2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

On Joint Convergence of Traffic State and Weight Vector in Learning-Based Dynamic Routing with Value Function Approximation

Yidan Wu, Jianan Zhang and Li Jin

Abstract— **Learning-based approaches are increasingly popular for traffic control problems, but they are applied typically as black boxes with limited theoretical guarantees and interpretability. In this paper, we address these challenges by analyzing dynamic routing over parallel servers, a representative traffic control task, through a semi-gradient on-policy control algorithm, a key reinforcement learning method. We consider a linear value function approximation on an unbounded state space and derive a Lyapunov function from the approximator. In particular, the structure of the approximator naturally enables idling policies, which is an interesting and useful advantage over existing dynamic routing schemes. Our results demonstrate that the convergence of the approximation weights is coupled with the convergence of the traffic state. Specifically, we show that if the system is stabilizable, then (i) the weight vector converges to a bounded region, and (ii) the traffic state is bounded in the mean. Additionally, empirical evidence shows that our proposed algorithm is computationally efficient with an insignificant optimality gap, which is effectively practical in real-world applications.**

Index terms: Dynamic routing, reinforcement learning, Lyapunov method, value function approximation.

I. Introduction

Dynamic routing is a classical control problem in transportation, manufacturing, and networking. This problem was conventionally challenging, because analytical characterization of the steady-state distributions of the traffic state and thus of the long-time performance metrics (e.g., queuing delay) are very difficult [1], [2]. Recently, there is a rapidly growing interest in applying reinforcement learning (RL) methods to dynamic routing and network control problems in general. RL methods are attractive because of their computational efficiency and adaptivity to unknown/non-stationary environments [3]. However, there is still a non-trivial gap between the demand for theoretical guarantees on key performance metrics and the black-box nature of RL methods. In particular, most existing theoretical results on RL are developed in the context of finite Markov decision processes (MDPs), while dynamic routing may be considered in infinite state spaces, especially for stability and throughput analysis.

In this paper, we make an effort to respond to the above challenge by studying the behavior of a parallel service system (Fig. 1) controlled by a class of semi-gradient SARSA (SGS) algorithms with linear function approximation; these methods are attractive because of (i) adaptivity to unknown model parameters and (ii) potential to obtain near-optimal policies. Importantly, we jointly consider the convergence of the algorithm training process and of the traffic state process. The specific research questions are:

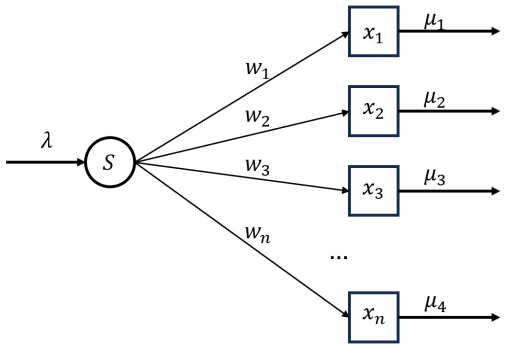

Fig. 1: A parallel service system.

1) How is the convergence of the weight vector coupled with the convergence of the traffic state?
2) Under what conditions does the proposed SGS algorithm ensure the joint convergence of the weights and the state?

The above questions involve two bodies of literature, viz. dynamic routing and reinforcement learning. Classical dynamic routing schemes rely on Lyapunov methods to study traffic stability and provide a solid foundation for our work [4], [5]. However, these methods are less powerful to search for routing policies that optimizing average system time. In particular, existing results usually assume non-idling policies, which may be quite restrictive. Recently, RL is used for finding optimal routing policies and provides important tools and insights for practice [6]. In particular, Liu et al. proposed a mixed scheme that uses learning over a bounded set of states and uses a known stabilizing policy for the other states [7]; Xu et al. proposed a deep RL-based framework for traffic engineering problems in communication networks [8]; Lin et al. proposed an adaptive routing algorithm utilizing RL in a hierarchical network [9]. However, existing theory on RL mostly, to the best of our knowledge, considers MDPs with finite or bounded state spaces [10], [11], [12]; the theory on infinite/unbounded state spaces is limited to very special problems (e.g., linear-quadratic regulation [13]). Hence, existing learning-based routing algorithms either rely on empirical evidence for

This work was in part supported by NSFC Project 62103260, SJTU UM Joint Institute, J. Wu & J. Sun Foundation, and US NSF CMMI-1949710.

Y. Wu and L. Jin are with the UM Joint Institute, Shanghai Jiao Tong University, China. J. Zhang is with the School of Electronics, Peking University, China. (Emails: wyd510@sjtu.edu.cn, li.jin@sjtu.edu.cn, zhangjianan@pku.edu.cn)

convergence or build on finite MDP theories. Thus, there is a lack of solid theory on the convergence of value function approximation over unbounded state spaces; such a theory is essential for developing interpretable and reliable routing schemes.

In response to the above research gaps, we jointly consider the convergence of traffic state and of the weight vector in dynamic routing. The traffic state characterizes the queuing process in the parallel service system illustrated in Fig. 1. The routing objective is to minimize the expected total system time, which includes waiting times and service times. The weights parameterize the approximate action value function, and thus determine the routing policy. In particular, the algorithm naturally makes possible idling policies, which is an advantage over existing methods. The weights are updated by a semi-gradient on-policy algorithm.

Our main result (Theorem 1) states that the proposed algorithm ensures joint convergence of the traffic state and the weight vector if and only if the system is stabilizable. Importantly, we study the coupling between the long-time behavior of the traffic state and that of the weight vector, which extends the existing theory on finite-state RL [11] to unbounded state spaces. The convergence of traffic state results from a Lyapunov function associated with the approximate value function [14] which verifies the drift criterion [15]. The convergence of weights results from stochastic approximation theory [16]. We compare the proposed algorithm with a much more sophisticated neural network-based algorithm and show that our algorithm converges much faster than the benchmark with only an 8% optimality gap.

In summary, the contributions of this paper are as follows.

1) We propose a structurally intuitive and technically sound algorithm to learn near-optimal routing policies over parallel servers.
2) We study joint convergence of traffic state and weight vector under the proposed algorithm; this is theoretically interesting in itself.
3) We show empirical evidence for the computational efficiency and near-optimality of the proposed algorithm.

The rest of this paper is organized as follows. Section II introduces the parallel service system model, the MDP formulation, and the SGS algorithm. Section III presents and develops the main result on joint convergence. Section IV compares the SGS algorithm with two benchmarks. Section V gives the concluding remarks.

II. Modeling and Formulation

Consider the system of parallel servers with infinite buffer sizes in Fig. 1. In this section, we model the dynamics of the system, formulate the dynamic routing problem as a Markov decision process (MDP), and introduce our semi-gradient SARSA (SGS) algorithm.

A. System modeling

Let $\mathcal{N} = \{1, 2, 3, \dots, N\}$ be the set of parallel servers. Each server n has an exponentially distributed service rate μ_n and job number $x_n(t)$ at time $t \in \mathbb{R}_{\geq 0}$. The state of the system is $x = [x_1, x_2, \dots, x_N]^T$, and the state space is $\mathbb{Z}_{\geq 0}^N$. Jobs arrive at origin S according to a Poisson process of rate $\lambda > 0$. When a job arrives, it will go to one of the N servers according to a routing policy

$$\pi : \mathcal{N} \times \mathbb{Z}_{\geq 0}^N \to [0, 1].$$

That is, $\pi(a|x)$ is the probability of routing the new job to server a conditional on state x. This paper focuses on a particular class of routing policies which we call the *weighted shortest queue* (WSQ) policy. WSQ is based on the approximation for the action value function $\hat{Q} : \mathcal{N} \times \mathbb{Z}_{\geq 0}^N \times \mathbb{R}_{>0}^N \to \mathbb{R}_{\geq 0}$ defined as:

$$\hat{Q}(x, a; w) := \sum_{n=1}^{N} w_n (x_n + \mathbb{I}_{\{n=a\}})^2, \tag{1}$$

where $w = [w_1, w_2, \dots, w_N]^T \in \mathbb{R}_{>0}^N$ is the weight vector. For technical convenience, we consider the softmax version of WSQ

$$\pi_w(a|x) = \frac{\exp(-\hat{Q}(x, a; w)/\iota)}{\sum_{b=1}^{N} \exp(-\hat{Q}(x, b, w)/\iota)}, \tag{2}$$

where $\iota \in (0, \infty)$ is the temperature of the softmax function. Note that $\pi_w(a|x)$ converges to a deterministic policy greedy w.r.t. $\hat{Q}$ as ι approaches 0 [11].

We say that the traffic in the system is *stable* if there exists a constant $M < \infty$ such that for any initial condition,

$$\lim_{t \to \infty} \frac{1}{t} \int_{s=0}^{t} \mathsf{E}[\|x(s)\|_1] ds < M, \tag{3}$$

which indicates that the length of the queues remains bounded in the mean for all time [17].

We say that the system is *stabilizable* if

$$\lambda < \sum_{n=1}^{N} \mu_n. \tag{4}$$

Note that the above ensures the existence of at least a stabilizing Bernoulli routing policy.

B. MDP formulation

Since routing actions are made only at transition *epochs* [18, p.72], the routing problem of the parallel queuing system can be formulated as a discrete-time (DT) MDP with countably infinite state space $\mathbb{Z}_{\geq 0}^N$ and finite action space $\mathcal{N}$. With a slight abuse of notation, we denote the state and action of the DT MDP as $x[k] \in \mathbb{Z}_{\geq 0}^N$ and $a[k] \in \mathcal{N}$, respectively. Specifically, $x[k] = x(t_k)$, where t_k is the k-th transition epoch of the continuous-time process. As indicated in Section II-A, the routing policy can be parameterized via weight vector $w \in \mathbb{R}_{>0}^N$.

The transition probability $\mathsf{p}(x'|x, a)$ of the DT MDP can be derived from the system dynamics in a straightforward manner. Let $e_i \in \{0, 1\}^N$ denote the unit vector such that $e_{i,i} = 1$ and $e_{i,j} = 0, j \neq i$. Then we have

$$\mathsf{p}(x'|x, a) = \begin{cases} \frac{\lambda}{\lambda + \sum_{n=1}^{N} \mu_n \mathbb{I}_{\{x_n > 0\}}} & x' \in \{x + e_a\}_{a=1}^{N}, \\ \frac{\mu_n \mathbb{I}_{\{x_n > 0\}}}{\lambda + \sum_{n=1}^{N} \mu_n \mathbb{I}_{\{x_n > 0\}}} & x' = x - e_n. \end{cases}$$

The one-step random cost of the MDP is given by

$$c[k+1] = \|x[k+1]\|_1 (t_{k+1} - t_k),$$

where $\|\cdot\|_1$ is the 1-norm for $\mathbb{R}^N$. Let $\bar{c}(x,a) = \mathsf{E}[c[k+1]|x[k]=x, a[k]=a]$ denote the expected value of cost. The total discounted cost over infinite-horizon process is thus given by

$$Q_\pi(x,a) = \mathsf{E}_\pi\Big[\sum_{\ell=0}^{\infty} \gamma^\ell \|x[\ell+1]\|_1 (t_{\ell+1} - t_\ell)\Big| x, a\Big],$$

where $\gamma \in (0,1)$ is the discount factor of infinite-horizon MDP. Let $Q^*(x,a)$ denote the solution of Bellman optimal equation, that

$$Q^*(x,a) = \bar{c}(x,a) + \gamma \min_{a'} \sum_{x'} \mathsf{p}(x'|x,a) Q^*(x',a'),$$

and let π^* denote the greedy policy with respect to Q^*.

Closed-form solution to Q_π is not easy. Hence, we use the $\hat{Q}$ function defined by (1) as a proxy for Q_π. Motivated by [11], [16], we consider the function approximator as

$$\hat{Q}(x,a;w) = \sum_{n=1}^{N} w_n \phi_n(x,a), \tag{5}$$
$$\phi_n(x,a) = (x_n + \mathbb{I}_{\{n=a\}})^2, \quad n = 1,2,\ldots,N,$$

where $\phi_n : \mathcal{N} \times \mathbb{Z}^N_{\geq 0} \to \mathbb{R}_{\geq 0}$ and $\{\{\phi_n\}_{x,a}\}_{n=1}^N$ are linearly independent basis functions. Let w^* denote the optimal solution to

$$\min_w \sum_{\substack{x \in \mathbb{Z}^N_{\geq 0}, \\ a \leq N}} d^*(x)\pi^*(a|x)\Big(Q^*(x,a) - \hat{Q}(x,a;w)\Big)^2,$$

where $d^*(x)$ is the invariant state distribution under policy π^*. We select quadratic basis functions because Q_π is nonlinear and the quadratic function is one of the simplest nonlinear functions. Besides, the analysis is generalizable to polynomials with higher order.

C. Semi-gradient SARSA algorithm

Inspired by [11], let $w[k]$ denote the weight vector at the k-th transition epoch, which is updated by an SARSA(0) algorithm

$$w[k+1] = \Gamma\Big(w[k] + \alpha_k \Delta[k] \nabla_w \hat{Q}(x[k], a[k]; w[k])\Big);$$

in the above, $\Gamma : \mathbb{R}^N_{>0} \to \mathbb{R}^N_{>0}$ is a projection operator, α_k is the stochastic step size, $\Delta[k]$ is the temporal-difference (TD) error, and $\nabla_w \hat{Q}(x[k], a[k]; w[k])$ is the gradient, which are specified as follows.

The projection $\Gamma(\cdot)$ is defined with a positive constant C_Γ:

$$\Gamma(w) = \begin{cases} |w| & \|w\| \leq C_\Gamma, \\ C_\Gamma \frac{|w|}{\|w\|} & \|w\| > C_\Gamma, \end{cases}$$

where $\|\cdot\|$ is the standard 2-norm, and $|w|$ is the vector that consists of the absolute value of the items in w. Besides, we use $\langle x, y\rangle := x^T y$ denote the standard inner product in Euclidean spaces.

The temporal difference (TD) error $\Delta[k]$ and the gradient $\nabla_w \hat{Q}(x[k], a[k]; w[k])$ are as follows. Let $\phi = [\phi_1, \phi_2, \ldots, \phi_N]^T$, then we can compactly write

$$\hat{Q}(x,a;w) = \phi^T(x,a) w,$$
$$\nabla_w \hat{Q}(x,a;w) = \phi(x,a).$$

Then, for any $k \in \mathbb{Z}_{\geq 0}$, the TD error and the gradient are collectively given by

$$\delta_{w[k]}(x[k], w[k]) = \Delta[k] \nabla_w \hat{Q}(x[k], a[k]; w[k])$$
$$= \Bigg(- \phi^T\Big(x[k], a[k]\Big) w[k] + c[k+1] + \gamma \phi^T\Big(x[k+1], a[k+1]\Big) w[k] \Bigg) \phi\Big(x[k], a[k]\Big).$$

The step sizes $\{\alpha_k\}$ are generated by the following mechanism. We define an auxiliary sequence $\{\tilde{\alpha}_{\tilde{k}}; \tilde{k} \in \mathbb{Z}^N_{\geq 0}\}$ satisfying the standard step size condition [10]

$$\sum_{k=0}^{\infty} \tilde{\alpha}_{\tilde{k}} = \infty, \quad \sum_{k=0}^{\infty} \tilde{\alpha}^2_{\tilde{k}} < \infty. \tag{6}$$

The step sizes can not be too small to stop the iteration process, while also can not be too large to impede convergence. Let B_α denote a finite positive constant and define

$$\tilde{k} = \max_{\substack{\delta_{w[k]}(x[k], w[k]) \leq B_\alpha \\ k' < k}} k'.$$

Then the step size sequence $\{\alpha_k\}$ can be constructed as

$$\alpha_k = \begin{cases} \tilde{\alpha}_{\tilde{k}} & \delta_{w[k]}(x[k], w[k]) \leq B_\alpha, \\ 0 & o.w., \end{cases}$$

where B_α is a finite positive constant. That is, $\{\alpha_k\}$ consists of zeros and elements from the deterministic sequence $\{\tilde{\alpha}_{\tilde{k}}\}$ as demonstrated in Table I. Thus the weight vector $w[k+1]$ is updated only when the constraint B_α is satisfied.

TABLE I: A sample of stochastic step sizes $\{\alpha_k\}$.

k	$\delta_{w[k]}(x[k], w[k]) \leq B_\alpha$?	$\alpha_k =$
0	Yes	$\tilde{\alpha}_0$
1	Yes	$\tilde{\alpha}_1$
2	No	0
3	Yes	$\tilde{\alpha}_2$
$\vdots$	$\vdots$	$\vdots$

The update equation thus becomes

$$w[k+1] = \Gamma\Big(w[k] + \alpha_k \delta_{w[k]}(x[k], w[k])\Big). \tag{7}$$

It is known that SARSA chatters when combined with linear function approximation [10]. We say that Algorithm 1 is *convergent to a bounded region* if there exists a positive finite constant B such that

$$\lim_{k \to \infty} \mathsf{E}[\|w[k] - w^*\|] \leq B, \tag{8}$$

for every initial traffic state $x[0] \in \mathbb{Z}^N_{\geq 0}$ and every initial weight $w[0] \in \mathbb{R}^N_{>0}$.

Algorithm 1 (SGS) Computation of $\hat{Q}$ for Q

Input: Initial weights $w[0], \|w[0]\| < C_\Gamma$, WSQ policy $\pi_{w[0]}$, step sizes sequence α_k, γ

Initialize weights $w[0] \leftarrow w[0]$
for $k = 0, 1, \ldots$ **do**
  Execute action $a[k]$
  Obtain new state $x[k+1]$ and immediate reward $c[k+1]$
  Select $a[k+1]$ according to policy $\pi_{w[k]}$
  Calculate $\delta_{w[k]}(x[k], w[k])$
  $w[k+1] \leftarrow \Gamma(w[k] + \alpha_k \cdot \delta_{w[k]}(x[k], w[k]))$
end for

III. Joint convergence guarantee

In this section, we develop the main result of this paper, which states that the proposed semi-gradient SARSA (SGS) algorithm ensures joint convergence of traffic state and weight vector if and only if the parallel service system is stabilizable.

Theorem 1. *(Joint convergence) Consider a stabilizable parallel service system with arrival rate $\lambda > 0$ and service rates $\mu_1, \mu_2, \ldots, \mu_N > 0$. Suppose that the step size condition* (6) *holds. Then, the traffic state $x[k]$ converges in the sense of* (3) *and the weight $w[k]$ converges in the sense of* (8)*.*

The above result essentially states that the joint convergence of $w[k]$ and $x[k]$ relies on step size constraint (6). They are standard for reinforcement learning methods, which ensures that (i) sufficient updates will be made, and (ii) randomness will not perturb the weights at steady state [16].

The rest of this section is devoted to the proof of the above theorem. Section III-A proves the stability of traffic state, section III-B presents unboundedness of $\sum_{k=0}^{\infty} \alpha_k$, and section III-C proves the convergence of the approximation weights.

A. Convergence of $x[k]$

In this section we construct a Lyapunov function and argue the drift to show the convergence of traffic state $x[k]$, with policy π and proper temperature parameter ι. In particular, we considering the Lyapunov function

$$\hat{V}_w(x) = \sum_{n=1}^{N} w_n x_n^2$$

with the same weight vector as (5). We show that there exist $\iota > 0,\ \epsilon_v > 0,\ B_v < \infty$ such that for all $x \in \mathbb{Z}_{\geq 0}^N$ and for all $w \in \mathbb{R}_{>0}^N$

$$\mathcal{L}\hat{V}_w(x) = -\epsilon_v \sum_{n=1}^{N} w_n x_n + B_v, \tag{9}$$

where $\mathcal{L}$ is the infinitesimal generator of the system under the (softmax) WSQ policy.

Let $l_n = \mathbb{I}_{\{x_n \geq 1\}},\ m = \arg\min_{n \leq N} w_n(2x_n + 1)$. We have

$$\mathcal{L}\hat{V}_w(x) = \sum\nolimits_{n=1}^{N} l_n \mu_n w_n(-2x_n + 1) + \sum\nolimits_{a=1}^{N} \pi_w(a|x)\lambda_a w_a(2x_a + 1).$$

Suppose that ι is sufficiently small, since $(l_n - 1)x_n = 0$, we have

$$\mathcal{L}\hat{V}_w(x) \leq \sum_{n=1}^{N} \Big(\lambda \frac{\mu_n}{\sum_{k=1}^{N} \mu_k} - \mu_n\Big) w_n(2x_n + 1) + B_v.$$

Then (9) holds when (4) holds. We can use Foster-Lyapunov criterion [15, Theorem 4.3.] conclude (3).

B. Unboundedness of $\sum_{k=0}^{\infty} \alpha_k$

In this section, we show that $\sum_{k=1}^{\infty} \alpha_k = \infty$ a.s..

Lemma 1. *Under the constraints in Theorem 1, let $W_p(x) = \sum_{n=1}^{N} e^{\nu w_n(2x_n+1)}/w_n,\ \nu > 0$, then there exists function $g_p \geq 1$ and finite non-negative constant B_w satisfying*

$$\Delta W_p(x) = \mathsf{E}[W_p(x[k+1]) - W_p(x[k])|x[k] = x] \leq -g_p(x) + B_w. \tag{10}$$

Furthermore, we have

$$\lim_{K \to \infty} \frac{1}{K} \sum\nolimits_{k=0}^{K-1} \mathsf{E}[g_p(x)] \leq B_w.$$

Proof: Considering similar definition of m, l_n in section III-A. Similarly, under the (softmax) WSQ policy, we have

$$\Delta W_p(x) = \sum_{a=1}^{N} \pi_w(a|x) \Big[\sum_{n \neq a}^{N} \frac{l_n}{w_n} \cdot \Big(e^{\nu w_n(2x_n - 2\mu_n + 1)} - e^{\nu w_n(2x_n+1)} \Big) - \frac{1}{w_a} \Big((1 - l_a) \cdot e^{\nu w_a(2x_a + 2\lambda + 1)} + l_a \cdot e^{\nu w_a(2x_a + 2(\lambda - \mu_a) + 1)} + e^{\nu w_a(2x_a+1)} \Big) \Big].$$

Note that there is $l_n \cdot e^{\nu w_n 2x_n} = (l_n - 1) + e^{\nu w_n 2x_n}$ and suppose that ι is sufficiently small, we have

$$\Delta W_p(x) = \sum_{n \neq m}^{N} e^{\nu w_n(2x_n+1)} \cdot (e^{-2\nu w_n \mu_n} - 1)/w_n + e^{\nu w_m(2x_m+1)} \cdot (e^{2\nu w_m(\lambda - \mu_m)} - 1)/w_m + B_0, \tag{11}$$

where $B_0 = \frac{1}{w_m}(e^{\nu w_m(2x_m+1)} - e^{\nu w_m(2\lambda_m - 2\mu_m + 1)}) + \sum_{n \neq m}^{N} e^{\nu w_n}(1 - e^{-2\nu w_n \mu_n}))\frac{1}{w_n}$ is a finite positive constant. In the case of $\lambda \leq \mu_m$, the drift equation (11) naturally satisfies (10). When $\lambda > \mu_m$, we have

$$\Delta W_p(x) \leq \sum_{n \neq m}^{N} e^{\nu w_n(2x_n+1)} \cdot \Lambda(\nu),$$

$$\Lambda(\nu) = \frac{\mu_n(e^{2\nu w_m(\lambda - \mu_m)} - 1)}{w_m \sum_{k \neq m}^{N} \mu_k} + \frac{(e^{-2\nu w_n \mu_n} - 1)}{w_n}.$$

Note that $\Lambda(0) = 0,\ \Lambda(\infty) \to \infty$. The derivate of $\Lambda(\nu)$ at $\nu = 0$ is calculated as

$$\frac{d\Lambda}{d\nu}\Big|_{\nu=0} = 2\mu_n\Big(\frac{\lambda - \mu_m}{\sum_{k \neq m}^{N} \mu_k} - 1\Big).$$

Then the derivative of Λ is negative when $\lambda < \sum_{n=1}^{N} \mu_n$, which implies that there exist $\nu_0 > 0$ as the second zero of $\Lambda(\nu)$ and $\Lambda(\nu) < 0,\ \nu \in (0, \nu_0)$. Now we can conclude that with a proper selection of ν, (10) is guaranteed. There exists a finite positive constant B_p satisfies

$$g_p(x) = B_p \sum\nolimits_{n=1}^{N} e^{\nu w_n (2x_n+1)}/w_n + 1.$$

Following the proof in [19], summing the inequality over epochs $k \in \{0, \ldots, K-1\}$ yields a telescoping series on the left hand side of (10), result in

$$\begin{aligned}&\mathsf{E}[W_p(x[K])] - \mathsf{E}[W_p(x[0])]\\ &\le K(B_0+1) - \sum_{k=0}^{K-1} \mathsf{E}[B_p \sum\nolimits_{n=1}^{N} e^{\nu w_n (2x_n+1)}/w_n + 1].\end{aligned}$$

Since $\mathrm{E}[W_p(x[0])] \ge 0$, we have

$$\lim_{K\to\infty} \frac{1}{K} \sum_{k=0}^{K-1} \mathsf{E}[B_p \sum\nolimits_{n=1}^{N} e^{\nu w_n (2x_n+1)}/w_n + 1] \le B_w,$$

where $B_w = B_0 + 1$. The above inequality implies the boundedness of $\mathsf{E}[\sum_{n=1}^{N} e^{\nu w_n (2x_n+1)}]$, thus the higher order stability of system states [17]. □

Proposition 1. *$\forall w \in \mathbb{R}^N_{>0}$, the chain induced by π_w is ergodic and positive Harris recurrent.*

Proof: To argue for the irreducibility of the chain, note that the state $x = \mathbf{0}$ can be accessible from any initial condition with positive probability. According to [20, Theorem 11.3.4], the proof of ergodic and positive Harris recurrent is straightforward with Lemma 1. □

Proposition 2. *With Proposition 1, the step size sequence $\{a_k\}$ in SGS satisfies* (6)*.*

Proof: Let $\tilde{\mathcal{X}} := \{x : \|x\| \le B_{\tilde{\mathcal{X}}}, x \in \mathbb{Z}^n_{\ge 0}\}$, where $B_{\tilde{\mathcal{X}}}$ is a finite positive constant and there is $\tilde{\mathcal{X}} \in \mathbb{Z}^n_{\ge 0}$. Then the boundedness $\Delta[k]\nabla_w \hat{Q}(a[k], x[k], w[k]) \le B_\alpha$ can be satisfied by constraining $x[k] \in \tilde{\mathcal{X}}$ and $t_{k+1} - t_k \le B_T$, where $B_T < \infty$ is a positive constant. That is, the weight vector $w[k+1]$ is updated only when the state and time interval (i.e., the immediate cost) are not too large.

Let use $\tau_{\tilde{\mathcal{X}}} := \sum_{k=1}^{\infty} \mathbb{I}_{x[k]\in\tilde{\mathcal{X}}}$ denote the occupation time of states $x \in \tilde{\mathcal{X}}$, since the chain is positive Harris recurrent, we have $P(\tau_{\tilde{\mathcal{X}}} = \infty) = 1$. Since the arrival rate and service rates are well-defined, we have $P(t_{k+1} - t_k \le B_T) \ge P_T > 0$. Then we have

$$\begin{aligned}&\sum_{k=0}^{\infty} \alpha_k \ge P_T \sum_{x_k\in\tilde{\mathcal{X}}}^{\tau_{\tilde{\mathcal{X}}}} \tilde{\alpha}_{\tilde{k}} + 0 = \infty \quad \text{a.s.},\\ &\sum_{k=0}^{\infty} \alpha_k^2 \le \sum_{k=0}^{\infty} \tilde{\alpha}_{\tilde{k}}^2 < \infty \quad \text{a.s.}\end{aligned}$$

as desired. □

C. Convergence of w[k]

In the following, we establish the convergence of $w[k]$ by showing (i) the convergence under fixed-policy evaluation and (ii) the difference among optimal weight vectors due to policy improvement is bounded.

Proposition 3. *(Lipschitz continuity of $\pi_w(a|x)$) For our WSQ policy, there exists $L_\pi > 0$ such that $\forall w, w', a, x$,*

$$\|\pi_w(a|x) - \pi_{w'}(a|x)\| \le L_\pi \|w - w'\|.$$

Proof: Note that the boundedness of derivative towards w implies the Lipschitz continuity. With the well constructed sequence $\{\alpha_k\}$, we have

$$|\pi_w(a|x) - \pi_{w'}(a|x)| = 0 = L_\pi \|w - w'\|, \quad x \notin \tilde{\mathcal{X}}.$$

For $x \in \tilde{\mathcal{X}}$, according to (2), we have $\pi_w(a|x) \in (0,1)$ and

$$\Big|\frac{d\pi_w(a|x)}{dw}\Big| \le \Big|\frac{N}{\iota} \max_{a,b\le N} 2 \cdot |x_a - x_b|\Big|,$$

which is bounded. □

With the above Proposition 1-3, we can analysis the fixed policy performance and bound the difference among distinct policies. For a better elucidation, we use subscript $w^*[k]$ denote the invariant steady-state dynamic of the chain under fixed policy $\pi_{w[k]}$, and use subscript k denote the real dynamic at the k-th transition. That is $d_{w^*[k]}(\cdot)$ denotes the invariant distribution of policy $\pi_{w[k]}$, and $d_k(\cdot)$ denotes the state distribution at the k-th transition. Suppose that under the fixed policy $\pi_{w[k]}$ and according to [21], we have

$$\bar{\delta}_{w[k]}(x[k], w[k]) = A_{w^*[k]} w[k] + b_{w^*[k]},$$

where $A_{w^*[k]}$ is defined as

$$\begin{aligned}&A_{w^*[k]}\\ &= \sum_{\substack{a'\le N,\\ x'\in\mathbb{Z}^N_{\ge 0}}} \Big(\sum_{\substack{a\le N,\\ x\in\mathbb{Z}^N_{\ge 0}}} d_{w^*[k]}(x)\pi_{w[k]}(a|x)\mathsf{p}_{w^*[k]}(x'|a,x)\\ &\times \gamma\phi(x,a) - d_{w^*[k]}(x')\phi(x',a')\Big)\pi_{w[k]}(a'|x')\phi^T(x',a').\\ &\le -\gamma_A \mathsf{E}_{w^*[k]}\Big[\phi(x',a')\phi^T(x',a')\Big], \quad \gamma_A \in \mathbb{R}_{>0},\end{aligned}$$

which is negative definite since $\gamma \in (0,1)$. Analogous, we have $b_{w^*[k]} = \mathsf{E}_{w^*[k]}[\phi^T(x,a)c], b_k = \mathsf{E}_k[\phi^T(x,a)c], A_k = \mathsf{E}_k[\phi(x,a)(\gamma\phi^T(x',a') - \phi^T(x,a))]$ and $\mathsf{E}[\delta_{w[k]}(x,w)] = A_k w + b_k$.

According to [20, Theorem 14.0.1], we have

$$\begin{aligned}\sup_{f_p:|f_p|\le g_p} &\sum_{t=0}^{\infty} \Big|\sum_{x'} \mathsf{p}^t_k(x'|x) f_p(x') - \sum_{x'} d_{w^*[k]}(x') f_p(x')\Big|\\ &< B_f(W_p(x)+1), \quad x[k] = x, x \in \tilde{\mathcal{X}}, \qquad (12)\end{aligned}$$

where B_f is a finite positive constant. $\mathsf{p}^t_k(x'|x)$ indicates the transition probability from state x to x' after t steps under policy $\pi_{w[k]}$, and there is $\mathsf{p}(x'|x) = \sum_{a=1}^{N} \pi(a|x)\mathsf{p}(x'|a,x)$. According to [16], with (12) holds, the iterative algorithm (7) has a unique solution w^* satisfies that $\bar{\delta}(x, w^*) = 0$ under

fixed policy. With a little abuse of notation, let w_k^* denote the solution of

$$\bar{\delta}_{w[k]}(x[k], w_k^*) = 0$$

in SGS under fixed policy $\pi_{w[k]}$.

According to the inequality $e^{kx} > \sum_{q=0}^{\infty} \frac{k^q x^q}{q!}$ and note that there is $\phi(x,a) = \mathcal{O}(x^2)$, the boundedness of $g_p(x)$ defined in Lemma 1 implies the boundedness of $A_{w^*[k]}, b_{w^*[k]}$. With the constructed step size $\{\alpha_k\}$, we have $\|A_{w^*[k]}\| < B_\phi$, $\|b_{w^*[k]}\| < B_r$, and $\mathsf{E}[\alpha_k^2 \|\delta_{w[k]}(x,w) - \bar{\delta}_{w[k]}(x,w)\|^2] \leq \alpha_k^2 B_\delta^2$, where B_ϕ, B_r, B_δ are finite positive constants.

Following [11], considering the weights update equation, the auxiliary sequence $\{u[k]\}$ is defined as

$$\begin{aligned} &u[0] := w[0], \\ &u[k+1] := \Gamma(u[k]) + \alpha_k \delta_{w[k]}(x[k], \Gamma(u[k])). \end{aligned}$$

Since the $\hat{Q}$ and policy π are both linearly related to w, we have $w[k] = \Gamma(u[k]), \forall k \in \mathbb{Z}_{\geq 0}$. Let $y' = u[k+1] - w_{k+1}^*$, $y = w[k] - w_k^*$, then we have

$$\begin{aligned} \mathsf{E}[\frac{1}{2}\|y'\|^2] &= \mathsf{E}\Big[\frac{1}{2}\|y\|^2\Big] \\ &+ \underbrace{\mathsf{E}\Big[\Big\langle y, \quad \alpha_k \delta_{w[k]}(x[k], w[k]) + w_k^* - w_{k+1}^* \Big\rangle\Big]}_{T_2} \\ &+ \underbrace{\mathsf{E}\Big[\frac{1}{2}\|\alpha_k \delta_{w[k]}(x[k], w[k]) + w_k^* - w_{k+1}^*\|^2\Big]}_{T_3}, \end{aligned}$$

where $\langle \cdot, \cdot \rangle$ is defined as section II-C. The second item can be rewritten as

$$T_2 = \Big\langle y, \mathsf{E}\Big[\underbrace{\alpha_k \delta_{w[k]}(x[k], w[k])}_{T_{21}}\Big]\Big\rangle + \langle y, w_k^* - w_{k+1}^* \rangle.$$

For T_3, we have

$$\begin{aligned} T_3 = \mathsf{E}\Big[\frac{1}{2}\|w_k^* - w_{k+1}^*\|^2\Big] + \mathsf{E}\Big[\Big\langle w_k^* - w_{k+1}^*, T_{21} \Big\rangle\Big] \\ + \mathsf{E}\Big[\frac{1}{2}\|T_{21}\|^2\Big]. \end{aligned}$$

Now we are ready to analyze the boundedness of each item. Note that $\alpha_k \neq 0$ only when $x[k] \in \tilde{\mathcal{X}}$, with [11, Theorem 4.4., Lemma C.5., Lemma D.10.] and the analysis of $\delta_{w[k]}$ and w_k^*, we have

$$\alpha_k \|\delta_{w[k]}(x[k], w)\| \leq \alpha_k (L_F \|w\| + U_F),$$

and

$$\begin{aligned} \|w_w^* - w_{w'}^*\| &\leq \underbrace{(B_A^2 B_\phi L_{DP} + B_A L_D) L_\pi B_r}_{L_w} \|w - w'\| \\ &\leq \alpha_k L_w (U_F + L_F C_\Gamma + C_\Gamma) = \alpha_k L_{BW}, \end{aligned}$$

where B_A, U_F, L_F are finite positive constants that $B_A > \sup_w \|A_{w^*}^{-1}\|$, $\alpha_k U_F \geq \|\alpha_k b_k\|$, $L_F \geq ((1+\gamma)(B_{\tilde{\mathcal{X}}} + 1)^2 + 1)$, and L_D, L_{DP} are Lipschitz constants of state distribution and transition probability. We have

$$T_{21} = \alpha_k \bar{\delta}_{w[k]}(\cdot) + \alpha_k \delta_{w[k]}(\cdot) - \alpha_k \bar{\delta}_{w[k]}(\cdot),$$

where $(\cdot)$ is short for $(x[k], w[k])$. By leveraging [11, Lemma D.2.], suppose that k is sufficient large, we have $\|\alpha_k \bar{\delta}_{w[k]}(\cdot)\| \leq (k_\alpha - 1)\|y\|$, $k_\alpha = \sqrt{1 - B_\phi \alpha_k} \in (0,1)$. Then we have

$$T_2 \leq \mathsf{E}[\Big((k_\alpha - 1)\|y\| + \alpha_k B_\delta + \alpha_k L_{BW}\Big)\|y\|],$$

where $k_\alpha = \sqrt{1 - B_\phi \alpha_k} \in (0,1)$. Analogously, we have

$$\begin{aligned} T_3 \leq \mathsf{E}[&\frac{1}{2}(k_\alpha - 1)^2 \|y\|^2 + \alpha_k (k_\alpha - 1)(B_\delta + L_{BW})\|y\| \\ &+ \alpha_k^2 (\frac{1}{2} B_\delta^2 + B_\delta L_{BW} + \frac{1}{2} L_{BW}^2)]. \end{aligned}$$

According to the above analysis, we have

$$\begin{aligned} &\mathsf{E}[\frac{1}{2}\|u[k+1] - w_{k+1}^*\|^2] \\ &= \frac{1}{2}\mathsf{E}\Big[\Big(k_\alpha \|w[k] - w_k^*\| + \alpha_k (L_{BW} + B_\delta)\Big)^2\Big]. \end{aligned}$$

Then we have

$$z_{k+1} \leq k_\alpha z_k + \alpha_k (L_{BW} + B_\delta),$$

where $z_{k+1} = \sqrt{\mathsf{E}[\|u[k+1] - w_{k+1}^*\|^2]}$. Since $k_\alpha \in (0,1)$ and $\|w[k] - w_k^*\| \leq \|u[k] - w_k^*\|$, by iteration, we have

$$\begin{aligned} z_{k+1} = &\sqrt{\prod_{\ell=k_0}^{k} (1 - \alpha_\ell B_\phi)} \cdot z_{k_0} \\ &+ (L_{BW} + B_\delta) \sum_{\ell=k_0}^{k} \alpha_\ell \sqrt{\prod_{j=\ell}^{k} (1 - \alpha_j B_\phi)}. \end{aligned}$$

Note that $\{\alpha_k\}$ is constrained by (6) and the inequality $1 - x \leq e^{-x}$ holds, we have

$$z_{k+1} \leq \frac{B_\delta + L_{BW}}{B_\phi}.$$

Considering the relationship between the policy and weight vector, we have

$$\begin{aligned} \mathsf{E}[\|w[k] - w^*\|] &\leq \mathsf{E}[\|w[k] - w_k^*\|] + \mathsf{E}[\|w_k^* - w_{w^*}^*\|] \\ &\leq \mathsf{E}[\|w[k] - w_k^*\|] + L_w \mathsf{E}[\|w[k] - w^*\|]. \end{aligned}$$

When the immediate cost c is constrained such that $L_w < 1$, we have

$$\mathsf{E}[\|w[k] - w^*\|] \leq \frac{1}{1 - L_w} \mathsf{E}[\|w[k] - w_k^*\|] \leq \frac{1}{1 - L_w} z_t.$$

Then finally we can conclude that

$$\mathsf{E}[\|w[k] - w^*\|] \leq \frac{B_\delta + L_{BW}}{(1 - L_w) B_\phi},$$

which yields (8).

IV. EXPERIMENTS

To evaluate the performance of the semi-gradient SARSA (SGS) algorithm with weighted shortest queue (WSQ) policy, we consider two benchmarks:

1) Neural network (NN)-WSQ: We constructed a NN for approximation of $Q(x,a)$. The algorithm is similar to Algorithm 1, except that the weights update is replaced by NN update with adaptive moment estimation (Adam)

algorithm [22]. Specifically, the NN has two fully connected layer with a rectified linear unit (ReLU) activation function. The loss function is the mean square error between the one-step predicted and calculated state-action-value. Since an exact optimal policy of the original MDP is not readily available, the policy computed by NN is used as an approximate optimal policy.

2) Join the shortest queue (JSQ) policy: For routing decisions under JSQ policy, we simply select the path with the shortest queue length, that is $a_{JSQ} = \arg\min_{n \le N} x_n$.

Consider the network in Fig. 1 with three parallel servers. Suppose that the service rate $\mu_1 = 0.5,\ \mu_2 = 2.5,\ \mu_3 = 5$ and the arrival rate $\lambda = 2$, all in unit sec^{-1}. The WSQ policy temperature parameter is set as $\iota = 0.01$. For SGS algorithm, we initialize the weight as $w_1 = w_2 = w_3 = 0.5$. For simulation, a discrete time step of 0.1 sec is used. All experiments were implemented in Google Colab [23], using Intel(R) Xeon(R) CPU with 12.7GB memory. We trained SGS for 10^6 epochs, NN for 4×10^6 epochs, and then evaluate them for 10^6 epochs each. The results are as follows.

For the performance, the weight of SGS converges to $w = [0.60\ 0.49\ 0.15]$, which is reasonable and consistent with the expectation of our policy that the weight of a slower server is higher. Note that there is a peak value of NN after the number of epochs growing larger than 10^2 epochs, which may be due to the explorative behavior of the transient NN based algorithm. Our WSQ policy is not restricted to non-idling conditions, since $w_1(2 \times 0 + 1) > w_3(2 \times 1 + 1)$, the server with higher service rate (i.e., server 3) is more likely to be selected, even the slower server (i.e., server 1) is empty.

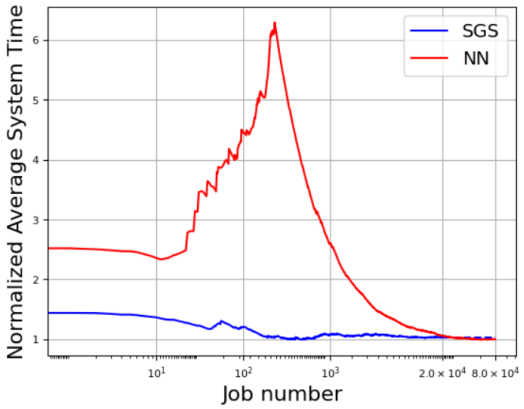

Fig. 2: The performance compare between SGS and NN.

TABLE II: Average system times of various schemes.

Algorithm	Normalized Average System Time
Neural network (NN)	1.00
Semi-gradient SARSA (SGS)	1.08
Join the shortest queue (JSQ)	2.78

Table II lists the normalized average system time results under various methods. The results are generated from test simulations of 10^5 sec. Although NN performs better in long terms of learning as expected, SGS performs better with just a few number of iterations as demonstrated in Fig. 2.

The job will spend more time going through the queuing network under JSQ policy at the average of 0.8581 sec. For WSQ, though the implementation efficiency of SGS is slightly worse than NN, SGS gives the best trade-off between computational efficiency and implementation efficiency: the average system time of SGS is 0.3318 sec, only 8% longer than the result of 0.3078 sec of NN, while with more than four times fewer training epochs. More importantly, SGS algorithm theoretically ensures the convergence of the optimal routing decision, while NN might be diverge and let alone the existence of the optimal decision.

V. CONCLUSION

In this paper, we propose a semi-gradient SARSA(0) (SGS) algorithm with linear value function approximation for dynamic routing over parallel servers. We extend the analysis of SGS to infinite state space and show that the convergence of the weight vector in SGS is coupled with the convergence of the traffic state, and the joint convergence is guaranteed if and only if the paralle service system is stabilizable. Specifically, the approximator is used as Lyapunov function for traffic state stability analysis; and the constraint and convergence analysis of weight vector is based on stochastic approximation theory. Besides, our analysis can be extended to polynomial approximator with higher order. We compare the proposed SGS algorithm with a neural network-based algorithm and show that our algorithm converges faster with a higher computationally efficiency and an insignificant optimality gap.

However, this paper focus on a general setting on the learning step sizes $\{\alpha_k\}$ and discount factor γ. We discover that the convergence performance of the learning algorithm is highly associated with the setting of the fine-tuning parameters $\{\alpha_k\}$ and γ, which may indicate that a better and more particular design of learning step sizes and discount factor may lead to a stronger theoretical guarantee of joint-convergence. Our ongoing work is trying to figure out the specific relationship between the parameter and the convergence. Possible future work includes (i) sensitivity analysis and precise design of $\{\alpha_k\}$ and γ; (ii) extension of the joint convergence result as well as SGS algorithm to a general service network and (iii) the analysis of fixed-point convergence condition.

REFERENCES

[1] J. G. Dai and M. Gluzman, "Queueing network controls via deep reinforcement learning," *Stochastic Systems*, vol. 12, no. 1, pp. 30–67, 2022.

[2] Q. Xie and L. Jin, "Stabilizing queuing networks with model data-independent control," *IEEE Transactions on Control of Network Systems*, vol. 9, no. 3, pp. 1317–1326, 2022.

[3] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[4] P. Kumar and S. P. Meyn, "Stability of queueing networks and scheduling policies," *IEEE Transactions on Automatic Control*, vol. 40, no. 2, pp. 251–260, 1995.

[5] J. G. Dai and S. P. Meyn, "Stability and convergence of moments for multiclass queueing networks via fluid limit models," *IEEE Transactions on Automatic Control*, vol. 40, no. 11, pp. 1889–1904, 1995.
[6] S. Bradtke and M. Duff, "Reinforcement learning methods for continuous-time markov decision problems," *Advances in Neural Information Processing Systems*, vol. 7, 1994.
[7] B. Liu, Q. Xie, and E. Modiano, "Rl-qn: A reinforcement learning framework for optimal control of queueing systems," *ACM Transactions on Modeling and Performance Evaluation of Computing Systems*, vol. 7, no. 1, pp. 1–35, 2022.
[8] Z. Xu, J. Tang, J. Meng, W. Zhang, Y. Wang, C. H. Liu, and D. Yang, "Experience-driven networking: A deep reinforcement learning based approach," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 1871–1879.
[9] S.-C. Lin, I. F. Akyildiz, P. Wang, and M. Luo, "Qos-aware adaptive routing in multi-layer hierarchical software defined networks: A reinforcement learning approach," in *2016 IEEE International Conference on Services Computing (SCC)*. IEEE, 2016, pp. 25–33.
[10] G. J. Gordon, "Reinforcement learning with function approximation converges to a region," *Advances in Neural Information Processing Systems*, vol. 13, 2000.
[11] S. Zhang, R. T. Des Combes, and R. Laroche, "On the convergence of sarsa with linear function approximation," in *International Conference on Machine Learning*. PMLR, 2023, pp. 41 613–41 646.
[12] D. P. De Farias and B. Van Roy, "On the existence of fixed points for approximate value iteration and temporal-difference learning," *Journal of Optimization Theory and Applications*, vol. 105, pp. 589–608, 2000.
[13] F. L. Lewis and D. Liu, *Reinforcement learning and approximate dynamic programming for feedback control*. John Wiley & Sons, 2013.
[14] Y. Wu, F. Shu, J. Zhang, and L. Jin, "Learning-based adaptive dynamic routing with stability guarantee for a single-origin-single-destination network," in *2024 43nd Chinese Control Conference (CCC)*. IEEE, 2024, pp. 00–00.
[15] S. P. Meyn and R. L. Tweedie, "Stability of markovian processes iii: Foster–lyapunov criteria for continuous-time processes," *Advances in Applied Probability*, vol. 25, no. 3, p. 518–548, 1993.
[16] J. Tsitsiklis and B. Van Roy, "Analysis of temporal-diffference learning with function approximation," *Advances in Neural Information Processing Systems*, vol. 9, 1996.
[17] S. Meyn, *Control techniques for complex networks*. Cambridge University Press, 2008.
[18] R. G. Gallager, *Stochastic Processes: Theory for Applications*. Cambridge University Press, 2013.
[19] L. Georgiadis, M. J. Neely, L. Tassiulas *et al.*, "Resource allocation and cross-layer control in wireless networks," *Foundations and Trends® in Networking*, vol. 1, no. 1, pp. 1–144, 2006.
[20] S. P. Meyn and R. L. Tweedie, *Markov chains and stochastic stability*. Springer Science & Business Media, 2012.
[21] F. S. Melo, S. P. Meyn, and M. I. Ribeiro, "An analysis of reinforcement learning with function approximation," in *Proceedings of the 25th International Conference on Machine Learning*, 2008, pp. 664–671.
[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[23] "Google Colaboratory," https://colab.research.google.com/, accessed: 2023-03-28.