Approximate Minimax Q Learning for Adversarial Markov Games with Unbounded State Spaces

Anonymous Authors[1]

Abstract

Learning systems deployed in adversarial or uncertain environments often require theoretical guarantees of robustness and stability. In reinforcement learning, such interactions can be modeled as two-player zero-sum Markov games, where an agent and an adversary alternately influence the system dynamics. We consider such a game over a dynamical system with unbounded state spaces (e.g., a data flow network). The attacker can alter the transition probabilities of the system at a non-zero attacking cost. The defender can reject such attacks at a non-zero defending cost. We show the existence of Markov perfect equilibrium for the game. We develop a minimax Q learning algorithm with linear function approximation that learns equilibrium strategies. We prove the convergence of the algorithm under rather mild assumptions on the system dynamics and on the basis functions. We also apply the learning algorithm and the convergence criterion to two representative control tasks in data flow management, viz. routing and polling. We demonstrate empirically that our algorithm converges faster than typical deep neural network-based approximation with an insignificant optimality gap.

1. Introduction

2. Model and Algorithm

In this section, we formulate the Markov security game, develop the function approximation scheme, and present the approximate minimax-Q learning algorithm.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

2.1. Security Game

We characterize the security problem as a two-player zero-sum game between a defender and an attacker.

Since we consider a version of off-policy learning algorithm, we differentiate the notations for the behavior policy and for the target policy. We use $\alpha(a|x) : \{0,1\} \times \mathbb{Z}^m_{\geq 0} \to [0,1]$ (resp. $\beta(b|x) : \{0,1\} \times \mathbb{Z}^m_{\geq 0} \to [0,1]$) to denote the probabilistic behavior policy for the attacker (resp. defender). We use $\pi(a|x) : \{0,1\} \times \mathbb{Z}^m_{\geq 0} \to [0,1]$ (resp. $\sigma(b|x) : \{0,1\} \times \mathbb{Z}^m_{\geq 0} \to [0,1]$) to denote the probabilistic target policy for the attacker (resp. defender). The transition rate of the state under the policy pair (α, β) is given by $q_{\alpha,\beta} : \mathbb{Z}^m_{\geq 0} \times \mathbb{Z}^m_{\geq 0} \to \mathbb{R}_{\geq 0}$.

The action space for the attacker is $\{0,1\}$, where $a(t) = 0$ (resp. $a(t) = 1$) means "not to attack" (resp. "to attack") at time t. The action space for the defender is also $\{0,1\}$, where $b(t) = 0$ (resp. $b(t) = 1$) means "not to defend" (resp. "to defend") at time t. The attacking cost is $c_1 > 0$ per unit time, while defending cost is $c_2 > 0$ per unit time. These costs account for the resources that attacking/defending actions have to consume. The instantaneous reward (resp. cost) for the attacker (resp. defender) at time t is defined as

$$\rho\Big(x(t), a(t), b(t)\Big) := f(x(t)) - c_1 a(t) + c_2 b(t), \quad (1)$$

where $f(x(t))$ is state-related costs. The action-induced costs are included in the reward/cost function, since both players may be interested in maximizing the opponent's costs. Note that the above reward/cost function assumes that both the traffic state and the opponent's action are observable to both players. This assumption is technologically reasonable in many scenarios.

Since the system state is countable and changes only at discrete epochs, we can reformulate the Markov security game in discrete time (DT). Note that a DT formulation also facilitates the design of learning algorithm.

Specifically, let t_k be the kth transition epoch of the continuous-tim (CT) process $\{x(t); t \geq 0\}$. With a slight abuse of notation, let

$$x_k = x(t_k),\ a_k = a(t_k),\ b_k = b(t_k), \quad k = 0, 1, ...$$

Thus, the transition probabilities $p(x'|x,a,b)$ for the DT process can be obtained from the transition rates $q_{\alpha,\beta}$.

Assumption 1. For transition probabilities $p(x'|x,a,b)$, there exists function $V(x)$ such that

$$\Delta V(x) \leq -c\|x\| + d, \qquad \forall x \in \mathbb{Z}^m_{\geq 0},$$

with constant $c > 0, d < \infty$.

The expected one-step reward/cost is given by

$$r\Big(x_k, a_k, b_k\Big) := \rho\Big(x(t_{k-1}), a(t_{k-1}), b(t_{k-1})\Big)\mathbb{E}[\Delta t_k | x_k, a_k, b_k], \tag{2}$$

where $\Delta t_k = t_k - t_{k-1}$ is the exponentially distributed inter-transition interval. Now we are ready to formally define the security game to be considered:

Definition 1. We consider a Markov game specified by a tuple $(\mathbb{Z}^m_{\geq 0}, \mathcal{A}, \mathcal{B}, p, r, \gamma)$ defined as follows.

1. $\mathbb{Z}^m_{\geq 0}$ is the state space.
2. $\mathcal{A}$ (resp. $\mathcal{B}$) is the space of (mixed) strategies for the attacker (resp. defender).
3. $p : (\mathbb{Z}^m_{\geq 0} \times \{0,1\}^2) \times \mathbb{Z}^m_{\geq 0} \to [0,1]$ is the transition probability of the state under a given pair of actions; these probabilities can be computed readily from the CT transition rates q.
4. $r : \mathbb{Z}^m_{\geq 0} \times \{0,1\}^2 \to \mathbb{R}$ is the one-step reward/cost function.
5. $\gamma \in (0,1)$ is the discount rate.

By the DT formulation, the value/cost function is thus given by

$$v_{\pi,\sigma}(x) = \mathbb{E}_{\pi,\sigma}\left[\sum_{k=0}^{\infty} \gamma^k r(x_k, a_k, b_k) \middle| x_0 = x\right].$$

In the zero-sum game, the attacker (resp. defender) attempts to maximize (resp. minimize) the above.

Definition 2. The Markov perfect equilibrium (MPE) for the security game is a strategy pair (π^*, σ^*) such that for any $x \in \mathbb{Z}^m_{\geq 0}$,

$$\pi^*(\cdot|x) = \arg\max_{\pi} v_{\pi,\sigma^*}(x),$$
$$\sigma^*(\cdot|x) = \arg\min_{\sigma} v_{\pi^*,\sigma}(x).$$

Hence, the MPE is characterized by the equilibrium state value function

$$v^*(x) = v_{\pi^*,\sigma^*}(x).$$

Note that the corresponding action value function is given by

$$Q_{\pi,\sigma}(x,a,b) = r(x,a,b) + \sum_{x' \in \mathbb{Z}^m_{\geq 0}} p(x'|x,a,b) v_{\pi,\sigma}(x').$$

By the Shapley theory (Shapley, 1953), v^* is associated with a unique action value function (also called the "minimax Q function") satisfying the minimax version of the Bellman equation (Szepesvári & Littman, 1999).

Following (Zhu & Zhao, 2020), we take the defender's perspective of minimax Bellman operator $\mathbf{T}$ on the space of functions $\{Q : \mathbb{Z}^m_{\geq 0} \times \{0,1\}^2 \to \mathbb{R}\}$ as

$$(\mathbf{T}Q)(x,a,b) = r(x,a,b) + \gamma \min_{\sigma \in \mathcal{B}} \max_{a' \in \{0,1\}} \sum_{\substack{x' \in \mathbb{Z}^m_{\geq 0} \\ b' \in \{0,1\}}} p(x'|x,a',b')\sigma(b'|x)Q(x',a',b').$$

Then the minimax Bellman equation can be written compactly as

$$Q^* = \mathbf{T}Q^*,$$

where Q^* is also the action value function associated with v^*.

2.2. Function Approximation

Consider a set of md linearly independent basis functions $\{\phi_i : \mathbb{Z}^m_{\geq 0} \times \{0,1\}^2 \to \mathbb{R}; 1 \leq i \leq m\}$. Let $\phi = [\phi_1, \ldots, \phi_m]^\top$ be the m-dimensional list of basis functions. We follow (Tanwani et al., 2015) and assume the following on regularity of the basis functions.

Assumption 2. The basis functions ϕ satisfy the following.

1. (Subexponential and non-negative) ϕ is such that
$$0 \leq \phi_i(x,a,b) \leq e^{x_i} \quad for\ i \in \{1, 2, \ldots, m\}.$$
2. (Dominance over gradient) There exists a constant $B > 0$ such that for x satisfying
$$\|x\|_2^2 \geq B,$$
it holds that
$$\left\| \frac{\partial \phi}{\partial x}(x,a,b) \right\|_1 < \|\phi(x,a,b)\|_1.$$

Let
$$\mathcal{Q} = \left\{\phi^\top w;\ w \in \mathbb{R}^m\right\}$$
be the space spanned by the basis functions. Then the approximate function $Q_w \in \mathcal{Q}$ is given by

$$Q_w(x,a,b) = \phi(x,a,b)^\top w, \tag{3}$$

where $w \in \mathbb{R}^m$ is the weight vector, with w_i being the weight of ϕ_i.

Assumption 3. There exists constant $\kappa > 0$ such that

$$|r(x,a,b)| \leq \kappa \|\phi(x,a,b)\|_\infty.$$

If the behavior policy pair $(\alpha, \beta) \in \mathcal{A} \times \mathcal{B}$ ensures ergodicity of the state process $\{x(t); t \geq 0\}$, let $\mu_{\alpha,\beta}$ be the invariant probability measure. We will discuss the qualifications for the behavior policy in the next subsection. With the linear function approximation, we in fact approximates the actual equilibrium value function Q^* with a projection Q_{w^*} in $\mathcal{Q}$.

Denote the orthogonal projection operator by $\mathbf{P}$ on the space of functions $\{Q : \mathbb{Z}^m_{\geq 0} \times \{0,1\}^2 \to \mathbb{R}\}$, which is given by

$$(\mathbf{P}Q)(x,a,b) = \phi^\top(x,a,b)\Sigma^{-1}\mathbb{E}_{\mu_{\alpha,\beta}}\left[\phi(x,a,b)Q(x,a,b)\right], \tag{4}$$

where $\mathbb{E}_{\mu_{\alpha,\beta}}$, with a slight abuse of notation, denotes the vector of expectations with respect to the invariant probability measure $\mu_{\alpha,\beta}$. We define the optimal weight vector w^* to verify

$$Q_{w^*}(x,a,b) = (\mathbf{PT}Q_{w^*})(x,a,b), \tag{5}$$

and approximate Q^* with Q_{w^*} as defined above. This Q_{w^*} is actually a fixed point of the projected Bellman operator $\mathbf{PT}$. Note that the corresponding optimal weight vector w^* can also be directly defined as a fixed point of a modified projected Bellman operator.

Accordingly, we follow van Eck and van Wezel (van Eck & van Wezel, 2008) and consider an approximated equilibrium as defined below:

Definition 3. The *linear approximated equilibrium* for the security game is a strategy pair $(\hat{\pi}^*, \hat{\sigma}^*)$ such that for any $x \in \mathbb{Z}^m_{\geq 0}$,

$$\begin{aligned}
&\hat{\pi}^*(\cdot|x) \\
&= \arg\max_{\hat{\pi}\in\mathcal{A}} \sum_{a\in\{0,1\}} \hat{\pi}(a|x) \sum_{b\in\{0,1\}} \hat{\sigma}^*(b|x)\phi^\top(x,a,b)w^*, \\
&\hat{\sigma}^*(\cdot|x) \\
&= \arg\min_{\hat{\sigma}\in\mathcal{B}} \sum_{b\in\{0,1\}} \hat{\sigma}(b|x) \sum_{a\in\{0,1\}} \hat{\pi}^*(a|x)\phi^\top(x,a,b)w^*.
\end{aligned}$$

There are multiple metrics for the quality of approximation. One is the mean error between the actual value Q^* and the approximated value Q_{w^*}. Another is the consistency between the MPE strategy profile (π^*, σ^*) and the approximated MPE strategy profile $(\hat{\pi}^*, \hat{\sigma}^*)$. We will study these metrics in numerical validation.

2.3. Learning Algorithm

We consider an approximate minimax-Q (AMQ) learning algorithm with the following update rule for the weights:

$$\begin{aligned}
w_{k+1} &= w_k + \eta_k \nabla_w Q_w(x_k, a_k, b_k)\Delta_k \\
&= w_k + \eta_k \phi(x_k, a_k, b_k)\Delta_k,
\end{aligned} \tag{6}$$

where Δ_k is the temporal difference at time t_k, given by

$$\begin{aligned}
\Delta_k =& r_k + \gamma \min_{\sigma\in\mathcal{B}} \max_{a\in\{0,1\}} \sum_{b\in\{0,1\}} \sigma(b|x)Q_{w_k}(x_{k+1},a,b) \\
&- Q_{w_k}(x_k, a_k, b_k).
\end{aligned} \tag{7}$$

To obtain σ at iteration k, we actually solve a linear programming as follows, where the optimum objective $c = \max_{a\in\{0,1\}} \sum_{b\in\{0,1\}} \sigma(b|x)Q_{w_k}(x_{k+1},a,b)$.

$$\begin{aligned}
\min \quad & c \\
s.t \quad & \sum_b \sigma(b|x)Q_{w_k}(x_{k+1},a,b) \leq c \qquad \forall a \in \{0,1\} \\
& \sigma(b|x) \geq 0, \qquad \sum_b \sigma(b|x) = 1 \qquad \forall b \in \{0,1\}
\end{aligned} \tag{8}$$

The initial weight w_0 is arbitrary. The pseudo-code is presented below.

Algorithm 1 AMQ learning for the security game

Require:
 Initial weights w_0, behavior policy α, β, step sizes sequence η_k, γ;
1: Initialize weights $w_0 \leftarrow w_0$
2: **for** $k = 0, 1, \cdots$ **do**
3: Sample $A_k \sim \alpha(\cdot|X_k)$, $B_k \sim \beta(\cdot|X_k)$
4: Receive R_{k+1} and observe X_{k+1}
5: Update Δ_k via (7) and (8)
6: Update w_k via (6)
7: **end for**

We assume the following conditions for the behavior policy pair and for the learning rates.

Assumption 4. Let $(\alpha, \beta) \in \mathcal{A} \times \mathcal{B}$ be the behavior policy pair.

1. $\alpha(a|x) > 0, \beta(b|x) > 0$ for $\mu_{\alpha,\beta}$-almost all $x \in \mathbb{Z}^m_{\geq 0}$.

2. There exist $\nu > 0, c > 0, d < \infty$ such that with $V(x) = \sum_{n=1}^m e^{\nu x_n}$,

$$\begin{aligned}
\mathcal{L}_{\alpha,\beta}V(x) &= \sum_{y\in\mathbb{Z}^m_{\geq 0}} q_{\alpha,\beta}(y|x)V(y) - V(x) \\
&\leq -cV(x) + d, \qquad \forall x \in \mathbb{Z}^m_{\geq 0},
\end{aligned}$$

 where $\mathcal{L}_{\alpha,\beta}$ is the infinitesimal generator under policy pair (α, β) and transition rate $q_{\alpha,\beta}(y|x)$.

Assumption 5. The learning rates satisfy

$$\sum_{k=1}^\infty \eta_k = \infty, \quad \sum_{k=1}^\infty \eta_k^2 < \infty.$$

Assumption 4 ensures ergodicity under the behavior policy pair. The class of policy pairs satisfying this assumption is fairly broad. In fact, any ϵ-greedy-type policy pair would verify part 1). Part 2) essentially ensures positive Harris of the traffic state process. An illustrative example is provided below.

$$\alpha(1|x) = e^{-\frac{|x|_1}{2}}, \quad \alpha(0|x) = 1 - e^{-\frac{|x|_1}{2}}, \tag{9a}$$

$$\beta(1|x) = \begin{cases} 1 - e^{-\frac{|x|_1}{2}} & \text{if } x \neq 0^m, \\ 0.5 & \text{if } x = 0^m. \end{cases} \tag{9b}$$

$$\beta(0|x) = \begin{cases} e^{-\frac{|x|_1}{2}} & \text{if } x \neq 0^m, \\ 0.5 & \text{if } x = 0^m. \end{cases} \tag{9c}$$

One can verify that this behavior policy pair satisfies Assumption 4; The assumptions on the learning rates are in fact the standard Robbins-Monro conditions for convergence analysis.

Finally, the AMQ learning algorithm is said to be convergent if $w_k \to w^*$ w.p.1, where w^* verifies the projected Bellman equation (5). The next section is devoted to show this.

2.4. Theoretical guarantee

The main result of this paper states that the approximate minimax-Q (AMQ) learning algorithm is guaranteed to converge to a solution to the projected minimax Bellman equation.

Theorem 1. Consider the Markov security game $(\mathbb{Z}^m_{\geq 0}, \mathcal{A}, \mathcal{B}, p, r, \gamma)$. Under Assumptions 1–5, for any initial weight $w_0 \in \mathbb{R}^d$ and any initial state $x_0 \in \mathbb{Z}^m_{\geq 0}$, the approximate minimax-Q learning algorithm (6) converges in the sense that $w_k \to w^*$ w.p.1., where w^* verifies the projected Bellman equation (5).

Theorem 1 provides a convergence guarantee for the proposed learning method under rather mild assumptions, viz. (i) stabilizability of the system, (ii) regularity of the basis functions, (iii) ergodicity under the behavior policy pair, and (iv) Robbins-Monroe conditions for the learning rates. Thus, the AMQ algorithm will reliably generate effective defense policies for managing data flow in practical scenarios.

3. Application

3.1. Routing System

3.1.1. System Model

Consider a parallel server system. Jobs arrive according to a Poisson process of rate $\lambda > 0$ and go to one of the m servers. The ith server has exponentially distributed service times with service rate $\mu_i > 0$. Let $x(t) \in \mathbb{Z}^m_{\geq 0}$ be the vector of the number of jobs in the servers, either waiting or being served. In the absence of attacks, we assume that an incoming job is routed to the server with the shortest queue; ties are broken uniformly at random. We select this policy because of its intuitiveness and popularity in practice (Singh & Kumar, 2018).

An attacker is able to manipulate the routing decision for an incoming job. A defender can defend the routing decision for an incoming job at a cost of $c_2 > 0$ per unit time. If a routing decision is attacked and is not defended, the job will go to the longest server, as the consequence of a misled decision. Otherwise, the job will join the shortest queue correctly. Ties are broken uniformly at random.

The instantaneous reward (resp. cost) for the attacker (resp. defender) at time t is defined as

$$\rho\Big(x(t), a(t), b(t)\Big) := \|x(t)\|_1 - c_1 a(t) + c_2 b(t), \tag{10}$$

where $\|\cdot\|_1$ is the 1-norm.

The transition rate $q_{\alpha,\beta} : \mathbb{Z}^m_{\geq 0} \times \mathbb{Z}^m_{\geq 0} \to \mathbb{R}_{\geq 0}$ of the traffic state under the policy pair (α, β) is given by

$$q_{\alpha,\beta}(y|x) = \begin{cases} \Big(\frac{\alpha(0|x)}{|\arg\min_j x_j|} + \frac{\alpha(1|x)\beta(1|x)}{|\arg\min_j x_j|}\Big)\lambda & \\ & \text{if } y \in \{x + e_i; i \in \arg\min_j x_j\}, \\ \frac{\alpha(1|x)\beta(0|x)}{|\arg\min_j x_j|}\lambda & \text{if } y \in \{x + e_i; i \in \arg\max_j x_j\}, \\ \mu_i & \text{if } y = x - e_i, \\ 0 & \text{otherwise}, \end{cases}$$

where $|\cdot|$ denotes the cardinality of a set. We exclude the case of self-transition since it does not affect our analysis.

3.1.2. Numerical Validation

In this section, we implement the approximate minimax-Q (AMQ) learning algorithm and numerically evaluate its performance. The objectives of this section is (i) to present and interpret the cost-aware defending strategy given by the AMQ method and (ii) to study the computational efficiency and approximation accuracy of the AMQ method.

We simulate two system models, one with three parallel servers and one with six; this is intended to study the impact of system complexity. The service rates are listed in Table 1: μ_1–μ_3 are used for the three-server model, while μ_1–μ_6 are used for the six-server model. The table also gives the other parameters. The policies given by (9a)–(9c) are used as the behavior policies. The initial target policies are set to be the random policies $\sigma(0|x) = \sigma(1|x) = 0.5$ and $\pi(0|x) = \pi(1|x) = 0.5$ for all $x \in \mathbb{Z}^m_{\geq 0}$. The initial traffic state is randomly generated.

We use a neural network Q (NNQ) learning as the benchmark for evaluate the AMQ method. The NNQ methods approximates the value function $Q(x, a, b)$ with a neural

Table 1. Experiment parameters.

PARAMETER	NOTATION	VALUE
ARRIVAL RATE	λ	5 PER UNIT TIME
SERVICE RATE 1	μ_1	2 PER UNIT TIME
SERVICE RATE 2	μ_2	3 PER UNIT TIME
SERVICE RATE 3	μ_3	4 PER UNIT TIME
SERVICE RATE 4	μ_4	2 PER UNIT TIME
SERVICE RATE 5	μ_5	0.5 PER UNIT TIME
SERVICE RATE 6	μ_6	1 PER UNIT TIME
ATTACKING COST	c_1	8 PER UNIT TIME
DEFENDING COST	c_2	6 PER UNIT TIME
DISCOUNT FACTOR	γ	0.9

network and trains it according to the minimax Bellman equation. Since NNs have extremely strong approximation performance, we use the NNQ function as a proxy for the ground truth of the equilibrium value, which cannot be analytically obtained. The architecture of the NN comprises two fully connected layers, employing a rectified linear unit (ReLU) as the activation function. The NN is updated via adaptive moment estimation. The loss function used is the mean squared error between the predicted one-step and calculated state-action value.

For the AMQ method, we consider two approximators with different dimensions. The first, named "AMQ1", is a collection of affine functions of the traffic states: for $i = 1, 2, \ldots, m$,

$$\begin{aligned}&\phi_{i,1}(x,a,b) = 1, \quad \phi_{i,2}(x,a,b) = x_i + \delta_i(x,a,b),\\ &\phi_{i,3}(x,a,b) = a, \quad \phi_{i,4}(x,a,b) = b,\end{aligned}$$

where $\delta_i(x,a,b)$ is given by

$$\delta_i(a,b) := \begin{cases} 1 & \text{if } i = \arg\max_i x_i, (a,b) = (1,0),\\ 1 & \text{if } i = \arg\min_i x_i, (a,b) \neq (1,0),\\ 0 & \text{otherwise.}\end{cases}$$

Note that we make $\phi_{i,1}, \phi_{i,2}, \ldots$ associated with server i; this construction incorporates particularly the parallel structure of server system and thus gives interpretability of the weights. Intuitively, the feature functions are motivated by the reward function in (10). The second, named "AMQ2", is a collection of second-order polynomials of the traffic states: for $i = 1, 2, \ldots, m$,

$$\begin{aligned}&\phi_{i,1}(x,a,b) = 1, \quad \phi_{i,2}(x,a,b) = x_i + \delta_i(x,a,b),\\ &\phi_{i,3}(x,a,b) = \Big(x_i + \delta_i(x,a,b)\Big)^2,\\ &\phi_{i,4}(x,a,b) = a, \quad \phi_{i,5}(x,a,b) = b.\end{aligned}$$

Hence, AMQ2 is more flexible than AMQ1 and will turn out to be more accurate than AMQ1.

Table 2. Algorithms to be compared.

ALGORITHM	APPROXIMATOR
NNQ (BASELINE)	TWO-LAYER NEURAL NETWORK WITH RELU
AMQ1 (OURS)	AFFINE FUNCTIONS OF TRAFFIC STATE
AMQ2 (OURS)	SECOND-ORDER POLYNOMIALS OF TRAFFIC STATE

Table 2 summarizes the three algorithms that we consider. Note that they are all off-policy temporal-difference learning methods.

We trained and evaluated the learning algorithms for 2×10^6 epochs. A discrete time step of 0.1 seconds was employed for simulation. All experiments were conducted using Jupyter Notebook, hosted on a system equipped with an Intel(R) Xeon(R) CPU with 36.7 GB of memory.

Every experiment that we conducted converged to an approximate equilibrium. As an illustration, consider the three-server setting. The weights for the AMQ2 in this setting turn out to be

$$\begin{aligned}&w_{1,1} = 6.55, \quad w_{2,1} = 5.55, \quad w_{3,1} = 4.55,\\ &w_{1,2} = 9.74, \quad w_{2,2} = 9.23, \quad w_{3,2} = 9.02,\\ &w_{1,3} = 0.46, \quad w_{2,3} = 0.41, \quad w_{3,3} = 0.34,\\ &w_{1,4} = 0.9, \quad w_{2,4} = 0.8, \quad w_{3,4} = 0.8,\\ &w_{1,5} = -1.1, \quad w_{2,5} = -1.0, \quad w_{3,5} = -0.89.\end{aligned}$$

Recall that the function $\phi^\top w^*$ is the (approximate) equilibrium cost for the defender; the first index in the subscript is actually the server index.

There are several insights about the weights associated with the same server worth mentioning. First, the first-order terms are associated with weights ($w_{i,3}$) greater than the second-order terms ($w_{i,2}$); this implies that the value function grows roughly linearly with the traffic states. Second, a non-trivial intercept exists ($w_{i,1}$) for every server, which implies that a server might be associated with a risk even if it is idling. Finally, the weights ($w_{i,4}, w_{i,5}$) associated with the player actions have the correct signs. In addition, attacks are associated with smaller weights than defenses, so the defender seems to have a stronger incentive to defend than the attacker to attack; this is probably due to that the defending cost is lower than the attacking cost.

Across various servers, it turns out that queues with lower service rates are in general associated with higher risks, which is intuitive. Interestingly, the greater intercepts ($w_{i,1}$) are consistently associated with higher service rates; that is, an incorrect routing to a slow server, even if it is idling, may still be costly. In addition, the weights ($w_{i,4}, w_{i,5}$) associated with the player actions directly indicates the benefit of attacking/defending a particular server; servers with slower service rates are associated with, without surprise, higher weights.

Table 3. Performance of various methods

METRIC	SYSTEM	AMQ1	AMQ2	NNQ
NORMALIZED MEAN COST	3-SERVER	1.079	1.043	1.000
POLICY CONSISTENCY	3-SERVER	94.2%	97.5%	100%
NORMALIZED MEAN COST	6-SERVER	1.082	1.045	1.000
POLICY CONSISTENCY	6-SERVER	94.1%	97.3%	100%

Table 3 presents the normalized learned values and policies with respect to the equilibrium state distribution. The initial state is sampled from this equilibrium distribution, and empirical data is obtained using the Monte Carlo method. The reported results represent the average of 10 repeated experiments. The findings indicate that the learned results of AMQ2 approximate optimal defense strategies with an average error of 2.5%, and approximate the optimal values with an average error of 4.3% under the equilibrium distribution, thus validating the precision of the proposed algorithm in approximating both optimal values and optimal policies. The performance of the AMQ2 algorithm further highlights that the inclusion of quadratic terms in the feature functions improves the empirical average cost by 3.6% and the empirical policy consistency by 3.3%. These results underscore the necessity of incorporating quadratic feature functions to achieve more accurate learning outcomes.

Fig. 1 illustrates the normalized l_2-norm difference between the weights w_t and the optimal weights w^* throughout the learning process for both the three methods. It is evident that the NN method converges after approximately 2.4×10^5 iterations, whereas the AMQ1 and the AMQ2 method achieves convergence after around 5×10^3 iterations. Hence, our proposed algorithm has a much higher convergence rate compared to NN, validating the efficiency of the AMQ learning algorithm.

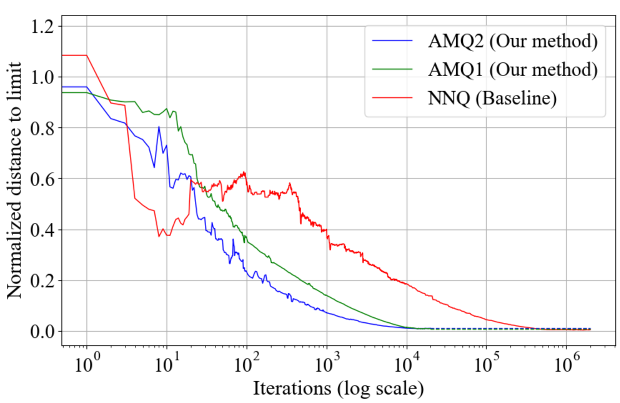

Figure 1. Performance comparison on distance to limit.

To test the scalability of AMQ, we further implement identical experiments on six servers. The results are also shown in Table 3. It can be seen that in six servers setting, the performance of AMQ degrades at most 0.2% in approximating optimal defense strategies and 0.3% in approximating the optimal values, compared to the three servers case. The results indicate that the computational advantage of linear approximation remains at more servers.

3.2. Polling System

3.2.1. SYSTEM MODEL

Consider a polling system with n queues. Jobs arrive at each queue i according to a Poisson process with arrival rate λ_i. A server services these queues sequentially. The service time for queue i follows an exponential distribution with service rate $\mu_i > 0$. Let $x(t) \in \mathbb{Z}^n_{\geq 0}$ be the vector of the number of jobs in each queue, and $p(t) \in \{1, 2, \ldots, n\}$ denotes the index of the currently polled queue. In the absence of attacks, we assume the server employs a longest queue polling policy: the server always serves the queue with the longest queue length. We select this strategy for its intuitiveness and widespread practical use.

Attackers can manipulate the polling decision with cost being c_a. When server prepares to move, they can alter the queue for next polling. Defenders can defend against the polling decision, with defence cost being c_b. If the polling decision is attacked and not defended, the server moves to the shortest queue; otherwise, the server correctly moves to the next queue. Ties are broken uniformly at random.

The instantaneous reward (resp. cost) for the attacker (resp. defender) at time t is defined as

$$\rho(x(t), a(t), b(t)) := \frac{(\sum_{i=1}^n x_i)^2}{n \cdot \sum_{i=1}^n x_i^2} - C_{switch} \cdot n_s - c_a a(t) + c_b b(t),$$

where $\frac{(\sum x_i)^2}{n \cdot \sum x_i^2}$ is the measure of load distribution fairness among queues. $C_{switch} \cdot n_s$ denotes cost associated with server switching between queues, where n_s is number of queue switches.

The transition rate $q_{\alpha,\beta} : \mathbb{Z}^n_{\geq 0} \times \mathbb{Z}^n_{\geq 0} \to \mathbb{R}_{\geq 0}$ of the traffic state under the policy pair (α, β) is given by

$$q_{\alpha,\beta}(y|x) = \begin{cases} \mu \left[\frac{\alpha(0|x)+\alpha(1|x)\beta(1|x)}{|\arg\max_k x_k|}\right] & \\ & \text{if } y \in \{x - e_i; i \in \arg\max_j x_j\}, \\ \mu \left[\frac{\alpha(1|x)\beta(0|x)}{|\arg\min_k x_k|}\right] & \text{if } y \in \{x - e_i; i \in \arg\min_j x_j\}, \\ \lambda_i & \text{if } y = x + e_i, \\ 0 & \text{otherwise,} \end{cases}$$

where e_i denotes the unit vector in the i-th dimension, $|\cdot|$ denotes the cardinality of a set. The cases for service completion reflect the probability of serving a specific queue based on the attacker's and defender's actions. We exclude self-transitions as they do not affect the analysis.

3.2.2. Numerical Validation

3.3. Scheduling System

3.3.1. System Model

Consider a task scheduling system with n servers. Tasks arrive according to a Poisson process of rate $\lambda > 0$ and need to be scheduled to one of the servers. The ith has exponentially distributed service times with service rate $\mu_i > 0$. Let $x(t) \in \mathbb{Z}^n_{\geq 0}$ be the vector of the number of tasks in the servers' queues, either waiting or being processed, and let $r(t) \in \mathbb{R}^m_{\geq 0}$ represent the current resource utilization (e.g., CPU, GPU, memory) of each server, where $r_i(t) \in [0, 1]$ denotes the normalized utilization level.

In the absence of attacks, we assume that an incoming task is scheduled to the server with the shortest expected completion time (ECT), which considers both queue length and server capability;

Its application scenarios includes the classical mobile edge computing (MEC). In MEC, scheduling refers to the intelligent decision-making process that manages and allocates limited resources to optimize system performance.

3.3.2. Numerical Validation

Accessibility

Software and Data

Acknowledgements

Impact Statement

References

Benveniste, A., Métivier, M., and Priouret, P. *Adaptive Algorithms and Stochastic Approximations*, volume 22. Springer Science & Business Media, 2012.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Shapley, L. S. Stochastic Games. *Proceedings of the National Academy of Sciences*, 39(10):1095–1100, 1953.

Singh, R. and Kumar, P. Throughput optimal decentralized scheduling of multihop networks with end-to-end deadline constraints: Unreliable links. *IEEE Transactions on Automatic Control*, 64(1):127–142, 2018.

Szepesvári, C. and Littman, M. L. A unified analysis of value-function-based reinforcement-learning algorithms. *Neural Computation*, 11(8):2017–2060, 1999.

Tanwani, A., Brogliato, B., and Prieur, C. Stability notions for a class of nonlinear systems with measure controls. *Mathematics of Control, Signals, and Systems*, 27:245–275, 2015.

van Eck, N. J. and van Wezel, M. Application of reinforcement learning to the game of Othello. *Computers & Operations Research*, 35(6):1999–2017, 2008.

Zhu, Y. and Zhao, D. Online minimax Q network learning for two-player zero-sum Markov games. *IEEE Transactions on Neural Networks and Learning Systems*, 33(3): 1228–1241, 2020.

A. Appendix

We will prove Theorem 1 in three steps. In Section A.1, we show that the state is geometrically ergodic under the behavior policy pair (Lemma 1) and that the basis function has a bounded first moment with respect to the corresponding invariant probability measure (Lemma 2). In Section A.2, we show that the first moment of the temporal-difference (TD) error is bounded by a linear function of the norm of the weight vector (Lemma 3). In Section A.3, we apply the ordinary differential equation-based argument to the first moment of the TD error and establish the convergence of the proposed algorithm.

A.1. Geometric ergodicity and boundedness of basis functions

Under a behavior policy pair (α, β) satisfying Assumption 4, the induced chain $(\mathcal{X}, P_{\alpha,\beta})$ is geometrically ergodic with corresponding equilibrium probability measure $\mu_{\alpha,\beta}$. To argue for the irreducibility of the induced chain, note that the state $x = 0$ can be accessible from any initial condition with positive probability. Hence, the induced chain is exponentially ergodic.

Let Φ be the matrix defined as

$$\Phi = \mathbb{E}_{\mu_{\alpha,\beta}}\left[\phi(x,a,b)\phi^\top(x,a,b)\right],$$

where $\mathbb{E}_{\mu_{\alpha,\beta}}$, with a slight abuse of notation, denotes the matrix of expectations with respect to the invariant probability measure $\mu_{\alpha,\beta}$. To prove the boundedness of infinity norm of Φ, we first derive Lemma 1 to show the quadratic version of Lyapunov function $V(x) = \sum_{n=1}^{m} e^{\nu x_n}$ has a negative drift with $\upsilon > 0$.

Lemma 1. Suppose that assumption $1, 4$ hold. Let $W(x) = (\sum_{n=1}^{m} e^{\nu x_n})^2$, $\nu > 0$. Then there exist some $d' < \infty$ such that

$$\mathcal{L}_{\alpha,\beta}W(x) = \sum_{y\in\mathbb{Z}^m_{\geq 0}} q_{\alpha,\beta}(y|x)W(y) - W(x) \leq -cW(x) + d', \qquad x \in \mathbb{Z}^m_{\geq 0},$$

where $\mathcal{L}_{\alpha,\beta}$ and constant c are defined in Assumption 4.

Proof. By Assumption 4 we obtain that

$$\mathcal{L}_{\alpha,\beta}\Big(\sum_{n=1}^{m} e^{\nu x_n}\Big) \leq -c\Big(\sum_{n=1}^{m} e^{\nu x_n}\Big) + d, \quad x \in \mathbb{Z}^m_{\geq 0},$$

where c, d is finite constant defined in Assumption 4. Note that $c > 0$. Then the infinitesimal generator of $W(x)$

$$\mathcal{L}_{\alpha,\beta}W(x) = 2\Big(\sum_{n=1}^{m} e^{\nu x_n}\Big)\mathcal{L}\Big(\sum_{n=1}^{m} e^{\nu x_n}\Big) \leq -2c\Big(\sum_{n=1}^{m} e^{\nu x_n}\Big)^2 + 2d\Big(\sum_{n=1}^{m} e^{\nu x_n}\Big) = -cW(x) + d',$$

where d' is a finite positive constant satisfying

$$d' = -c\Big(\sum_{n=1}^{m} e^{\nu x_n}\Big)^2 + 2d\Big(\sum_{n=1}^{m} e^{\nu x_n}\Big) \leq \frac{d^2}{c}.$$

$\square$

Lemma 2. Suppose that assumption $1-3$ hold, then

$$\left\|\mathbb{E}_{\mu_{\alpha,\beta}}\Big[\phi(x,a,b)\phi^\top(x,a,b)\Big]\right\|_\infty \leq \frac{d'}{c}, \tag{11}$$

where c, d' is the constant in Lemma 1.

Proof. By Lemma 1 we obtain that

$$\lim_{t\to\infty}\frac{1}{t}\int_{s=0}^{t}\mathbb{E}\Big[\sum_{i=1}^{m}\sum_{j=1}^{m} e^{\nu\left(x_i(s)+x_j(s)\right)}\Big]ds \leq \frac{d'}{c} < \infty.$$

Hence, by Assumption 2, since

$$\phi_i(x) \leq e^{x_i} \quad for\ i \in \{1, 2, \dots, m\},$$

then with $\psi(x) = \mathbb{E}_{\mu_{\alpha,\beta}}\left[\sum_{i=1}^m \phi_i(x,a,b)\right] \leq \mathbb{E}_{\mu_{\alpha,\beta}}\left[\sum_{i=1}^m e^{x_i}\right]$ that

$$\lim_{t\to\infty} \frac{1}{t}\int_{s=0}^{t} \mathbb{E}[\psi^2(x(s))]ds \leq \frac{d'}{c}$$

for any initial condition $x(0)$. Then we can conclude that

$$\lim_{t\to\infty} \mathbb{E}[\psi^2(x(t))] \leq \frac{d'}{c},$$

which means

$$\left\|\mathbb{E}_{\mu_{\alpha,\beta}}\Big[\phi(x,a,b)\phi^\top(x,a,b)\Big]\right\|_\infty \leq \frac{d'}{c},$$

where $\mu_{\alpha,\beta}$ is the equilibrium state-action distribution under policy α,β. □

A.2. Boundedness of gradient

We write (6) in the form

$$w_{k+1} = w_k + \eta_k H(w_k, Y_{k+1}),$$

where $Y_{k+1} = (x_k, a_k, b_k)$, and

$$H(w,Y) = \phi(x,a,b)\Big(r(x,a,b) + \gamma \min_{\sigma\in\mathcal{B}} \max_{a'\in\{0,1\}} \sum_{b'\in\{0,1\}} \sigma(b')Q_w(y,a',b') - Q_w(x,a,b)\Big). \tag{12}$$

Lemma 3. The function H satisfies

$$\left\|\mathbb{E}_{\mu_{\alpha,\beta}}\left[H(w,x,a,b)\right]\right\|_\infty \leq C(1+\|w\|_\infty), \tag{13}$$

for any w, where C is a finite constant.

Proof. Denote e_i as the unit vector with only the ith element equals 1. Also denote the longest queue as $x_{\max}$ and its corresponding index as i. Define similarly the shortest queue $x_{\min}$ and its index j.

Denote by $g(x,a,b,y)$ the vector as

$$g(x,a,b,y) = \max_{\substack{a'\in\{0,1\}\\ b'\in\{0,1\}}} \phi(y,a',b') - \phi(x,a,b)$$

Hence, we can obtain by definition of (12) that

$$\begin{aligned}
\left\|\mathbb{E}_{\mu_{\alpha,\beta}}\left[H(w,x,a,b)\right]\right\|_\infty &\leq \left\|\mathbb{E}_{\mu_{\alpha,\beta}}\Big[\phi(x,a,b)\big[r(x,a,b,y) + g^\top(x,a,b,y)\cdot w\big]\Big]\right\|_\infty \\
&\leq \left\|\mathbb{E}_{\mu_{\alpha,\beta}}\left[\phi(x,a,b)\cdot r(x,a,b,y)\right]\right\|_\infty + \left\|\mathbb{E}_{\mu_{\alpha,\beta}}\left[\phi(x,a,b)\cdot g^\top(x,a,b,y)\right]\right\|_\infty \cdot \|w\|_\infty
\end{aligned} \tag{14}$$

When $\sum_{k=1}^m x_k^2 \geq B$, it can be deduced that $\|g^\top(x,a,b,y)\|_1 \leq \|\phi^\top(x,a,b)\|_1$ by Assumption 2. Thus by Lemma 2,

$$\begin{aligned}
&\left\|\mathbb{E}_{\mu_{\alpha,\beta}}\Big[\phi(x,a,b)g^\top(x,a,b,y)\Big|\sum_{k=1}^m x_k^2 \geq B\Big]\cdot P\Big(\sum_{k=1}^m x_k^2 \geq B\Big)\right\|_\infty \\
&+ \left\|\mathbb{E}_{\mu_{\alpha,\beta}}\Big[\phi(x,a,b)g^\top(x,a,b,y)\Big|\sum_{k=1}^m x_k^2 < B\Big]\cdot P\Big(\sum_{k=1}^m x_k^2 < B\Big)\right\|_\infty < \frac{d'}{c} + \left\|\mathbb{E}_{\mu_{\alpha,\beta}}[\phi(x,a,b)(\zeta(B))^2]\right\|_\infty
\end{aligned}$$

where c, d' is the constant in Lemma 1, $\zeta(B)$ is a finite positive constant related to the specific form of ϕ. It can be easily derived according to different combination of state action pairs, e.g. $\zeta(B) = (\sqrt{B}+1)^2$ when adopting polynomial approximators. By Assumption 4, we obtain

$$\lim_{t\to\infty}\frac{1}{t}\int_{s=0}^{t}\mathbb{E}\Big[\sum_{i=1}^m e^{\nu\left(x_i(s)\right)}\Big]ds \leq \frac{d}{c} < \infty.$$

where d is the constant in Assumption 4. Hence, we can derive with $\psi(x) = \mathbb{E}_{\mu_{\alpha,\beta}}\left[\|\phi(x,a,b)\|_1\right]$,

$$\lim_{t\to\infty}\frac{1}{t}\int_{s=0}^{t}\mathbb{E}[\psi(x(s))]ds \le \frac{d}{c}$$

for any initial condition $x(0)$. Then we conclude that

$$\lim_{t\to\infty}\mathbb{E}[\psi(x(t))] \le \frac{d}{c},$$

which implies

$$\left\|\mathbb{E}_{\mu_{\alpha,\beta}}[\phi(x,a,b)]\right\|_\infty \le \frac{d}{c}.$$

Hence,

$$\left\|\mathbb{E}_{\mu_{\alpha,\beta}}[\phi(x,a,b)g^\top(x,a,b,y)]\right\|_\infty < \frac{1}{c}\Big(d' + d(\zeta(B))^2\Big). \tag{15}$$

Then by Assumption 3, we can conclude

$$\left\|\mathbb{E}_{\mu_{\alpha,\beta}}\left[\phi(x,a,b)\cdot r(x,a,b)\right]\right\|_\infty \le \kappa\left\|\mathbb{E}_{\mu_{\alpha,\beta}}\left[\phi(x,a,b)\phi^\top(x,a,b)\right]\right\|_\infty \le \frac{\kappa d'}{c}.$$

Hence, (13) can be satisfied by selecting $C = \max\left\{\frac{1}{c}\Big(d' + d(\zeta(B))^2\Big), \frac{\kappa d'}{c}\right\}$. □

A.3. Proof of Theorem 1

We first prove the convergence of approximate minimax Q learning w.p.1. Let μ_x be the corresponding invariant probability measure, $\mu_{\alpha,\beta}$ be the invariant state-action distribution under the given behavior policy pair (α,β). It verifies the existence of function

$$h(w) = \int H(w,Y)\mu_{\alpha,\beta}(dY)$$

by bound of function $H(w,Y)$ derived in Lemma 3.

Since the chain is geometrically ergodic, it follows that so is the chain Y_k. The geometric ergodicity of Y_k and the fact that α,β do not depend on w ensure that the requirements are satisfied. Hence, by Theorem 17 of (Benveniste et al., 2012) , the convergence of w_k w.p.1 is established as long as the ODE

$$\dot{w}_k = h(w_k) \tag{16}$$

with

$$h(w) = \mathbb{E}_{\mu_{\alpha,\beta}}\Big[\phi(x,a,b)\Big(r(x,a,b) - \phi^\top(x,a,b)w + \gamma\min_{\sigma\in\mathcal{B}}\max_{a'\in\{0,1\}}\sum_{b'\in\{0,1\}}\sigma(b')\phi^\top(y,a',b')w\Big)\Big],$$

has a globally asymptotically stable equilibrium w^*.

We can write h as

$$h(w) = h_1(w) - h_2(w),$$

with

$$h_1(w) = \mathbb{E}_{\mu_{\alpha,\beta}}[\phi(x,a,b)(r(x,a,b) + +\gamma\min_{\sigma\in\mathcal{B}}\max_{a'\in\{0,1\}}\sum_{b'\in\{0,1\}}\sigma(b')\phi^\top(y,a',b')w)]$$

and

$$h_2(w) = \mathbb{E}_{\mu_{\alpha,\beta}}[\phi(x,a,b)\phi^\top(x,a,b)w)].$$

Then using the non-expansiveness of *min* and *max* operator, we can conclude

$$\|h_1(w_1) - h_1(w_2)\|_\infty \leq \gamma \|\mathbb{E}_{\mu_{\alpha,\beta}}[\phi(x,a,b) \max_{\sigma\in\mathcal{B}} \max_{a'\in\{0,1\}} \sum_{b'\in\{0,1\}} \sigma(b')\phi^\top(y,a',b')(w_1-w_2)]\|_\infty$$

$$\leq \gamma\Big(\|\mathbb{E}_{\mu_{\alpha,\beta}}[\phi(x,a,b)\phi^\top(x,a,b)]\|_\infty + \|\mathbb{E}_{\mu_{\alpha,\beta}}[\phi(x,a,b)g^\top(x,a,b,a',b')]\|_\infty\Big)\cdot\|w_1-w_2\|_\infty,$$

where $g(x,a,b,y)$ is defined in Lemma 3.

Actually, we can scale the feature function $\phi(x,a,b)$ arbitrarily to make h_1 be γ-contraction. Scale $\phi(x,a,b)$ by a constant factor $\varepsilon \leq \frac{\sqrt{[d'+d(\zeta(B))^2]^2+4d'c}-[d'+d(\zeta(B))^2]}{2d'}$, where B is the constant defined in Assumption 2. Then by Lemma 2 and condition (15) in Lemma 3 we can ensure

$$\|h_1(w_1) - h_1(w_2)\|_\infty = \gamma\Big(\varepsilon^2\|\mathbb{E}_{\mu_{\alpha,\beta}}[\phi(x,a,b)\phi^\top(x,a,b)]\|_\infty + \varepsilon\|\mathbb{E}_{\mu_{\alpha,\beta}}[\phi(x,a,b)g^\top(x,a,b,y)]\|_\infty\Big)\cdot\|w_1-w_2\|_\infty$$
$$\leq \gamma\Big[\frac{\varepsilon^2 d'}{c} + \varepsilon\Big(\frac{d'}{c} + \frac{d}{c}(\zeta(B))^2\Big)\Big]\cdot\|w_1-w_2\|_\infty \leq \gamma\|w_1-w_2\|_\infty. \tag{17}$$

Also, we can conclude by Lemma 2 that

$$\|h_2(w_1) - h_2(w_2)\|_\infty = \|\mathbb{E}_{\mu_{\alpha,\beta}}[\phi(x,a,b)\phi^\top(x,a,b)(w_1-w_2)]\|_\infty \leq \|(w_1-w_2)\|_\infty. \tag{18}$$

Next we calculate the derivative of p-norm of term $(w_k - w^*)$, where w^* is the equilibrium point of (16) which verifies $h(w^*) = 0$.

$$\frac{d}{dk}\|w_k - w^*\|_p = \|w_k - w^*\|_p^{1-p}\cdot\Big(\sum_{i=1}^d (w_k(i) - w^*(i))^{p-1}\cdot\big((h_1(w_k))_i - (h_1(w^*))_i\big) + \sum_{i=1}^d (w_k(i) - w^*(i))^{p-1}\cdot\big((h_2(w^*))_i - (h_2(w_k))_i\big)\Big),$$

where we denote by $(h_1(w))_i$ the i^{th} component of $h_1(w)$ and similarly for h_2. Applying Hölder's inequality to the above summations yields

$$\frac{d}{dk}\|w_k - w^*\|_p \leq \|h_1(w_k) - h_1(w^*)\|_p + \|h_2(w^*) - h_2(w_k)\|_p.$$

Taking the limit as $p \to \infty$ and using (17) and (18) leads to

$$\frac{d}{dk}\|w_k - w^*\|_\infty \leq (\gamma - 1)\|w_k - w^*\|_\infty. \tag{19}$$

Let $\lambda = 1 - \gamma > 0$. Integrate w.r.t k, (19) becomes

$$\|w_k - w^*\|_\infty \leq e^{-\lambda k}\|w_0 - w^*\|_\infty,$$

which establishes the existence of a globally asymptotically stable equilibrium point for (16). And it is clear that $h(w^*) = 0$ leads to

$$w^* = \Sigma^{-1}\mathbb{E}_{\mu_{\alpha,\beta}}[\phi(x,a,b)(r(x,a,b,y) + \gamma \min_{\sigma\in\mathcal{B}} \max_{a'\in\{0,1\}} \sum_{b'\in\{0,1\}} \sigma(b')\phi^\top(y,a',b')w^*)]. \tag{20}$$

Hence, the sequence w_k converges w.p.1 to the globally asymptotically stable equilibrium point w^*.

Then we further prove that the limit of approximate minimax-Q function is the fixed point of projected Bellman operator.

Given w^* as (20), the corresponding approximate Q function

$$Q_{w^*}(x,a,b) = \phi^\top(x,a,b)\Sigma^{-1}\mathbb{E}_{\mu_{\alpha,\beta}}\left[\phi(x,a,b)(\mathbf{T}Q_{w^*})(x,a,b)\right] = (\mathbf{PT}Q_{w^*})(x,a,b).$$

This implies that Q_{w^*} verifies the fixed point equation in (5).