CHAO LU, Ph.D.[1]
E-mail: chaolu@bit.edu.cn
JIE HUANG, Ph.D.[2]
(Corresponding author)
E-mail: huangjie@igsnrr.ac.cn
JIANWEI GONG, Ph.D.[1]
E-mail: gongjianwei@bit.edu.cn
[1] School of Mechanical Engineering, Beijing Institute of Technology
5 Zhongguancun South Street, Haidian District, Beijing 100081, China
[2] Institute of Geographic Sciences and Natural Resources Research, Chinese Academy of Sciences, 11 A Datun Road, Chaoyang District, Beijing 100101, China

Traffic on Motorways
Preliminary Communication
Submitted: May 12, 2015
Accepted: Feb. 10, 2016

REINFORCEMENT LEARNING FOR RAMP CONTROL: AN ANALYSIS OF LEARNING PARAMETERS

ABSTRACT

Reinforcement Learning (RL) has been proposed to deal with ramp control problems under dynamic traffic conditions; however, there is a lack of sufficient research on the behaviour and impacts of different learning parameters. This paper describes a ramp control agent based on the RL mechanism and thoroughly analyzed the influence of three learning parameters; namely, learning rate, discount rate and action selection parameter on the algorithm performance. Two indices for the learning speed and convergence stability were used to measure the algorithm performance, based on which a series of simulation-based experiments were designed and conducted by using a macroscopic traffic flow model. Simulation results showed that, compared with the discount rate, the learning rate and action selection parameter made more remarkable impacts on the algorithm performance. Based on the analysis, some suggestions about how to select suitable parameter values that can achieve a superior performance were provided.

KEY WORDS

reinforcement learning; Q-learning; ramp control; agent; macroscopic traffic flow model;

1. INTRODUCTION

After more than 50 years of application, ramp control (or ramp metering) has been identified as one of the most effective control methods on motorways [1]. The ramp control mentioned here refers to the on-ramp control. This control method uses signal devices named ramp meters at on-ramps to regulate the ramp metering rate which is usually defined as the number of vehicles entering the motorway mainline during each signal cycle. Through suitable regulations on the metering rate, a ramp control strategy aims to alleviate motorway congestions, improve motorway throughput, and thus reduce the travel time spent by road users [2]. Over the last decades, a number of control strategies have been proposed to achieve this goal, such as capacity-density method [3], ALINEA [4] and the model-based optimization approaches (e.g. model predictive control methods [5, 6]). Among these strategies, the model-based optimization method has become increasingly popular in recent studies, as it is sound and can solve the ramp control problems based on the optimization theory. However, this method is dependent on the model accuracy and usually requires high computational demand, which limits its field of application [7].

In order to overcome these limitations, reinforcement learning (RL) was recently proposed by Jacob and Abdulhai [7, 8] to solve ramp control problems based on the Markov decision process (MDP) and dynamic programming (DP). After this contribution, some recent studies have also shown the effectiveness of RL for ramp control under different settings and conditions. For instance, coordinated ramp control using RL is considered in [9], continuous state space was analyzed in [10], and indirect RL was tested in [11, 12]. Although some efforts have been made to explore the application of RL in the ramp control domain, the issues of how to set the parameters for RL based ramp control strategies and how these settings influence the algorithm performance have not been widely studied. In most of these studies, the learning parameters are set according to experience without analysis. To our knowledge, the only published work related to the analysis of learning parameters for ramp control is shown in [13]. This work provides some useful suggestions about how to select suitable parameters in a continuous-state case with some adaptive settings. However, the behaviour of different parameter values

and their impacts on the algorithm performance have not been thoroughly analyzed. Moreover, this work is based on a microscopic simulation package which may affect the observed results for different parameter settings because of its stochastic behaviour. In this paper, the aim is to develop a ramp control agent following the RL mechanism, based on which the influence of learning parameters with different value settings is well analyzed. Unlike [13], a macroscopic traffic flow model with deterministic demand is used to avoid the stochastic influence of models, so that we can examine the algorithm tractably.

The remainder of this paper is organized as follows. Section 2 introduces the basic knowledge of RL including the Q-learning algorithm and relevant parameters. The description of the analyzed ramp control agent and applied traffic flow model is shown in Section 3. After that, Section 4 presents the simulation test for the algorithm performance under different parameter settings. Finally, Section 5 gives some conclusions and possible directions of the future work.

2. REINFORCEMENT LEARNING

In an RL problem, the learning process is conducted through the interaction between an agent and its external environment as shown in *Figure 1*. The environment is usually represented by a group of states, and the agent can capture the environment changes through these states. The reward can be either positive or negative which can be considered as an encouragement or a penalty for the actions executed by the agent. The objective of an agent is to obtain the maximum cumulative reward (the sum of all rewards received) after executing a sequence of actions following some sort of policy [14].

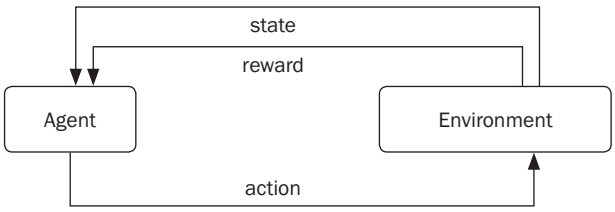

Figure 1 – Agent-environment interaction

2.1 Q-learning

Given a policy π, the cumulative reward can be estimated for each state and action pair (s,c) by the action-value function (or Q function) $Q^{\pi}(s,c)$:

$$Q^{\pi}(s,c) = E\left\{ \sum_{k=0}^{\infty} \gamma^{(k)} R(s^k,c^k) \middle| s^0 = s, c^0 = c, \pi \right\} \quad (1)$$

where $E\{\cdot\}$ gives the expected value of the discounted sum of rewards, k is the time index. $R(s^k,c^k)$ is the reward function which generates the immediate reward for the agent performing action c^k at state s^k and $\gamma^{(k)}$ denotes the discount rate to the power of k. In this case, the problem of obtaining the maximum cumulative reward can be transferred to the problem of maximizing the values of Q functions (Q values). To this end, the Q-learning algorithm has been developed and used in a broad range of applications [14, 15]. In a typical Q-learning problem, all Q values are stored in a table named Q table. Following the mechanism of temporal difference learning and Bellman equation, an updating rule (*Equation 2*) can be derived to maximize Q values and update the Q table (refer to [14] for details).

$$\begin{aligned} Q^k(s^{k-1},c^{k-1}) = {} & Q^{k-1}(s^{k-1},c^{k-1}) + \\ & + \alpha\left[R(s^{k-1},c^{k-1}) + \right. \\ & + \gamma \max_{c^k} Q^{k-1}(s^k,c^k) - \\ & \left. - Q^{k-1}(s^{k-1},c^{k-1}) \right] \end{aligned} \quad (2)$$

where $Q^k(s^{k-1},c^{k-1})$ and $Q^{k-1}(s^{k-1},c^{k-1})$ are the Q values for state-action pair (s^{k-1},c^{k-1}) at the k-th step and k-1-th step, respectively, $Q^{k-1}(s^k,c^k)$ is the Q value for the state-action pair (s^k,c^k) at the k-1-th step.

α is named learning rate or step-size parameter that is used to determine how fast Q values can be updated to approach their maximum values [16]. Typically, α is a small positive fraction value within the range between 0 and 1 ($\alpha \in (0,1)$). The discount rate γ takes values in the same interval, i.e. $\gamma \in (0,1)$. In a Q-learning problem, the Q values are updated by combing the immediately received reward and estimated future Q values from experience. The discount rate is used here to determine to what extent the agent will take the future rewards into account.

2.2 Action selection

For an RL-based agent, exploitation and exploration are two basic behaviours [14]. Exploitation means the agent always takes the greedy action that can obtain the maximum Q values according to the existing experience. Exploration is the behaviour that the agent tries non-greedy actions with smaller Q values. These two behaviours are essential for the continuous learning of RL. Exploration can help the agent update the information of greedy actions by capturing new experience. In the meanwhile, exploitation can keep the agent from being interrupted by the new experience.

In order to balance these two behaviours, ε-greedy action selection strategy can be used, which is also one of the most widely used action selection strategies [14]. Specifically, this strategy takes a random non-greedy action ($c^k \neq c^k_{greedy}$) with probability ε and chooses the greedy action ($c^k = c^k_{greedy}$) with probability 1-ε at each state s^k (as shown in *Equation 3*). The greedy action c^k_{greedy} at state s^k is the action corresponding to the maximum Q value at this state.

$$p(c^k \mid s^k) = \begin{cases} \varepsilon, if\ c^k \neq c^k_{greedy}, c^k_{greedy} = \arg\max_{c^k}(Q^{k-1}(s^k, c^k)) \\ 1-\varepsilon, otherwise \end{cases} \quad (3)$$

When ε is bigger, the agent will be more adventurous and always try to explore the unknown actions. This kind of exploration may be good, and better actions may be found much faster than using a conservative strategy. However, it may also interrupt the learning process by trying worse actions too much. Therefore, for using ε-greedy action selection strategy, how to set the action selection parameter ε is very important to the algorithm performance.

Therefore, for any applications of RL with ε-greedy strategy, three learning parameters, learning rate α, discount rate *y* and action selection parameter ε, should be well designed and tested, which is also the main aim of our work.

3. DESCRIPTION OF RAMP AGENT

In this paper, a ramp agent is designed to control the motorway traffic, based on which the analysis of learning parameters is conducted. Section 3 firstly introduces the basic ramp agent design and applied traffic flow model.

3.1 Asymmetric cell transmission model

For the ramp control application, the environment of *Figure 1* refers to the dynamic traffic situation on motorways. Considering the difficulties of real site evaluation, a macroscopic traffic flow model asymmetric cell transmission model (ACTM) is selected for our test. ACTM has been used for ramp control strategies development and evaluation in some recent studies [17-19] because of the low computational demand and stable performance on mimicking real traffic on motorways. Furthermore, this deterministic model can avoid the stochastic influence of the traffic flow model when the ramp agent is analyzed. In this study, we use a discontinuous version of ACTM as shown in [20].

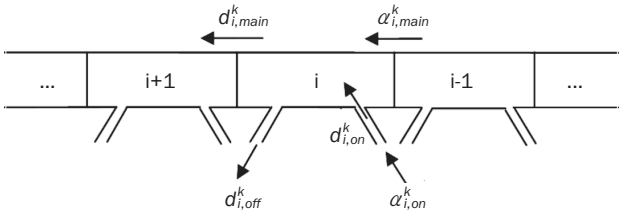

Figure 2 – A typical motorway segment

To apply ACTM, the studied motorway segment should be divided into short stretches that are named cells. Each cell may only contain the mainline, and may also be linked with on- or/and off-ramps. We simplify the expression by naming the cell with on- or/and off-ramps "on- or/and off-ramp cell", the cell without any ramps "normal cell". A typical cell (cell *i*) with one on-ramp and one off-ramp is shown in *Figure 2*, according to which the modified model can be written as follows.

$$d^k_{i,main} = \begin{cases} (1-\beta_i)\cdot\left(\frac{v_i}{l_i}\right)\cdot(q^k_{i,main} + \theta_i \cdot d^k_{i,on}\cdot\Delta t), \\ if\ \rho^k_i \le \rho^c_i\ and\ \rho^k_{i+1} \le \rho \\ \min\{(1-\beta_i)\cdot\left(\frac{v_i}{l_i}\right)\cdot(q^k_{i,main} + \theta_i \cdot d^k_{i,on}\cdot\Delta t); \\ \left(\frac{w_{i+1}}{l_{i+1}}\right)\cdot(q^{\max}_{i+1,main} - q^k_{i+1,main} - \\ \quad - \theta_{i+1}\cdot d^k_{i+1,on}\cdot\Delta t\} \\ if\ \rho^k_i \le \rho^c_i\ and\ \rho^k_{i+1} > \rho^c_{i+1} \\ \lambda \cdot d^{\max}_{i,\min}, if\ \rho^k_i > \rho^c_i\ and\ \rho^k_{i+1} \le \rho \\ \left(\frac{w_{i+1}}{l_{i+1}}\right)\cdot(q^{\max}_{i+1,main} - q^k_{i+1,main} - \\ \quad - \theta_{i+1}\cdot d^k_{i+1,on}\cdot\Delta t), \\ if\ \rho^k_i > \rho^c_i\ and\ \rho^k_{i+1} > \rho^c_{i+1} \end{cases} \quad (4)$$

$$d^k_{i,on} = \begin{cases} \min\left\{\frac{q^k_{i,main} + a^k_{i,on}\cdot\Delta t}{\Delta t}; \right. \\ \left.\frac{\eta_i\cdot(q^{\max}_{i,main} - q^k_{i,main})}{\Delta t}; \frac{c^k_i}{\Delta t}\right\} \\ \text{if } i \text{ is metered on - ramp cell} \\ \min\left\{\frac{q^k_{i,main} + a^k_{i,on}\cdot\Delta t}{\Delta t}; \right. \\ \left.\frac{\eta_i\cdot(q^{\max}_{i,main} - q^k_{i,main})}{\Delta t}\right\}; \\ \text{if } i \text{ is unmetered on - ramp cel} \\ 0, \text{otherwise} \end{cases} \quad (5)$$

$$\begin{cases} q^{k+1}_{i,main} = q^k_{i,main} + \Delta t\cdot(a^k_{i,main} + d^k_{i,on} - d^k_{i,main} - d^k_{i,off}) \\ d^k_{i,off} = \frac{d^k_{i,main}}{1-\beta_i} \end{cases} \quad (6)$$

$$q^{k+1}_{i,on} = q^k_{i,on} + \Delta t\cdot(a^k_{i,on} - d^k_{i,on}) \quad (7)$$

where v_i is the free-flow speed of cell *i*, w_i is the congestion wave speed, l_i is the cell length, $a^k_{i,main}$, $d^k_{i,main}$ are the mainline arrival and departure rates for the cell *i* at step *k*. $a^k_{i,on}$, $d^k_{i,on}$ are the on-ramp arrival and departure rates in cell *i* at step *k*. $d^k_{i,off}$ is the off-ramp departure rate for cell *i* at step *k*. $q^k_{i,main}$ represents the number of vehicles on the mainline of cell *i* at step *k*. $q^{\max}_{i,main}$ is the maximum number of this value limited by the mainline space of cell *i*. Similarly, $q^k_{i,on}$ and $q^{\max}_{i,on}$ denote the current (at step *k*) and maximum number of vehicles in the on-ramp of cell *i*, respectively. $\rho^k_i(\rho^k_i = q^k_{i,main}/l_i)$ and $\rho^k_{i+1}(\rho^k_{i+1} = q^k_{i+1,main}/l_{i+1})$ are mainline densities of cell *i* and *i*+1 at step *k*, while ρ^c_i and ρ^c_{i+1} are critical densities of these two cells. When the mainline density exceeds the critical density, congestions will occur on motorways. Under this situation, capacity drop phenomenon may arise. $\lambda \in (0,1]$ is

the capacity drop parameter that is used to reproduce capacity drop phenomenon. If capacity drop happens, this parameter should be set as a value between 0 and 1. Δt is the time duration of each simulation step. c_i^k is the metering rate for the on-ramp cell i at step k. $\beta_i \in [0,1]$ is the split ratio of cell i. $\eta i \in [0,1]$ is the flow allocation parameter of cell i. $\theta_i \in [0,1]$ is the flow blending parameter of traffic flow from the on-ramp to the mainline of cell i.

3.2 Agent design

To design an agent, three elements, namely, action, state and reward shown in *Figure 1* should be defined for any specific application. In a ramp control application, these three elements are defined as follows.

As mentioned earlier, the control action of a ramp meter is to regulate the ramp metering rate. In practical applications, a suitable lower (usually 240 to 360 veh/h) and upper bound (usually 900 to 1,200 veh/h) of ramp metering rate is set according to different traffic and road conditions [21], and the optimal ramp metering rate within the permitted range can be calculated by different control strategies. In this study, the control action is represented by vector C with n metering rates between the minimum and maximum permitted values, $C=\{c_1, c_2,... c_n\}$ (veh/Δt). At each control step k, value is selected as the ramp metering rate for this step. For ease of calculation, we use 30 seconds as the control interval. The minimum and maximum metering rates are set as: 240 veh/h and 1,200 veh/h. Then the control action can be $C=\{2,3,4,5,6,7,8,9,10\}$ (veh/30s) with 9 discrete metering rates.

To represent the traffic states on both motorway mainline and on-ramp, a four-dimensional state space $(s_{q,main}^k, s_{a,main}^k, s_{q,on}^k, s_{a,on}^k)$ is used in this study. Here, the state space can represent the state of one cell or a group of cells; thus, the cell index i is omitted in the following equations.

For the mainline traffic:

$$s_{q,main}^k = \begin{cases} 0, if\ q_{main}^k \le q_{main}^{\min} \\ \left\lceil \frac{q_{main}^k - q_{main}^{\min}}{\Delta q_{main}} \right\rceil, if\ q_{main}^{\min} < q_{main}^k \le q_{main}^{\max} \\ \left\lceil \frac{q_{main}^{\max} - q_{main}^{\min}}{\Delta q_{main}} \right\rceil + 1, \text{otherwise} \end{cases} \quad (8)$$

$$s_{a,main}^k = \begin{cases} 0, if\ a_{main}^k \le a_{main}^{\min} \\ \left\lceil \frac{a_{main}^k - a_{main}^{\min}}{\Delta a_{main}} \right\rceil, if\ a_{main}^{\min} < a_{main}^k \le a \\ \left\lceil \frac{a_{main}^{\max} - a_{main}^{\min}}{\Delta a_{main}} \right\rceil + 1, \text{otherwise} \end{cases} \quad (9)$$

$$s_{main}^k = s_{q,main}^k \cdot \left\lceil \frac{a_{main}^{\max} - a_{main}^{\min}}{\Delta a_{main}} + 2 \right\rceil + s_{a,main}^k \quad (10)$$

$s_{q,main}^k$, $s_{a,main}^k$ and s_{main}^k are all integer numbers that are used as state index to represent the traffic state of mainline. *Equation 7* means the number of vehicles on the motorway ranging from its minimum value $q_{main}^{\min}$ and maximum value $q_{main}^{\max}$ is uniformly divided into several intervals according to Δq_{main}. Each interval corresponds to a state index ranging from 1 to $\lceil (q_{main}^{\max} - q_{main}^{\min})/\Delta q_{main} \rceil$. When q_{main}^k exceeds two boundary values (the maximum and minimum value), two additional state index 0 and $\lceil (q_{main}^{\max} - q_{main}^{\min})/\Delta q_{main} \rceil + 1$ are added into the state set. Similarly, the on-ramp state can be calculated by:

$$s_{q,on}^k = \begin{cases} 0, if\ q_{on}^k \le q_{on}^{\min} \\ \left\lceil \frac{q_{on}^k - q_{on}^{\min}}{\Delta q_{on}} \right\rceil, if\ q_{on}^{\min} < q_{on}^k \le q_{on}^{\min} \\ \left\lceil \frac{q_{on}^{\max} - q_{on}^{\min}}{\Delta q_{on}} \right\rceil + 1, \text{otherwise} \end{cases} \quad (11)$$

$$s_{a,on}^k = \begin{cases} 0, if\ a_{on}^k \le a_{on}^{\min} \\ \left\lceil \frac{a_{on}^k - a_{on}^{\min}}{\Delta a_{on}} \right\rceil, if\ a_{on}^{\min} < a_{on}^k \le a_{on}^{\min} \\ \left\lceil \frac{a_{on}^{\max} - a_{on}^{\min}}{\Delta a_{on}} \right\rceil + 1, \text{otherwise} \end{cases} \quad (12)$$

$$s_{on}^k = s_{q,on}^k \cdot \left\lceil \frac{a_{on}^{\max} - a_{on}^{\min}}{\Delta a_{on}} + 2 \right\rceil + s_{a,on}^k \quad (13)$$

Therefore, the integrated state index is calculated by:

$$s^k = s_{main}^k \cdot \left\lceil \frac{q_{on}^{\max} - q_{on}^{\min}}{\Delta q_{on}} + 2 \right\rceil \cdot \left\lceil \frac{a_{on}^{\max} - a_{on}^{\min}}{\Delta a_{on}} + 2 \right\rceil + s_{on}^k \quad (14)$$

Here, the number of vehicles on the mainline and on-ramp is divided into 20 and 10 intervals, respectively. Both mainline and on-ramp arrival rates are divided into 10 intervals. There are thus a total of 38,016 (22·12·12·12) states in the state space. As the state partition is not the concern here, we have adopted a relatively simple partition method that can be calculated efficiently. The impacts of different partitions are beyond the scope of this paper.

A reward function is used to calculate the immediate reward after executing a specific action at each time step, which guides the agent to achieve its objective. The formal reward function is defined as:

$$R(s^{k-1}, c^{k-1}) = \frac{r^k - r^{\min}}{r^{\max} - r^{\min}} \quad (15)$$

r^k is the raw reward value collected directly from external environment at step k. This value is normalized at each control step and saved as the immediate reward $R(s^{k-1},c^{k-1})$ for the state-action pair (s^{k-1},c^{k-1}). r^{max} and r^{min} are upper and lower bounds for the raw reward value, which are used for normalization.

This study considers the most commonly used objective for traffic control system, i.e. minimizing the total time spent (TTS) by road users. For a motorway segment, this TTS contains the time spent on travelling through the motorway mainline and queuing on the on-ramp [2]. Thus, a typical TTS can be obtained by:

$$TTS = \Delta t \cdot \sum_{k=0}^{K} (q_{main}^{k} + q_{on}^{k}) \quad (16)$$

In the above equation, K is the total number of simulation steps. Since Δt is a fixed value, minimizing TTS is equivalent to minimizing the number of vehicles on the network $\sum_{k=0}^{K} (q_{main}^{k} + q_{on}^{k})$. To minimize TTS, the reward defined here is a negative value as shown below.

$$r^{k} = \begin{cases} -(q_{main}^{k} + q_{on}^{k}), \\ if\ q_{main}^{k} < q_{main}^{\max}\ and\ q_{on}^{k} < q_{on}^{\max} \\ -(q_{main}^{\max} + q_{on}^{\max}), \text{otherwise} \end{cases} \quad (17)$$

For normalization, the boundary values for the reward are set as: r^{max}=0 and $r^{\min} = -(q_{main}^{\max} + q_{on}^{\max})$. In this way, the immediate reward can fall into the range between 0 and 1.

3.3 Control algorithm

The control algorithm for ramp agent is developed based on the standard Q-learning described in section 2.1. Two loops related to episode e and control step k are maintained in the algorithm. In our application, the ramp control problem is modelled as a discrete task. One learning run is composed of a sequence of episodes (E episodes in our case) and each episode represents a learning process starting from the initial state to the end state. At each control step, one control action should be taken by the ramp agent, which leads to a one-step state transition. Therefore, one episode of the algorithm contains a number of control steps (K steps in our case). The details of the control algorithm can be found in *Figure 3*.

4. SIMULATION ANALYSIS

After the definition of agent elements and control algorithm, a series of simulation-based experiments are conducted to test the algorithm performance, especially the influence of different parameters. Both the ramp agent and traffic flow model (ACTM) are implemented with C++ using Visual C++ 6.0.

4.1 Experiment design

A simple network similar to the one proposed by [20] is used for the test. This network contains four

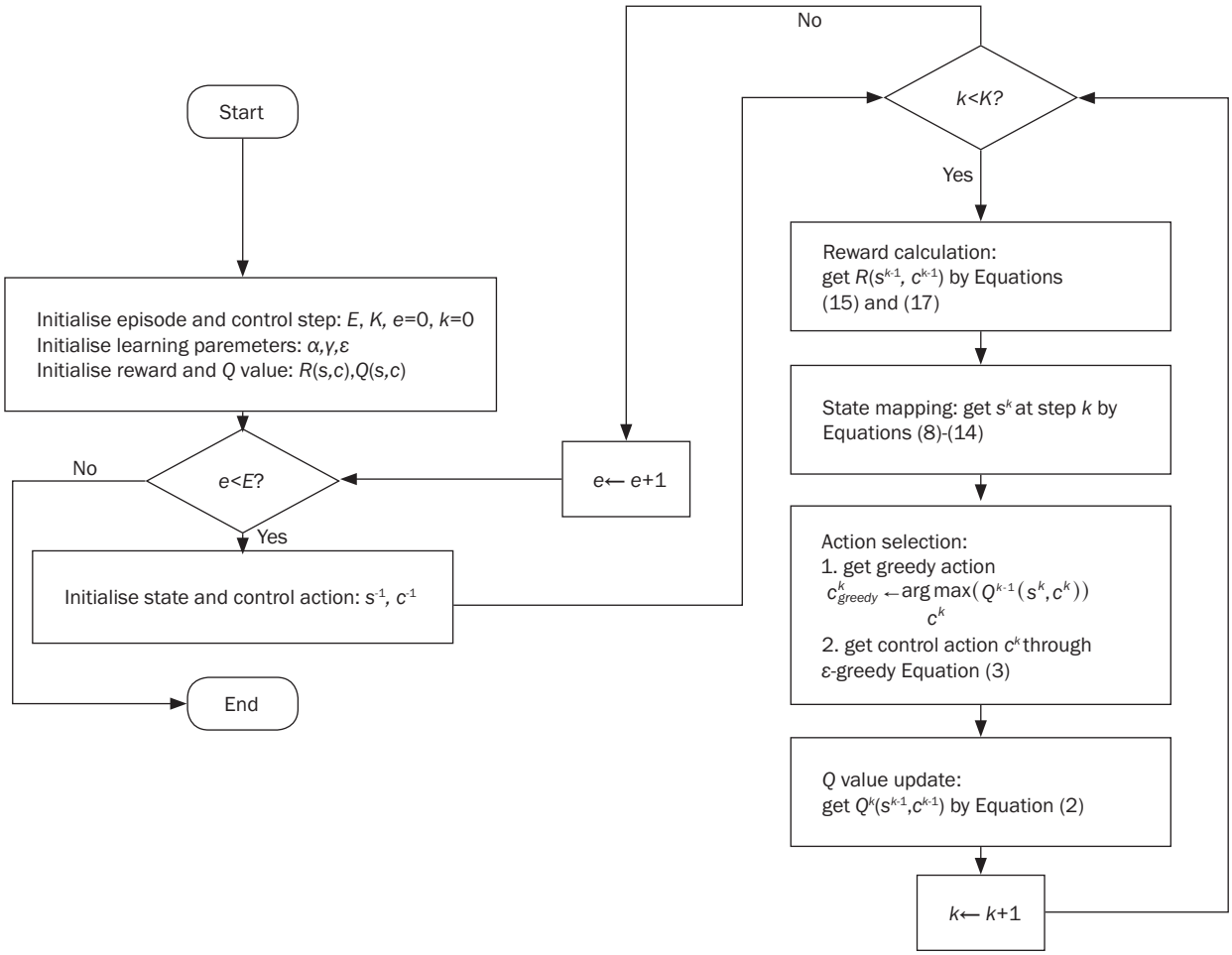

Figure 3 – Algorithm for the ramp agent

cells including one on-ramp cell (the cell with one linked on-ramp) and three normal cells. The aim of the single ramp agent is to minimize TTS of the on-ramp cell 2. The network layout can be found in *Figure 4*, which is a typical motorway with a three-lane mainline and one single-lane on-ramp.

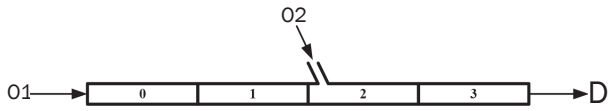

Figure 4 - Network layout

For the ease of calculation, all cells have the same length of 1 km, and therefore have the same $q_{i,main}^{max}$. We assume $q_{i,main}^{max}$ =600 veh, which means each lane can contain up to 200 vehicles per km. The capacity of each lane is set as a typical value of 2,000 veh/h [5]; thus, for 3-lane mainline $d_{i,main}^{max}$ =6,000 veh/h. The free flow speed v_i=100 km/h, flow blending parameter θ_i=0, flow allocation parameter η_i =0.16 are selected from [18]. Thus, the congestion wave speed can be calculated from these parameters as w_i=11.4 km/h. Δt is set as 30 s to guarantee that $\Delta t \leq \min\{l_i/v_i\}$. When congestion happens in cell 2, the road capacity will drop to 90% of the original capacity of uncongested situations with λ=0.9.

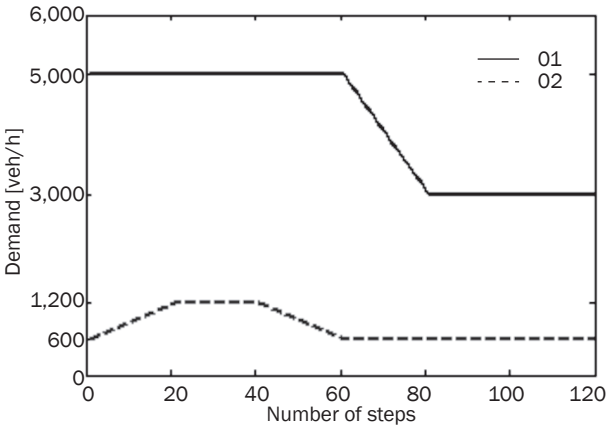

Figure 5 – Demand profile

A demand profile similar to the one used by [5] is chosen for our analysis (as shown in *Figure 5*). Although such a demand profile can guarantee the effectiveness of ramp control on reducing TTS, we do not know if this reduction is the optimal result. To verify the result generated by ramp agent, the commonly used ramp control strategy ALINEA is applied here as a comparison. ALINEA has been proven to be a high-quality strategy for local ramp control problems in many studies [2, 4]. In our simple case, the ramp agent with suitable parameter settings can obtain almost the same result of ALINEA (see *Figure 6*). Here, the optimal parameters of ALINEA are set according to [20].

Under the control of ramp agent, TTS of the whole test period is 3,920 veh min, which is almost 55% of the situation without control (7,160 veh min). This TTS (3,920 veh min) is used as the benchmark in our test. Once the ramp agent reaches this result, it will be said that it has learned the optimal control strategy. The number of episodes (or time) spent for archiving this result is used to measure the learning speed.

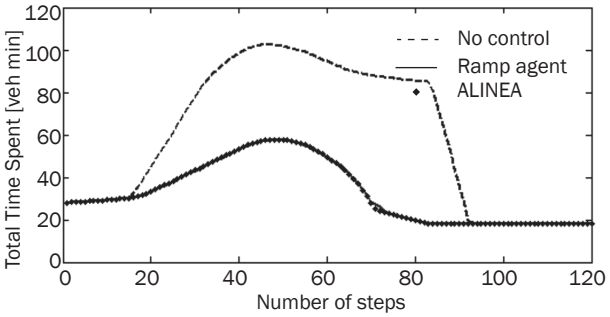

Figure 6 – TTS comparison

4.2 Results and discussion

The influence of three parameters on the algorithm performance has been tested according to two aspects: learning speed and convergence stability. As introduced in the above section, the number of episodes (NE) spent is used as the indicator for the learning speed. The higher NE is, the slower the agent learns to obtain the optimal result. The convergence stability is measured by the variance of results (VR) which is calculated by:

$$VR = \frac{1}{N-1}\sum_{i}^{N}(TTS_i - \overline{TTS})^2 \tag{18}$$

where N is the number of episodes after the benchmark is reached, TTS_i is the total time spent corresponding to the i-th episode after the benchmark is reached, $\overline{TTS}$ is the average value of TTS_i collected from the last N episodes, i.e. $\overline{TTS} = \left(\sum_i^N TTS_i\right)/N$. This indicator can be used to measure how steadily the algorithm can converge to the optimal result, which means higher VR corresponds to lower convergence stability. A simple sensitivity analysis method OAT (one-factor-at-a-time) [22] is used here. This method is conducted by regulating one parameter at a time, while keeping others fixed. For instance, if α is the parameter analyzed, γ and ε will be set as their baseline values for the whole test period. Then, we will change α slightly from its baseline value to observe the changes of NE and VR.

Two sensitivity indices SI(NS)=|ΔNE/$\Delta\alpha$|and SI(VR)=|ΔVR/ $\Delta\alpha$| are used to measure the changes of NE and VR, respectively. For γ and ε, the same method can be used to test their influence. A commonly used value 0.8 is chosen as the baseline of γ. The other two baselines for α and ε are set as 0.05 and 0.01 which are their minimum values for the test. Based

on this method and the experiment design shown in Section 4.1, a series of experiments are conducted in this section. Each experiment runs for one million episodes taking about 25 minutes to guarantee the convergence.

Although the theoretical range for three learning parameters is (0,1), not all parameter values within this range are valuable for the sensitivity analysis. In the ramp control scenario, the RL algorithm with some parameter settings performed quite poorly and even could not converge to the optimal solution. These parameter values should not be involved in the sensitivity analysis. To identify these defective parameter settings, an initial test considering the full theoretical range of parameters is carried out.

As shown in *Figure 7*, when α>0.5, the algorithm performance is very unstable with highly fluctuant TTS. When α reaches 0.9, the algorithm even cannot converge to the benchmark line. Thus, in the following sensitivity analysis, the range for α is set as $\alpha \in [0.05,0.5]$.

Figure 8 shows that γ performs better with higher values. With low γ values such as 0.3 and 0.1, the algorithm has a very poor performance and fails to find the optimal solution. In this case, $\gamma \in [0.5,0.9]$ is set as the test range for γ. On the other hand, as shown in *Figure 9*, lower ε values outperform the higher ones. When ε reaches 0.1, the algorithm becomes unstable. If the algorithm selects higher ε values such as 0.5 and 0.9, it will fail to reach the benchmark line. Thus, in contrast to γ, the suitable values for ε should be much lower and set as $\varepsilon \in [0.01,0.1]$ in the sensitivity analysis. Therefore, the test ranges for three parameters in this study are: $\alpha \in [0.05,0.9]$, $\gamma \in [0.5,0.9]$ and $\varepsilon \in [0.01,0.1]$, which are reasonable according to the literature. In previous studies relating to the RL algorithm [7-13], α is often less than 0.5, γ is between 0.7 and 0.9 and ε is usually around 0.1.

After determining the parameter ranges, an OAT sensitivity analysis is conducted to test the performance of different parameters. The test result is shown in *Figure 10*. From *Figure 10* a and b, we can see that the learning speed is very sensitive with α when it is less than 0.2. For learning rates bigger than 0.3, the learning speed cannot be increased too much by increasing α, and the number of episodes (NE) spent is kept at the same level around 150,000 episodes. The convergence stability, on the other hand, continues to decrease (with increased VR) with the growth of α. Therefore, α is suggested to be set close to 0.2 to avoid the low stability and keep a relatively high learning speed.

With the increasing γ, the number of episodes required to approach the benchmark grows gradually (see *Figure 10* c). For the stability test, one interesting finding is that VR does not fall all the time with the growth of γ. Indeed, one flexion point arises between

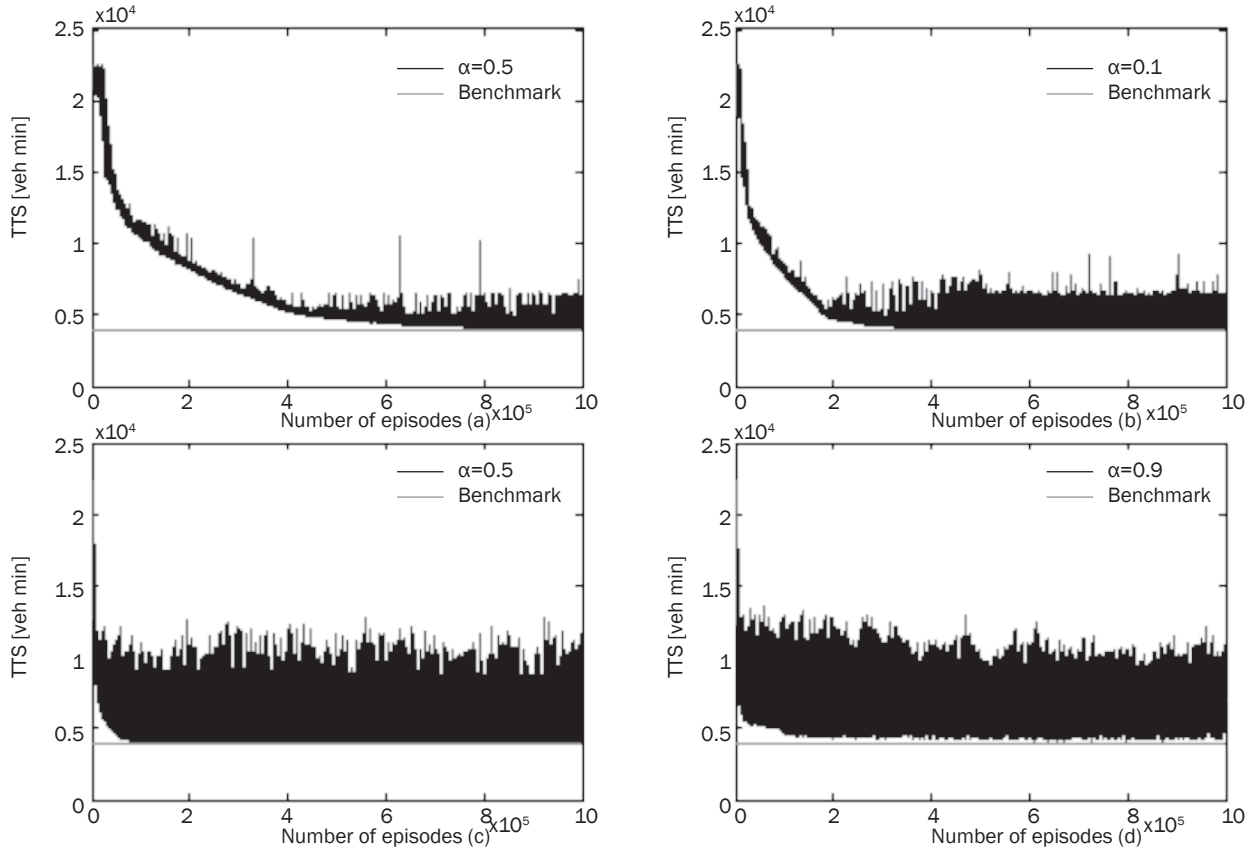

Figure 7 – TTS convergence for different α

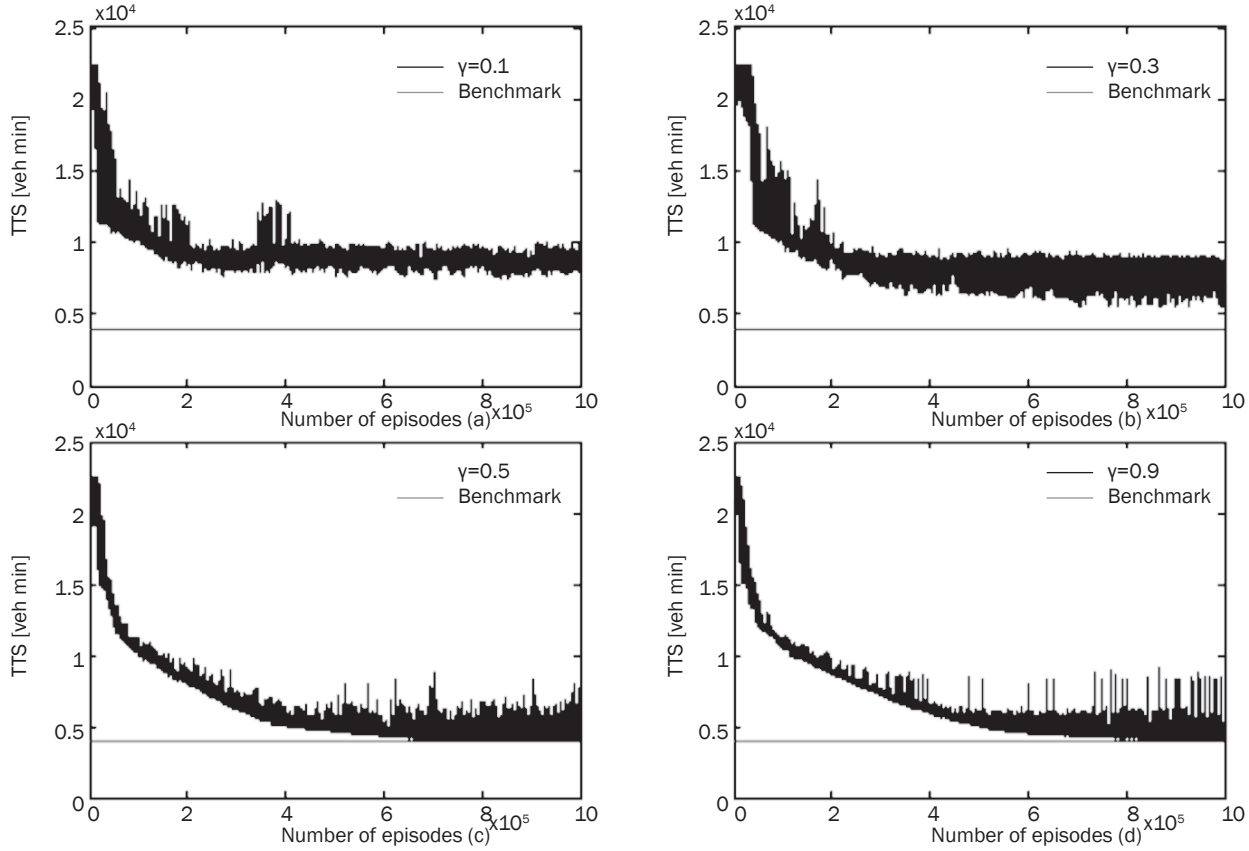

Figure 8 – TTS convergence for different γ

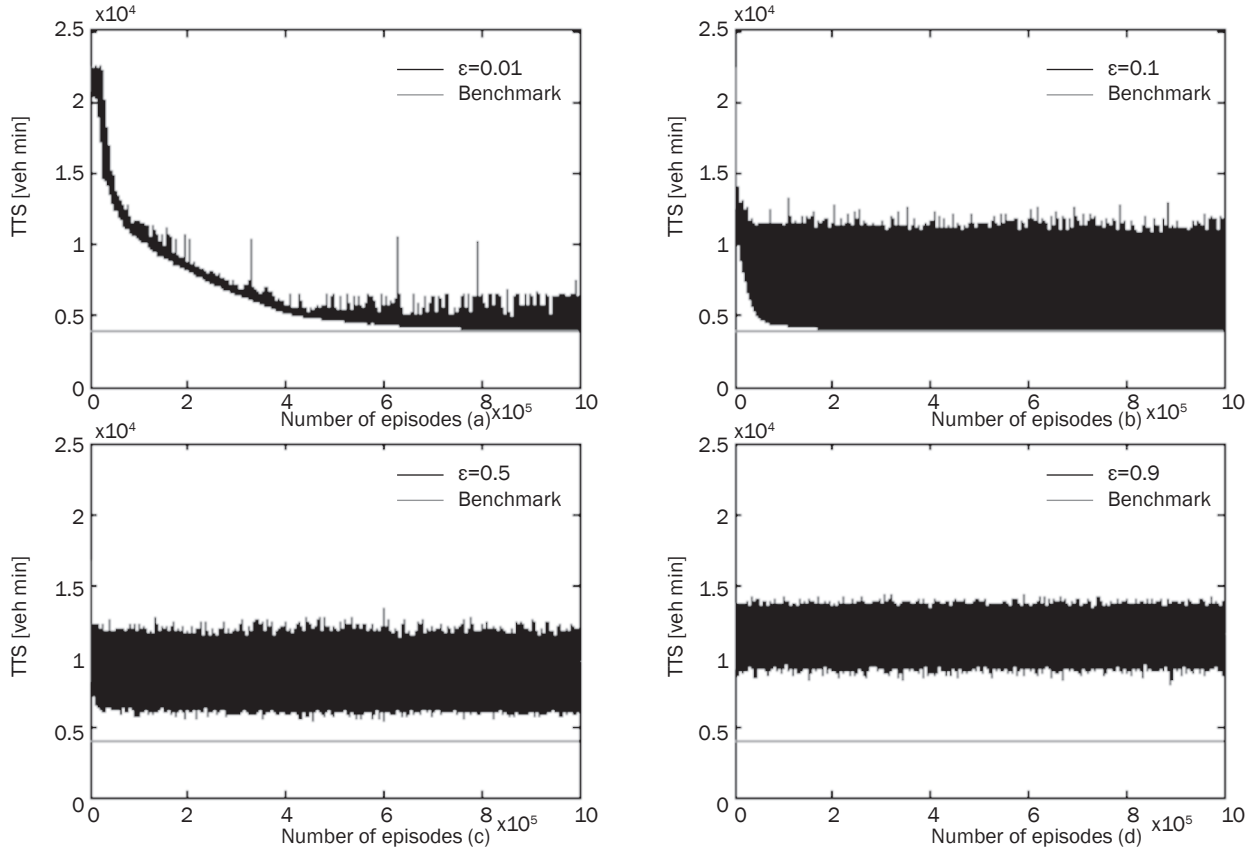

Figure 9 – TTS convergence for different ε

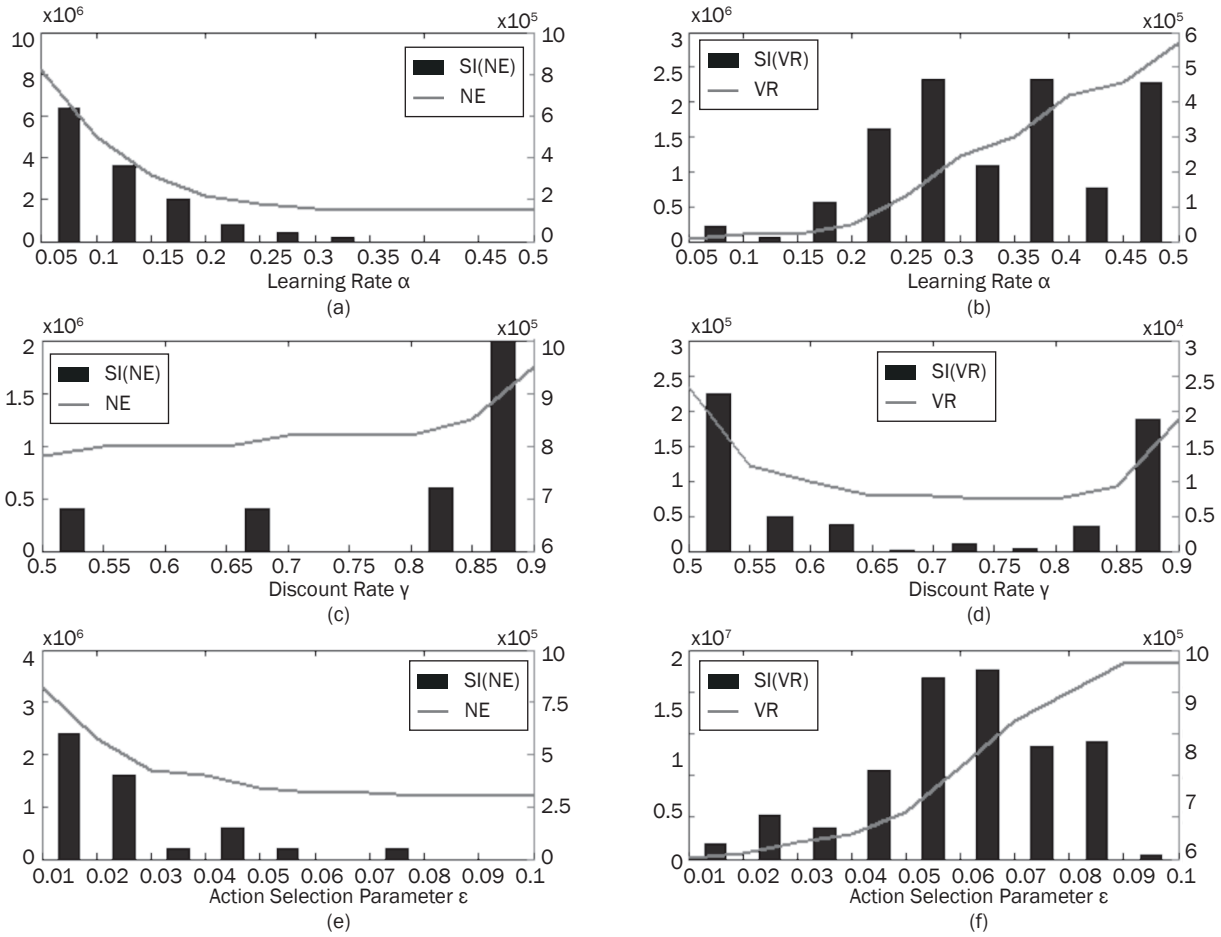

Figure 10 – Sensitivity analysis for different parameters

0.7 and 0.8 (in our case this point is 0.75), around which the highest stability can be obtained (see *Figure 10* d). From the test of γ, it can be concluded that learning speed is not sensitive with the discount rate, and the highest γ cannot guarantee the best stability. If the learning speed is not a concern, a value between 0.7 and 0.8 should be chosen for γ to achieve the highest stability.

From *Figures 10 e* and *f*, it can be seen that both NE and VR are very sensitive to ε. Higher ε leads to lower stability. When ε reaches 0.1, the VR is already around 90×10^5. On the other hand, NE decreases with the growth of ε, while after ε=0.05, the learning speed cannot be improved greatly with NE around 3.0×10^5. Therefore, it is better to set ε as a very small value such as 0.01 to obtain the acceptable convergence stability.

In summary, the most sensitive parameter for both learning speed and convergence stability is ε with the highest indices around 2.5×10^5. α has less impacts on the algorithm than ε and it has the highest indices 6.0×10^6 for learning speed and 2.5×10^6 for convergence stability. Compared with α and ε, the discount rate γ seems to be less important with the highest sensitivity indices 2.0×10^6 and 2.5×10^5 for the learning speed and convergence stability, respectively. Through the comparison of different parameter values, one possible parameter setting is given as: α=0.2, γ =0.75, ε=0.01 that can guarantee high learning speed without losing too much stability.

5. CONCLUSIONS

A ramp control agent with relevant control algorithm has been developed in this paper to examine the influence of different learning parameters. Through the comparison with ALINEA, the control algorithm of ramp agent has been proven to be effective on finding the optimal solution. Using this optimal result as a benchmark, we have analyzed the impacts of three learning parameters including learning rate, discount rate and action selection parameter on the algorithm performance in terms of learning speed and convergence stability. Through the analysis we can obtain the following conclusions: (1) within the test range ($\alpha \in [0.05.0.5]$, $\gamma \in [0.5.0.9]$, $\varepsilon \in [0.01.0.1]$) these three parameters had no obvious effects on the optimal solution itself (the benchmark line), and they only affected how fast and steadily this solution can be obtained; (2) the most sensitive parameter for using

Q-learning with ε-greedy strategy is the action selection parameter ε which makes the most remarkable effect on both learning speed and stability among the three parameters; (3) compared to the other two parameters, the discount rate *γ* is the least sensitive to learning speed and stability; (4) for parameter settings, the learning rate *α* is suggested to be a small value close to 0.2, the discount rate *γ* should be between 0.7 and 0.8, and it is better to set the action selection parameter ε around 0.01.

In the current stage, only deterministic parameters and ε-greedy action selection strategy are examined. Some adaptive parameter settings and other action selection strategies such as -decreasing and soft-max strategies will be considered in the future research. Another limitation of the work shown here is that it is focused on the local ramp control problem which does not consider the coordination of different controllers. When the coordination of multiple ramp agents are taken into account, different parameter settings that may affect the cooperation of different ramp agents should be further investigated.

ACKNOWLEDGEMENT

This study is supported by the China Scholarship Council, University of Leeds (CSC-University of Leeds scholarship) and the National Natural Science Foundation of China (Grant No. 91420203 and 51275041). The authors would like to thank the institutions that support this study.

吕超，博士，讲师，单位：中国北京理工大学机械与车辆学院
黄洁，博士，单位：中国科学院地理科学与资源研究所
龚建伟，博士，教授，单位：中国北京理工大学机械与车辆学院

基于强化学习的匝道控制算法参数分析

摘要
强化学习算法（RL）已经被用于解决匝道控制问题，但是关于学习算法参数特性的研究还并不完善。本文基于强化学习建立了匝道控制智能体，并以此为基础分析了三种不同学习算法参数（即学习率，折扣率和动作选择参数）对算法性能的影响。本文采用学习速度和收敛稳定性来评价算法的性能，并使用宏观交通流模型进行仿真实验。实验结果表明，相对于折扣率，学习率和动作选择参数对算法性能有更明显的影响。基于分析结果，本文给出了关于如何选取算法参数的建议。

关键字
强化学习，Q学习，匝道控制，智能体，宏观交通流模型

REFERENCES

[1] Zhang G, Wang Y. Optimizing coordinated ramp metering: a preemptive hierarchical control approach. Comput-Aided Civ Inf. 2013;28(1):22-37. doi: 10.1111/j.1467-8667.2012.00764.x

[2] Papageorgiou M, Kotsialos A. Freeway ramp metering: an overview. IEEE Trans Intell Transport Syst. 2002;3(4):271-281. doi: 10.1109/TITS.2002.806803

[3] Masher DP, Ross DW, Wong PJ, Tuan PL, Zeidler HM, Petracek S. Guidelines for design and operation of ramp control systems. Stanford: Stanford Research Institute; 1975.

[4] Papageorgiou M, Hadj-Salem H, Blosseville JM. ALINEA: A local feedback control law for on-ramp metering. Transport Res Rec, 1991;1320:58-64.

[5] Hegyi A, De Schutter B, Hellendoorn H. Model predictive control for optimal coordination of ramp metering and variable speed limits. Transport Res C-Emer. 2005;13(3):185-209. doi: 10.1016/j.trc.2004.08.001

[6] Papamichail I, Kotsialos A, Margonis I, Papageorgiou M. Coordinated ramp metering for freeway networks – a model-predictive hierarchical control approach. Transport Res C- Emer. 2010;18(3):311-331. doi: 10.1016/j.trc.2008.11.002

[7] Jacob C, Abdulhai B. Machine learning for multi-jurisdictional optimal traffic corridor control. Transport Res A-Pol. 2010;44(2):53-64. doi: 10.1016/j.tra.2009.11.001

[8] Jacob C, Abdulhai B. Automated adaptive traffic corridor control using reinforcement learning: approach and case studies. Transport Res Rec. 2006;1959:1-8. doi: 10.3141/1959-01

[9] Veljanovska K, Bombol K, Maher T. Reinforcement learning technique in multiple motorway access control strategy design. Promet – Traffic & Transportation. 2010;22(2):117-123. doi: 10.7307/ptt.v22i2.170

[10] Rezaee K, Abdulhai B, Abdelgawad H. Application of reinforcement learning with continuous state space to ramp metering in real-world conditions. Proceedings of the 15th International IEEE Conference on Intelligent Transportation Systems. 2012 Sept 16-19; Anchorage, USA. IEEE; 2012.

[11] Lu C, Chen H, Grant-Muller S. An indirect reinforcement learning approach for ramp control under incident-induced congestion. Proceedings of the 16th International IEEE Conference on Intelligent Transportation Systems. 2013 Oct 6-9; The Hague, the Netherlands. IEEE; 2013.

[12] Lu C, Chen H, Grant-Muller S. Indirect reinforcement learning for incident-responsive ramp control. Procedia Soc Behav Sci. 2014;111:1112-1122. doi: 10.1016/j.sbspro.2014.01.146

[13] Rezaee K, Abdulhai B, Abdelgawad H. Self-learning adaptive ramp metering: analysis of design parameters on a test case in Toronto, Canada. Transport Res Rec. 2013;2013:10-18. doi: 10.3141/2396-02

[14] Sutton RS, Barto AG. Reinforcement learning: an introduction. Cambridge: the MIT press; 1998.

[15] [Watkins CCH, Dayan P. Q-learning. Mach Learn. 1992;8(3-4):279-292. doi: 10.1007/BF00992698

[16] Even-Dar E, Mansour Y. Learning rates for Q-learning. J Mach Learn Res. 2004;5:1-25.

[17] Sun X, Horowitz R. Set of new traffic-responsive ramp-metering algorithms and microscopic simulation results. Transport Res Rec. 2006;1959:9-18. doi: 10.3141/1959-02

[18] Gomes G, Horowitz R. Optimal freeway ramp metering using the asymmetric cell transmission model. Transport Res C-Emer. 2006;14(4):244-262. doi: 10.1016/j.trc.2006.08.001

[19] Haddad J, Ramezani M, Geroliminis N. Cooperative traffic control of a mixed network with two urban regions and a freeway. Transport Res B-Meth. 2013; 54:17-36. doi: 10.1016/j.trb.2013.03.007

[20] Gomes G, Horowitz R. A study of two onramp metering schemes for congested freeways. in, 2003. Proceedings of the 2003 American Control Conference. 2003 June 4-6; Denver, USA. IEEE; 2003.

[21] Arnold ED. Ramp metering: a review of the literature. Virginia: Virginia Transportation Research Council; 1998.

[22] Saltelli A. Sensitivity analysis: Could better methods be used. J Geophys Res-Atmos. 1999;104(D3): 3789-3793. doi: 10.1029/1998JD100042