{S}} \mathbf{P}(s'|s,d,a)\,\|v_k^n(s') - \bar{v}_k^n(s')\|_1. \end{aligned} \quad (17)$$

Eqn. (17) follows from triangle inequality. To find an upper bound for the term $\mathbf{P}(s'|s,d,a)$ in Eqn. (17), we construct a Stochastic Shortest Path Problem (SSPP) with the same state space and transition probability structure as in the game, and a player whose action set is given by $\mathcal{A}_D \times \mathcal{A}_A$. Further set the rewards corresponding to all the state transition in SSPP to be -1. Then by Proposition 2.2 in [29], $\sum_{s'\in\mathbf{S}} \mathbf{P}(s'|s,d,a)\varepsilon(s') \leq \eta\varepsilon(s)$ holds for all $s \in \mathbf{S}$ and $(d,a) \in \mathcal{A}_D(s) \times \mathcal{A}_A(s)$, where $\varepsilon \in [0,1]^{|\mathbf{S}|}$ and $0 \leq \eta < 1$. Rewrite Eqn. (17) as $|F(v_k^n)(s) - F(\bar{v}_k^n)(s)|$

$$\leq \sum_{d\in\mathcal{A}_D(s),\ a\in\mathcal{A}_A(s)} \pi(s,d,a) \sum_{s'\in\mathbf{S}} \mathbf{P}(s'|s,d,a)\varepsilon(s') \frac{|v_k^n(s') - \bar{v}_k^n(s')|}{\varepsilon(s')}$$
$$\leq \sum_{d\in\mathcal{A}_D(s),\ a\in\mathcal{A}_A(s)} \pi(s,d,a) \sum_{s'\in\mathbf{S}} \mathbf{P}(s'|s,d,a)\varepsilon(s') \parallel v_k^n - \bar{v}_k^n \parallel_\varepsilon$$
$$\leq \sum_{d\in\mathcal{A}_D(s),\ a\in\mathcal{A}_A(s)} \pi(s,d,a)\eta\varepsilon(s) \parallel v_k^n - \bar{v}_k^n \parallel_\varepsilon = \eta\varepsilon(s) \parallel v_k^n - \bar{v}_k^n \parallel_\varepsilon.$$

$$\parallel F(v_k^n) - F(\bar{v}_k^n) \parallel_\varepsilon \leq \eta \parallel v_k^n - \bar{v}_k^n \parallel_\varepsilon. \qquad \square$$

Below we prove boundedness of Algorithm III.1 iterates.

Lemma III.7. *Consider the RL-ARNE algorithm presented in Algorithm III.1. Then, the iterates $v_k^n(s)$ and ρ_k^n, for $s \in \mathbf{S}$ and $k \in \{D,A\}$, in Eqn.*(9) *and Eqn.*(10) *are bounded.*

Proof. Recall Eqns. (9) and (10). We know $\delta_\rho^n \ll \delta_v^n$. In order to prove the result we first show the errors introduced in the slow iterates $v_k^n(s)$ by the transient errors of the fast iterates ρ_k^n approach to zero as $n \to \infty$. For a positive integer Δ, with slight abuse of notation, we let $v_k^{n+\Delta}(s)$ to denote the value function at the iterate n computed by replacing ρ_k^n at Eqn.(9) with $\rho_k^{n+\Delta}$. As a consequence, an error

$$Err(v;\rho) = v_k^{n+\Delta}(s) - v_k^n(s) = \delta_v^n(\rho_k^n - \rho_k^{n+\Delta}), \tag{18}$$

is introduced in $v_k^n(s)$ iterates, where the term $\rho_k^n - \rho_k^{n+\Delta}$ captures the transient errors of the fast iterate ρ_k^n. It has been shown that $\rho_k^n - \rho_k^{n+\Delta} = O(\delta_\rho^n)$ in [25], [30]. Then, from Eqn. (18) we get $Err(v;\rho) = O(\delta_v^n\delta_\rho^n)$. This proves that $Err(v;\rho) \to 0$ when $n \to \infty$ as $\delta_v^n, \delta_\rho^n \ll 1$ and $\delta_\rho^n \to 0$ at a faster rate compared to δ_v^n due to $\delta_\rho^n \ll \delta_v^n$, when $n \to \infty$. Similarly, since $\delta_\pi^n \ll \delta_v^n$ and $\delta_\pi^n \ll \delta_\rho^n$, we can show $Err(v;\pi)$, $Err(\rho;\pi) \to 0$ as $n \to \infty$. Therefore, the error introduced in the slow iterates due to the transient errors of the fast iterates are asymptotically bounded.

Lemma III.6 proved that $F(v_k^n)$ is a pseudo-contraction w.r.t some weighted sup-norm. By choosing step-size, δ_v^n to satisfy Assumption II.6 and observing that the noise parameter, w_v^n is zero mean with bounded variance, all the conditions in Theorem 1 in [31] hold for the game. Hence, by Theorem 1 in [31], the iterates $v_k^n(s)$ in Eqn. (9) are bounded for all $s \in \mathbf{S}$.

Finally, we show the boundedness of the ρ_k^n iterates. From Proposition II.3, for a fixed policy pair (π_D,π_A) and $n >> 0$, the average reward payoff values ρ_k^n depend only on the rewards due to the state transitions that occur within the recurrent classes of induced MC. Recall that under Assumption II.2 the induced Markov chain, $\mathbf{P}(\pi_D,\pi_A)$, contains only a single recurrent class. Let $\mathbf{S}_1$ be the set of states in the recurrent class of $\mathbf{P}(\pi_D,\pi_A)$. Then there exists a unique stationary distribution p for $\mathbf{P}(\pi_D,\pi_A)$ restricted to states in $\mathbf{S}_1$. Thus for $n >> 0$ and each $k \in \{D,A\}$,

$$\rho_k^n = \sum_{s\in\mathbf{S}_1} p(s) r_k(s,\pi), \tag{19}$$

where $p(s)$ is the probability of being at state $s \in \mathbf{S}_1$ and $r_k(s,\pi) = \sum_{d\in\mathcal{A}_D(s),\ a\in\mathcal{A}_A(s)} \pi(s,d)\pi(s,a) \sum_{s'\in\mathbf{S}} \mathbf{P}(s'|s,d,a) r(s,d,a,s')$ is the expected reward at the state $s \in \mathbf{S}_1$ for player $k \in \{D,A\}$. Since $\mathbf{S_1}$ has finite cardinality and the rewards r_ks are finite for the game, ρ_k^n converges to a globally asymptotically stable critical point given in Eqn. (19) and ρ_k^n iterates are bounded. $\square$

Theorem III.8 proves the convergence of the iterates $v_k^n(s)$ and ρ_k^n.

Theorem III.8. *Consider the RL-ARNE algorithm presented in Algorithm III.1. Then the iterates $v_k^n(s)$, for all $s \in \mathbf{S}$, and ρ_k^n for $k \in \{D,A\}$ in Eqn.* (9) *and Eqn.* (10) *converge.*

Proof. By Proposition II.8, an SA-based algorithm converges under the conditions (I)-(VI). Lemma III.3 and Lemma III.7 showed that conditions (I) and (II) in Proposition II.8 are satisfied, respectively.

To show that condition (III) is satisfied, we first show that there exists a Lipschitz function $\psi_k(\rho_k)$ that characterizes the critical points of $\dot{v}_k = f(v_k,\rho_k)$ in Eqn. (15). Note that v_k is a critical point of $\dot{v}_k = f(v_k,\rho_k)$ if and only if $v_k = F(v_k,\rho_k)$. Let $|\mathbf{S}|$ be the cardinality of the state space associated with the game. Let $\mathbf{1}_{|\mathbf{S}|\times 1}$ and $\mathbf{I}_{|\mathbf{S}|\times|\mathbf{S}|}$ denote all ones vector with length $|\mathbf{S}|$ and $|\mathbf{S}| \times |\mathbf{S}|$ identity matrix, respectively. Then using Eqn. (11), we get $v_k = \bar{r} - \rho_k \mathbf{1}_{|\mathbf{S}|\times 1} + \mathbf{P}(\pi_D,\pi_A)v_k$, which can be rewritten as

$$\big[\mathbf{I}_{|\mathbf{S}|\times|\mathbf{S}|} - \mathbf{P}(\pi_D,\pi_A)\big] v_k = \bar{r} - \rho_k \mathbf{1}_{|\mathbf{S}|\times 1}, \tag{20}$$

where $\bar{r}$ is a length $|\mathbf{S}|$ vector whose entries are given by $\sum_{s'\in\mathbf{S}} \mathbf{P}(s'|s,\pi) r_k(s,d,a,s')$. The set of linear equations defined in Eqn. (20) has infinite number of solutions under Assumption II.2 [7], [29]. Let $\mathbf{J} = \mathbf{I}_{|\mathbf{S}|\times|\mathbf{S}|} - \mathbf{P}(\pi_D,\pi_A)$. Then for an arbitrary vector ω, we define $\psi_k(\rho_k) = \mathbf{J}^+\big[\bar{r} - \rho_k \mathbf{1}_{|\mathbf{S}|\times 1}\big] + \big[\mathbf{I}_{|\mathbf{S}|\times|\mathbf{S}|} - \mathbf{J}^+\mathbf{J}\big]\omega$, where $\mathbf{J}^+$ denotes the generalized inverse [32] of $\mathbf{J}$. Consider two distinct ρ_k and $\bar{\rho}_k$ with a fixed ω.

$$\begin{aligned} \parallel \psi_k(\rho_k) - \psi_k(\bar{\rho}_k) \parallel_1 &= \parallel \mathbf{J}^+(\bar{\rho}_k - \rho_k)\mathbf{1}_{|\mathbf{S}|\times 1} \parallel_1 \\ \leq \parallel \mathbf{J}^+ \parallel_1 \parallel (\rho_k - \bar{\rho}_k)\mathbf{1}_{|\mathbf{S}|\times 1} \parallel_1 &= |\mathbf{S}| \parallel \mathbf{J}^+ \parallel_1 \parallel \rho_k - \bar{\rho}_k \parallel_1. \end{aligned}$$

Hence, $\psi_k(\rho_k)$ is Lipschitz. Next we show $F(v_k^n)(s)$ is a non-expansive map to prove the convergence of the $v_k^n(s)$ iterates. Consider two distinct v_k^n and $\bar{v}_k^n$. Since $P(s'|s,\pi_D,\pi_A) \leq 1$, from Eqn. (17), $\parallel F(v_k^n)(s) - F(\bar{v}_k^n)(s) \parallel \leq \parallel v_k^n(s') - \bar{v}_k^n(s') \parallel$. Thus $F(v_k^n)(s)$ is a non-expansive map and hence from Theorem 2.2 in [33] iterates $v_k^n(s)$, for all $s \in \mathbf{S}$ and $k \in \{D,A\}$, converge to an asymptotically stable critical point given by $\psi_k(\rho_k)$. Hence, condition (III) is verified.

Lemma III.7, showed that ρ_k^n, for $k \in \{D,A\}$, converge to a globally asymptotically stable critical point which implies that condition (IV) is satisfied. From Eqns. (13) and (14), the noise measures have zero mean. The variance of these noise measures are bounded by the fineness of the rewards in the game and the boundedness of the iterates $v_k^n(s)$ and ρ_k^n. Thus condition (V) is satisfied. Finally, the choice of step-sizes satisfies condition (VI). Therefore the results follows by Proposition II.8. $\square$

Next theorem proves the convergence of gradient estimates.

Theorem III.9. *Consider $\Omega_{k,\pi_{-k}}^{s,a_k}$ and $\Delta(\pi)$ given in Eqns.* (2) *and* (3)*, respectively. Then gradient estimation iterate, $\varepsilon_k^n(s,a_k)$ in line 11 corresponding to any $k \in \{D,A\}$, $s \in \mathbf{S}$, and $a_k \in \mathcal{A}_k(s)$, converge to $-\frac{\partial\Delta(\pi)}{\partial\pi_k(s,a_k)} = -\sum_{\bar{k}\in\{A,D\}} \Omega_{\bar{k},\pi_{-k}}^{s,a_k}$.*

Proof. Rewrite gradient estimation in line 11 as follows.

$$\varepsilon_k^{n+1}(s,a_k) = \varepsilon_k^n(s,a_k) + \delta_\varepsilon^n\big[-\sum_{\bar{k}\in\{D,A\}} \Omega_{\bar{k},\pi_{-k}}^{s,a_k} - \varepsilon_k^n(s,a_k) + w_\varepsilon^n\big], \tag{21}$$

where $w_\varepsilon^n = \sum_{k\in\{D,A\}} \bar{\Omega}_k^s + \sum_{\bar{k}\in\{D,A\}} \Omega_{\bar{k},\pi_{-k}}^{s,a_k}$, and $\bar{\Omega}_k^s = r_k(s,d,a,s') - \rho_k^n + v_k^n(s') - v_k^n(s)$. Since $\mathbb{E}(w_\varepsilon^n) = 0$, the ODE associated with Eqn. (21) is given by, $\dot{\varepsilon}_k(s,a_k) = -\sum_{\bar{k}\in\{D,A\}} \Omega_{\bar{k},\pi_{-k}}^{s,a_k} - \varepsilon_k(s,a_k)$.

We use Proposition II.7 to prove the convergence of gradient estimation iterates, $\varepsilon_k^n(s,a_k)$. Step-size δ_ε^n is chosen such that condition 1) in Proposition II.7 is satisfied. Validity of condition 2) can be shown as follows.

$$\mathbb{E}\left(\lim_{n\to\infty}\left(\sup_{\bar{n}>n}\left|\sum_{l=n}^{\bar{n}}\delta_\varepsilon^l w_\varepsilon^l\right|^2\right)\right)\leq 4\lim_{n\to\infty}\sum_{l=n}^{\infty}(\delta_\varepsilon^l)^2\mathbb{E}(|w_\varepsilon^l|^2)=0. \quad (22)$$

Inequality in Eqn. (22) follows by Doob inequality [27]. Equality in Eqn. (22) follows by choosing δ_ε^n to satisfy Assumption II.6 and observing $\mathbb{E}(|w_\varepsilon^l|^2)<\infty$ as r_k, v_k^n, and ρ_k^n are bounded in the game. Comparing Eqn. (21) with Eqn. (6), $\kappa=0$ in Eqn. (21). Therefore, from Proposition II.7, as $n\to\infty$, $\varepsilon_k^n(s,a_k)\to-\sum_{\bar{k}\in\{A,D\}}\Omega_{\bar{k},\pi_{-k}}^{s,a_k}=-\frac{\partial\Delta(\pi)}{\partial\pi_k(s,a_k)}$. This completes the proof showing the convergence of gradient estimation iterates $\varepsilon_k^n(s,a_k)$. □

Next, we prove the convergence of the policy iterates. In order to do so, we proceed in the following manner.

1) We rewrite the conditions in Prop II.5 that characterize ARNE of the game as a non-linear optimization problem (Problem III.10).
2) Then we show the policies are updated in a valid decent direction, $\sqrt{\pi_k^n(s,a_k)}\left|\Omega_{k,\pi_{-k}}^{s,a_k}\right|\text{sgn}\left(\frac{\partial\Delta(\pi^n)}{\partial\pi_k^n(s,a_k)}\right)$, w.r.t the objective function (TD error), $\Delta(\pi)$, of Problem III.10 (Lemma III.11).
3) Using steps 1) and 2), we characterize the stable and unstable equilibrium points associated with the ODE corresponding to the policy iterates in line 12 (Lemma III.12).
4) Invoking Prop II.7 we prove the convergence of policy iterates to stable equilibrium points found in step 3) (Theorem III.13).

Below we elaborate steps 1)-4). The non-linear program below characterizes an ARNE of the game (step 1).

Problem III.10. *The necessary and sufficient conditions given in Proposition II.5 that characterize the ARNE of the game can be reformulated as the following non-linear program using* $\Omega_{k,\pi_{-k}}^{s,a_k}$ *and* $\Delta(\pi)$ *introduced in Eqns.* (2) *and* (3).

$$\min_{v,\rho,\pi}\quad \Delta(\pi)\ s.t.\quad \Omega_{k,\pi_{-k}}^{s,a_k}\geq 0;\ \sum_{a_k\in\mathcal{A}_k(s)}\pi_k(s,a_k)=1;\ \pi_k(s,a_k)\geq 0,$$

where $v=(v_D,v_A)$, $v_k=[v_k(s)]_{s\in\mathbf{S}}$, $\rho=(\rho_D,\rho_A)$, $\pi=(\pi_D,\pi_A)$, $\pi_k=[\pi_k(s)]_{s\in\mathbf{S}}$, $\pi_k(s)=[\pi_k(s,a_k)]_{a_k\in\mathcal{A}_k(s)}$, *for* $k\in\{D,A\}$.

Lemma III.11 proves policy iterates are updated in a valid descent direction w.r.t the objective function, $\Delta(\pi)$ (step 2).

Lemma III.11. *Consider* $\Omega_{k,\pi_{-k}}^{s,a_k}$, $\Delta(\pi)$ *given in Eqs.* (2), (3), *respectively. For any* $k\in\{D,A\}$, $s\in\mathbf{S}$, *and* $a_k\in\mathcal{A}_k(s)$, *policy iterate,* $\pi_k^n(s,a_k)$, *in line 12 of Algorithm III.1 is updated in a valid descent direction,* $\sqrt{\pi_k^n(s,a_k)}\left|\Omega_{k,\pi_{-k}}^{s,a_k}\right|sgn\left(\frac{\partial\Delta(\pi^n)}{\partial\pi_k^n(s,a_k)}\right)$, *of* $\Delta(\pi)$ *when* $\Omega_{k,\pi_{-k}}^{s,a_k}\geq 0$ *and* $\Delta(\pi)>0$.

Proof. First we rewrite policy iteration in line 12 as follows.

$$\pi_k^{n+1}(s,a_k)=\Gamma\left(\pi_k^n(s,a_k)-\delta_\pi^n\left(\sqrt{\pi_k^n(s,a_k)}\left|\Omega_{k,\pi_{-k}}^{s,a_k}\right|\text{sgn}\left(\frac{\partial\Delta(\pi^n)}{\partial\pi_k^n(s,a_k)}\right)+w_\pi^n\right)\right), \quad (23)$$

where $w_\pi^n=\sqrt{\pi_k^n(s,a_k)}\left[\left|\bar{\Omega}_k^s\right|-\left|\Omega_{k,\pi_{-k}}^{s,a_k}\right|\right]\text{sgn}\left(\frac{\partial\Delta(\pi^n)}{\partial\pi_k^n(s,a_k)}\right)$, and $\bar{\Omega}_k^s=r_k(s,d,a,s')-\rho_k^n+v_k^n(s')-v_k^n(s)$. Policy iterate updates in the slowest time scale when compared to the other iterates. Thus, all the terms except $\Omega_{k,\pi_{-k}}^{s,a_k}$ in Eqn. (23) use the converged values of v_k, ρ_k, and $\frac{\partial\Delta(\pi^n)}{\partial\pi_k^n(s,a_k)}$ w.r.t policy $\pi^n=(\pi_D^n,\pi_A^n)$.

Consider a policy π_k^{n+1} whose entries are same as π_k^n except the entry $\pi_k^{n+1}(s,a_k)$ which is chosen as in Eqn. (23), for small $0<\delta_\pi^n<<1$. Let $\bar{\pi}=(\pi_k^{n+1},\pi_{-k}^n)$ and $\hat{\pi}=(\pi_k^n,\pi_{-k}^n)$. Also note that $\mathbb{E}(w_\pi^n)=0$. Thus ignoring the term w_π^n and using Taylor series expansion yields,

$$\Delta(\bar{\pi})=\Delta(\hat{\pi})+\delta_\pi^n\left(-\sqrt{\pi_k^n(s,a_k)}\left|\Omega_{k,\pi_{-k}}^{s,a_k}\right|\left|\frac{\partial\Delta(\hat{\pi})}{\partial\pi_k^n(s,a_k)}\right|\right)+o(\delta_\pi^n),$$

where $o(\delta_\pi^n)$ represents the higher order terms corresponding to δ_π^n. We ignore $o(\delta_\pi^n)$ in the second equality above since the choice of δ_π^n is small. Notice that the term $\delta_\pi^n\left(-\sqrt{\pi_k^n(s,a_k)}\left|\Omega_{k,\pi_{-k}}^{s,a_k}\right|\left|\frac{\partial\Delta(\hat{\pi})}{\partial\pi_k^n(s,a_k)}\right|\right)$ is negative. Since $\Delta(\pi)>0$ for any π, we get $\Delta(\bar{\pi})<\Delta(\hat{\pi})$. This proves policies are updated in a valid descent direction. □

Notice that the ODE associated with Eqn. (23) is,

$$\dot{\pi}_k(s,a_k)=\bar{\Gamma}\left(-\sqrt{\pi_k(s,a_k)}\left|\Omega_{k,\pi_{-k}}^{s,a_k}\right|\text{sgn}\left(\frac{\partial\Delta(\pi)}{\partial\pi_k(s,a_k)}\right)\right), \quad (24)$$

where $\bar{\Gamma}$ is the continuous version of the projection operator Γ which is defined analogous to the continuous projection operator in Eqn. (7). Let Π denotes the set of limit points associated with the system of ODEs in Eqn. (24). Let the feasible set of Problem III.10 be

$$H=\{\pi\in L|\Omega_{k,\pi_{-k}}^{s,a_k}\geq 0,\ \text{for all}\ a_k\in\mathcal{A}_k(s),\ s\in\mathbf{S},\ k\in\{D,A\}\}, \quad (25)$$

where the set $L=\{\pi|\sum_{a_k\in\mathcal{A}_k(s)}\pi_k(s,a_k)=1,\pi_k(s,a_k)\geq 0,\ \text{for all}\ a_k\in\mathcal{A}_k(s),\ s\in\mathbf{S}\}$. The set Π can be partitioned using the set H as $\Pi=\Pi_1\cup\Pi_2$, where $\Pi_1=\Pi\cap H$ and $\Pi_2=\Pi\setminus\Pi_1$. Using these notations and steps 1) and 2), we characterize the stable and unstable equilibrium points of the system of ODEs in Eqn. (24) in Lemma III.11 (step 3).

Lemma III.12. *The following statements are true for the set of equilibrium policies* $\pi^\star$ *of ODE in Eqn.* (24).

1) All $\pi^\star\in\Pi_1$ *form a set of stable equilibrium points.*
2) All $\pi^\star\in\Pi_2$ *form a set of unstable equilibrium points.*

Proof. First we show statement 1) holds. Since the set Π_1 is in the feasible set H of Problem III.10 defined in Eqn. (25), for any $\pi^\star\in\Pi_1$, there exists some $a_k\in\mathcal{A}_k(s)$, $s\in\mathbf{S}$ that satisfy $\Omega_{k,\pi_{-k}}^{s,a_k}\geq 0$. Let $B_\zeta(\pi^\star)=\{\pi\in L|\,\|\pi-\pi^\star\|<\zeta\}$. Then, for any $\pi\in B_\zeta(\pi^\star)\setminus\Pi_1$, there exists a $\zeta>0$ such that $\Omega_{k,\pi_{-k}}^{s,a_k}>0$ which yields $\frac{\partial\Delta(\pi)}{\partial\pi_k(s,a_k)}>0$. This implies $\text{sgn}\left(\frac{\partial\Delta(\pi)}{\partial\pi_k(s,a_k)}\right)>0$.

Hence, $\bar{\Gamma}\left(-\sqrt{\pi_k(s,a_k)}\left|\Omega_{k,\pi_{-k}}^{s,a_k}\right|\text{sgn}\left(\frac{\partial\Delta(\pi)}{\partial\pi_k(s,a_k)}\right)\right)<0$ for any $\pi\in B_\zeta(\pi^\star)\setminus\Pi_1$. This implies that $\pi_k(s,a_k)$ will decrease when moving away from $\pi^\star\in\Pi_1$. This proves $\pi^\star\in\Pi_1$ is an stable equilibrium point of the system of ODEs given in Eqn. (24).

To show statement 2) is true, we first note that for any $\pi^\star\in\Pi_2$, there exists some $a_k\in\mathcal{A}_k(s)$, $s\in\mathbf{S}$ such that $\Omega_{k,\pi_{-k}}^{s,a_k}<0$. Then, for any $\pi\in B_\zeta(\pi^\star)\setminus\Pi_2$, there exists a $\zeta>0$ such that $\Omega_{k,\pi_{-k}}^{s,a_k}<0$ which yields $\frac{\partial\Delta(\pi)}{\partial\pi_k(s,a_k)}<0$. This implies $\text{sgn}\left(\frac{\partial\Delta(\pi)}{\partial\pi_k(s,a_k)}\right)<0$. Therefore, $\bar{\Gamma}\left(-\sqrt{\pi_k(s,a_k)}\left|\Omega_{k,\pi_{-k}}^{s,a_k}\right|\text{sgn}\left(\frac{\partial\Delta(\pi)}{\partial\pi_k(s,a_k)}\right)\right)>0$ for any $\pi\in B_\zeta(\pi^\star)\setminus\Pi_2$. This implies that $\pi_k(s,a_k)$ will increase when moving away from $\pi^\star\in\Pi_2$. This proves $\pi^\star\in\Pi_2$ is an unstable equilibrium point of the system of ODEs in Eqn. (24) and completes the proof. □

Theorem III.13 gives the convergence of the policy iterates to the set of stable equilibrium points in step 3) (step 4).

Theorem III.13. *The policy iterates* $\pi_k^n(s,a_k)$ *for all* $a_k\in\mathcal{A}_k(s)$, $s\in\mathbf{S}$, *and* $k\in\{D,A\}$ *in Algorithm III.1 converge to a stable equilibrium point* $\pi^\star=(\pi_D^\star,\pi_A^\star)\in\Pi_1$.

Proof. Recall $w_\pi^n=\sqrt{\pi_k^n(s,a_k)}\left[\left|\bar{\Omega}_k^s\right|-\left|\Omega_{k,\pi_{-k}}^{s,a_k}\right|\right]\text{sgn}\left(\frac{\partial\Delta(\pi^n)}{\partial\pi_k^n(s,a_k)}\right)$. We invoke Proposition II.7 to prove the convergence of policy iterates,

$\pi_k^n(s,a_k)$. Step-size δ_π^n is chosen such that condition 1) in Proposition II.7 is satisfied. Validity of condition 2) can be shown as follows.

$$\mathbb{E}\left(\lim_{n\to\infty}\left(\sup_{\bar{n}>n}\left|\sum_{l=n}^{\bar{n}}\delta_\pi^l w_\pi^l\right|^2\right)\right) \le 4\lim_{n\to\infty}\sum_{l=n}^{\infty}(\delta_\pi^l)^2\mathbb{E}(|w_\pi^l|^2) = 0. \quad (26)$$

In Eqn. (26), inequality follows by Doob inequality [27] and equality follows by choosing δ_π^n to satisfy Assumption II.6 and observing $\mathbb{E}(|w_\pi^l|^2) < \infty$ as r_k, v_k^n, and ρ_k^n are bounded in the game. Comparing Eqs. (23) and (6), $\kappa = 0$. Therefore, from Proposition II.7, as $n \to \infty$, the policy iterates $\pi_k^n(s,a_k)$ for all $a_k \in \mathcal{A}_k(s)$, $s \in \mathbf{S}$, and $k \in \{D,\ A\}$ converge to a stable equilibrium point $\pi^\star \in \Pi_1$. □

Next theorem proves the convergence of $\pi_k^n(s,a_k)$ given in line 12 of Algorithm III.1, to an ARNE in the game.

Theorem III.14. *Consider $\Omega_{k,\pi_{-k}}^{s,a_k}$ and $\Delta(\pi)$ given in Eqns.* (2) *and* (3)*, respectively. A converged policy $(\pi_D^\star, \pi_A^\star)$ of Algorithm III.1 forms an ARNE in the game.*

Proof. In the following, we show any converged policy $\pi^\star = (\pi_D^\star, \pi_A^\star)$ returned by Algorithm III.1 will satisfy conditions (4a)-(4c) in Proposition II.5 and thus $\pi^\star$ forms an ARNE in the game.

Recall from Theorem III.13, the policy iterates $\pi_k^n(s,a_k)$ for all $a_k \in \mathcal{A}_k(s)$, $s \in \mathbf{S}$, and $k \in \{D,\ A\}$ converge to a stable equilibrium point $\pi^\star \in \Pi_1$. Also, recall Π denotes the set of limit points associated with the system of ODEs in Eqn. (24) and $L = \{\pi | \sum_{a_k\in\mathcal{A}_k(s)} \pi_k(s,a_k) = 1, \pi_k(s,a_k) \ge 0,\ \text{for all } a_k \in \mathcal{A}_k(s),\ s \in \mathbf{S}\}$. Then, from the definition of the set Π_1, any converged $\pi^\star$ will satisfy conditions (4a) and (4c), since $\pi^\star \in \Pi_1 = \Pi \cap H$ yields $\pi^\star \in H$, where $H = \{\pi \in L | \Omega_{k,\pi_{-k}}^{s,a_k} \ge 0,\ \text{for all } a_k \in \mathcal{A}_k(s),\ s \in \mathbf{S},\ k \in \{D,A\}\}$.

Then it suffices to show any $\pi^\star \in \Pi_1$ will yield $\sqrt{\pi_k(s,a_k)}\Omega_{k,\pi_{-k}}^{s,a_k} = 0$ since this proves condition (4b) in Proposition II.5. We show this by contradiction arguments. Note that $\bar{\Gamma}\left(-\sqrt{\pi_k(s,a_k)}|\Omega_{k,\pi_{-k}}^{s,a_k}|\text{sgn}\left(\frac{\partial\Delta(\pi)}{\partial\pi_k(s,a_k)}\right)\right) = 0$ as $\pi^\star$ forms a set of equilibrium polices associated with the system of ODEs in Eqn. (24). Suppose there exists a policy $0 < \pi_k(s,\bar{a}_k) \le 1$ for some $\bar{a}_k \in \mathcal{A}_k(s)$, $s \in \mathbf{S}$, and $k \in \{D,\ A\}$ such that $\sqrt{\pi_k(s,\bar{a}_k)}\Omega_{k,\pi_{-k}}^{s,\bar{a}_k} \ne 0$. Consider the following two cases.

Case I: $\pi_k(s,\bar{a}_k) = 1$ and $\Omega_{k,\pi_{-k}}^{s,\bar{a}_k} \ne 0$. Recall $F(v_k,\rho_k) = [F(v_k,\rho_k)(s)]_{s\in\mathbf{S}}$ and $F(v_k,\rho_k)(s) = \sum_{s'\in\mathbf{S}} \mathbf{P}(s'|s,\pi)[r_k(s,d,a,s') - \rho_k^n + v_k(s')]$. Then under Case I $\sum_{a_k\in\mathcal{A}_k(s)} \pi_k(s,a_k)\Omega_{k,\pi_{-k}}^{s,a_k} = \pi_k(s,\bar{a}_k)\Omega_{k,\pi_{-k}}^{s,\bar{a}_k} = 0$, where the first equality is due to $\pi_k(s,\bar{a}_k) = 1$ and the second equality is due to the convergence of the value iterates to their true values (i.e., as $n \to \infty$, $v_k \to F(v_k,\rho_k)$) which is proved in Theorem III.8. Further, as $\pi_k(s,\bar{a}_k) = 1$ this yields $\Omega_{k,\pi_{-k}}^{s,\bar{a}_k} = 0$, which contradicts the condition $\Omega_{k,\pi_{-k}}^{s,\bar{a}_k} \ne 0$ in Case I.

Case II: $0 < \pi_k(s,\bar{a}_k) < 1$ **and** $\Omega_{k,\pi_{-k}}^{s,\bar{a}_k} \ne 0$. Under this case we get $\bar{\Gamma}\left(-\sqrt{\pi_k(s,\bar{a}_k)}|\Omega_{k,\pi_{-k}}^{s,\bar{a}_k}|\text{sgn}\left(\frac{\partial\Delta(\pi)}{\partial\pi_k(s,\bar{a}_k)}\right)\right) = -\sqrt{\pi_k(s,\bar{a}_k)}|\Omega_{k,\pi_{-k}}^{s,\bar{a}_k}|\text{sgn}\left(\frac{\partial\Delta(\pi)}{\partial\pi_k(s,\bar{a}_k)}\right) \ne 0$, due to conditions given in the Case II and assuming $\text{sgn}(\cdot) \ne 0$. However this contradicts with our initial observation of $\bar{\Gamma}\left(-\sqrt{\pi_k(s,a_k)}|\Omega_{k,\pi_{-k}}^{s,a_k}|\text{sgn}\left(\frac{\partial\Delta(\pi)}{\partial\pi_k(s,a_k)}\right)\right) = 0$.

Therefore, by contradiction, there does not exist any policy $0 < \pi_k(s,\bar{a}_k) \le 1$ for some $\bar{a}_k \in \mathcal{A}_k(s)$, $s \in \mathbf{S}$, and $k \in \{D,\ A\}$ such that $\sqrt{\pi_k(s,\bar{a}_k)}\Omega_{k,\pi_{-k}}^{s,\bar{a}_k} \ne 0$. This proves condition (4b) in Proposition II.5 holds. Since conditions (4a)-(4c) in Proposition II.5 hold, a converged policy $(\pi_D^\star, \pi_A^\star)$ of Algorithm III.1 forms an ARNE in the game. □

Remark III.15. *RL-ARNE algorithm presented in Algorithm III.1 and the associated convergence proofs given in Section III-B extend to K-player, non-zero sum, average reward unichain stochastic games. Unichain property is a mild regularity assumption compared to other regularity conditions such as ergodicity or irreducibility [34].*

IV. Simulations

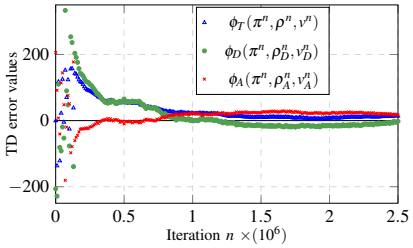
In this section we test Algorithm III.1 on a real-world attack dataset of a ransomware attack by an advanced persistant threat (APT). We collected the attack data using RAIN framework [24] and modeled the interaction between APT and a dynamic information flow tracking (DIFT)-based defense mechanism as a non-zero sum average reward stochastic game. The dataset consists of system logs with both benign and malicious information flows recorded in a *Linux* computer threatened by a ransomware attack. We first obtained a graphical representation of the dataset, referred as information flow graph (IFG). Immediate conversion of the system logs resulted in an IFG with 173 nodes and 2426 edges. We then performed a pruning technique to obtain a pruned IFG consisting of 18 nodes and 29 edges. Nodes in the IFG symbolize system locations, such as processes, files, and network sockets. These nodes form the state space for the DIFT-APT game. The action space for DIFT is focused on making decisions about inspecting or bypassing an incoming information flow at a specific system location. The action space for APT involves making strategic choices about moving to an adjacent system location linked via edges in the IFG or opting to cease the attack by discontinuing the information flow. For more details about the pruning technique and the DIFT-APT game model please refer to [35].

We used the following learning rates: $\delta_v^n = \delta_\varepsilon^n = 0.5$ if $n < 7000$ and $\delta_v^n = \delta_\varepsilon^n = \frac{1.6}{\kappa(s,n)}$, otherwise. $\delta_\rho = \delta_\pi^n = 1$, if $n < 7000$ and $\delta_\rho = \frac{1}{1+\tau(n)\log(\tau(n))}$, $\delta_\pi^n = \frac{1}{\tau(n)}$, otherwise. The learning rates remain constant until iteration 7000 and then start decaying. We observed that setting learning rates in this fashion helps the finite time convergence of the algorithm. Here, the term $\kappa(s,n)$ in δ_v^n and δ_ε^n denotes the total number of times a state $s \in \mathbf{S}$ is visited from 7000$^{\text{th}}$ iteration onwards in Algorithm III.1. Hence, δ_v^n and δ_ε^n depend on the iteration n and the state visited at iteration n. The term $\tau(n) = n - 6999$.

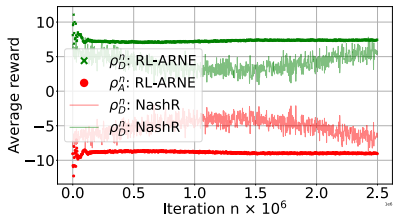
Let $\phi_T(\pi,\rho,v) = \phi_D(\pi,\rho_D,v_D) + \phi_A(\pi,\rho_A,v_A)$, where $\pi = (\pi_D,\pi_A)$, $\rho = (\rho_D,\rho_A)$, $v = (v_D,v_A)$. Here, $\phi_k(\pi,\rho_k,v_k)$, for $k \in \{D,A\}$, is given by $\phi_k(\pi,\rho_k,v_k) = \sum_{s\in\mathbf{S}}\sum_{a_k\in\mathcal{A}_k(s)}\Big(\rho_k + v_k(s) - r_k(s,a_k,\pi_{-k}) - \sum_{s'\in\mathbf{S}}\mathbf{P}(s'|s,a_k,\pi_{-k})v_k(s')\Big)\pi_k(s,a_k)$. We refer to $\phi_T(\pi,\rho,v)$, $\phi_D(\pi,\rho_D,v_D)$, and $\phi_A(\pi,\rho_A,v_A)$ as the total TD error, defender's TD error, and adversary's TD error, respectively. Then conditions in Proposition II.5 imply that a policy pair forms an ARNE if and only if $\phi_D(\pi,\rho_D,v_D) = \phi_A(\pi,\rho_A,v_A) = 0$. Consequently, at ARNE $\phi_T(\pi,\rho,v) = 0$.

Figure 2a plots ϕ_T, ϕ_D, ϕ_A corresponding to the policies given by Algorithm III.1 at iterations $n = 1,500,\ldots,2.5\times10^6$. The figure shows that ϕ_T, ϕ_D and ϕ_A converge close to 0 ($\approx 10^{-3}$) as n increases. This suggests that RL-ARNE algorithm is converging to an ARNE of the DIFT-APT game. Figure 2b illustrates the comparison of the convergence trends for average reward values of DIFT and APT (ρ_D^n and ρ_A^n) as implemented in Algorithm III.1 (RL-ARNE), against the NashR algorithm [12], [13]. The outcomes demonstrate that ρ_D^n and ρ_A^n achieved via RL-ARNE converge more rapidly and with lower variance compared to those attained through the NashR.

Figure 2c presents a comparison of the average rewards for players under converged policies from Algorithm III.1 (RL-ARNE policy) versus the average reward values obtained using the NashR policy and two distinct DIFT policies: i) the uniform policy and ii) the cut policy. In the uniform policy scenario, DIFT selects actions across all states following a uniform distribution. Under the cut policy, DIFT conducts security analyses at bottleneck states within the pruned IFG with a probability of one. It is important to note that the APT's policy remains consistent with the ARNE policy case for both the uniform and cut policy scenarios. The results indicate that DIFT secures a higher average reward utilizing the ARNE policy in comparison to



(a)



(b)

Policy\Player	DIFT	APT
RL-ARNE	*7.45*	*-9.06*
NashR	6.24	-7.26
Uniform	5.48	-6.61
Cut	4.44	-5.87

(c)

Figure 2: (a) Plots of total TD error, $\phi_T(\pi^n,\rho^n,v^n)$, DIFT's TD error $\phi_D(\pi^n,\rho_D^n,v_D^n)$, and APT's TD error $\phi_A(\pi^n,\rho_A^n,v_A^n)$ of Algorithm III.1 (RL-ARNE) for ransomware attack. (b) Plots comparing the average rewards of DIFT (ρ_D^n) and APT (ρ_A^n) obtained from RL-ARNE and NashR [12], [13]. (c) Comparison of the ρ_D and ρ_A obtained by the converged policies in RL-ARNE against ρ_D and ρ_A obtained by NashR and two other policies of DIFT. Uniform: Security analysis at every state under a uniform distribution. Cut: Security analysis at bottleneck states in the pruned IFG with probability one.

the NashR, uniform, and cut policies. Additionally, under the ARNE policy deployed by DIFT, APT receives a lower reward relative to the rewards it garners under the other mentioned policies.

V. Conclusion

In this paper we studied a competitive multi-agent stochastic decision making problem (nonzero-sum stochastic game) with incomplete and imperfect information structure. We proposed an RL-based algorithm, RL-ARNE, to learn an Average Reward Nash Equilibrium (ARNE) of the game. The proposed algorithm is a multiple-time scale stochastic approximation algorithm. We proved the convergence of RL-ARNE algorithm to an ARNE of the game. We evaluated the proposed RL-ARNE algorithm using a attaker-defender game grounded on a real-world ransomware attack dataset collected using RAIN framework. Our simulation results validated the convergence of the proposed algorithm to an ARNE of the attacker-defender game.

References

[1] L. S. Shapley, "Stochastic games," *Proceedings of the national academy of sciences*, vol. 39, no. 10, pp. 1095–1100, 1953.

[2] R. Amir, "Stochastic games in economics and related fields: An overview," *Stochastic Games and Applications*, pp. 455–470, 2003.

[3] D. Foster and P. Young, "Stochastic evolutionary game dynamics," *Theoretical Population Biology*, vol. 38, no. 2, p. 219, 1990.

[4] Q. Zhu and T. Başar, "Robust and resilient control design for cyber-physical systems with an application to power systems," *IEEE Decision and Control and European Control Conference (CDC-ECC)*, 2011.

[5] K.-w. Lye and J. M. Wing, "Game strategies in network security," *International Journal of Information Security*, vol. 4, no. 1-2, 2005.

[6] J. F. Nash, "Equilibrium points in n-person games," *Proceedings of the national academy of sciences*, vol. 36, no. 1, pp. 48–49, 1950.

[7] J. Filar and K. Vrieze, *Competitive Markov Decision Processes*. Springer Science & Business Media, 2012.

[8] M. Sobel, "Noncooperative stochastic games," *The Annals of Mathematical Statistics*, vol. 42, no. 6, pp. 1930–1935, 1971.

[9] J.-F. Mertens and T. Parthasarathy, "Equilibria for discounted stochastic games," *Stochastic Games and Applications*, pp. 131–172, 2003.

[10] T. Raghavan and J. A. Filar, "Algorithms for stochastic games—A survey," *Zeitschrift für Operations Research*, vol. 35, no. 6, 1991.

[11] K. Zhang, Z. Yang, and T. Başar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," *Handbook of Reinforcement Learning and Control*, pp. 321–384, 2021.

[12] J. Li, "Learning average reward irreducible stochastic games: Analysis and applications," *Ph.D. dissertation, Dept. Ind. Manage. Syst. Eng., Univ. South Florida, Tampa, FL, USA*, 2003.

[13] J. Li, K. Ramachandran, and T. K. Das, "A reinforcement learning (nash-R) algorithm for average reward irreducible stochastic games," *Journal of Machine Learning Research*, 2007.

[14] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou, "The complexity of computing a nash equilibrium," *SIAM Journal on Computing*, vol. 39, no. 1, pp. 195–259, 2009.

[15] X. Chen, X. Deng, and S.-H. Teng, "Settling the complexity of computing two-player nash equilibria," *Journal of the ACM*, vol. 56, 2009.

[16] J. Hu and M. P. Wellman, "Nash Q-learning for general-sum stochastic games," *Journal of Machine Learning Research*, vol. 4, 2003.

[17] M. L. Littman *et al.*, "Friend-or-foe Q-learning in general-sum games," in *ICML*, vol. 1, 2001, pp. 322–328.

[18] H. L. Prasad, L. A. Prashanth, and S. Bhatnagar, "Two-timescale algorithms for learning Nash equilibria in general-sum stochastic games," *International Conference on Autonomous Agents and Multiagent Systems*, pp. 1371–1379, 2015.

[19] J. Pérolat, F. Strub, B. Piot, and O. Pietquin, "Learning nash equilibrium for general-sum Markov games from batch data," in *Artificial Intelligence and Statistics*. PMLR, 2017, pp. 232–241.

[20] G. Arslan and S. Yüksel, "Decentralized Q-learning for stochastic teams and games," *IEEE Transactions on Automatic Control*, 2016.

[21] M. Sayin, K. Zhang, D. Leslie, T. Basar, and A. Ozdaglar, "Decentralized Q-learning in zero-sum Markov games," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[22] C. Martin and T. Sandholm, "Efficient exploration of zero-sum stochastic games," *arXiv preprint arXiv:2002.10524*, 2020.

[23] A. Ozdaglar, M. O. Sayin, and K. Zhang, "Independent learning in stochastic games," *arXiv preprint arXiv:2111.11743*, 2021.

[24] Y. Ji, S. Lee, E. Downing, W. Wang, M. Fazzini, T. Kim, A. Orso, and W. Lee, "RAIN: Refinable attack investigation with on-demand inter-process information flow tracking," *ACM SIGSAC Conference on Computer and Communications Security*, pp. 377–390, 2017.

[25] A. Gosavi, "Reinforcement learning for long-run average cost," *European Journal of Operational Research*, vol. 155, no. 3, 2004.

[26] H. J. Kushner and D. S. Clark, *Stochastic approximation methods for constrained and unconstrained systems*. Springer Science & Business Media, 2012, vol. 26.

[27] M. Metivier and P. Priouret, "Applications of a Kushner and Clark lemma to general classes of stochastic algorithms," *IEEE Transactions on Information Theory*, vol. 30, no. 2, pp. 140–151, 1984.

[28] V. S. Borkar, *Stochastic Approximation: A Dynamical Systems Viewpoint*. Springer, 2009, vol. 48.

[29] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Athena Scientific, 1996.

[30] M. Kaledin, E. Moulines, A. Naumov, V. Tadic, and H.-T. Wai, "Finite time analysis of linear two-timescale stochastic approximation with Markovian noise," in *Conference on Learning Theory*. PMLR, 2020.

[31] J. N. Tsitsiklis, "Asynchronous stochastic approximation and Q-Learning," *Machine learning*, vol. 16, no. 3, pp. 185–202, 1994.

[32] M. James, "The generalised inverse," *The Mathematical Gazette*, vol. 62, no. 420, pp. 109–114, 1978.

[33] K. Soumyanath and V. S. Borkar, "An analog scheme for fixed-point computation-part ii: Applications," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 46, no. 4, 1999.

[34] S. Bhatnagar, R. S. Sutton, M. Ghavamzadeh, and M. Lee, "Natural actor–critic algorithms," *Automatica*, vol. 45, no. 11, 2009.

[35] D. Sahabandu, S. Moothedath, J. Allen, L. Bushnell, W. Lee, and R. Poovendran, "A reinforcement learning approach for dynamic information flow tracking games for detecting advanced persistent threats," *arXiv preprint arXiv:2007.00076*, 2020.