Hindawi
Journal of Advanced Transportation
Volume 2020, Article ID 8813467, 13 pages
https://doi.org/10.1155/2020/8813467

Research Article

Ramp Metering for a Distant Downstream Bottleneck Using Reinforcement Learning with Value Function Approximation

Yue Zhou,[1] **Kaan Ozbay**,[1] **Pushkin Kachroo**,[2] **and Fan Zuo**[1]

[1]*C2SMART Center, New York University, NYU Civil Engineering, 6 Metrotech Center, Brooklyn 11201, NY, USA*
[2]*Department of Electrical and Computer Engineering, University of Nevada, Las Vegas, 4505 S. Maryland Pkwy, Las Vegas 89154-4026, NV, USA*

Correspondence should be addressed to Yue Zhou; zhouyue30@msn.com

Received 13 July 2020; Revised 19 August 2020; Accepted 18 September 2020; Published 28 October 2020

Academic Editor: Ruimin Li

Ramp metering for a bottleneck located far downstream of the ramp is more challenging than for a bottleneck that is near the ramp. This is because under the control of a conventional linear feedback-type ramp metering strategy, when metered traffic from the ramp arrive at the distant downstream bottleneck, the state of the bottleneck may have significantly changed from when it is sampled for computing the metering rate; due to the considerable time, these traffic will have to take to traverse the long distance between the ramp and the bottleneck. As a result of such time-delay effects, significant stability issue can arise. Previous studies have mainly resorted to compensating for the time-delay effects by incorporating predictors of traffic flow evolution into the control systems. This paper presents an alternative approach. The problem of ramp metering for a distant downstream bottleneck is formulated as a Q-learning problem, in which an intelligent ramp meter agent learns a nonlinear optimal ramp metering policy such that the capacity of the distant downstream bottleneck can be fully utilized, but not to be exceeded to cause congestion. The learned policy is in pure feedback form in that only the current state of the environment is needed to determine the optimal metering rate for the current time. No prediction is needed, as anticipation of traffic flow evolution has been instilled into the nonlinear feedback policy via learning. To deal with the intimidating computational cost associated with the multidimensional continuous state space, the value function of actions is approximated by an artificial neural network, rather than a lookup table. The mechanism and development of the approximate value function and how learning of its parameters is integrated into the Q-learning process are well explained. Through experiments, the learned ramp metering policy has demonstrated effectiveness and benign stability and some level of robustness to demand uncertainties.

1. Introduction

A genuine motivation behind ramp metering strategies is to reduce the total time spent within the freeway network of interest [1]. Minimization of the total time spent can be shown to be equivalent to maximizing time-weighted discharging flow from the network, i.e., encouraging early discharge of flow [1]. This motivation, combined with the knowledge of traffic flow theory, implies that the objective of a ramp metering strategy is to maintain the flow rate into the most restrictive bottleneck of the network to be close to the capacity of the bottleneck, but not to exceed it, so that congestion will not be caused. This objective can be achieved by regulating the traffic density (or occupancy) of the bottleneck to stay close to the critical density (or critical occupancy) through metering the ramp flow. This is the principle behind many conventional linear feedback-type ramp metering strategies, e.g., [2–5]. For this kind of ramp metering strategies, the control target bottleneck is usually near the ramp, and in most cases, the bottleneck is incurred by the merging of the mainline and ramp traffic itself. In some other cases, however, the control target bottleneck is located far away from the metered ramp, for example, a lane-drop that is a few kilometers downstream. In these latter cases, conventional linear feedback-type ramp metering strategies can perform poorly in stability due to the long

distance between the ramp and the bottleneck. Specifically, when metered traffic from the ramp arrive at the distant downstream bottleneck, the traffic density (or occupancy) of the bottleneck may have significantly changed from when it is sampled for computing the metering rate. To overcome this issue, many previous studies have resorted to compensating for the time-delay effects by incorporating predictors of traffic flow evolution into the control systems.

This study presents an alternative approach. The proposed approach formulates the problem of ramp metering for a distant downstream bottleneck as a Q-learning problem, in which an intelligent ramp meter agent learns an optimal ramp metering policy such that the capacity of the distant downstream bottleneck can be fully utilized but not to be exceeded to cause congestion. To our best knowledge, this is the first such effort in the literature. The learned policy is in pure feedback form in that only the current state of the environment is needed to determine the optimal metering rate for the current time. No prediction is needed, as anticipation of traffic flow evolution has been instilled into the learned nonlinear feedback policy. To deal with the intimidating computational cost associated with the multidimensional continuous state space of the formulated Q-learning problem, the value function of ramp metering rates is approximated by an artificial neural network (ANN), rather than a lookup table.

In the remainder of this paper, Section 2 reviews previous studies in ramp metering for distant downstream bottlenecks and Q-learning applications in freeway control. Section 3 develops the proposed approach, including formulation of the Q-learning problem with value function approximation and the algorithm to solve the problem. Section 4 evaluates the proposed approach by experiments. Section 5 concludes this study.

2. Literature Review

2.1. Ramp Metering for a Distant Downstream Bottleneck. Compared with the richness of the literature in ramp metering strategies for bottlenecks near ramps, studies in ramp metering for distant downstream bottlenecks are much fewer. These studies include [6–13]. In [6], the notable ALINEA strategy, which is a linear "proportional" control strategy, was extended by adding to it an "integral" term, resulting in the so-called PI-ALINEA strategy. The authors theoretically proved the stability of the PI-ALINEA strategy. Later, Kan et al. [7] evaluated the performance of PI-ALINEA in controlling a distant downstream bottleneck by simulation. The simulation model employed was METANET [14], a second-order discrete-time macroscopic model of traffic flow dynamics. The simulation evaluation showed that PI-ALINEA outperformed ALINEA in terms of stability. In [8], to deal with the time-delay effects of ramp metering for distant lane-drop bottlenecks, the authors incorporated a Smith predictor [15] into ALINEA and termed the resulting strategy as SP-ALINEA. Through simulation, they showed that the stability region of SP-ALINEA is much broader than the PI-ALINEA. The simulation model employed by Felipe de Souza and Jin [8] was the cell transmission model (CTM) [16], a first-order discrete-time macroscopic model of traffic flow dynamics. Similar to [8], Frejo and De Schutter [9] added a feedfoward term to ALINEA to incorporate anticipated evolutions of the bottleneck density in order to improve the performance of ALINEA. The resulting strategy is termed FF-ALINEA. Similar to [8, 9], Yu et al. [10] coupled a predictor to an extremum-seeking controller for controlling a distant downstream lane-drop bottleneck by metering upstream mainline flow. In [12, 13], fuzzy theory was applied to a proportional-integral-derivative- (PID-) type ramp metering controller to learn the PID gains in real time. The resulting controller has the capability of anticipation, hence performs better in controlling a distant downstream bottleneck than a controller with fixed gains. Stylianopoulou et al. [11] proposed a linear-quadratic-integral (LQI) regulator-type ramp metering strategy for controlling a distant downstream bottleneck. Unlike all the studies that were summarized above which only take measurements near the bottleneck, in [11], however, measurements which spread along the whole stretch between the ramp and the downstream bottleneck are utilized by the controller, so the controller has a better sense of traffic flow evolutions along the stretch, hence possessing better stability and robustness.

2.2. Q-Learning Applications in Freeway Control. Application of Q-learning to freeway control has been widely studied. However, to our best knowledge, no effort has been made to apply Q-learning to ramp metering for distant downstream bottlenecks. Notwithstanding this, this section summarizes previous studies in Q-learning applications to ramp metering (RM) control for nearby bottlenecks and to variable speed limit (VSL) control. These studies are summarized in Table 1. Although this summary may not be thorough, it should have included most previous studies in freeway control by Q-learning approaches. Among these studies, [18–22, 27, 28, 32] were concerned with ramp metering. [23, 30, 31, 33] studied variable speed limits (VSL). Ramp metering and variable speed limits were jointly applied by [29]. [17, 24–26] simultaneously used ramp metering and variable message signs (VMS) for dynamic routing. Most of these studies aimed to achieve one of the following three objectives: minimization of the total time spent by vehicles [17, 19, 27, 28, 31, 33], maximization of early discharge of flow [24–26], and minimization of deviations of the traffic density of the control target section from the critical density [20, 23, 29, 30]. As discussed in Section 1, these three objectives are equivalent.

By the type of the applied Q-learning method, these studies can be classified into two categories. The first category consists of those that used lookup table methods, i.e., [17, 18, 20–31]; the second category includes those that employed value function approximation-based methods, i.e., [31–33].

Lookup table methods, also known as tabular methods [34], as suggested by the name, maintain a lookup table that stores the values for all state-action pairs (known as Q-values). The Q-learning process can be viewed as the

Table 1: Summary of Q-learning applications in freeway control.

Work	Control method	Lookup table method or value function approximation method	State variables	Action	Reward	Simulation model
[17]	RM-VMS	Lookup table	Speed, density, flow diversion splits	Increment in metering rate, increment in flow diversion split	Total time spent	Macro (METANET)
[18]	RM	Lookup table	Density of bottleneck, ramp queue length, ramp demand, current metering rate	Whether to increase, decrease, or not change the current metering rate	Outflow, ramp queue length	Macro (METANET)
[19]	RM	Lookup table	Not clear	Metering rates	Total time spent	Micro (VISSIM)
[20]	RM	Lookup table	Number of vehicles in mainline, number of vehicles entered from the ramp, ramp signal of the last step	Red/green signal	Deviation of density from critical density	Macro (not clear)
[21]	RM	Lookup table	Number of vehicles in the area of interest	Red/green signal	Not clear	Macro (not clear)
[22]	RM	Lookup table	Mainline speeds, ramp queue lengths, ramp metering signal status	Red/green signal	Ramp queue length, mainline average speed	Micro (VISSIM)
[23]	VSL	Lookup table	Densities of mainline and ramp	Speed limits	Deviation of density from critical density	Macro (CTM)
[24–26]	RM-VMS	Lookup table with state-space approximation by the cerebellar model articulation controller	Average speeds, occupancies, status of VMS and ramp, incident presence/absence	Increments in red phase length, VMS for routing	Time-weighted exit flow	Micro (Paramics)
[27, 28]	RM	Lookup table with state-space approximation by k-nearest neighbors	Density, ramp flow	Direct red phase lengths	Total time spent	Micro (Paramics)
[29]	RM-VSL	Lookup table with state-space approximation by k-nearest neighbors	Densities, ramp flow, average speeds, speed differences	Direct red phase lengths	Deviation from critical density	Micro (AnyLogic)
[30]	VSL	Lookup table with state-space approximation by k-nearest neighbors	Densities and speeds	Speed limits	Deviations of densities from critical density, times to collision	Micro (MOTUS)
[31]	VSL	Value function approximation by the neural network; lookup table with state-space approximation by tile coding	Current and predicted densities and speeds	Speed limits	Total time spent	Macro (METANET)
[32]	RM	Value function approximation by the deep neural network	Densities, ramp queue lengths, off-ramp presence/absence	Metering rates	Number of discharged vehicles	Macro (CTM)
[33]	VSL	Value function approximation by the deep neural network	Lane-specific occupancies in mainline and ramp	Lane-specific speed limits	Total time spent, bottleneck speed, emergency brake, emissions	Micro (SUMO)

process of updating the lookup table. Lookup table methods can only handle discrete state-action pairs. They may also deal with the continuous state space; however, the continuous state space needs to be approximated (discretized) first so that any continuous state the learning agent encounters can be mapped to a representative discrete state that is indexed in the lookup table. Most of the studies in Table 1 belong to lookup table methods. Since state variables of

freeway control problems are usually continuous, e.g., traffic densities and ramp queue lengths, those studies that have applied lookup table methods all have involved some kind of state-space approximation. The simplest state-space approximation method is aggregation, which divides a continuous state space into discrete intervals that do not overlap with each other. Many studies in Table 1 are of this kind, i.e., [17, 18, 20–23]. Some other studies employed more sophisticated methods, e.g., k-nearest neighbors, to approximate continuous state spaces. These studies include [24–31].

It is important to note that state-space approximation is not primarily a tool for reducing the computational cost of reinforcement learning. For a multidimensional continuous state-space problem, the lookup table after state-space approximation can still be very large. Admittedly, if the state-space approximation is made very coarse, the table size can be decreased (hence the computational cost), however, at the expense of undermining the effectiveness of the learned policy. Such a difficulty is born with lookup table methods because they aim at directly updating the value of each state-action pair, hence cannot avoid the curse of dimensionality of the state space [35].

The above difficulty can be circumvented by introducing value function approximation. A value function approximation-based reinforcement learning method uses a parameterized function to replace the lookup table to serve as the approximate value function [34]. Consequently, the reinforcement learning process entails learning the unknown parameters of the approximate value function instead of learning the values of state-action pairs. Compared with the number of state-action pairs of a lookup table for a (discretized) multidimensional continuous state-space problem, the number of unknown parameters of an approximate value function is usually profoundly smaller, hence making the learning computationally affordable. Only three studies in Table 1, i.e., [31–33], applied value function approximation-based reinforcement learning methods. The approximate value functions used by these three studies were all artificial neural networks.

An outstanding feature of reinforcement learning that distinguishes it from supervised and unsupervised learning is that, for reinforcement learning, data from which the intelligent agent learns an optimal policy are generated from within the learning process itself. Specifically, the intelligent agent learns through a great amount of interactions with the environment which are enabled by simulation. Hence, simulation models play an important role in reinforcement learning. Among the studies summarized in this section, [19, 22, 24, 30, 33] employed microscopic traffic simulation models such as VISSIM, Paramics, and SUMO; [17, 18, 20, 21, 23, 31, 32] used macroscopic dynamic traffic flow models such as CTM [16] and METANET [14] as the simulation tools.

3. A Q-Learning Problem with Value Function Approximation

3.1. Multidimensional Continuous State Space. Consider the freeway section depicted in Figure 1. A lane-drop bottleneck exists far downstream of the metered ramp. The ramp meter is supposed to regulate the traffic flow into the bottleneck by metering the ramp inflow so that the bottleneck capacity can be fully utilized but not to be exceeded. To this end, the objective of the ramp metering policy is such that it can maintain the per-lane traffic density of the control target location to stay close to a predetermined desired value, which is $(\lambda_2/\lambda_1)\rho^{\mathrm{cr}}$, where λ_1 and λ_2 denote the number of lanes before and after the lane-drop, respectively, and ρ^{cr} is the per-lane critical density. As discussed before, due to the long distance between the metered ramp and the downstream bottleneck, a conventional ramp metering strategy that only senses and utilizes traffic condition near the bottleneck can perform poorly due to the lack of anticipation capability. Therefore, one main requirement in designing our reinforcement learning approach is that it needs to take into account traffic densities measured along the long stretch between the metered ramp and the downstream bottleneck so that an anticipation capability can be built by learning. Since the computational cost of Q-learning grows exponentially with the increase of the dimension of the state space, it would not be computationally cost-effective to take into account measurements at too many places. As a result, three representative places are selected. They are located at the two ends and the middle of the stretch, respectively. Such a treatment, on the one hand, enables the intelligent ramp meter agent to learn to anticipate traffic flow evolution on the stretch, and on the other hand, it limits the computational cost associated with learning. Note that the place of the downstream end of the stretch happens to be the control target location, whose traffic density will be regulated to stay close to the desired value by ramp metering. Therefore, the first three state variables of the proposed Q-learning problem are traffic densities of the three representative places, denoted by ρ_1, ρ_2, and ρ_3, respectively. Note that when the distance between the metered ramp and the downstream bottleneck is sufficiently long and meanwhile the traffic demand pattern is complicated enough in terms of having frequent and large fluctuations, the resulting temporal-spatial traffic flow pattern may be too complicated for the three mainline sampling locations to effectively represent the environment state for the purpose of learning. Under such a circumstance, more sampling locations may be needed. What kind of combinations of the stretch length and traffic demand pattern may yield complicated enough temporal-spatial traffic flow patterns that would cause the three representative mainline sampling locations to result in suboptimal solutions and, accordingly, how many sampling locations should be taken under these circumstances are considered beyond the scope of this paper.

The fourth and also the last state variable is known as the estimated traffic demand on the ramp, denoted by D_{ramp}. This state variable is needed because to learn how much flow from the ramp should be released into the mainline, the intelligent ramp meter agent needs to know not only the traffic conditions of representative mainline places but also the current (estimated) traffic demand on the ramp so as to avoid picking up a metering rate that is too high. The estimated traffic demand on the ramp over the current time

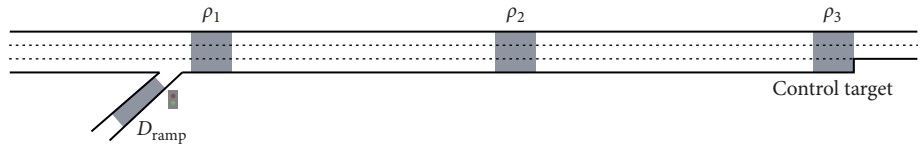

Figure 1: The formulated Q-learning problem having four state variables.

step is computed by (1), where $D_{\text{ramp}}(t)$ denotes the estimated traffic demand on the ramp (in vehicles per hour) for the current time step; $l_{\text{ramp\_queue}}(t)$ represents the queue length on the ramp at the current time step; Δt is the time step length (in seconds); and $q_{\text{ramp\_arrival}}(t-1)$ represents the arrival flow rate at the ramp over the previous time step.

$$D_{\text{ramp}}(t) \doteq \frac{l_{\text{ramp\_queue}}(t)}{(\Delta t/3600)} + q_{\text{ramp\_arrival}}(t-1). \quad (1)$$

The reason to use the arrival flow rate at the ramp over the previous time step rather than the current time step is for the following realistic consideration. Ramp metering rate for the current time step needs to be computed at the end of the previous time step (or, equivalently, at the beginning of the current time step) so that it can be implemented over the current time step; however, by that time, the actual arrival flow rate over the current time step is unknown because it has not yet happened. Therefore, the arrival flow rate at the ramp over the previous time step is used as a proxy to the arrival flow rate at the ramp over the current time step. Such a treatment that brings anticipation of the ramp condition into learning and thus may enhance the learning efficiency appears to be first used by Davarynejad et al. [18]. Note that the queue length on the ramp of the current time step does not need a proxy because it can be readily calculated at the end of the previous time step.

To summarize, the state vector contains four continuous variables, i.e., $\mathbf{s} \doteq \begin{bmatrix} \rho_1 & \rho_2 & \rho_3 & D_{\text{ramp}} \end{bmatrix}$, resulting in a four-dimensional continuous state space.

3.2. State-Dependent Action Space. The actions in the proposed approach are composed of discrete ramp metering rates, as in [29], ranging from the lowest allowable metering rate, $a_{\min}$, to the highest allowable metering rate, $a_{\max}$. The values of $a_{\min}$ and $a_{\max}$ and the number of discrete metering rates are up to the user's specification. In Section 4.1, an example of such a specification is given which is consistent with the requirements of the so-called "full traffic cycle" signal policy for ramp metering [36] so that the results can be implemented by a traffic light. At any time step, the set of admissible actions may not necessarily consist of all the specified discrete metering rates; it is bounded from above by the estimated traffic demand on the ramp introduced in Section 3.1. Such a treatment can prevent the agent from picking up a metering rate that is higher than the ramp traffic demand, hence may enhance the learning efficiency. Thus, the action space at any time step is state-dependent. To emphasize this point, the action space in this paper is written as $A(\mathbf{s})$, as will be seen in the remainder of this paper.

3.3. Reward. The rewards earned by the intelligent ramp meter agent during learning should reflect the objective of the ramp metering policy to be learned. As introduced in Section 3.1, the objective of the ramp metering policy to be learned is to maintain the traffic density of the control target location, ρ_3, to stay close to the desired value, $(\lambda_2/\lambda_1)\rho^{\text{cr}}$. Therefore, the reward function can be defined as

$$r = k\left|\rho_3 - \frac{\lambda_2}{\lambda_1}\rho^{\text{cr}}\right|. \quad (2)$$

In (2), r is the reward received by the agent for resulting in ρ_3; k is a user-defined negative constant value, serving as a scaling factor; the other notations have been defined earlier. The implication of this reward is straightforward: it penalizes the traffic density of the control target location for deviating from the desired value. Similar reward designs have been applied by [20, 23, 29, 30]. In our approach, the reward is a function of the state resulting from taking an action; but, in general, depending on needs, the reward can be a function of the states both before and after taking an action, as well as the action itself [34].

Note that although the reward defined by (2) is based on the state of the current time step, reinforcement learning aims to maximize the total of these rewards over the entire control horizon. There also exist traffic flow optimization methods which optimize performance measures that are solely based on the current traffic state but repeat the optimization at every time step, e.g., [37, 38]. These two approaches are different.

3.4. Value Function Approximation by an Artificial Neural Network. If a lookup table method was to be used, the four-dimensional continuous state space needs to be approximated (discretized) first. If, for example, using the simple aggregation method for approximating the continuous state space, the range of the traffic density is aggregated into 40 intervals and the range of the estimated traffic demand on the ramp is aggregated into 20 intervals, then there will be as many as $40^3 \times 20$, i.e., 1.28 million discrete states. Then, if the action space consists of 20 metering rates, it implies that the dimension of the resulting lookup table will be 1.28 million $\times$ 20. This means that there will be a total of 25.6 million action values (i.e., Q-values) to learn, which will be computationally very demanding. This motivates the introduction of value function approximation.

We use an artificial neural network (ANN) to serve as the approximate value function. The role of this approximate value function in the Q-learning process is at each time step, it takes as inputs all the state variables, i.e., ρ_1, ρ_2, ρ_3, and

D_{ramp}, based on which it computes the values for all the available actions, as outputs. That is, the approximate value function maps the state vector to another vector, each element of which is the value of the pair of that state and a candidate action. In general, a value function approximated by an ANN is a nonlinear mapping:

$$\text{ANN: } \mathbb{R}^{|S|} \longrightarrow \mathbb{R}^{|A|}. \tag{3}$$

In (3), ANN represents the value function approximated by an ANN and $|S|$ and $|A|$ denote the dimensions of the state space and action space, respectively.

3.4.1. State Encoding. In many cases, the state variables are not directly fed into ANNs; they are first transformed into some other variables called features [34, 39], which will then be taken by ANNs. Such a transformation is known as state encoding or feature extraction [34, 39]. As pointed out by Bertsekas [39], state encoding can be instrumental in the success of value function approximation, and with good state encoding, an ANN need not to be very complicated. The state encoding method used by this study is a simple tile coding method [34], which is described as follows. For each of the four continuous state variables, its value range is divided into equal intervals that do not overlap with each other; as a result, at any time step, the value of a state variable will fall into one of the intervals that collectively cover the value range of this state variable; the interval into which the value of this state variable falls will be given value 1, while all the others will be given value 0. Such a state encoding treatment can give the ANN stronger stimuli than a treatment that normalizes state variables to have continuous values between 0 and 1. To emphasize the fact that the feature vector is a function of the state vector, in this paper, the feature vector is written as $\mathbf{x}(\mathbf{s})$, as can be seen in the remainder of this paper.

3.4.2. Structure of the Value Function Approximated by the ANN. The feature vector, $\mathbf{x}(\mathbf{s})$, is then taken by the ANN. The ANN works in the following way. First, through a linear mapping which is specified by a weight matrix, $\mathbf{W}$, it generates the so-called raw values [40]. Subsequently, each of these raw values is transformed by a nonlinear function, e.g., a sigmoid function, to obtain the so-called threshold values [40]. Such a nonlinear transformation is also known as activation [41]. Then, the threshold values are transformed again through a linear mapping which is specified by another weight matrix, $\mathbf{V}$. Finally, the newly transformed values are added by a vector of coefficients, $\mathbf{c}$, known as the bias coefficients [40], yielding the outputs from the ANN, i.e., the vector of action values, $\mathbf{q}$. Note that the dimension of $\mathbf{c}$ is equal to the number of actions. Therefore, we see that the ANN is characterized by three sets of parameters, i.e., $\mathbf{W}$, $\mathbf{V}$, and $\mathbf{c}$. In other words, the value function approximated by the ANN is parameterized by $\mathbf{W}$, $\mathbf{V}$, and $\mathbf{c}$. The mapping from the input state vector to the output action-value vector can thus be written in a compact form as

$$\mathbf{q} = \text{ANN}(\mathbf{x}(\mathbf{s}); \mathbf{W}, \mathbf{V}, \mathbf{c}). \tag{4}$$

The structure of the ANN described above is presented in Figure 2. The three sets of parameters, $\mathbf{W}$, $\mathbf{V}$, and $\mathbf{c}$, are unknown and need to be learned through the Q-learning process. The algorithm used for achieving this is presented in Section 3.5.

3.4.3. Benefit in Computational Cost. It is worth demonstrating the benefit in computational cost brought by introducing the ANN approximate value function. Recall that we have estimated the computational cost of the lookup table method in the beginning of Section 3.4. To enable a "fair" comparison with the lookup table method, for the ANN approximate value function, we also assume that the value range of each traffic density variable is divided into 40 intervals, and the value range of the estimated traffic demand on the ramp is divided into 20 intervals. This implies that there will be a total of $40 \times 3 + 20$, i.e., 140 state features. We further assume that the number of hidden nodes is specified as 3 times of the number of features, which has been found to be sufficient to yield good learning outcomes in this study. This implies that the dimension of the weight matrix $\mathbf{W}$ will be 140×420. We still assume that there are 20 available metering rates, as in the lookup table case. This implies that the dimension of the weight matrix $\mathbf{V}$ will be 420×20, and the dimension of the bias coefficient vector $\mathbf{c}$ will be 20. As a result, there will be a total of 67,220 unknown parameters to learn. Compared with the 25.6 million action values (i.e., Q-values) to learn for the lookup table method, the benefit in computational cost brought by the value function approximation is tremendous.

3.5. The Learning Algorithm. As shown above, thanks to the approximate value function, the computational cost of learning can be profoundly reduced. The price is that the learning algorithm will no longer be as straightforward as lookup table methods. For a lookup table method, for any encountered state-action pair, the new Q-value computed by the so-called temporal difference (TD) rule is directly used to replace the original Q-value in the lookup table. In general, the TD rule of Q-learning is defined as [34]

$$Q_{\text{new}}(\mathbf{s}, a) = (1 - \alpha) Q_{\text{old}}(\mathbf{s}, a) + \alpha \left(r(\mathbf{s}, a, \mathbf{s}')) + \gamma \max_{b \in A(\mathbf{s}')} Q(\mathbf{s}', b) \right). \tag{5}$$

In (5), $\mathbf{s}$ and $\mathbf{s}'$ denote states before and after taking the action, respectively; a and b denote actions; A is the state-dependent action space; r represents the reward received by the agent moving from state $\mathbf{s}$ to state $\mathbf{s}'$ by taking action a; α is the learning rate; and γ is the discounting factor. In our approach, the reward r depends only on the state after taking the action, as described in Section 3.3.

For a value function approximation-based method, however, replacements of Q-values in a lookup table are no longer applicable as there is not a lookup table at all; instead, at each time step, the original and new Q-values are jointly

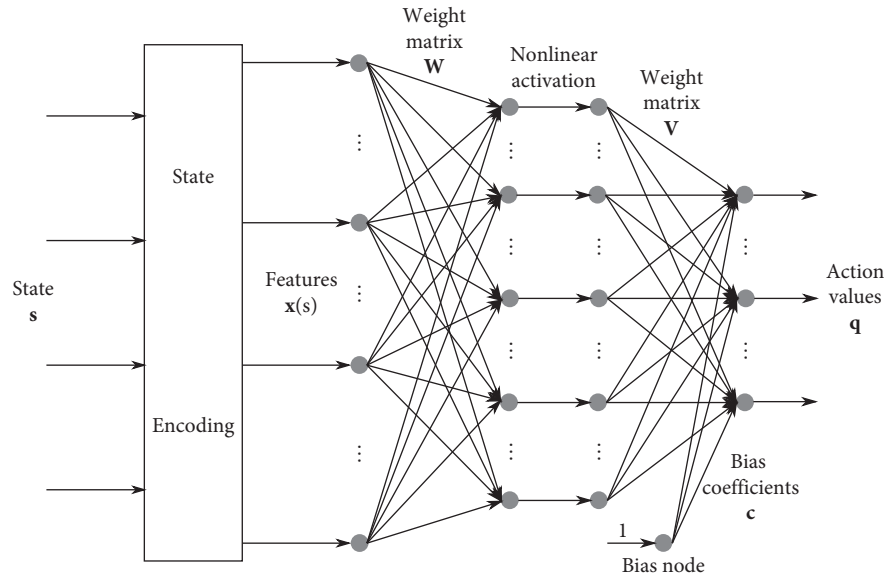

Figure 2: Structure of the artificial neural network that serves as the approximate value function.

used to update the parameters of the approximate value function. In other words, unlike a lookup table method for which a final lookup table filled by converged Q-values will be the ultimate outcome of the learning process, a value function approximation-based method uses Q-values as training data to calibrate the parameters of the approximate value function, and the Q-values will not be part of the ultimate outcome of the learning process. This is a distinct difference between the two kinds of methods. It is worth noting that the calibration of the parameters of the approximate value function is itself a learning problem. Specifically, it is an incremental supervised learning problem. It is incremental as information encapsulated in the datum generated at each time step (i.e., the new Q-value) needs to be absorbed by the parameters as soon as it becomes available. It is supervised as the target output (i.e., the new Q-value) for the approximate value function (i.e., the ANN in this study) is specified at each time step. The ANN calibration method employed in this study is the so-called incremental backpropagation algorithm [40].

The above process is formally presented by Algorithm 1, the pseudocode of the algorithm of Q-learning with ANN value function approximation used for this study. There are two minor abuses of notations in Algorithm 1 for convenience of presentation. First, by $\text{argmax}_{a\in A(\mathbf{s})}\text{ANN}(\mathbf{x}(\mathbf{s}); \mathbf{W}, \mathbf{V}, \mathbf{c})$, we mean the metering rate of the highest action-value among all admissible metering rates under the current state $\mathbf{s}$. Second, similarly, by $\max_{a\in A(\mathbf{s})}\text{ANN}(\mathbf{x}(\mathbf{s}); \mathbf{W}, \mathbf{V}, \mathbf{c})$, we mean the highest admissible action-value under the current state $\mathbf{s}$.

4. Assessments

4.1. Experiment Settings. This section evaluates the effects of the proposed reinforcement learning approach. The layout of the experiment freeway section is illustrated in Figure 3. As shown in Figure 3, a lane-drop is located as far as 3500 meters downstream of the metered ramp. Before the lane-drop, there are 3 lanes in the mainline, and after that, there are 2 lanes in the mainline. The ramp has only one lane.

The classical first-order discrete-time macroscopic model of traffic flow dynamics, the cell transmission model (CTM) [16], is employed as the simulation model. The free-flow speed is set as 120 km/h, the critical density is set as 20 veh/km/lane, and the jam density is set as 100 veh/km/lane. The flow-density fundamental diagram employed is triangular. Thus, the capacity of one lane is $120 \times 20 = 2400$ veh/h. Since the number of lanes before and after the lane-drop is 3 and 2, respectively, and the critical density is 20 veh/km/lane, the desired traffic density for the control target cell is $(2/3) \times 20 = 13.33$ veh/km/lane.

In general, it may not be possible to quantify the threshold distance value between the metered ramp and the downstream bottleneck that will fail a conventional linear feedback-type ramp metering controller, as this value may vary from case to case, depending on factors including the free-flow speed and design of the linear feedback controller. For the specific experiment environment as described above, we found that a proportional-integral (PI) controller, which is a conventional linear feedback-type controller and can work well for close bottlenecks, will no longer be stable if the distance between the metered ramp and the downstream lane-drop location exceeds 1000 meters.

Traffic demands of the mainline and ramp are given in Figure 4. This demand profile is similar to what was used in [18, 23, 29–31]. It is assumed in this study that the traffic flow is composed of only passenger cars. Multiclass traffic flow cases are not considered in this study. Note that, in order for the problem to be meaningful, the mainline demand should not exceed the mainline capacity after the lane drop, for otherwise the ramp metering cannot help in anyway.

The method described in Section 3.4.1 is applied for state encoding. The value range of each of the three traffic density

Data: mainline and ramp traffic demands
Result: calibrated parameters of the artificial neural network
Initialization: set $\mathbf{W}$, $\mathbf{V}$, and $\mathbf{c}$ to small random numbers [40].
while *episode reward not yet converged* **do**
  Set the freeway network of interest as empty
  Initialize the state $\mathbf{s}$
  while *not the end of this episode* **do**
(1) Determine ramp metering rate a according to the ϵ−greedy strategy: $a \longleftarrow \text{argmax}_{a \in A(\mathbf{s})} \text{ANN}(\mathbf{x}(\mathbf{s}); \mathbf{W}, \mathbf{V}, \mathbf{c})$ or $a \longleftarrow a$ is a random element in $A(\mathbf{s})$
(2) Simulate to obtain the new state $\mathbf{s}'$, with a implemented.
(3) Compute reward r based on $\mathbf{s}'$
(4) Compute Q_{old} by the ANN: $Q_{\text{old}} \longleftarrow \max_{a \in A(\mathbf{s})} \text{ANN}(\mathbf{x}(\mathbf{s}); \mathbf{W}, \mathbf{V}, \mathbf{c})$
(5) Compute Q_{next} by the ANN: $Q_{\text{next}} \longleftarrow \max_{a \in A(\mathbf{s}')} \text{ANN}(\mathbf{x}'(\mathbf{s}'); \mathbf{W}, \mathbf{V}, \mathbf{c})$
(6) Compute Q_{new} by updating Q_{old} using the temporal difference rule $Q_{\text{new}} \longleftarrow (1 - \alpha) Q_{\text{old}} + \alpha (r + \gamma Q_{\text{next}})$
(7) Update the parameters of the ANN by the incremental backpropagation algorithm using Q_{old} as the input to the ANN and Q_{new} as the desired output [40]: $\mathbf{W}, \mathbf{V}, \mathbf{c} \longleftarrow \text{Backpropagation}(Q_{\text{old}}, Q_{\text{new}}, \mathbf{W}, \mathbf{V}, \mathbf{c})$
(8) Update the state $\mathbf{s} \longleftarrow \mathbf{s}'$
  end
end

Algorithm 1: Pseudocode of the algorithm of Q-learning with value function approximated by an artificial neural network.

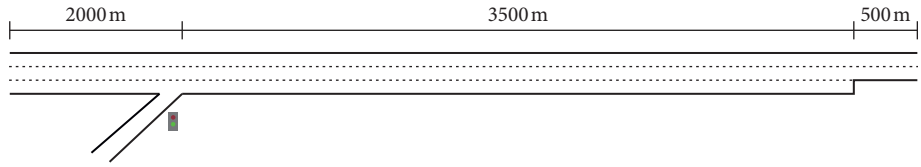

Figure 3: Layout of the freeway section used for assessment.

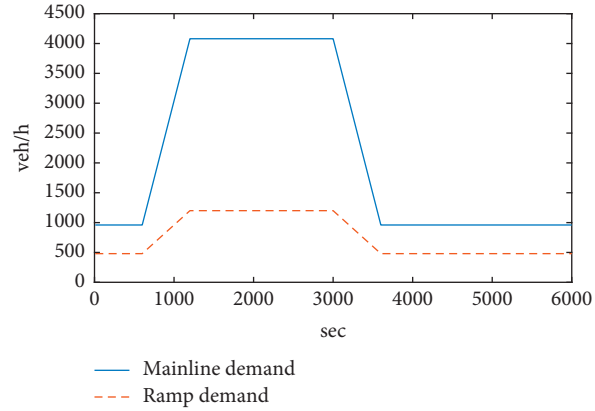

Figure 4: Traffic demands for the mainline and the ramp.

variables, $[0, \rho^{\text{jam}}]$, is equally divided into 40 intervals. The value range of the estimated traffic demand on the ramp is divided into 20 intervals. Unlike the value range of any traffic density variable which has an explicit fixed upper bound (i.e., ρ^{jam}), it is not that straightforward to specify a proper upper bound for the value range of the estimated traffic demand on the ramp. We could specify a very large upper bound to ensure that any estimated traffic demand on the ramp will fall within the value range. However, this can cause the estimated traffic demand on the ramp to be much lower than the specified upper bound for most of the times, hence may not be efficient. To handle this issue, it is worth recalling the purpose of state encoding: to facilitate the efficiency of learning through translating the state variable into some other variable(s) that is(are) more representable under the specific learning task. Here, the learning task is to determine the ramp metering rate which is bounded by the highest allowable value, $a_{\max}$, regardless of the traffic demand on the

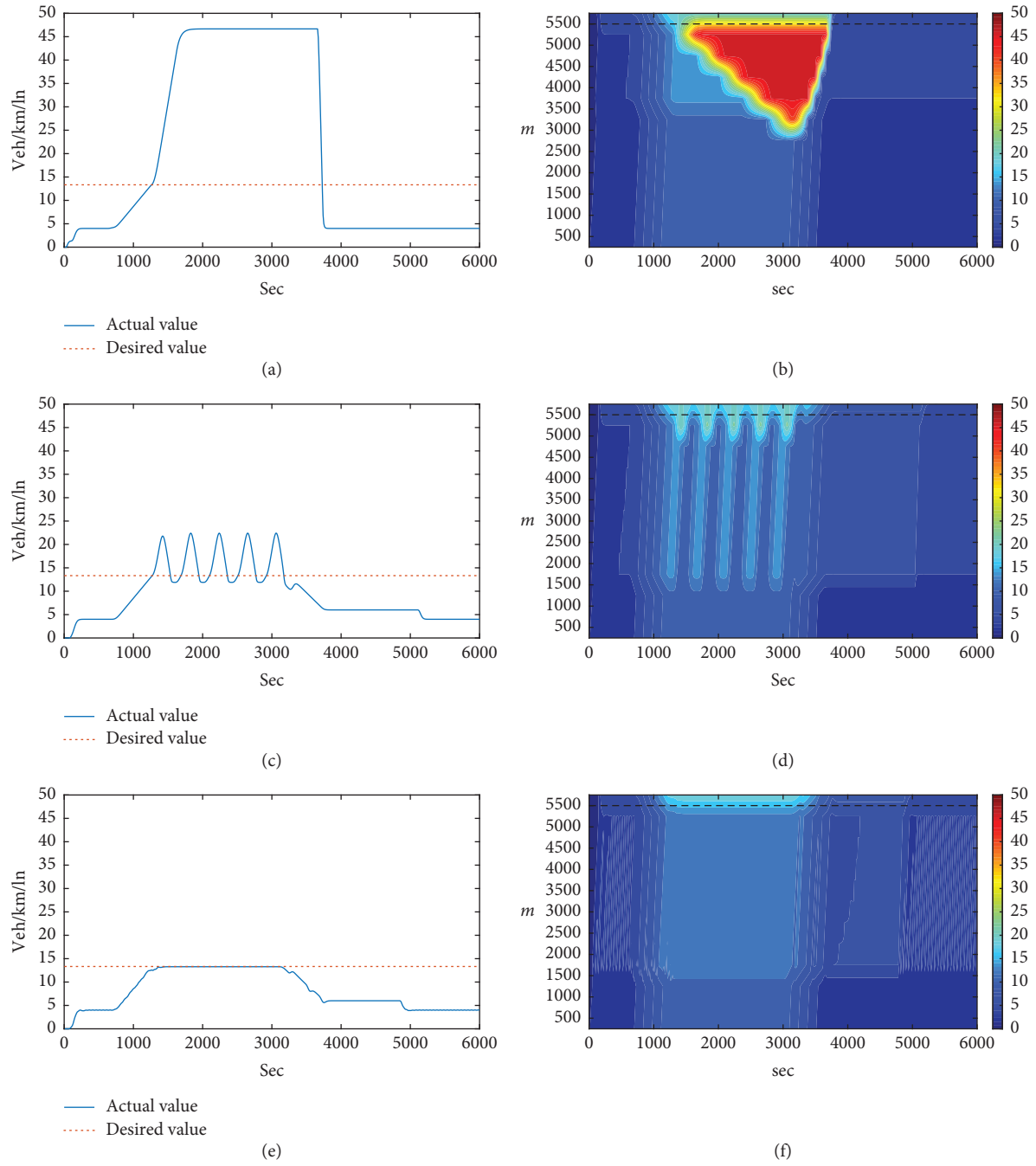

FIGURE 5: Comparison of traffic density time series of the control target cell (left column) and traffic density contours (right column), respectively, among the no control case (top row), the PI controller case (middle row), and the case of the proposed approach (bottom row).

ramp. Therefore, a reasonable way to discretize the value range of the estimated traffic demand on the ramp is as follows: the range $[0, a_{\max}]$ is equally divided into 19 intervals; the range $(a_{\max}, \infty)$ accounts for the last interval. The above state encoding treatment converts the four-dimensional state vector of continuous variables into a 140-dimensional $(40 \times 3 + 20 = 140)$ feature vector of binary variables.

In this experiment, the lowest allowable metering rate, $a_{\min}$, is set as 200 veh/h, and the highest allowable metering rate, $a_{\max}$, is set as 1200 veh/h. The range $[a_{\min}, a_{\max}]$ is equally divided into 10 intervals, resulting in a total of 11 discrete metering rates: $\{200, 300, \ldots, 1100, 1200\}$ veh/h. This specification for the action space is determined following the so-called "full traffic cycle" signal policy for ramp metering [36] to ensure that the optimal metering rates learned through the proposed method can be implemented by a traffic light. Note that $\{200, 300, \ldots, 1100, 1200\}$ veh/h is the largest admissible action space. As introduced in Section 3.2, in the proposed approach, at any time step, the

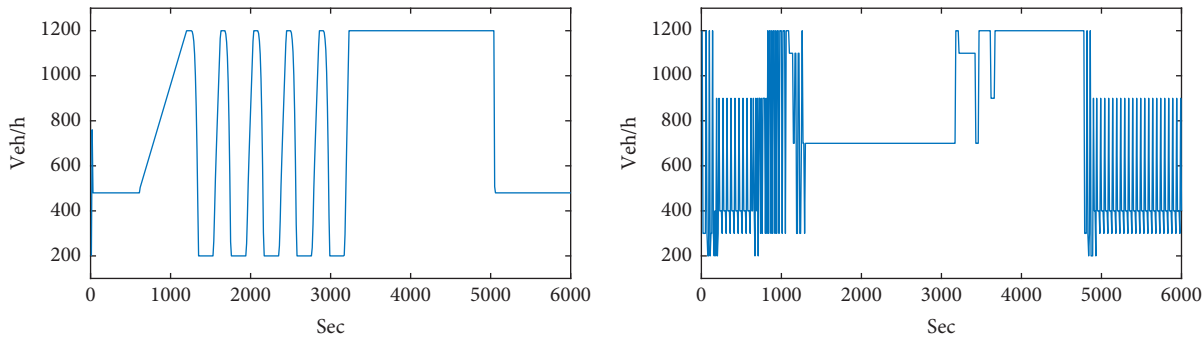

Figure 6: Comparison of ramp metering rates computed by the PI controller (a) and by the policy learned through the proposed approach (b).

admissible action space can be smaller than the largest set because it is constrained by the estimated traffic demand on the ramp.

The hyperparameters used in the experiments are specified as follows. The number of hidden neurons is set as 3 times of the features, i.e., $3 \times 140 = 420$. The determination of this number was based on a considerable amount of trial-and-error experiments. If this number is set too big, the training time would be excessively long; if it is set too small, the approximate value function would not be able to effectively discriminate state inputs. The learning rate, α, of TD updating rule (5) is set as such that, for the first 0.1 million episode iterations, it is equal to 0.05, and it is equal to 0.01 afterwards. The discounting factor, γ, of TD updating rule (5) is set as 0.95. The exploration rate, ε, in the ε-greedy policy in Algorithm 1 is set as decaying with the increase of the number of iterated episodes [34].

4.2. Results. The experiment was coded and executed by MATLAB R2019a. Learning converged after about 0.7 million of episodic iterations. The left column of Figure 5 presents the resulting traffic density time series of the control target cell for the case of no control, the case of a PI controller (which is a conventional linear feedback-type controller), and the case of the proposed reinforcement learning approach; the right column of Figure 5 illustrates the traffic density contours of the entire freeway section for the three cases. The black dash line in each traffic density contour indicates the location of the lane-drop; the origin of the y-axis of each traffic density contour corresponds to the beginning location of the concerned freeway section as depicted in Figure 3. From Figure 5, it can be seen that, without any control measure, as traffic demands increase, the traffic density of the control target cell soon grows beyond the desired value, and hence, congestion initiates from the bottleneck and grows into the upstream. Under the PI ramp metering control, the traffic density of the control target cell can be maintained around the desired value in the large, however, with severe oscillations which propagate into the upstream and influence the whole section. Under the ramp metering policy learned through the proposed reinforcement learning approach, the traffic density of the control target cell is managed to stay close to the desired value with almost no fluctuations, and accordingly, the traffic density contour of the entire section is much smoother than the case of the PI controller.

Figure 6 compares the ramp metering rates computed by the PI controller (Figure 6(a)) and by the policy learned through the proposed reinforcement learning approach (Figure 6(b)). It indicates that the patterns of the two sets of metering rates are quite different. Moreover, microscopically, the metering rates given by the learned policy are very shredded in order to avoid the potential time-delay effects due to the long distance, thanks to the facts that it is a highly nonlinear feedback policy and takes in traffic conditions at multiple locations along the stretch. It is these shredded metering rates that manage to stabilize the traffic density of the control target cell around the desired value with almost no fluctuations, as shown in Figure 5. By contrast, the metering rates given by the PI controller lack subtle variations but can only constantly oscillate with large amplitudes, which results in quite unstable traffic densities of the control target cell, as shown in Figure 5.

4.3. Robustness. It is of interest to what extent the learned ramp metering policy can tolerate uncertainties in traffic demands. To this end, the traffic demands are corrupted by white noise. Figure 7 presents the results for the cases in which the standard deviation of the white noise of the traffic demands is 50, 100, 150, 200, and 250 veh/h, respectively. It can be seen that the metering policy learned from the proposed approach can perform satisfactorily up to the noise level of 200 veh/h; its performance starts to go down as the demand noise grows bigger.

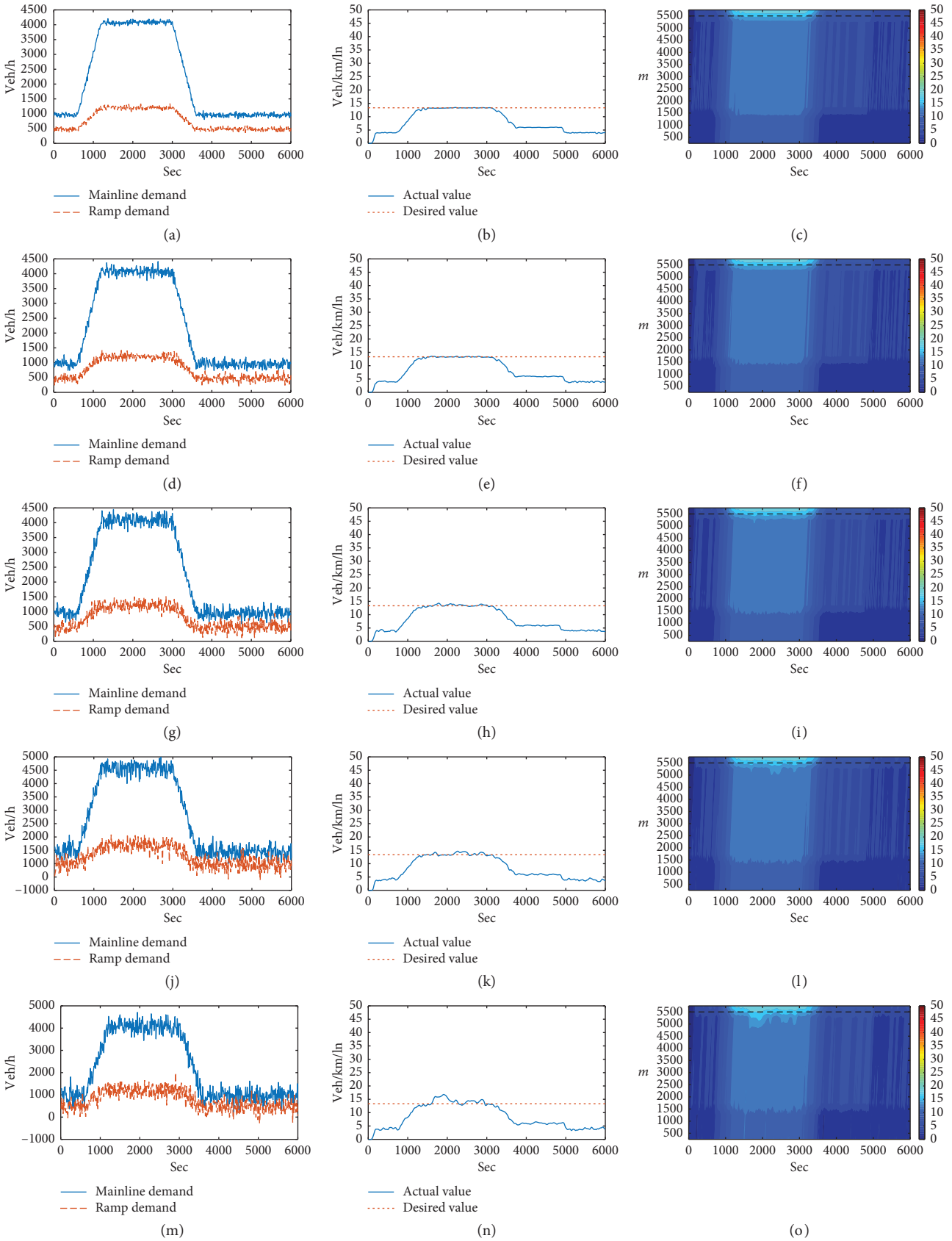

Figure 7: Performances of the ramp metering policy learned through the proposed approach under traffic demands with white noise. From the top row to the bottom row, the standard deviation of the white noise is 50, 100, 150, 200, and 250 veh/h, respectively. The left column is traffic demands, the middle column is traffic density time series of the control target cell, and the right column is the traffic density contours of the entire section.

5. Conclusions

This paper proposes a reinforcement learning approach to learn an optimal ramp metering policy controlling a downstream bottleneck that is far away from the metered ramp. An artificial neural network replaces the lookup table in the ordinary Q-learning approach to serve as the approximate value function. The state vector is chosen so that a tradeoff between the capability to anticipate traffic flow evolution and the computational cost is achieved. The action space is state-dependent to enhance the learning efficiency. A simple tile coding method is employed to convert the continuous state vector to a binary feature vector to give stronger stimuli to the artificial neural network. The experiment results indicate that the ramp metering policy learned through the proposed approach is able to yield clearly more stable results than a conventional linear feedback-type controller. Specifically, under the learned ramp metering policy, the traffic density of the control target cell is successfully maintained to stay close to the desired value with almost no fluctuations. As a result, traffic flow evolution over the entire freeway section is also smooth. In comparison, under a conventional linear feedback-type ramp metering strategy, the traffic density of the control target cell oscillates significantly around the desired value. Consequently, traffic flow evolution over the entire freeway section also suffers from significant instability. The metering policy learned through the proposed approach has also demonstrated some level of robustness in terms of yielding satisfactory results under uncertain traffic demands.

For the next step, we plan to extend the proposed method so that it can manage queue length on the ramp at the expense of trading off some mainline efficiency. Another interesting direction is to replace the artificial neural network approximate value function by a simpler linear approximate value function but with employing more sophisticated state encoding techniques to better capture the interactions among the state variables so that a sophisticated approximate value function such as an ANN may be avoided. It will also be interesting to examine the impact of the number of representative mainline sampling locations, especially under the circumstances of excessively long distance between the ramp and the downstream bottleneck and complicated traffic demand patterns. Finally, we will also look into the approach of policy approximation as an alternative to the action-value approximation approach in this paper.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors thank Prof. Abhijit Gosavi of Missouri University of Science and Technology for his comments on the manuscript. This work was funded by the C2SMART University Transportation Center under USDOT award no. 69A3551747124.

References

[1] M. Papageorgiou and A. Kotsialos, "Freeway ramp metering: an overview," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 4, pp. 271–281, 2002.

[2] M. Papageorgiou, H. Hadj-Salem, J.-M. Blosseville et al., "Alinea: a local feedback control law for on-ramp metering," *Transportation Research Record*, vol. 1320, no. 1, pp. 58–67, 1991.

[3] P. Kachroo and K. Kumar, "System dynamics and feedback control design problem formulations for real time ramp metering," *Journal of Integrated Design and Process Science*, vol. 4, no. 1, pp. 37–54, 2000.

[4] K. Ozbay, I. Yasar, and P. Kachroo, "Modeling and paramics based evaluation of new local freeway ramp metering strategy that takes into account ramp queues," *Transportation Research Record*, vol. 2004, pp. 89–97, 1867.

[5] P. Kachroo and K. Ozbay, "Feedback Ramp Metering for Intelligent Transportation System," Kluwer Academics, New York, NY, USA, 2003.

[6] Y. Wang, E. B. Kosmatopoulos, I. M. Papageorgiou, and I. Papamichail, "Local ramp metering in the presence of a distant downstream bottleneck: theoretical analysis and simulation study," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2024–2039, 2014.

[7] Y. Kan, Y. Wang, M. Papageorgiou, and I. Papamichail, "Local ramp metering with distant downstream bottlenecks: a comparative study," *Transportation Research Part C: Emerging Technologies*, vol. 62, pp. 149–170, 2016.

[8] Felipe de Souza and W. Jin, "Integrating a smith predictor into ramp metering control of freeways," in *Proceedings of the 2017 96th Transportation Research Board Annual Meeting*, New York, NY, USA, 2017.

[9] J. R. D Frejo and B. De Schutter, "Feed-forward alinea: a ramp metering control algorithm for nearby and distant bottlenecks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 7, pp. 2448–2458, 2018.

[10] H. Yu, S. Koga, T. Roux Oliveira, and M. Krstic, "Extremum seeking for traffic congestion control with a downstream bottleneck," 2019.

[11] E. Stylianopoulou, M. Kontorinaki, M. Papageorgiou, and I. Papamichail, "A linear-quadratic-integral regulator for local ramp metering in the case of distant downstream bottlenecks," *Transportation Letters*, vol. 1, 2019.

[12] L. Zhao, Z. Li, Ke Zemian, and Li Meng, "Distant downstream bottlenecks in local ramp metering: comparison of fuzzy self-adaptive pid controller and pi-alinea," in *Proceedings of the 2019 19th COTA International Conference of Transportation Professionals*, pp. 2532–2542, New York, NY, USA, 2019.

[13] L. Zhao, Z. Li, Z. Ke, and M. Li, "Fuzzy self-adaptive proportional-integral-derivative control strategy for ramp metering at distance downstream bottlenecks," *IET Intelligent Transport Systems*, vol. 14, no. 4, pp. 250–256, 2020.

[14] M. Papageorgiou, J.-M. Blosseville, and H. Hadj-Salem, "Modelling and real-time control of traffic flow on the southern part of boulevard peripherique in paris: Part i:

Modelling," *Transportation Research Part A: General*, vol. 24, no. 5, pp. 345–359, 1990.

[15] C. Meyer, D. E. Seborg, and R. K. Wood, "A comparison of the smith predictor and conventional feedback control," *Chemical Engineering Science*, vol. 31, no. 9, pp. 775–778, 1976.

[16] C. Daganzo, "The cell transmission model. part i: a simple dynamic representation of highway traffic," *Transportation Research Part B: Methodology*, vol. 31, 1994.

[17] K. Wen, S. Qu, and Y. Zhang, "A machine learning method for dynamic traffic control and guidance on freeway networks," in *Proceedings of the 2009 International Asia Conference on Informatics in Control, Automation and Robotics*, vol. 67–71, New York, NY, USA, 2009.

[18] M. Davarynejad, A. Hegyi, J. Vrancken, and J. van den Berg, "Motorway ramp-metering control with queuing consideration using q-learning," in *Proceedings of the 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, New York, NY, USA, 2011.

[19] K. Veljanovska, Z. Gacovski, and S. Deskovski, "Intelligent system for freeway ramp metering control," in *Proceedings of the 2012 6th IEEE International Conference Intelligent Systems*, pp. 279–282, New York, NY, USA, 2012.

[20] F. Ahmed and W. Gomaa, "Freeway ramp-metering control based on reinforcement learning," in *Proceedings of the 11th IEEE International Conference on Control & Automation (ICCA)*, pp. 1226–1231, New York, NY, USA, 2014.

[21] F. Ahmed and W. Gomaa, "Multi-agent reinforcement learning control for ramp metering," in *Progress in Systems Engineering*, pp. 167–173, Springer, Berlin, Germany, 2015.

[22] E. Ivanjko, D. Koltovska Nečoska, M. Gregurić, M. Vujić, G. Jurković, and S. Mandžuka, "Ramp metering control based on the q-learning algorithm," *Cybernetics and Information Technologies*, vol. 15, no. 5, pp. 88–97, 2015.

[23] Z. Li, P. Liu, C. Xu, H. Duan, and W. Wang, "Reinforcement learning-based variable speed limit control strategy to reduce traffic congestion at freeway recurrent bottlenecks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 11, pp. 3204–3217, 2017.

[24] C. Jacob and B. Abdulhai, "Integrated traffic corridor control using machine learning," in *Proceedings of the 2005 IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, pp. 3460–3465, New York, NY, USA, 2005.

[25] C. Jacob and B. Abdulhai, "Automated adaptive traffic corridor control using reinforcement learning: approach and case studies," *Transportation Research Record*, vol. 1, no. 1, pp. 1–8, 2006.

[26] C. Jacob and B. Abdulhai, "Machine learning for multi-jurisdictional optimal traffic corridor control," *Transportation Research Part A: Policy and Practice*, vol. 44, no. 2, pp. 53–64, 2010.

[27] K. Rezaee, B. Abdulhai, and H. Abdelgawad, "Application of reinforcement learning with continuous state space to ramp metering in real-world conditions," in *Proceedings of the 2012 15th International IEEE Conference on Intelligent Transportation Systems*, New York, NY, USA, 2012.

[28] K. Rezaee, B. Abdulhai, and H. Abdelgawad, "Self-learning adaptive ramp metering," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2396, no. 1, pp. 10–18, 2013.

[29] T. Schmidt-Dumont and J. H van Vuuren, "Decentralised reinforcement learning for ramp metering and variable speed limits on highways," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 8, p. 1, 2015.

[30] C. Wang, J. Zhang, L. Xu, L. Li, and B. Ran, "A new solution for freeway congestion: cooperative speed limit control using distributed reinforcement learning," *IEEE Access*, vol. 7, pp. 41947–41957, 2019.

[31] E. Walraven, M. T. J. Spaan, and B. Bakker, "Traffic flow optimization: a reinforcement learning approach," *Engineering Applications of Artificial Intelligence*, vol. 52, p. 203, 2016.

[32] F. Belletti, D. Haziza, G. Gomes, and A. M. Bayen, "Expert level control of ramp metering based on multi-task deep reinforcement learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 4, pp. 1198–1207, 2017.

[33] Y. Wu, H. Tan, and B. Ran, "Differential variable speed limits control for freeway recurrent bottlenecks via deep reinforcement learning," 2018.

[34] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, New York, NY, USA, 2018.

[35] R. E. Bellman, *Adaptive Control Processes: A Guided Tour*, Princeton University Press, New York, NY, USA, 2015.

[36] M. Papageorgiou and I. Papamichail, "Overview of traffic signal operation policies for ramp metering," *Transportation Research Record*, vol. 204, no. 1, pp. 28–36, 2008.

[37] J. Zhao, W. Ma, Y. Liu, and K. Han, "Optimal operation of freeway weaving segment with combination of lane assignment and on-ramp signal control," *Transportmetrica A: Transport Science*, vol. 12, no. 5, pp. 413–435, 2016.

[38] C. Zhang, N. R. Sabar, E. Chung, A. Bhaskar, and X. Guo, "Optimisation of lane-changing advisory at the motorway lane drop bottleneck," *Transportation Research Part C: Emerging Technologies*, vol. 106, pp. 303–316, 2019.

[39] D. P. Bertsekas, *Reinforcement Learning and Optimal Control*, Athena Scientific Belmont, Massachusetts, MA, USA, 2019.

[40] A. Gosavi, *Simulation-Based Optimization: Parametric Optimization Techniques and Reinforcement Learning*, Springer, Berlin, Germany, 2015.

[41] J.-A. Gosavi, *Probabilistic Machine Learning for Civil Engineers*, MIT Press, New York, NY, USA, 2020.