Adiabatic Markov Decision Process with Application to Queuing Systems

Thai Duong, Duong Nguyen-Huu, Thinh Nguyen
School of Electrical Engineering and Computer Science
Oregon State University

***Abstract*—Markov Decision Process (MDP) is a well-known framework for devising the optimal decision making strategies under uncertainty. Typically, the decision maker assumes a stationary environment which is characterized by a time-invariant transition probability matrix. However, in many real-world scenarios, this assumption is not justified, thus the optimal strategy might not provide the expected performance. In this paper, we study the performance of the classic Value Iteration (VI) algorithm for solving an MDP problem under non-stationary environments. Specifically, the non-stationary environment is modeled as a sequence of time-variant transition probability matrices governed by an adiabatic evolution inspired from quantum mechanics. We characterize the performance of the VI algorithm subject to the rate of change of the underlying environment. The performance is measured in terms of the convergence rate to the optimal average reward. We show two examples of queuing systems that make use of our analysis framework.**

***Keywords*—Markov Decision Process, Adiabatic, Value Iteration.**

I. Introduction

The theory of Markov Decision Process (MDP) aims to study optimal decision making processes under uncertainty. It is widely used in economics, engineering, operation research, and artificial intelligence. In an MDP setting, there is a controller who interacts with its environment by taking actions based on its observations at every discrete time step. Each action by the controller induces a change in the environment. Typically, the environment is described by a finite set of states. An action will move the environment from the current state to some other states with certain probabilities. Associated with each action in each state is a reward given to the controller. The goal of the controller is to maximize the expected cumulative reward or average reward over some finite or infinite number of time steps by making sequential decisions based on its current observations.

It is not difficult to find many applications of the MDP framework. A classic application of MDP is the warehouse example in operation research. In this setup, a company's business is to buy and sell a number of merchandises. To operate smoothly, it uses a warehouse to store the merchandises that allows shipments to the buyers promptly. Everyday, it has to make the decision on how many and which items it should buy and store in its warehouse subject to the uncertainty of the market demands. Buying too many items would incur high storage costs while buying too little would run the risk of not having the items ready for shipping, and thus reducing profits. The MDP framework enables the company to decide on the optimal action, i.e., how many and which items it should buy on a given day in order to maximize the expected cumulative reward, i.e., its profits over a month, a year, or indefinitely. Naturally, the optimal action should based on the environmental states, i.e., the current status of different items in the stock and the current market demands.

A solution of an MDP problem is an optimal policy. A policy/decision rule is a mapping from the states to the action. The optimal policy would produce the maximum expected cumulative reward. For the infinite-horizon MDP models, to be discussed subsequently, there is a number of classic algorithms for finding such optimal policies. These algorithms include Value Iteration, Policy Iteration, Linear Programming, all are based on the Bellman equations [1][2]. All also assume a stationary policy, i.e., a policy that does not change with time. This assumption is justified as it is well-known that for a stationary environment, there exists an optimal policy that is also stationary. Fundamentally, the MDP framework relies on the assumption that a given policy will induce a stationary dynamic on the states. Moreover, the state changes are characterized by a time-invariant transition probability matrix P.

For many real-world scenarios, this assumption is not justified, thus the optimal policy might not provide the expected performance. In this paper, we study the performance of the classic Value Iteration (VI) algorithm for solving an MDP problem under non-stationary environments. Specifically, the non-stationary environment is modeled as a sequence of time-variant transition probability matrices governed by an adiabatic evolution inspired from quantum mechanics [3] [4] [5]. Formally, the transition probability matrix P_i^d at time step i induced by decision rule d is determined by:

$$P_i^d = \Phi(i)P_0^d + (1 - \Phi(i))P_f^d, \tag{1}$$

where P_0^d and P_f^d are the transition probability matrices induced by the decision rule d at time step 0 and ∞, and $\Phi(.)$ characterizes the rate of change of the system with $\Phi(0) = 1$ and $\Phi(\infty) = 0$.

The above transition model can be applied in two interesting scenarios. In the first scenario, P_i^d model the actual dynamics of the underlying non-stationary environment. In other words, the environment is initially characterized by P_0, then over time it converges to P_f. In the second scenario, the environment is always stationary, and is characterized by P_f. However the estimation of the environmental parameters is initially inaccurate, and thought to be P_0. Thus, the actions/decisions are made based on inaccurate knowledge of the environment. However, over time, the estimations of the environmental parameters become increasingly more accurate, i.e.,

P_i getting closer to P_f. Thus, the decisions are closer to the optimal ones, and eventually converge to the optimal policy. That said, we characterize the performance of the VI algorithm subject to the rate of change of the environment ($\Phi(.)$). The performance is measured in terms of the convergence rate to the optimal average reward. We present two queuing system examples illustrating the two scenarios above that make use of our analysis framework.

This paper is organized as follows. The Section II provides some background on the theory of Markov Decision Process, Value Iteration and Adiabatic Settings which are necessary for the next Sections. In Section III, we formulate the problem in term of the distance from the average reward to the optimal one. In Section IV, the theoretical results on the convergence rate of Adiabatic Value Iteration Algorithm are presented. Section V formulates Markov Decision Process for two examples of queuing system and applies the theoretical results to them. Finally, some conclusions are provided in the Section VI.

II. Preliminaries

A. Markov Decision Process

A typical discrete-time MDP represents a dynamic system and is specified by a finite set of states S, representing the possible states of the system, a set of control actions A, a transition probability matrix $P^{|S|\times|S|}$, and a reward function r. The transition probability specifies the dynamics of the system whose each entry $P_{ij} \triangleq P(s_{t+1} = j|s_t = i, a_t = a)$ represents the conditional probability of the system moving to state $s_{t+1} = j$ in the next time step after taking an action a in the current state $s_t = i$. The dynamics are Markovian in the sense that the probability of the next state j depends only on the current state i and the action a, and not on any previous history. The reward function $r(s, a)$ assigns a real number to the state s the action a, so that $r(s, a)$ represents the immediate reward of being in state s and taking action a. A policy $\pi = \{d_1, d_2, \ldots\}$ is a sequence of decision rules. Each decision rule d_t is a mapping from states to actions at each time step: $d_t : S \to A$, and induces a corresponding transition probability matrix. The policy π is called stationary if its actions depend only on the state s, independent of time, i.e., $\pi = \{d, d, d, \ldots\}$. A stationary policy induces a time-invariant transition probability matrix. Every policy π is associated with a value function $V^\pi(s)$ such that $V^\pi(s)$ gives the expected cumulative reward achieved by π when starting in state s. The solution to an MDP problem is an optimal policy π^* that maximizes the expected cumulative reward over some finite or infinite number of time steps. The former and latter are termed finite-horizon MDP and infinite-horizon MDP, respectively. An infinite-horizon model has two typical forms of reward functions: the discounted and the average reward functions. The discounted reward function is defined as:

$$V^\pi_{dis}(s) = E^\pi_s \left\{ \sum_{t=1}^{\infty} \alpha^t r_t(s_t, a_t) \right\}, \tag{2}$$

where $0 < \alpha < 1$ denotes a given discount factor that provides convergence of $V^\pi(s)$, but also carries the notion of discounting the future reward, i.e., putting less emphasis on the rewards in the far future than those in the near future. The average reward function is defined as:

$$V^\pi_{ave}(s) = \lim_{N\to\infty} v_N, \tag{3}$$

where

$$v^\pi_N(s) = E^\pi_s \left[\sum_{t=1}^{N} r(s_t, a_t) \right]. \tag{4}$$

Under a number of conditions such as $r(s, a)$ is bounded and the "environment" is stationary, $V^\pi_{ave}(s)$ is finite. In such cases, there are many algorithms for finding an optimal policy. Since our analysis in this paper is on the classic Value Iteration (VI) algorithm for the infinite-horizon model with the average reward objective under "non-stationary environments", we briefly discuss the VI algorithm for the average reward objective.

B. Value Iteration Algorithm

The VI algorithm is an iterative algorithm for finding an ε-optimal policy for the infinite-horizon MDP. More precisely, given an ε, the VI algorithm guarantees to produce a reward value within an ε of the optimal value. The key to the VI algorithm is that each step of the algorithm can be viewed as applying a contracting operating L on v. Running the algorithm iteratively, or equivalently, applying the operator L repeatedly, will guarantee that v will converge to the optimal value based on Bellman equation. Specifically, For a unichain, at each iteration n, we have:

$$v_{n+1} = Lv_n,$$

where L is defined as:

$$Lv = \max_{d\in D}\{r_d + P_d v\},$$

r_d and P_d denote the reward and the transition probability matrix induced by the decision rule d. The pseudo-code for the VI algorithm with average reward objective is shown below.

Definition 1 (The Value Iteration): [1]. The algorithm for the Value Iteration with Average Reward Criteria is shown below:

- Choose any initial reward vector v_0, for a given $\varepsilon > 0$, n=0
- For each $s \in S$, we have:

$$v_{n+1}(s) = \max_{a\in A}\{r(s, a) + \sum_{j\in S} p(j|s, a)v^n(j)\}.$$

- Increasing n until $sp(v_{n+1} - v_n) < \varepsilon$, then choose:

$$d_\varepsilon \in \arg\max\{r(s, a) + \sum_{j\in S} p(j|s, a)v_n(j)\}.$$

where $sp(v) = \max_{s\in S} v(s) - \min_{s\in S} v(s)$ is the span seminorm of vector v.

We note that the ε-optimal policy approaches to an optimal policy as ε reduces to zero when the number of iterations goes to infinity.

III. Adiabatic Markov Decision Process and Value Iteration Algorithm

A. Adiabatic Setting

Typically, the VI algorithm is used to find an ε-optimal stationary policy in an offline manner using a number of iterations, assuming a stationary environment such that every stationary policy π induces a time-invariant transition probability matrix. The resulted policy is then used in an online manner with the assumption that the environment is stationary and governed by the time-invariant transition probability matrices in the VI algorithm. In this paper, we study an adiabatic MDP setting in which, we assume that the "environment" is no longer stationary. Instead, it might change at every iteration of the VI algorithm, resulting in a sequence of time-variant transition probability matrices under a stationary policy. The precise meaning of the "environment" will be clear shortly.

Instead of running the VI algorithm offline to find an ε-optimal policy, we apply the decision rule found after each iteration immediately and repeatedly in an online manner. Our goal is to determine how good the reward is for such a scheme. The analysis of such a setting is useful in the rapidly-changing environments where decisions must be made quickly. Unlike the traditional MDP setting where for each decision rule d, there is a time-invariant transition probability matrix P^d, in our setting, for a fixed decision rule d, there is sequence of time-variant transition probability matrices:

$$P_i^d = \Phi(i)P_0^d + (1-\Phi(i))P_f^d = P_f^d + \Phi(i)(P_0^d - P_f^d), \quad (5)$$

where Φ is a function such that:

- $\Phi(i) : [0, +\infty) \to [0, 1]$.
- $\Phi(0) = 1$.
- $\lim_{i\to\infty} \Phi(i) = 0$.

$\Phi(i)$ characterizes the change in the environment at the iteration i of the VI algorithm, and P_i^d is the induced transition probability matrix due to the decision rule found at the iteration i of the VI algorithm. A slowly changing $\Phi(i)$ implies a slow change in the environment. We note that the notion of optimal reward is not well defined if the environment fluctuates arbitrarily. Thus in the model above, we assume that the environment will approach to a final stationary environment characterized by P_f^d to ensure a well-defined reward. This can be seen as $\lim_{i\to\infty} P_i^d = P_f^d$.

We now articulate a bit more on the meaning of the "environment." We note that the induced P_i^d depends on both actions and environments. Therefore, a change in the environment implies possible changes in the underlying environments, or the set of actions, or the combination of both over time. For example, let us consider a queuing system in which the controller attempts to send packets (actions) at some varying rates based on the number of packets in the queue in order to maximize a given reward. In one scenario, we assume the traffic arrival rate at the queue increases steadily from a initial rate of λ_0 to a final rate of λ_f. As a result, P_i^d varies as the underlying environment changes over time. In another scenario, the arrival rate of the packets remains the same, however, the controller has inaccurate estimation of the arrival rate initially due to few observations. Consequently, it makes the decision on what rate it should send based on an inaccurate arrival rates, and P_i^d characterizes the change based on its decision rule d at iteration i. However, over time with more observations, its estimation of the arrival rate becomes more accurate. Therefore, its decision rule approaches the optimal one for which, the state dynamic is characterized by P_f^d. We will discuss these two examples in more detail in the later section.

B. Convergence Rate of Adiabatic MDP

It is important to emphasize again that the environment will be asymptotically stationary corresponding to an induced transition probability matrix P_f^d for any decision rule d. In addition, there exists an optimal decision rule d^* corresponding to a transition probability matrix $P_f^{d^*}$ which can be obtained when running the VI algorithm under a stationary environment [1]. Importantly, running the VI algorithm in an adiabatic setting for a sufficiently large number of steps would produce the same optimal decision rule d_f^* as that of the classic VI algorithm, and also the same average reward:

$$g^* = P_{d^*}^{\infty} r^{d_f^*} = \pi_f^{d^*} r^{d_f^*} e,$$

where by Cesaro mean,

$$P_{d^*}^{\infty} = \lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} (P_f^{d^*})^{n-1} = \lim_{n\to\infty} (P_f^{d^*})^n = \begin{bmatrix} \pi_f^{d_f^*} \\ \pi_f^{d_f^*} \\ \dots \\ \pi_f^{d_f^*} \end{bmatrix},$$

$\pi_f^{d_f^*}$ is the stationary distribution corresponding to $P_f^{d_f^*}$, $e = [11\dots1]^T$.

However, the convergence rates to this final reward g^* are quite different for the traditional MDP and adiabatic MDP settings. One would expect that the rate in the former setting would be faster since the environment does not change, thus the VI algorithm can learn faster than that of the latter. Therefore, our primary focus of this paper is to characterize the convergence rate of the VI algorithm in an adiabatic setting given the dynamics specified by $\Phi(i)$. Specifically, we want to find an integer N such that $\forall n > N$,

$$E = \left\| \frac{v_n}{n} - g^* \right\|_{\infty} \leq \varepsilon$$

Finding N is not trivial. Therefore, we will provide a lower a bound on N which depends on $\Phi(.)$ as well as the set of all possible matrices P_f^d.

IV. Main Results

We first give a number of definitions and small linear algebra results to help us establish the main result.

Definition 2 (Gamma coefficient): The gamma coefficient of a matrix is defined as follows [1]:

$$\gamma = \max_{s\in S, a\in A_s, s'\in S, a'\in A_{s'}} \left[1 - \sum_{j\in S} \min\{p(j|s,a), p(j|s',a')\} \right],$$

where A_s denote the set of possible actions can be taken in state s. From [6], the delta coefficient or Hajnal measure of a transition matrix P^d defined as $\gamma_d = \max_{s\in S, s'\in S}\left[1-\sum_{j\in S}\min\{p^d(j|s), p^d(j|s')\}\right]$ is an upper bound on the second largest eigenvalue modulus (SLEM) λ^* of the matrix P^d. Easily, we can see the gamma coefficient is the maximum value of all γ_d over the set of all decision rules. Therefore, the gamma coefficient is an upper bound of SLEM λ^* of all transition matrices P^d for all decision rule d.

Proposition 1: [1]. For any reward vectors u, v,

$$sp(Lu - Lv) \leq \gamma sp(u - v).$$

Proposition 2: [1]. For any reward vectors v and any decision rule d,

$$sp(P^d v) \leq \gamma sp(v).$$

Proposition 3 (The bound on gamma coefficients): Given $P_i^d = \Phi(i)P_0^d + (1-\Phi(i))P_f^d, \Phi(i) \in [0,1]$ for all $d \in D$:

$$\gamma_i \leq \Phi(i)\gamma_0 + (1-\Phi(i))\gamma_f.$$

Proof: We omit the proof due to limited space. ■

Corollary 1: Given $P_i^d = \Phi(i)P_0^d + (1-\Phi(i))P_f^d$ with decreasing function $\Phi(i) > 0$, for all $d \in D$, then for any $i \geq n_0$:

$$\gamma_i \leq \max(\gamma_{n_0}, \gamma_f). \tag{6}$$

Proof: We omitted the proof due to limited space. ■

Note that one way to ensure that $0 < \gamma < 1$ is that for each decision rule d, there exists a column of the corresponding P with all positive entries.

Theorem 1 (Main Result 1): Consider a unichain adiabatic-time MDP with S and A_s are both finite, $|r(s,a)|$ is bounded by a number M. Suppose $0 < \gamma = \max(\gamma_f, \gamma_{n_0}) < 1$ and $e_{n_0} < \infty$, then

$$E \leq \frac{||v_{n_0} - n_0 g^*||_\infty}{n} + \frac{1}{n}\sum_{i=n_0}^{n-1}\left[\left(\prod_{k=n_0}^{i-1}\gamma_k\right) sp(v_{n_0+1} - v_{n_0}) + YM\left(e_i + 2\sum_{j=n_0+1}^{i} sp(v_j)|\Delta\Phi_{j-1}|(\prod_{k=j}^{i-1}\gamma_k)\right)\right], \tag{7}$$

for any n_0, where $\prod_{k=u}^{v}(\cdot) = 1$ if $v < u$, $e_i = \sum_{j=i+1}^{\infty}(sp(v_j)|\Delta\Phi_{j-1}|), 2Y = \max_{d\in D}\|P_0 - P_f\|_\infty$.

Proof: From (5), for any $d \in D$ we have:

$$\begin{aligned} \Rightarrow ||P_i^d - P_{i-1}^d||_\infty &\leq |\Phi(i) - \Phi(i-1)|||P_0^d - P_f^d||_\infty, \\ &\leq 2Y|\Delta\Phi_{i-1}|, \\ &\text{where } 2Y = \max_{d\in D}\|P_0 - P_f\|_\infty. \end{aligned} \tag{8}$$

Consider $E = \left\|\frac{v_n}{n} - g^*\right\|_\infty$ where $v_{i+1} = L_i v_i$:

$$\begin{aligned} E &= \left\|\sum_{i=n_0}^{n-1}\left(\frac{v_{i+1} - v_i - g^*}{n}\right) + \frac{v_{n_0} - n_0 g^*}{n}\right\|_\infty, \\ &\leq \sum_{i=n_0}^{n-1}\frac{\|v_{i+1} - v_i - g^*\|_\infty}{n} + \frac{||v_{n_0} - n_0 g^*||_\infty}{n}. \end{aligned}$$

Firstly, we bound $\|v_{i+1} - v_i - g^*\|_\infty$.
Let $x_i = \arg\max_{s\in S}(L_i v_i - L_{i-1}v_{i-1})$, $y_i = \arg\min_{s\in S}(L_i v_i - L_{i-1}v_{i-1})$.
Let d_i^*, d_{i-1}^* be the optimal decision rule corresponding to the operator L_i, L_{i-1}, respectively. Then,

$$\begin{aligned} L_i v_i(x_i) - L_{i-1}v_{i-1}(x_i) &\leq L_i^{d_i^*} v_i(x_i) - L_{i-1}^{d_i^*} v_{i-1}(x_i), \\ &= P_i^{d_i^*} v_i(x_i) - P_{i-1}^{d_i^*} v_{i-1}(x_i), \\ &= (P_i^{d_i^*} - P_{i-1}^{d_i^*})v_i(x_i) + P_{i-1}^{d_i^*}(v_i(x_i) - v_{i-1}(x_i)). \end{aligned}$$

where L^d is the operator that we apply the decision rule at that step: $L^d v = r_d + P_d v$.

Let $\alpha_i = \arg\max_{s\in S}(v_i), \beta_i = \arg\min_{s\in S}(v_i), \Delta P_i^{d_i^*} = P_i^{d_i^*} - P_{i-1}^{d_i^*}$. Since $P_i^{d_i^*}, P_{i-1}^{d_i^*}$ are stochastic matrices, then $\sum_{s\in S}(\Delta P_i^{d_i^*}(x_i, s)) = 0$. Let $a = \Delta P_i^{d_i^*} v_i(x_i) = (\Delta P_i^{d_i^*}(x_i, \cdot))v_i = (\Delta P_i^{d_i^*}(x_i, \cdot))(v_i - v_i(\beta_i)e)$ where $e = [11\ldots1]^T$. Therefore:

$$\begin{aligned} a &= \sum_{s\in S}\Delta P_i^{d_i^*}(x_i, s))(v_i(s) - v_i(\beta_i)), \\ &\leq \sum_{\Delta P_i^{d_i^*}(x_i,s))\geq 0}\Delta P_i^{d_i^*}(x_i, s))(v_i(s) - v_i(\beta_i)), \\ &\leq (v_i(\alpha_i) - v_i(\beta_i))\sum_{\Delta P_i^{d_i^*}(x_i,s))\geq 0}\Delta P_i^{d_i^*}(x_i, s)), \\ &\leq (v_i(\alpha_i) - v_i(\beta_i))\frac{||P_i^{d_i^*} - P_{i-1}^{d_i^*}||_\infty}{2}, \\ &\leq Y|\Delta\Phi_{i-1}|sp(v_i) \qquad \text{from (8).} \end{aligned}$$

Hence,

$$\begin{aligned} L_i v_i(x_i) - L_{i-1}v_{i-1}(x_i) \leq YMsp(v_i)|\Delta\Phi_{i-1}| + \\ +P_{i-1}^{d_i^*}(v_i(x_i) - v_{i-1}(x_i)). \end{aligned} \tag{9}$$

Similarly,

$$\begin{aligned} L_i v_i(y_i) - L_{i-1}v_{i-1}(y_i) \geq -YMsp(v_i)|\Delta\Phi_{i-1}| + \\ +P_{i-1}^{d_{i-1}^*}(v_i(y_i) - v_{i-1}(y_i)). \end{aligned} \tag{10}$$

Since $b = P_{i-1}^{d_i^*}(v_i(x_i) - v_{i-1}(x_i)) \leq v_i(x_{i-1}) - v_{i-1}(x_{i-1})$, then $L_i v_i(x_i) - L_{i-1}v_{i-1}(x_i) \leq YMsp(v_i)|\Delta\Phi_{i-1}| + v_i(x_{i-1}) - v_{i-1}(x_{i-1})$. Similarly, $L_i v_i(y_i) - L_{i-1}v_{i-1}(y_i) \geq -YMsp(v_i)|\Delta\Phi_{i-1}| + v_i(y_{i-1}) - v_{i-1}(y_{i-1})$.

Now, by keep expanding, we have:
$v_{t+1}(x_t) - v_t(x_t) = L_t v_t(x_t) - L_{t-1}v_{t-1}(x_t) \leq YM\sum_{j=i+1}^{t}(sp(v_j)|\Delta\Phi_{j-1}|) + (v_{i+1}(x_i) - v_i(x_i))$.
Let $t \to \infty$. When $t = \infty$, we know exactly P_f and run VI for it, we will have a reward received at one time step $\lim_{t\to\infty}(v_{t+1}(x_t) - v_t(x_t)) = g^*$ which is the optimal average reward [1]. Therefore,
$g^* \leq YM\sum_{j=i+1}^{\infty}(sp(v_j)|\Delta\Phi_{j-1}|) + (v_{i+1}(x_i) - v_i(x_i))$ for any i.

Similarly,
$g^* \geq -YM\sum_{j=i+1}^{\infty}(sp(v_j)|\Delta\Phi_{j-1}|) + (v_{i+1}(y_i) - v_i(y_i))$ for any i.
Let $e_i = \sum_{j=i+1}^{\infty}(sp(v_j)|\Delta\Phi_{j-1}|)$ which represents the total

error from the time step i to ∞. This error comes from the fact that we use the matrix P_i^d matrix at each time step instead of P_f^d for all d.

Using $\infty-$norm,

$$\begin{aligned}\|v_{i+1}-v_i-g^*\|_\infty &\leq YMe_i+(v_{i+1}(x_i)-v_i(x_i))\\ &\quad -(v_{i+1}(y_i)-v_i(y_i)),\\ &\leq sp(v_{i+1}-v_i)+YMe_i.\end{aligned}$$

Now, we upper bound $sp(v_{i+1}-v_i)$. From (9), (10), we have:

$$\begin{aligned}sp(v_{i+1}-v_i) &= (L_iv_i(x_i)-L_{i-1}v_{i-1}(x_i))-\\ &\quad -(L_iv_i(y_i)-L_{i-1}v_{i-1}(y_i)),\\ &\leq 2YMsp(v_i)|\Delta\Phi_{i-1}|+\\ &\quad +P_{i-1}^{d_i^*}(v_i(x_i)-v_{i-1}(x_i))-\\ &\quad -P_{i-1}^{d_{i-1}^*}(v_i(y_i)-v_{i-1}(y_i)),\\ &\leq 2YMsp(v_i)|\Delta\Phi_{i-1}|+\\ &\quad +sp([P_{i-1}^{d_i^*}/P_{i-1}^{d^{i-1}}](v_i-v_{i-1})),\\ &\leq 2YMsp(v_i)|\Delta\Phi_{i-1}|+\\ &\quad +\gamma_{i-1}sp(v_i-v_{i-1}). \qquad (11)\end{aligned}$$

where $[P1/P2]$ denotes the stacked matrix in which the rows of $P1$ follow the rows of $P2$. Based on the Definition 2, the gamma coefficient of the set of stacked matrices at time step $i-1$ is at most γ_{i-1}. (11) is similar to Proposition 1 except there is an error $2YMsp(v_i)|\Delta\Phi_{i-1}|$ which goes to 0 when $i\to\infty$.

Since $0<\gamma_i\leq\gamma=\max(\gamma_{n_0},\gamma_f)<1$, $sp(v_{i+1}-v_i)\leq(\prod_{k=n_0}^{i-1}\gamma_k)sp(v_{n_0+1}-v_{n_0})+2YM\left(\sum_{j=n_0+1}^{i}sp(v_j)|\Delta\Phi_{j-1}|(\prod_{k=j}^{i-1}\gamma_k)\right)$ for all $i\geq n_0$. Then,
$\sum_{i=n_0}^{n-1}\frac{\|v_{i+1}-v_i-g^*\|_\infty}{n}\leq\frac{1}{n}\sum_{i=n_0}^{n-1}[sp(v_{i+1}-v_i)+YMe_i]$
$\sum_{i=n_0}^{n-1}\frac{\|v_{i+1}-v_i-g^*\|_\infty}{n+1}\leq\frac{1}{n}\sum_{i=n_0}^{n-1}\left[(\prod_{k=n_0}^{i-1}\gamma_k)sp\left(v_{n_0+1}-v_{n_0}\right)+YM\left(e_i+2\left(\sum_{j=n_0+1}^{i}sp(v_j)|\Delta\Phi_{j-1}|(\prod_{k=j}^{i-1}\gamma_k)\right)\right)\right].$

Therefore, we have an upper bound of A as follows:

$$E\leq\frac{||v_{n_0}-n_0g^*||_\infty}{n}+\frac{1}{n}\sum_{i=n_0}^{n-1}\left[\left(\prod_{k=n_0}^{i-1}\gamma_k\right)sp(v_{n_0+1}-v_{n_0})+YM\left(e_i+2\sum_{j=n_0+1}^{i}sp(v_j)|\Delta\Phi_{j-1}|(\prod_{k=j}^{i-1}\gamma_k)\right)\right].$$

■

Theorem 2 (Main Result 2): Consider a unichain adiabatic-time MDP with S and A_s are both finite, $|r(s,a)|$ is bounded by a number M. Suppose $0<\gamma=\max(\gamma_f,\gamma_{n_0}),\gamma'=\max(\gamma_f,\gamma_0)<1$ and $\Phi(i)$ is a positive decreasing function on $[n_0,+\infty)$, then for

$$n\geq\frac{2}{\varepsilon}\left(n_0M+||v_0||_\infty+\frac{\varepsilon}{1-\gamma}+\frac{M+(1+\gamma)\left[\frac{M(1-(\gamma')^{n_0})}{1-\gamma'}+(\gamma')^{n_0}(sp(v_0))\right]}{1-\gamma}\right), \qquad (12)$$

we guarantee: $E=\left\|\frac{v_n}{n}-g^*\right\|_\infty<\varepsilon$,
where $2Y=\max_{d\in D}\|P_0-P_f\|_\infty$, n_0 is the smallest integer satisfying $\left[\frac{M}{1-\gamma'}+\gamma'^{(n_0)}sp(v_0)\right]\Phi(n_0)\leq\frac{\varepsilon}{2YM}$.

Proof: Suppose we can find n_0 so that $e_{n_0}<\frac{\varepsilon}{2YM}$. We will show how to find n_0 later. By applying the Theorem 1 with n_0 and the Corollary 1,
$E\leq\frac{||v_{n_0}-n_0g^*||_\infty}{n}+\frac{1}{n}\frac{sp(v_{n_0+1}-v_{n_0})}{1-\gamma}+\frac{1}{n}\sum_{i=n_0}^{n-1}YM\left(e_i+2\sum_{j=n_0+1}^{i}(\gamma^{i-j}sp(v_j)|\Delta\Phi_{j-1}|)\right)$.

We have the following facts:

1) $||v_{n_0}-n_0g^*||_\infty\leq||r_{d_{n_0-1}^*}+P_i^{d_{n_0-1}^*}v_{n_0-1}-n_0g^*||_\infty\leq||(r_{d_{n_0-1}^*}-g^*)||_\infty+||P_i^{d_{n_0-1}^*}(v_{n_0-1}-(n_0-1)g^*)||_\infty\leq||(r_{d_{n_0-1}^*}-g^*)||_\infty+||(v_{n_0-1}-(n_0-1)g^*)||_\infty\leq\sum_{k=1}^{n_0}||r_{d_{k-1}^*}-g^*||_\infty+||v_0||_\infty\leq n_0\max(M-g^*,g^*)+||v_0||_\infty\leq n_0M+||v_0||_\infty,$

2) $sp(v_{n_0+1}-v_{n_0})=sp(r_{d_{n_0}^*}+P_{n_0}^{d_{n_0}^*}v_{n_0}-v_{n_0})\leq sp(r_{d_{n_0}^*})+sp(P_{n_0}^{d_{n_0}^*}v_{n_0})+sp(v_{n_0})\leq M+(1+\gamma)sp(v_{n_0})$ (from Proposition 2).
Now, since $v_i=r^{d_{i-1}^*}+P_{i-1}^{d_{i-1}^*}v_{i-1}$ and $\gamma_i\leq\gamma',\forall i\geq 0$, then

$$\begin{aligned}sp(v_i) &\leq \left[sp(r^{d_{i-1}^*})+sp(P_{i-1}^{d_{i-1}^*}v_{i-1})\right],\\ &\leq [M+\gamma_{i-1}sp(v_{i-1})] \quad \text{(Proposition 2)},\\ &\leq M\left[1+\sum_{j=1}^{i-1}\prod_{k=1}^{j}\gamma_{i-k}\right]+\prod_{k=1}^{i}\gamma_{i-k}(sp(v_0)),\\ &\leq \frac{M(1-(\gamma')^i)}{1-\gamma'}+(\gamma')^i(sp(v_0)).\end{aligned}$$

Then, $sp(v_{n_0+1}-v_{n_0})\leq M+(1+\gamma)sp(v_{n_0})\leq M+(1+\gamma)\left[\frac{M(1-(\gamma')^{n_0})}{1-\gamma'}+(\gamma')^{n_0}(sp(v_0))\right]$.

3) Let $y_i=\sum_{j=n_0+1}^{i}(\gamma^{i-j}sp(v_j)|\Delta\Phi_{j-1}|)$. We have:

$$\begin{aligned}y_{n_0+1} &= sp(v_{n_0+1})\Delta\Phi_{n_0},\\ y_{n_0+2} &= \gamma y_{n_0+1}+sp(v_{n_0+2})\Delta\Phi_{n_0+1},\\ &\cdots\\ y_n &= \gamma y_{n-1}+sp(v_n),\Delta\Phi_{i-1}\\ &\cdots\end{aligned}$$

Then, $\sum_{i=n_0}^{\infty}y_i=\frac{e_{n_0}}{1-\gamma}$.

4) Now, we find conditions on n_0 so that $e_{n_0}=\sum_{j=n_0+1}^{\infty}(sp(v_j)|\Delta\Phi_{j-1}|)\leq\frac{\varepsilon}{2YM}$. Since $sp(v_j)\leq\frac{M(1-(\gamma')^j)}{1-\gamma'}+(\gamma')^jsp(v_0)\leq\frac{M}{1-\gamma'}+\gamma'^{(n_0)}sp(v_0)$, for all $j>n_0$,

$$\begin{aligned}e_{n_0} &\leq \sum_{j=n_0+1}^{\infty}(sp(v_j)|\Delta\Phi_{j-1}|),\\ &\leq \left[\frac{M}{1-\gamma'}+(\gamma')^{n_0}sp(v_0)\right]\sum_{j=n_0+1}^{\infty}(|\Delta\Phi_{j-1}|),\\ &\leq \left[\frac{M}{1-\gamma'}+(\gamma')^{n_0}sp(v_0)\right]\Phi(n_0).\end{aligned}$$

since Φ_i is decreasing, $i \geq n_0$, then $|\Delta\Phi_{j-1}| = \Phi(j-1) - \Phi(j)$. Easily, we can see $\frac{M}{1-\gamma'} + (\gamma')^{n_0} sp(v_0)$ is bounded. Therefore, there exists n_0 so that $e_{n_0} \leq \left[\frac{M}{1-\gamma'} + (\gamma')^{n_0} sp(v_0)\right] \Phi(n_0) \leq \frac{\varepsilon}{2YM}$. Then for all $i \geq n_0$, $e_i \leq e_{n_0} \leq \frac{\varepsilon}{2YM}$.

Now, for $n \geq n_0$,

$$\begin{aligned} E &\leq \frac{||v_{n_0} - n_0 g^*||_\infty}{n} + \frac{1}{n}\frac{sp(v_{n_0+1} - v_{n_0})}{1-\gamma} \\ &\quad + \frac{YM\sum_{i=n_0}^{n-1} e_i}{n} + \frac{1}{n} 2YM \frac{e_{n_0}}{1-\gamma}, \\ &\leq \frac{1}{n}\Bigg(n_0 M + ||v_0||_\infty + \\ &\quad \frac{M + (1+\gamma)\left[\frac{M(1-(\gamma')^{n_0})}{1-\gamma'} + (\gamma')^{n_0}(sp(v_0))\right]}{1-\gamma} + \\ &\quad \frac{\varepsilon}{1-\gamma}\Bigg) + \frac{\varepsilon}{2}. \end{aligned}$$

Let $\frac{1}{n}\left(\frac{M+(1+\gamma)\left[\frac{M(1-(\gamma')^{n_0})}{1-\gamma'}+(\gamma')^{n_0}(sp(v_0))\right]}{1-\gamma} + + n_0 M + \right.$

$$\left. + ||v_0||_\infty + \frac{\varepsilon}{1-\gamma}\right) \leq \frac{\varepsilon}{2},$$

or

$$n \geq \frac{2}{\varepsilon}\left(n_0 M + ||v_0||_\infty + \frac{\varepsilon}{1-\gamma} + \right.$$
$$\left. + \frac{M + (1+\gamma)\left[\frac{M(1-(\gamma')^{n_0})}{1-\gamma'} + (\gamma')^{n_0}(sp(v_0))\right]}{1-\gamma}\right).$$

Then, $E \leq \varepsilon$. ■

As shown above, the gamma coefficient of a matrix is an upper bound of the second largest eigenvalue modulus λ^*. Then, the term $\frac{1}{1-\gamma}$ is a upper bound of the relaxation time $t_{rel} = \frac{1}{1-\lambda*}$ of P_f^d for all $d \in D$. Moreover, the relaxation time is proportional to the the mixing time of P_f^d or the convergence rate of the corresponding Markov chain [7]. Therefore, the convergence rate of the Adiabatic-Time MDP is proportional to the convergence rate of a Markov chain with $P_f^{d_f^*}$. This is intuitively plausible that when n is large, the environment becomes approximately stationary under the decision rule d_f^*.

V. Applications

A. Application to Queuing System with Progressive Arrival Rate Estimation

Consider an M/M/1/K queue with a unknown packet arrival rate λ per unit time. We estimate λ at time $i\Delta t$ denoted as $\hat{\lambda}_i$ and decide the packet departure rate, $\mu_i = f(\hat{\lambda}_i)$ as follows:

$$\begin{aligned} \hat{\lambda}_i &= \frac{1}{i\Delta t}\sum_{k=1}^{i} X_k, \\ \mu_i &= f(\hat{\lambda}_i) = (1+\delta_i)\hat{\lambda}_i, \end{aligned}$$

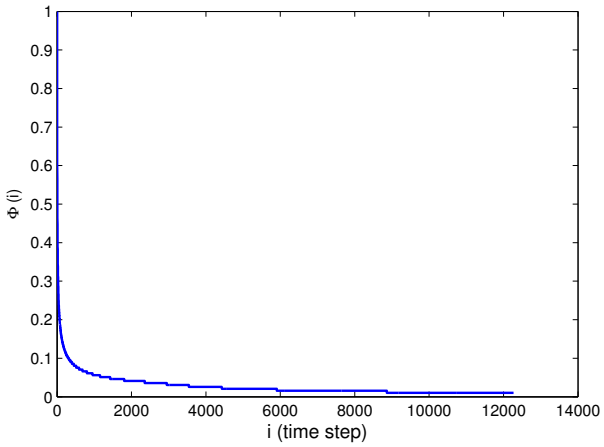

Figure 1. The $\Phi(i)$ function.

where $X_k \sim \text{Poisson}(\lambda\Delta t)$ is the number of packets in the kth slot of duration Δt and $\delta_i > 0$ is an action we choose from a set of constant numbers.

We define the state s as the number of packets in the queue, therefore, $s \in S = \{0, 1, 2, \ldots, K-1, K\}$. Let d be the decision rule which maps each state to a value, i.e., $d_i(s) = \delta_i(s)$. The generator matrix in time interval $(i\Delta t, (i+1)\Delta t]$ is:

$$Q_i = \begin{pmatrix} -\lambda_i & \lambda_i & 0 & 0 & \ldots \\ \mu_i & -(\mu_i+\lambda_i) & \lambda_i & 0 & \ldots \\ \ddots & \ddots & \ddots & \ddots & \ddots \\ \ldots & 0 & \mu_i & -(\mu_i+\lambda_i) & \lambda \\ \ldots & 0 & 0 & \mu_i & -\mu_i \end{pmatrix}. \tag{13}$$

The corresponding transition probability matrix $P(i\Delta t, (i+1)\Delta t)$:

$$P_i = P(i\Delta t, (i+1)\Delta t) = e^{Q_i \Delta t}. \tag{14}$$

If we only care about the delay of packets in the queue and the cost of implementing fast serving rates, we can set the immediate reward: $r(s,\delta) = -\frac{M(s-K\delta)^2}{\max_{s,\delta}(s-K\delta)^2}$. Our goal is to maximize the average reward $\frac{1}{N}E\left[\sum_{t=1}^{N} r(s_t, a_t)\right]$

Using Chernoff bound, we can find a number a_i so that with probability at least $1-2\alpha, (1-a_i)\lambda \leq \hat{\lambda}_i \leq (1+a_i)\lambda$. Based on that, we can find $\Phi(i) = \frac{|P_i - P_f|_\infty}{|P_0 - P_f|_\infty}$ which is now a function of a_i. Easily, we can see that a_i decreases to 0 when i goes to ∞. Thus, $\Phi(i)$ is a decreasing function satisfying our conditions (5) on $\Phi(i)$ as shown in the Figure 1.

Now, we verify the theoretical bound provided by Theorems 1 and 2 via simulation using the following parameters: $\varepsilon = 0.01, \lambda = 40, \Delta_t = 1, \alpha = 0.1, M = 1, K = 10, A = \{0.2, 0.4, 0.6, 0.8\}, v_0 = 0e$. The function $\Phi(i)$ is shown on the Figure 1. From the Theorem 2, we obtained

$$n_0 = 58, N = 12270.$$

Figure 2 shows the actual the distance to the optimal reward obtained from the VI algorithm and its upper bound by

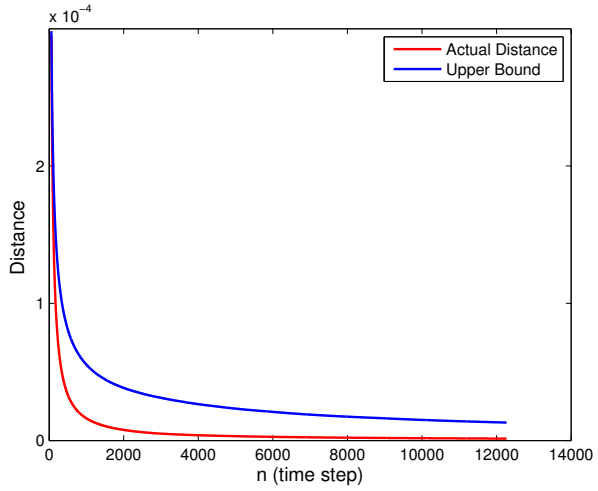

Figure 2. The simulated distance to the reward and its upper bound.

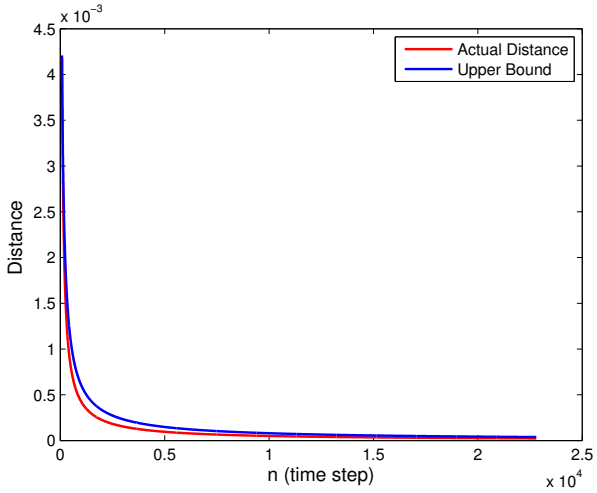

Figure 3. The simulated distance to the otimal reward and its upper bound.

Theorem 2. As seen, they are well correlated with each other.

B. Bernoulli Discrete Queuing System with Underlying Time-variant Environment

In this section, we show a toy example illustrating the application of our framework for the time-varying underlying environment. Specifically, we consider a Bernoulli queuing system of size $K = 2$. Assume at each time step, there are probabilities p and q that a packet will be arriving and departing the queue, respectively. In this case, the state space $S = \{0, 1, 2\}$, the time-varying environment is described by changing the values of p over time, while the action is the value of q. The transition matrix has the following form:

$$P = \begin{bmatrix} 1-p(1-q) & p(1-q) & 0 \\ q(1-p) & pq+(1-p)(1-q) & p(1-q) \\ 0 & q(1-p) & 1-q(1-p) \end{bmatrix}$$

Because all entries in the matrix P are linear functions of p, if we set $p_i = \phi(i)p_0 + (1-\phi(i))p_f$ to model the change in the arrival rates, then:

$$P_i = \Phi(i)P_0 + (1-\Phi(i))P_f$$

where $\Phi(i) = \phi(i)$ for all i. Note that $\Phi(i)$ satisfies the conditions for the adiabatic setting. We also choose the reward function shown in the Table I.

To examine the theoretical results, we run simulation using

$r(s,a)$		a								
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
s	0	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1
	1	0.1	0.2	0.3	0.4	0.5	0.4	0.3	0.2	0.1
	2	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9

Table I
THE REWARD $r(s,a)$

the following parameters: $\varepsilon = 0.1, p_0 = 0.4, p_f = 0.6, \phi(i) = \frac{1}{i+1}, A = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}, v_0 = 0e$.

From the Theorem 2, we obtained $n_0 = 82, N = 22801$. Figure 3 shows the simulated distance to the optimal reward obtained from the VI algorithm and its upper bound predicted by our main results. As seen, they are very correlated.

VI. CONCLUSION

We provide an analysis framework for studying the VI algorithm under the adiabatic setting. We provide theoretical bounds on the convergence rate of the VI algorithm with the average reward objective. Specifically, our work provide a lower bound on the number of time iterations in the VI algorithm needed in order to ensure that the resulted policy produces an average reward that is ε-close to the optimal average reward value.

REFERENCES

[1] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic, Dynamic Programming*, 3rd ed. John Wiley & Sons, 2005.

[2] R. Bellman, *Dynamic Programming*. Priceton University Press, 1957.

[3] M. Born and V.Fork, "Beweis des adiabatensatzes," *Zeitschrift Fur Physik A Hadrons and Nuclei*, vol. 51, pp. 165–180, 1928.

[4] A. Messiah, *Quantum mechanics*, 1st ed. John Wiley & Sons, 1962, vol. 2.

[5] Y. Kovchegov, "A note on adiabatic theorem for markov chains," *Statistics and Probability Letters*, vol. 80, pp. 186–190, February 2010.

[6] E. Seneta, *Non-negative Matrices and Markov Chains*. Springer-Verlag, 1981.

[7] D. A.Levin, Y. Peres, and E. L.Wilmer, *Markov Chains and Mixing Times*. American Mathematical Society, 2008.