Decentralised reinforcement learning for ramp metering and variable speed limits on highways

Thorsten Schmidt-Dumont and Jan H van Vuuren

Abstract—Ramp metering and variable speed limits are the best-known control measures for effective traffic flow on highways. In most approaches towards solving the control problems presented by these control measures, optimal control methods, or online feedback control theory have been employed. Feedback control does not, however, guarantee optimality with respect to the metering rate or the speed limit chosen, while optimal control approaches are limited to small networks due to their large computational burden. Reinforcement learning is a promising alternative, providing the tools and the framework required to achieve optimal control policies associated with a smaller computational burden than that of optimal control approaches. In this paper, a decentralised reinforcement learning approach is adopted towards simultaneously solving both the ramp metering and variable speed limit control problems. A simple, microscopic traffic simulation model is employed for the evaluation of the relative effectiveness of the control policies proposed by the reinforcement learning agents.

Index Terms—Reinforcement learning, ramp metering, variable speed limits, decentralised traffic flow control.

I. INTRODUCTION

HIGHWAYS were originally built to provide virtually unlimited mobility to road users. As a result, traffic control measures on highways were initially implemented mainly for safety reasons [1]. The ongoing drastic expansion of car ownership and travel demand have, however, led to increasing traffic congestion, especially within well-developed metropolitan areas. In the United States, for example, travel delays increased by a factor of five from 1.1 billion hours to 5.5 billion hours from 1982 to 2011 [2]. This dramatic increase in delay time brought about a corresponding increase in wasted fuel from 1.9 billion litres to 11 billion litres which, in turn, was associated with a total cost of congestion of 121 billion US dollars in 2011. Although drivers’ perception of congestion is generally linked to increased travel times only, a more important concern is the corresponding capacity decline on highways which, in turn, causes further congestion accumulation. Increasing highway capacity is not always an option due to space limitations, especially within metropolitan areas. Furthermore, the theory of induced travel demand suggests that increases in highway capacity will induce additional traffic demand, thus not alleviating the congestion as envisioned [3]. The alternative to capacity expansion is more effective control of the existing infrastructure. This includes dynamic traffic control measures such as ramp metering (RM), variable speed limits (VSLs) and dynamic route guidance (DRG). Of these dynamic control measures, RM and VSLs are considered to be the most effective [4], [5]. These two control methods are therefore considered in this paper.

The main contributions of this paper may be summarised as follows. First, the reinforcement learning (RL) approach to solving the VSL problem is addressed within a microscopic traffic simulation context for the first time. Furthermore, this paper demonstrates the first application of the κNN-TD RL algorithm to the VSL problem. Secondly, the decentralised RL approach employed to solve the RM and VSL problems simultaneously is applied for the first time in this paper. Furthermore, a thorough comparison of the well-known Q-Learning and κNN-TD RL algorithms is performed within the context of the RL and VSL problems. Finally, a multi-agent approach towards solving the RM and VSL problems simultaneously is presented for the first time.

The remainder of the paper is organised as follows. Section II contains a brief literature review on RM and VSLs, as well as several implementations thereof. Thereafter, the notion of reinforcement learning is reviewed in the third section, with a specific focus on the Q-Learning and κNN-TD learning algorithms. In Section IV, a microscopic traffic simulation model, which is used as a test bed for the algorithmic implementations, is introduced and discussed. Section V is devoted to a description of the RM and VSL problems as RL problem formulations, which serve as the blueprint for the aforementioned algorithmic implementations. The findings and results of a number of computational experiments conducted as a part of this study are presented in Section VI, before the paper closes in Section VII with a discussion on the findings of this paper.

II. LITERATURE REVIEW

RM improves highway traffic flow by effectively regulating the flow of vehicles that enter a highway traffic stream at an on-ramp, thereby increasing the mainline throughput and served traffic volume as a result of an avoidance of capacity loss and blockage of on-ramps due to congestion [4]. Fixed-time RM strategies based on historical demand were first introduced by Wattleworth [6]. In order to be able to determine metering rates in an effective, online manner, Papageorgiou et al. [7] introduced the well-known Asservissement Linéaire d’Entrée Autoroutière (ALINEA) RM control mechanism. Other RM solution approaches include model predictive control (MPC), an optimal control strategy proposed by Hegyi et al. [8], and the hierarchical control approach of Papamichail et al. [9]. One of the first attempts at incorporating RL into
the RM problem so as to be able to learn optimal control policies in an online manner is due to Dauverjnad et al. [10] who employed Q-Learning in a macroscopic traffic simulation modelling environment in order to discover optimal metering rates while taking the build-up of on-ramp queues into account. The first application of RL in a microscopic modelling environment is due to Rezaee et al. [5] who employed the kNN-TD learning algorithm to the RM problem in the context of a case study performed on a portion of Highway 401 in Toronto, Canada.

Initially, VSLs were employed mainly to improve traffic safety on highways, due to the resulting increased homogenisation of traffic flow. More recently, however, VSLs have been employed within a traffic flow optimisation context so as to improve traffic flows along highways. One of the first applications of VSLs as an optimisation technique is due to Smulders [11] who formulated the VSL control problem as an optimal control problem, based on a macroscopic simulation model, with the aim of finding the maximum expected time until congestion occurs. The optimal control problem approach was later extended and refined by Alessandri et al. [12], [13]. In an attempt at simplifying the VSL problem, Carlson et al. [14] proposed a feedback controller which takes as input real-time traffic flow and density measurements in order to calculate, in real time and within a closed loop, appropriate speed limits so as to maintain a stable traffic flow that is close to a pre-specified reference value, with the aim of achieving maximal throughput for any demand realisation. Applications of RL to the VSL control problem have been demonstrated by Zhu and Ukkusuri [15] as well as by Walraven et al. [16]. Zhu and Ukkusuri [15] applied the R-MART RL algorithm within the context of a link-based dynamic network loading model, while the application of RL to the VSL problem demonstrated by Walraven et al. [16] involved application of Q-Learning in conjunction with a neural network for function approximation within the context of a METANET macroscopic traffic flow model. One limitation worth noting in respect of the studies mentioned above is that the VSL problem has up to this point been addressed within a macroscopic traffic modelling paradigm. In this paradigm it is often difficult to capture some of the important, realistic characteristics of traffic flow, such as shockwave propagation, or the spill-back effect of heavy congestion [15].

III. REINFORCEMENT LEARNING

In RL, an agent attempts to learn an optimal control policy by trial and error [17]. At each time step, the agent receives information about the current state of the environment it finds itself in, based on which it takes an action, transforming the environment into a new state. The mapping according to which the agent chooses its action is called the policy, which is used to define the agent’s behaviour. Based on the action chosen, as well as the resulting next state, the agent receives a scalar reward which provides an indication of the quality of the action chosen. In RL, the aim of an agent is to discover a policy which maximises the accumulated reward over time (i.e. finding a policy according to which the agent always chooses a best action with respect to the long-term reward achieved) [17].

A. Q-Learning

Numerous reinforcement algorithms which exhibit favourable learning speeds and contain easily customised parameters have been proposed in the literature. One of the most notable of these is Q-Learning, proposed by Watkins and Dayan [18]. In Q-Learning, a scalar value, denoted by $Q(s, a)$, is assigned to each state-action combination $(s, a)$ so as to provide an indication of the quality of that combination. In an infinite horizon discounted reward problem, the aim is then to

$$\text{maximise } \sum_{t=0}^{\infty} \gamma^t R_t,$$

where $R_t$ denotes the reward obtained at time step $t$, and $\gamma^t$ denotes the discount factor employed to determine the relative importance of future rewards. The $Q$-value is then updated with each new training sample according to the rule

$$Q^{t+1}(s_t, a_t) = Q^t(s_t, a_t) + \alpha (R_{t+1} + \gamma \max_{a_{t+1}} Q^t(s_{t+1}, a_{t+1}) - Q^t(s_t, a_t)),$$

where $R_{t+1}$ denotes the reward received after performing action $a_t$ when the system is in state $s_t$, resulting in the new state $s_{t+1}$, and where $\alpha \in (0, 1]$ is a user-defined parameter known as the learning rate. Watkins and Dayan [18] provided a more detailed description of the Q-Learning algorithm. Furthermore, they have shown that, if designed correctly, Q-Learning is guaranteed to converge to optimal $Q$-values, regardless of the policy employed for action selection.

B. Continuous State Spaces and Function Approximation

In its conventional form, Q-Learning employs a lookup table in order to implement $Q$-values. Using the lookup table-based approach, however, limits the practicality of the methodology, especially when applied to problems exhibiting complex, continuous state spaces. This is caused by the so-called curse of dimensionality according to which significant increases in computation time are observed as the size of the state space increases. In order to extend the applicability of reinforcement learning algorithms, such as Q-Learning, to problems with large, complex, continuous state spaces, function approximators have been introduced, which provide a direct representation of the state space [5], [17]. In this paper, a function approximator, based on the $k$-nearest neighbour clustering technique proposed by Martin et al. [19], is employed. The resulting algorithm has been called the $k$NN-TD learning algorithm [19]. In $k$NN-TD learning, a set of centres $X$ is generated in the state space. Each of the centres $X \in Y$ is assigned an explicit $Q$-value, as in Q-Learning. In order to estimate the $Q$-value of a new point $s$ in the state space, the $k$ nearest neighbours of $s$ in $X$, denoted by $k$NN, are determined based on the euclidean distance from $s$ to each of the centres
microscopic traffic modelling, where each vehicle is simulated individually within an agent-based modelling paradigm. In the microscopic traffic simulation model, the behaviour of individual vehicles is defined. This definition of the vehicle behaviour is bounded by a number of user-defined properties, such as the vehicle lengths, initial and preferred speeds, as well as maximum acceleration and deceleration rates of vehicles. The AnyLogic model implementation may be downloaded electronically [21].

V. RL FOR HIGHWAY TRAFFIC CONTROL

This section is devoted to a description of the implementation of both RM and VSLs as highway traffic control methods in the context of the benchmark simulation model described above.

A. The RM Agent

Davarjenad et al. [10] and Rezaee et al. [5] have shown that the RM problem may be formulated as an RL problem and that it may subsequently be solved using RL techniques. In the benchmark model described above, an RM agent is employed at the single on-ramp, as shown in Figure 2.

As may be seen in the figure, the network has two demand nodes, denoted by O1 and O2, which occur in the mainline and at a single on-ramp, respectively. The stretch of highway before the on-ramp consists of four sections, denoted by S1.1–S1.4 which are all 1 km in length. After the on-ramp there are two further 1 km sections of highway, denoted by S2.1 and S2.2, which lead to a single destination node, denoted by D1. All highway sections have two lanes in the forward direction, while the on-ramp has only a single lane joining into the highway system.

This benchmark simulation model of the network in Figure 1 was implemented within the AnyLogic 7.3.5 simulation software suite [20], making specific use of its built-in Road Traffic Library. The Road Traffic Library allows for microscopic traffic modelling, where each vehicle is simulated individually within an agent-based modelling paradigm. In the microscopic traffic simulation model, the behaviour of individual vehicles is defined. This definition of the vehicle behaviour is bounded by a number of user-defined properties, such as the vehicle lengths, initial and preferred speeds, as well as maximum acceleration and deceleration rates of vehicles. The AnyLogic model implementation may be downloaded electronically [21].

V. RL FOR HIGHWAY TRAFFIC CONTROL

This section is devoted to a description of the implementation of both RM and VSLs as highway traffic control methods in the context of the benchmark simulation model described above.

A. The RM Agent

Davarjenad et al. [10] and Rezaee et al. [5] have shown that the RM problem may be formulated as an RL problem and that it may subsequently be solved using RL techniques. In the benchmark model described above, an RM agent is employed at the single on-ramp, as shown in Figure 2.

As may be seen in the figure, the network has two demand nodes, denoted by O1 and O2, which occur in the mainline and at a single on-ramp, respectively. The stretch of highway before the on-ramp consists of four sections, denoted by S1.1–S1.4 which are all 1 km in length. After the on-ramp there are two further 1 km sections of highway, denoted by S2.1 and S2.2, which lead to a single destination node, denoted by D1. All highway sections have two lanes in the forward direction, while the on-ramp has only a single lane joining into the highway system.

This benchmark simulation model of the network in Figure 1 was implemented within the AnyLogic 7.3.5 simulation software suite [20], making specific use of its built-in Road Traffic Library. The Road Traffic Library allows for microscopic traffic modelling, where each vehicle is simulated individually within an agent-based modelling paradigm. In the microscopic traffic simulation model, the behaviour of individual vehicles is defined. This definition of the vehicle behaviour is bounded by a number of user-defined properties, such as the vehicle lengths, initial and preferred speeds, as well as maximum acceleration and deceleration rates of vehicles. The AnyLogic model implementation may be downloaded electronically [21].

V. RL FOR HIGHWAY TRAFFIC CONTROL

This section is devoted to a description of the implementation of both RM and VSLs as highway traffic control methods in the context of the benchmark simulation model described above.

A. The RM Agent

Davarjenad et al. [10] and Rezaee et al. [5] have shown that the RM problem may be formulated as an RL problem and that it may subsequently be solved using RL techniques. In the benchmark model described above, an RM agent is employed at the single on-ramp, as shown in Figure 2.

As may be seen in the figure, the network has two demand nodes, denoted by O1 and O2, which occur in the mainline and at a single on-ramp, respectively. The stretch of highway before the on-ramp consists of four sections, denoted by S1.1–S1.4 which are all 1 km in length. After the on-ramp there are two further 1 km sections of highway, denoted by S2.1 and S2.2, which lead to a single destination node, denoted by D1. All highway sections have two lanes in the forward direction, while the on-ramp has only a single lane joining into the highway system.

This benchmark simulation model of the network in Figure 1 was implemented within the AnyLogic 7.3.5 simulation software suite [20], making specific use of its built-in Road Traffic Library. The Road Traffic Library allows for microscopic traffic modelling, where each vehicle is simulated individually within an agent-based modelling paradigm. In the microscopic traffic simulation model, the behaviour of individual vehicles is defined. This definition of the vehicle behaviour is bounded by a number of user-defined properties, such as the vehicle lengths, initial and preferred speeds, as well as maximum acceleration and deceleration rates of vehicles. The AnyLogic model implementation may be downloaded electronically [21].

V. RL FOR HIGHWAY TRAFFIC CONTROL

This section is devoted to a description of the implementation of both RM and VSLs as highway traffic control methods in the context of the benchmark simulation model described above.

A. The RM Agent

Davarjenad et al. [10] and Rezaee et al. [5] have shown that the RM problem may be formulated as an RL problem and that it may subsequently be solved using RL techniques. In the benchmark model described above, an RM agent is employed at the single on-ramp, as shown in Figure 2.

As may be seen in the figure, the network has two demand nodes, denoted by O1 and O2, which occur in the mainline and at a single on-ramp, respectively. The stretch of highway before the on-ramp consists of four sections, denoted by S1.1–S1.4 which are all 1 km in length. After the on-ramp there are two further 1 km sections of highway, denoted by S2.1 and S2.2, which lead to a single destination node, denoted by D1. All highway sections have two lanes in the forward direction, while the on-ramp has only a single lane joining into the highway system.

This benchmark simulation model of the network in Figure 1 was implemented within the AnyLogic 7.3.5 simulation software suite [20], making specific use of its built-in Road Traffic Library. The Road Traffic Library allows for microscopic traffic modelling, where each vehicle is simulated individually within an agent-based modelling paradigm. In the microscopic traffic simulation model, the behaviour of individual vehicles is defined. This definition of the vehicle behaviour is bounded by a number of user-defined properties, such as the vehicle lengths, initial and preferred speeds, as well as maximum acceleration and deceleration rates of vehicles. The AnyLogic model implementation may be downloaded electronically [21].

V. RL FOR HIGHWAY TRAFFIC CONTROL

This section is devoted to a description of the implementation of both RM and VSLs as highway traffic control methods in the context of the benchmark simulation model described above.

A. The RM Agent

Davarjenad et al. [10] and Rezaee et al. [5] have shown that the RM problem may be formulated as an RL problem and that it may subsequently be solved using RL techniques. In the benchmark model described above, an RM agent is employed at the single on-ramp, as shown in Figure 2.
backwards along the highway. As with the downstream density, the upstream density \( \rho_{us} \) is discretised into \( n_{\rho_{us}} = 10 \) equi-spaced intervals for the Q-Learning implementation. For the kNN-TD learning implementation, the centres for the upstream density are chosen as \{12, 20, 25, 30, 70, 75, 80\}.

The third and final state variable is the on-ramp queue length \( w \). This variable is included so as to be able to provide the RM agent with information on the prevailing traffic conditions on the on-ramp, as well as provide information about the on-ramp demand. For the Q-Learning implementation, the on-ramp queue length \( w \) is discretised into \( n_w = 9 \) intervals according to

\[
n_w = \begin{cases} 
2.50 & \text{if } w \geq 200, \\
2.00 & \text{if } 175 \leq w < 200, \\
1.75 & \text{if } 150 \leq w < 175, \\
1.50 & \text{if } 125 \leq w < 150, \\
1.25 & \text{if } 100 \leq w < 125, \\
1.00 & \text{if } 75 \leq w < 100, \\
0.75 & \text{if } 50 \leq w < 75, \\
0.50 & \text{if } 25 \leq w < 50, \text{ and} \\
0.25 & \text{if } 0 \leq w < 25.
\end{cases}
\]

Finally, the centres for the queue length in the kNN-TD learning implementation are chosen as \{3, 5, 7, 9, 11, 20, 40, 60\}.

2) The action space: In order to improve the state of traffic flow, the RM agent may select a suitable action based on the prevailing traffic conditions. Rezaee et al. [5] showed that the use of a direct action selection policy (i.e. choosing a red phase duration directly from a set of pre-specified red times) instead of an incremental action selection policy (i.e. adjusting the red phase duration incrementally) yields better results when applied to the RM problem. Direct action selection is therefore also adopted for implementation in this study.

As stated above, red phase durations are varied in order to control the flow of vehicles that enter the highway from the on-ramp. Direct action selection then implies that the RM agent chooses pre-specified red phase durations from a set of actions. In this study, the actions available to the agent are \( a \in \{0, 2, 3, 4, 6, 8, 10, 13\} \), where each action represents a corresponding red phase duration in seconds. These red phase durations correspond to on-ramp flows of \( q_{OR} \in \{1650, 720, 600, 514, 400, 327, 277, 225\} \) vehicles per hour, assuming a green phase duration of three seconds in each case.

3) The reward function: The objective when designing a traffic control system is typically to minimise the combined total travel time spent in the system by all transportation users. From traffic flow theory it follows that the maximum throughput, which corresponds to the maximum flow, occurs at the critical density [4]. Density is usually the variable that the RM agent aims to control. This is the case in ALINEA, the most celebrated RM technique. As a result of the successful implementation of ALINEA in several studies and real-world applications [22], the reward function adopted in order to provide feedback to the RM agent was inspired by the ALINEA control law. In ALINEA, the metering rate is adjusted based on the difference in the measured density downstream of the on-ramp and a desired downstream density [7]. The reward awarded to the agent is calculated as

\[
r(t) = -\left(\hat{\rho} - \rho_{ds}(t)\right)^2,
\]

where \( \hat{\rho} \) denotes the desired density that the RL agent aims to achieve directly downstream of the on-ramp, and \( \rho_{ds}(t) \) denotes the density measured downstream of the on-ramp during the last control interval \( t \), as indicated in Figure 3. This difference is squared in order to amplify large deviations from the desired density, thereby providing amplified negative feedback for actions which result in such large deviations.

4) Other parameters: Watkins and Dayan [18] have shown that Q-Learning suppresses uncertainties and converges to optimal Q-values if a decreasing learning rate is employed, as long as the sum

\[
\sum_{i=1}^{\infty} \alpha_i n_i(s,a) = \sum_{i=1}^{\infty} [\alpha n_i(s,a)]^2
\]

diverges (whether or not the sum

diverges) for all state-action pairs, where \( n_i(s,a) \) denotes the index of the \( i \)-th time that the state-action pair \( (s,a) \) has been visited. As a result, the learning rate

\[
\alpha_{n_i(s,a)} = \left[ \frac{1}{1 + i(1 - \gamma^i)} \right]^{0.85}
\]

is employed, which is a decreasing function of the number of visits to state-action pairs, where \( i \) denotes the index of the \( i \)-th visit to the state-action pair \( (s,a) \), and \( \gamma^i \) denotes the discount factor.

When solving RL problems, it is of critical importance to find a good balance between exploration of the state-action space, and exploitation of what has already been learnt [17]. In order to achieve this balance, an adaptive \( \epsilon \)-greedy policy is employed, where a random action is chosen with a probability \( \epsilon \), while the best known action is chosen with a probability \( 1 - \epsilon \). As with the learning rate above, the adaptive \( \epsilon \)-value is determined as a function of the number of prior visits to each state. This state-dependent \( \epsilon \)-value is calculated as

\[
\epsilon(s) = \max \left\{ 0.05, \frac{1}{1 + \frac{\epsilon}{N_a(s)} \sum_{i=1}^{\epsilon} i(s) \right\}
\]

where \( N_a(s) \) denotes the number of available actions \( a \) when the system is in state \( s \) and \( i(s) \) denotes the number of prior
visits to state \( s \). Employing such a state-dependent \( \epsilon \)-value encourages exploration in the case where a state has not yet been visited, but encourages exploitation as the number of visits to the state increases and the \( \epsilon \)-value decreases to a minimum value of 0.05. For the \( k \text{NN-TD} \) implementation, the calculation of the learning rate \( \alpha \) remains unchanged, except that it is now determined for centre-action pairs rather than for state-action pairs. The calculation of the state-dependent \( \epsilon \)-value, however, changes slightly to

\[
\epsilon(s) = \max \left\{ 0.05, \left[ 1 + \frac{1}{\text{NN}_R(s)} \sum_{i=1}^{k} C^{\text{kNN}}(s) \right] \right\}, \tag{13}
\]

where \( C^{\text{kNN}}(s) \) is the estimated number of visits to state \( s \), given by

\[
C^{\text{kNN}}(s) = \sum_{i=1}^{k} p_i C(X_i, a). \tag{14}
\]

In (14), \( p_i \) is the weighted probability linked to each of the \( k \) nearest neighbours, as determined in (3), and \( C(X_i, a) \) denotes the number of visits to the centre-action pair \( (X_i, a) \).

B. Variable Speed Limits

Zhu and Ukkusuri [15] and Walraven et al. [16] have shown that the VSL problem may be formulated as an RL problem and solved using RL techniques. In the benchmark model considered in this study, VSLs are applied from the start of \( S_{1.3} \) until the start of \( S_{2.1} \), where the normal speed limit of 120 km/h is again restored after the bottleneck location, as shown in Figure 4.

1) The state space: As with the state space for the RM application, the state space for the VSL implementation comprises three main components, as illustrated graphically in Figure 4. The first state variable is the density \( \rho_{us} \) directly downstream of the on-ramp. This variable is chosen so as to provide the VSL agent with information on the state of traffic flow at the bottleneck location.

The second state variable is the vehicle density on \( S_{1.4} \) at the application area, denoted by \( \rho_{app} \). This variable is chosen since it is expected to provide the agent with an indication of the effectiveness of the action chosen, as the most immediate response to the action will be reflected on this section of the highway.

The third and final state variable is the upstream density \( \rho_{us} \). In the case of VSLs, the upstream density is the density on \( S_{1.3} \). This variable is chosen so as to provide the learning agent with a predictive component in terms of highway demand, as well as an indication of the severity of the congestion, should it have spread beyond the application area.

For the Q-Learning application, the downstream density, application density and upstream density are discretised into \( n_{\rho_{us}} = n_{\rho_{app}} = n_{\rho_{us}} = 10 \) equi-spaced intervals. For the \( k \text{NN-TD} \) learning implementation, the downstream density centres are chosen as \( \{12, 19, 24, 29, 32, 35, 38, 45, 55, 60\} \), while the centres for the density at the application area are placed at \( \{12, 20, 26, 30, 35, 40, 45, 55\} \). Finally, the centres for the upstream density are \( \{12, 20, 26, 30, 35, 40, 45, 55\} \).

2) The action space: As in the RM implementation, a direct action selection policy is adopted by the VSL agent in pursuit of a fast learning rate. The VSL to be applied is at \( S_{1.4} \) is then determined by

\[
\text{VSL}_{1.4} = 90 + 10a, \tag{15}
\]

where \( a \in \{0, 1, 2, 3\} \). This results in minimum and maximum variable speed limits of 90 km/h and 120 km/h, respectively.

As shown in (15), the learning agent directly adjusts the speed limit at \( S_{1.4} \). In order to reduce the difference in speed limit from 120 km/h to VSL_{1.4}, the speed limit at \( S_{1.3} \) is adjusted according to

\[
\text{VSL}_{1.3} = \max([\text{VSL}_{1.4} + \delta], 120). \tag{16}
\]

This more gradual reduction in the speed limit has been introduced in order to minimise the probability of shock-waves propagating backwards along the highway due to the sudden, sharp reduction in the speed limit.

3) The reward function: As in the case of RM, the objective of the VSL agent remains to minimise the total time spent in the system by vehicles. As stated above, this may be achieved by maximising the system throughput. As a result, the reward function chosen for the VSL agent is the flow rate out of the bottleneck location.

VI. NUMERICAL EXPERIMENTATION

The RL algorithms presented above for the RM and VSL problems have been trained in respect of four scenarios of varying traffic demand, as shown in Figure 5. As may be seen in the figure, a rush hour is imitated in all of the scenarios, initially accommodating free-flowing traffic due to low demand. This is followed by a 30-minute period of steady increase in demand, until the demand reaches a peak, after which it remains constant for an hour. Thereafter, the demand decreases steadily back to the free-flow demand over a 30 minute period. Finally, in order to account for congestion which has built up over the peak demand period, a time of 90 minutes is allowed at the end of every experimental run for the system to once again reach the initial free-flow traffic conditions. As may be seen in Figure 5, scenario 1 presents the heaviest traffic demand, with the demand at \( O_1 \) peaking at 3500 vehicles per hour, and the demand at \( O_2 \) peaking at 750 vehicles per hour. In scenario 2, the demand at \( O_1 \) remains unchanged, while the peak demand at \( O_2 \) is reduced to 500 vehicles per hour. Scenario 3 has a reduced demand of 3000
vehicles per hour at $O_1$, while the on-ramp demand at $O_2$ remains as in scenario 1. Finally, scenario 4 presents lower demands at both $O_1$ and $O_2$, with peaks at 3 000 vehicles per hour and 500 vehicles per hour, respectively.

A. Algorithmic performance measures

The relative performances of the RL algorithms are measured in the contexts of the four scenarios described above according to the following performance measures:

1) the total time spent in the system by all vehicles (TTT) measured in vehicle hours,
2) the total time spent in the system by vehicles travelling along the highway only (TTTHW) measured in vehicle hours,
3) the total time spent in the system by vehicles joining the highway from the on-ramp (TTTOR) measured in vehicle hours,
4) the mean time spent in the system by vehicles travelling along the highway only (MTTHW) measured in seconds,
5) the mean time spent in the system by vehicles joining the highway from the on-ramp (MTTOR) measured in seconds,
6) the maximum time spent in the system by a vehicle travelling along the highway only (MaxHW) measured in seconds, and
7) the maximum time spent in the system by a vehicle joining the highway from the on-ramp (MaxOR) measured in seconds.

The reason for breaking down the total time spent in the system performance measure into the two further performance measures is that it is expected that there may be an increase in the total time spent in the system by vehicles that join the network from the on-ramp due to ramp metering, which may not be reflected sufficiently in the single total time in the system measure. In order to provide an indication of the expected travel time for each of the routes in the network, the respective mean travel time values are recorded. Due to the fact that there may also be interest in a worst-case scenario in terms of travel time, the maximum travel time along each route is also recorded as a final performance measure.

For purposes of comparison of the relative performance of the RL algorithms in respect of RM, the ALINEA ramp metering control strategy, which is widely regarded as the benchmark RM control strategy [5], is also implemented. ALINEA has, however, been designed for application in a macroscopic modelling environment. As a result, a number of minor adjustments had to be made to the control strategy in order to facilitate its successful application within a microscopic traffic simulation model. In the ALINEA strategy, the metering rate is adjusted based on the density directly downstream of the on-ramp. In the macroscopic case, this is
achieved simply by adjusting the maximum allowable flow entering the highway from the on-ramp. The proposed change to the control law results in an update protocol

\[ R(t+1) = R(t) + K_R (\rho_{\text{out}}(t) - \hat{\rho}), \]

(17)

where \( R(t) \) denotes the red phase duration during control interval \( t \), \( K_R \) is a nonnegative control parameter, and \( \hat{\rho} \) and \( \rho_{\text{out}} \) denote the desired and measured downstream traffic densities, respectively. As in the original ALINEA control law, a downstream density greater than the desired value will lead to a reduced metering rate (\( i.e. \) increased red phase times) in the adapted version, allowing fewer vehicles to join the highway stream, while a density lower than the desired value will result in an increased metering rate (\( i.e. \) decreased red phase times), allowing more vehicles to enter the highway stream.

B. Vehicle configuration

For the purpose of this study, the default values for the vehicle properties, as suggested in the AnyLogic Road Traffic Library [20], are employed in order to demonstrate the working of the algorithms in this hypothetical scenario. All vehicle lengths are thus fixed at 5 metres, while the initial speeds of vehicles entering the highway and the on-ramp are set to 120 km/h and 60 km/h, respectively. In order to account for variation in driver aggressiveness, the preferred speed of vehicles on the highway is uniformly distributed between 110 km/h and 130 km/h, while maximum acceleration and deceleration values are set to 1.8 m/s\(^2\) and 4.2 m/s\(^2\), respectively. Vehicles are generated according to an arrival rate following a Poisson distribution with a mean corresponding to the desired traffic volume. For the scenarios described above, this mean is therefore equal to the traffic demand at each of the origins. Vehicles appear in a randomly selected lane when multiple lanes exist at the entry point. In the case where a vehicle has to wait before entering the road network (such as in the case of a long on-ramp queue, or severe congestion spill-back past the boundaries of the simulated environment) the vehicle is stored in a queue buffer until sufficient space has opened up on the road network for the vehicle to enter.

C. Numerical Results

1) Ramp metering: In the case where only RM was employed as a control measure, four cases are considered for each of the four scenarios. In the first case no control is applied, while in the second case RM is enforced according to the modified ALINEA control law in (17). In the third case RM is enforced by the Q-Learning agent. Finally, in the fourth case, RM is enforced by the \( k \)NN-TD agent. A summary of the performance results for RM.

![Fig. 6. The learning progression over the course of 1000 training epochs for Scenario 2 shown for both the Q-learning and the \( k \)NN-TD learning agents. In order to filter out some simulation noise, a moving average over 30 epochs is shown.](image)

As may be seen in Table I, all the RM implementations
are able to improve on the no control case in respect of the TTT, apart from ALINEA in Scenario 4. The results presented in the table indicate that according to the ALINEA control strategy, the greatest focus is placed on protecting the highway flow, resulting in the most significant improvements in terms of the travel times along the highway, while compromising with very long on-ramp waiting times. The kNN-TD RM implementation is consistently able to achieve the smallest TTT value. In Scenarios 1 and 2, the performance in terms of the waiting times for vehicles held up on the on-ramp is similar to that of ALINEA. Interestingly, in Scenario 3, the behaviour of the kNN-TD implementation changes, resulting in a sharp reduction of the on-ramp waiting times, while not protecting the highway flow as much as it did in Scenarios 1 and 2. Q-Learning seems to be able to find a middle ground by balancing the highway and on-ramp waiting times more effectively than the other RM implementations in all four scenarios.

2) Variable speed limits: In the case where VSLs are employed as a highway control measure, three cases are considered for each of the four scenarios, as may be seen in Table II. The first case is again the no-control case. In the second and third cases, Q-Learning and kNN-TD learning agents are implemented, respectively.

### Table II

<table>
<thead>
<tr>
<th>PMI</th>
<th>No control</th>
<th>Q-Learning</th>
<th>kNN-TD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scenario 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TTT</td>
<td>1753.01</td>
<td>1746.54</td>
<td>1738.33</td>
</tr>
<tr>
<td>TTTHW</td>
<td>1707.70</td>
<td>1700.68</td>
<td>1692.11</td>
</tr>
<tr>
<td>TTTOR</td>
<td>45.31</td>
<td>45.86</td>
<td>46.22</td>
</tr>
<tr>
<td>MTTHW</td>
<td>657.83</td>
<td>656.61</td>
<td>653.05</td>
</tr>
<tr>
<td>MTTOR</td>
<td>99.32</td>
<td>100.60</td>
<td>101.06</td>
</tr>
<tr>
<td>MaxHW</td>
<td>1935.24</td>
<td>1942.55</td>
<td>1908.69</td>
</tr>
<tr>
<td>MaxOR</td>
<td>140.37</td>
<td>141.25</td>
<td>148.01</td>
</tr>
<tr>
<td>Scenario 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TTT</td>
<td>1141.80</td>
<td>1067.69</td>
<td>1052.84</td>
</tr>
<tr>
<td>TTTHW</td>
<td>1107.88</td>
<td>1033.87</td>
<td>1019.16</td>
</tr>
<tr>
<td>TT TOR</td>
<td>33.92</td>
<td>33.82</td>
<td>33.67</td>
</tr>
<tr>
<td>MTTHW</td>
<td>424.86</td>
<td>398.65</td>
<td>393.36</td>
</tr>
<tr>
<td>MTTOR</td>
<td>94.68</td>
<td>96.38</td>
<td>97.21</td>
</tr>
<tr>
<td>MaxHW</td>
<td>1166.96</td>
<td>1103.57</td>
<td>1087.60</td>
</tr>
<tr>
<td>MaxOR</td>
<td>127.85</td>
<td>138.09</td>
<td>136.56</td>
</tr>
<tr>
<td>Scenario 3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TTT</td>
<td>932.46</td>
<td>888.97</td>
<td>877.73</td>
</tr>
<tr>
<td>TTTHW</td>
<td>887.06</td>
<td>843.66</td>
<td>832.62</td>
</tr>
<tr>
<td>TT TOR</td>
<td>45.40</td>
<td>45.31</td>
<td>45.11</td>
</tr>
<tr>
<td>MTTHW</td>
<td>370.72</td>
<td>353.21</td>
<td>350.16</td>
</tr>
<tr>
<td>MTTOR</td>
<td>97.94</td>
<td>99.25</td>
<td>99.19</td>
</tr>
<tr>
<td>MaxHW</td>
<td>1331.29</td>
<td>1227.12</td>
<td>1215.50</td>
</tr>
<tr>
<td>MaxOR</td>
<td>141.92</td>
<td>145.24</td>
<td>147.05</td>
</tr>
<tr>
<td>Scenario 4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TTT</td>
<td>550.00</td>
<td>547.41</td>
<td>533.67</td>
</tr>
<tr>
<td>TTTHW</td>
<td>517.07</td>
<td>514.50</td>
<td>500.88</td>
</tr>
<tr>
<td>TT TOR</td>
<td>32.93</td>
<td>33.91</td>
<td>32.79</td>
</tr>
<tr>
<td>MTTHW</td>
<td>216.21</td>
<td>215.65</td>
<td>209.89</td>
</tr>
<tr>
<td>MTTOR</td>
<td>92.46</td>
<td>94.01</td>
<td>93.12</td>
</tr>
<tr>
<td>MaxHW</td>
<td>489.89</td>
<td>358.04</td>
<td>354.46</td>
</tr>
<tr>
<td>MaxOR</td>
<td>128.05</td>
<td>133.46</td>
<td>131.71</td>
</tr>
</tbody>
</table>

As was the case in the RM implementations, the kNN-TD VSL implementation is able to achieve the greatest improvements in respect of the TTT in all four scenarios, as shown in Table II. Interestingly, the reduction in the mean values presented in the table does not stem from sharply reduced travel times. Instead the travel times on the highway exhibit reduced variance, clustering towards the lower whisker, as may be seen from the box plots in Figure 7. This observation suggests that

![Figure 7](image_url)

Fig. 7. MTTHW values for the different VSL implementations in Scenario 2.

As was the case in the RM implementations, the kNN-TD VSL implementation is able to achieve the greatest improvements in respect of the TTT in all four scenarios, as shown in Table II. Interestingly, the reduction in the mean values presented in the table does not stem from sharply reduced travel times. Instead the travel times on the highway exhibit reduced variance, clustering towards the lower whisker, as may be seen from the box plots in Figure 7. This observation suggests that...
4) Only the agent able to achieve the largest joint gain is allowed to change its action, while the neighbour’s action remains unchanged. Then the process is repeated from Step 2 until no agent is able to achieve a larger combined gain by changing its action.

5) This process is repeated during each learning iteration. For the maximax MARL approach, the state-action space for each agent $i$ increases by the factor of $|\mathcal{A}^i|$, where $|\mathcal{A}^i|$ denotes the set of all actions available to its neighbour $j$.

For all three MARL approaches, three combinations of reward functions were assessed. For Combination 1, both agents were rewarded as in the single-agent implementations. For Combination 2, both agents were rewarded based on the density downstream of the on-ramp as in the reward function in (8). Finally, for Combination 3, both agents were rewarded based on the flow out of the bottleneck location, as in the single agent VSL implementation. In both the independent and hierarchical MARL implementations, using the original reward functions (i.e. Combination 1) yielded the best results, while Combination 2 was found to return the best performance in the maximax MARL implementation. For the evaluation of the three combinations of reward functions in the maximax MARL implementation, the combined gain of the RM and VSL agents was calculated as the combined proportional gain. The combined gain of the RM and VSL agents is thus calculated as

$$G_{a_{i+1}} = \max_{a_{i+1}} \left[ \frac{Q^i(s_{t+1}, a_{i+1}) - Q^i(s_{t+1}, a_i)}{Q^i(s_{t+1}, a_i)} \right] + \frac{Q^j(s_{t+1}, a_{t+1})}{Q^j(s_{t+1}, a_t)}$$

when both agents are rewarded based on (8).

Due to its superior performance in the single-agent paradigm, only the kNN-TD algorithm was selected for implementation in the MARL systems. The results obtained by the algorithms employing the three different coordination methods are presented in Table III.

As may be seen in Table III, the MARL approaches are always able to improve on the no-control case in respect of the TTS. Furthermore, the MARL approaches are able to improve upon the performance of the single VSL agent in respect of the TTS, the VSL implementations have shown that reductions in total travel times due to the influence of VSLs, as may be seen in Figure 8. The independent MARL approach returns a performance that is typically very similar to that of the single kNN-TD RM agent, while the maximax MARL agent provides less protection of the highway flow, but thereby returns lower on-ramp waiting times than the other RM approaches without compromising on the TTS-value achieved.

VII. CONCLUSION

The results obtained from the RM implementations show that RM may be effectively employed so as to reduce the total travel time spent by vehicles in the system in varying conditions of traffic demand. The RL approach to RM yields marginally better results than ALINEA, often due to finding a better balance between protecting the highway flow and achieving acceptable on-ramp queueing times. Finally, the kNN-TD RM algorithm is effective in identifying the lower traffic demand on the highway in Scenarios 3 and 4, and effectively adjusting the RM strategy according to these conditions.

Although they are not as effective as RM in reducing the TTS, the VSL implementations have shown that reductions in the TTS are possible when VSLs are applied effectively. The main reason for this may be homogenisation of traffic flow, since the traffic flow becomes more stable, as may be seen in...
Finally, employing a MARL approach to solving the RM and VSL problems simultaneously has shown that further reductions in the TTS are possible, although these reductions are relatively small. It was, however, found that employing a MARL approach again leads to a more stable traffic flow in the presence of RM, as may be seen in Figure 8. Furthermore, the maximax MARL approach was typically able to find a better balance between protecting the highway flow and achieving an acceptable on-ramp queue than the other RM approaches, which may be favourable as it may prevent congestion on the on-ramp spilling back into arterial systems.

REFERENCES


