Reinforcement Learning for Road Pricing: A Review and Future Directions

Otto Vermeulen[1*], Arno Siebes[1] and Yannis Velegrakis[1]

[1]Department of Information and Computing Sciences, Utrecht University, Princetonplein 5, Utrecht, 3584 CC, The Netherlands.

*Corresponding author(s). E-mail(s): o.a.vermeulen@uu.nl;
Contributing authors: a.p.j.m.siebes@uu.nl; i.velegrakis@uu.nl;

Abstract

Demand for mobility is growing, and traffic on roads has increased substantially, leading to suboptimal traffic flow and congestion. Road pricing can encourage vehicles to change their behavior by charging for road use. Because traffic is not static, dynamic road pricing can help dynamically control traffic. Reinforcement Learning is an effective approach to optimizing the performance of a system. It has already been applied to control traffic signals and has recently found an application in dynamic road pricing for traffic optimization. We survey recent solutions and find that the methods proposed demonstrate the usefulness of reinforcement learning for road pricing. We compared how common challenges in reinforcement learning were approached in the works. Challenges which remain little explored are generalizability and scalability of solution approaches. Approaches to partial observability, credit assigment and non-stationarity are not in all cases taking full account of existing solutions for these common challenges. We further note the need for standardized benchmarks to allow comparisons between the performance of the provided solutions.

Keywords: Reinforcement Learning, Pricing and Resource Allocation, Optimization, Simulation, Transportation

1 Introduction

As traffic on road networks grows, so too does traffic congestion (Mokbel et al., 2024; Prieto Curiel et al., 2021). Congestion occurs when vehicles moving on a road network are forced to reduce their speed as a result of the presence of other vehicles. Congestion

has significant productivity, environmental, and health consequences, therefore it is important to be able to manage it.

Intelligent Transportation Systems (Haydari and Yilmaz, 2020) (ITS) are systems that have been designed to optimize traffic by reducing the presence of congestions. One type of such systems are those imposing constraints to which vehicles must adhere. Examples include traffic signals, ramp metering, or speed limits. Another type is those that provide incentives for vehicles to follow a specific behavior. Examples include route guidance and road pricing.

Road pricing is the mechanism to determine the tolls that a vehicle must pay to use a specific road at a specific time. It is a mechanism that can contribute to alleviating congestion. This is based on the insight that people tend to make choices to optimize their utility, taking into account costs and benefits of their actions. In contrast to the other ITS mechanisms, road pricing uses a price mechanism, with the benefits of clarity, universality and efficiency (Lindsey and Verhoef, 2001). Its potential to influence vehicles' route choice and travel mode were advocated early on from an economic perspective (Beckmann et al., 1956; Knight, 1924; Pigou, 1924; Vickrey, 1969). Farias et al. (2024) provide a recent comprehensive review of dynamic pricing for toll roads in the U.S. In the literature, there have been variations of this idea under the term of congestion pricing, congestion charging, tolling, toll charging, or road pricing. We use the term road pricing. With road pricing, roads with higher tolls tend to be less preferable to some of vehicle drivers. This means that, by controlling the tolls, one may indirectly control the flow of vehicles on the road network.

Effects of road pricing may be unanticipated if vehicles change their behavior. Other ITS, like ramp metering and variable speed limits, may also influence the effectiveness of road pricing. Assuming that vehicles change their behavior if roads are priced and that other mechanisms, like traffic signals and variable speed limits, are at play in road networks as well, we note that the effects of road pricing may be unanticipated. Because of this, we require a system that is able to learn from the effects of road pricing decisions and adjust this accordingly. Given the inherent dynamic aspect of road pricing, it seems ideal for a Reinforcement Learning approach.

Reinforcement Learning (RL) is a machine learning technique that learns agents to take actions to optimize their performance. In RL, agents interact with an environment that can be in a number of states. These agents want to achieve some objective. Agents take actions according to a policy, which leads to a change in the state of the environment. Agents learn by observing the effect of their actions in the environment through reward signals, to improve the policy to an optimal one (Sutton and Barto, 2018). RL methods can use tables or functions. If these functions are neural networks we associate this with deep learning and correspondingly use the term Deep RL.

Despite its potential, RL has received limited attention within the research community for road pricing applications. This is a notable gap, as RL presents a strong candidate for optimizing the tolling process, alongside offering numerous compelling research challenges. Traditional approaches like fixed tolling, rule-based heuristics or optimization-based techniques often rely on access to tractable traffic models and assume stationary demand patterns. However, in reality, traffic systems are dynamic, stochastic and are subject to incidents and human behavior, which evolves over time

in response to (toll-) pricing but also other policies and control measures. These limitations hinder the effectiveness of conventional methods in practice.

The complex and analytically intractable nature of traffic propagation, influenced by a multitude of factors, makes the model-free learning capabilities of RL particularly well-suited for dynamic tolling optimization. By learning from real-time traffic data, RL agents can develop effective tolling policies that incentivize behavioral changes in road users through price adjustments. RL mechanisms can also adapt online to evolving traffic patterns and non-stationary conditions like incidents or changes in vehicle preferences.

Furthermore, the application of RL to road pricing raises significant research questions concerning scalability, generalization, and off-line learning. The interactions between multiple road pricing agents, diverse road user types (including human-driven and autonomous vehicles), and agents controlling other traffic elements (such as traffic signals, ramp meters, and variable speed limits) warrant in-depth investigation.

While RL has been extensively explored in the context of traffic signal control, its application to road pricing remains relatively underdeveloped. We attribute this to principal differences between the two domains. Traffic signal control typically benefits from more immediate and localized feedback, as well as greater availability of data and standardized interfaces for widely used simulation platforms. In contrast, RL for road pricing involves challenges such as delayed and indirect feedback, stochastic vehicle behavior, limited access to suitable datasets, and a lack of standardized interfaces. These factors have likely contributed to the slower adoption of RL in this domain. However, they also highlight opportunities for impactful research.

By addressing these, and the other challenges noted above, this survey aims to underscore and advance the use of RL in road pricing and stimulate further research in this promising domain.

1.1 Recent Related Surveys

Various works in the wider Intelligent Transportation Systems domain exist which use RL to solve traffic problems. Illustrative recent surveys covering such works are listed in Table 1. As motivated in the previous section, road pricing solutions cannot occur in isolation, but ultimately need to take account of other ITS mechanisms. We distinguish therefore two categories: surveys providing an overview of the use of RL in the wider ITS domain and surveys focusing on the use of RL in a specific ITS domain like traffic signal control.

1.2 Contribution of this Survey

Our contributions can be summarized as follows:

- We describe a traffic model and give an introduction to road pricing. We provide a high level overview of RL and different algorithms that have been used in the surveyed works.
- We formulate the problem how traffic can be managed by road pricing using RL and identify important specific research challenges. At the best of our knowledge this structured approach is unique.

Table 1: Surveys on ITS solutions through Reinforcement Learning on multiple aspects of a road network (top) or on specific aspects of the road network collectively (bottom). A survey for RL in road pricing does not yet exist.

Ref.	Features and relevance
Han et al. (2023)	Provides a comprehensive overview of RL based traffic control strategies, highlighting that few studies separate learning and testing. Identifies challenges for field implementation.
Haydari and Yilmaz (2020)	Categorizes application types, control methods and RL algorithms. Both for Single Agent and Multi Agent RL. Strong focus on traffic signal controls.
Farazi et al. (2021)	Broad overview of RL in the wider transportation domain. Includes mapping of problems to RL methods. Highlights simultaneous decision making of both vehicles and infrastructure.
Yan et al. (2022)	A comprehensive review of RL in logistics and supply chain management, including the Vehicle Routing Problem and urban logistics. Highlights the need to use different agents in complex cases.
Schmidt et al. (2022)	A review of MARL in autonomous mobility in four application areas: traffic control, autonomous vehicles, resource optimization and unmanned aerial vehicles.
Saharan et al. (2020)	Dynamic pricing techniques for ITS, importance of evaluation parameters, limitations of techniques.
Lombardi et al. (2021)	Overview of design, simulation, implementation and evaluation of tolling schemes.
Wei et al. (2021)	Recent advances in the use of RL for traffic signal control, including analysis of environments, experiment settings and evaluation of approaches.
Kumar and Raubal (2021)	Traffic congestion alleviation using Deep Learning. Systematic overview of recurring and non-recurring congestion types including RL solutions.
Qin et al. (2022)	RL in ride sharing business processes. Amongst other for pricing and dynamic routing.
Bogyrbayeva et al. (2024)	Machine Learning to Solve the Vehicle Routing Problems. Covers RL next to DL, heuristics and combinations.

- We analyze how RL is applied to solve the road pricing problem and what objectives are achieved and which challenges are addressed.
- Based on our model and the scope of the various works we identify the most important challenges. We provide additional areas for future work.

1.3 Methodology and Structure of the Survey

Research papers were identified through searches for reinforcement learning, tolling, road pricing, congestion pricing and congestion charging within prominent conference proceedings and journals (as indexed by Scopus, Google Scholar, DBLP and similar sources). We excluded works that were not sufficiently specific, such as those using pricing for electric vehicle charging as an incentive for changing user behavior or those considering the relationship with autonomous vehicles. This process yielded 19 works, specifically addressing RL and road pricing, for inclusion in this survey with 17 selected for detailed analysis. We used as source for the different theories the original and seminal papers as they are referenced in traffic and road pricing published works. For reinforcement learning theory, we consulted and included original works and existing surveys. The total number of works included in this survey is 144.

The remainder of this survey is structured as follows. Preliminaries and background are provided in three sections. Section 2 introduces modeling traffic and three traffic models which provide a basis for analyzing the traffic models in the works on RL and road pricing. In Section 3 we introduce road pricing. Section 4 provides a high level introduction to RL and a detailed introduction is provided in Appendix A. Section 5 formulates the traffic congestion problem in a reinforcement learning setup and, in the context of pricing, important research challenges. In the sequel, Section 6 discusses recent works applying reinforcement learning as solutions to the problem defined in Section 5. Section 7 describes how the challenges have been addressed and future research directions for reinforcement learning in the road pricing domain. Section 8 concludes the survey.

2 Modeling Traffic

2.1 Traffic Model - Network

We assume that we operate on a network (Dafermos and Sparrow, 1969). A network $G(N, L)$ is a graph where N is a set of nodes and L is a set of directed edges between nodes in N, referred to as links. A link is typically denoted as a pair of nodes, the first being referred to as the start-point and the second as the end-point, (s, e), or we may write $l^{\langle s,e\rangle}$, instead. A network models a real life network, where the nodes correspond to real road junctions and the links to the real roads that connect the different junctions. In some cases synthetic networks are used, that are more abstract, to illustrate a concept or idea. A network is commonly governed by a network authority, which can impose a range of centralized and/or decentralized controls. A path is a sequence of links where the end-point of each link (apart from the one of the last) is the start-point of the subsequent link. The start-point of the first link and the end-point of the last are the origin and destination of the path, respectively. A path from a node o to a node d is denoted as $p^{o\to d}$, that is, $p^{o\to d}=\{l^{\langle o,x_1\rangle}, l^{\langle x_1,x_2\rangle}, \ldots, l^{\langle x_{n-1},x_n\rangle}, l^{\langle x_n,d\rangle}\}$. Note that there may be more than one path from an origin node o to a destination node d. The set of such paths is denoted as $P^{o\to d}$. We denote by $\mathbb{P}^G$ the set of all possible paths in a network G.

2.2 Traffic Model - Vehicles - State and Demand

Vehicles move on the network. Let $\mathcal{V}$ be the universe of vehicles. At time $\underline{t}$ a vehicle v is in position s on a link, moving towards its destination node o. We assume the final destination to be always a node, that is they never end in the middle of a road. The state of a vehicle v on the network at a specific time $\underline{t}$ can be modeled as a tuple $\langle \underline{t}, v, l^{\langle b,o\rangle}, s, p^{o\to d}\rangle$, where $l^{\langle b,o\rangle}$ is the link on which the vehicle is found, s is the position of the vehicle on the link l, and $p^{o\to d}$ is the path that the vehicle intends to follow after it reaches the end-point of the link $l^{\langle b,o\rangle}$ in order to arrive to its final destination d. Although time is a continuous quantity, we consider discrete ordered time points $\underline{t} = 0, 1, \ldots, T$. Let $\mathcal{T}$ denote the set of these time points. Note that we use $\underline{t}$ as t is reserved for episode steps, to be introduced in Section A.

The set of all the states of all vehicles at all different time points, denoted as $\mathbb{D}$, represents the amount of traffic on the network. $\mathbb{D}$ is referred to as traffic states set. We will use the notation $\mathbb{D}[cond]$ to select the states that satisfy certain conditions *cond*. For instance, $\mathbb{D}_{[\underline{t}\leq 5 \wedge l^{\langle b,o\rangle}=l]}$ denotes the states of the vehicles before time $\underline{t} \leq 5$ on the link l. If we assume an analysis period of traffic $[\underline{t}_a, \underline{t}_b\rangle$ and consider an origin o and destination d we define demand $D_{od}(\underline{t}_a, \underline{t}_b)$ as the number of vehicles willing to travel from o to d during that period.

2.3 Link and Vehicle Properties

Every link in the network has a set of properties. There are three types of such properties. First, there are properties related to physical characteristics of the link that are naturally invariant. An example is the length le of the link. Second, there are properties that are invariant as the traffic progresses, but the network authority has the ability to explicitly set them to a different value. Examples of these adaptable properties include the maximum capacity of vehicles allowed on a link, the maximum speed limit u_{max}, or a possible toll price τ. The network authority that controls these values may adapt them when it sees fit, or may have them also change based on some function of time, for example having higher tolls during business hours and much lower tolls after midnight. Finally there are characteristics that depend on the traffic at each specific moment, which means that these characteristics are dynamically changing over time. Illustrative examples are the density k (vehicles/m), the average speed of vehicles u (m/s) and the flow q (vehicles/s). The values of these dynamic properties of the links over time are referred to as a traffic pattern.

A vehicle also has some invariant properties, for example its maximum speed, maximum acceleration/deceleration, and length. Dynamic properties are driving speed, steering direction, and position.

2.4 Cost Function and Route and Departure Time Choice

When a vehicle is about to start moving from a point in the network towards a final destination node, it selects a route. A number of different parameters are taken into consideration, like the number of vehicles on the network, their speed, as well as other preferences of the vehicle.

To model this situation, we assume the existence of a cost function which a vehicle wants to minimize. It provides each vehicle a quantitative value, the predicted cost, for each possible route from the current position until the destination. The function takes into consideration the physical characteristics of the network, e.g. the number of links to traverse, the length or travel time at maximum allowed speed of links, in order to opt for shortest routes. It may also consider the dynamic characteristics like the observed traffic in order to opt for less congested links. It is also possible that the function takes the experienced costs into account which is their history of costs experienced on the network. In most cases the elements contributing to the cost are transformed to a single number. Thus, in its most general form the cost function for a vehicle v is $c_v{:}\mathbb{P}^G \times \mathbb{D} \rightarrow \mathbb{R}$. While this function, consisting of two arguments, is defined on all possible paths $\mathbb{P}^G$ and all history is kept in $\mathbb{D}$, it is in practice always applied

to a small subset. For example, at decision time, a vehicle could assume the cost of traversing a link to be the free flow travel time (time required when no other vehicles are present). Often the formula is assumed the same for the vehicles, but varies in a set of parameters that identify how much the various factors affect the final costs. For instance a multiplication factor converting travel time to money, commonly called value of time (vot) is usually not homogeneous across vehicles. We illustrate the use of the abstract cost function c_v by relating it to the link cost function (11, Section 6.2.1) from Chen et al. (2018) which provides practical context. Recall the cost function:

$$c_v = \mathbb{P}^G \times \mathbb{D} \to \mathbb{R}. \tag{1}$$

Assume that vehicle v chooses a path $p \in \mathbb{P}^G$ at decision time t. The path p is decomposed in its constituent links: $p = \{l^{\langle o,x_1\rangle}, l^{\langle x_1,x_2\rangle}, \ldots, l^{\langle x_{n-1},x_n\rangle}, l^{\langle x_n,d\rangle}\}$.

We now consider the per-link cost (as defined in Section 6.2.1):

$$c_v(l,\underline{t}) = tt(l,\underline{t}) * vot + \tau(l,\underline{t}). \tag{11}$$

This depends on the toll $\tau(l,\underline{t})$ and $tt(l,\underline{t})$ which is itself a function of $\mathbb{D}_{[\underline{t},l]}$ the state of traffic on that link at that time, which depends on the paths chosen by other vehicles, which were based on their evaluations, based on $\mathbb{D}$. We now evaluate the cost for the full path by aggregating the per-link costs:

$$c_v(p,\mathbb{D}) = \sum_{l\in p} c_v(l,t). \tag{2}$$

Substituting the expression for $c_v(l,\underline{t})$ we get:

$$c_v(p,\mathbb{D}) = \sum_{l\in p} tt(l,\underline{t}) * vot + \tau(l,\underline{t}). \tag{3}$$

Each $tt(l,\underline{t})$ is estimated based on vehicle speeds on link l at time $\underline{t}$, which recursively depends on the network-wide vehicle distribution on prior decisions of other vehicles. Hence the cost function incorporates, albeit implicitly, the full network traffic pattern encoded in $\mathbb{D}$.

The computation of the cost and the planning of the route to follow can take place either once when starting its route and keep the route that was decided, or may be recomputed every time the vehicle is at a node. The choice of the path based on the cost may be deterministic or stochastic. In case of stochastic path choice, a softmax function is commonly used to translate path costs to probabilities:

$$p_{v_i} = \frac{e^{-\theta c_{v_i}}}{\sum_j e^{-\theta c_{v_j}}}, \tag{4}$$

where θ is a scaling parameter and c_{v_i} represents the cost for vehicle v when taking path i. In some cases the departure time is not fixed, meaning the vehicle can opt to

depart earlier or later, or even refrain from departing. We denote this process route and departure time choice as $\mathcal{R}$. For a network G and a set of vehicles V with a traffic demand set $\mathbb{D}$ we refer to the tuple $\langle G, V, \mathbb{D} \rangle$ as a traffic network.

2.5 Network Loading

At any new time point, the vehicles that are on the network need to move towards their planned destinations as dictated by their planned paths. This process is known as network loading, and is denoted as $\mathcal{M}$. Given route choices for vehicles, network loading provides the resulting route travel times. After a network loading step, the network has new characteristics, and as such, a path refresh phase follows where the planned routes of the different vehicles get recalculated. After the network loading and the planned route refresh phase, the network gets into a new state:

$$\mathbb{D}_{[t=k+1]} = \mathcal{R}(\mathcal{M}(\mathbb{D}_{[t=k]})). \tag{5}$$

See Figure 1 (modified from Boyles et al. (2023)) for an illustration.

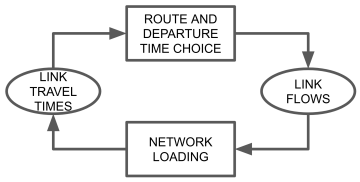


Fig. 1: Traffic Model without Toll

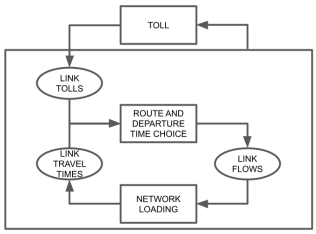


Fig. 2: Traffic Model with Toll

There are different types of model for network loading, to represent the movement of vehicles on a network of which we will describe three types. In a microscopic model, every vehicle is considered an individual entity and its behavior, characteristics and interactions are studied independently. Characteristics like position, speed, and distance from other vehicles are computed for each vehicle individually. In each network loading step, these characteristics are updated for every vehicle. For instance, in the Intelligent Driver Model (IDM) (Treiber et al., 2000), the acceleration of a vehicle is computed using vehicle and other vehicles' characteristics, leading to its position update. Other microscopic models (Ahmed et al., 2021; Chowdhury et al., 2000; Diallo et al., 2021; Knorr, 2013; Nguyen et al., 2021) exist but the main principle to calculate vehicle movement remains the same. Traffic simulators which use IDM are among other SUMO (Lopez et al., 2018), CityFlow (Zhang et al., 2019) and CBLab (Liang et al., 2022). Widely used commercial software for microscopic traffic simulation using different models are PTV VISSIM (Wiedemann model), Paramics (Fritzsche model) and AIMSUM (Gipps model) (Ahmed et al. (2021)).

To study traffic in a dynamic macroscopic representation, the Cell Transmission Model (CTM) (Daganzo, 1994) is often used. A link is divided into a finite number of

sections (called cells), and the traffic is studied by evaluating each cell at every time step, by transferring part of the traffic from each cell to the next. A key distinction from the microscopic model is that this model propagates vehicles not individually but in the aggregate. The CTM has its origins in a hydrodynamic model of traffic flow, the Lighthill-Whitham-Richards (LWR) model (Lighthill and Whitham, 1955; Richards, 1956). It uses three concepts: traffic flow q, speed u, density k, and their relationship on a link, the fundamental diagram, $q = uk$. A corresponding node model for CTM facilitates connections, merges and diverges (Daganzo, 1995; Tampère et al., 2011).

A third model is the static macroscopic model. In a static macroscopic model the congestion properties on a link are usually described using a Link Performance Function (LPF). This function expresses the average or steady state travel time on a link as a function of the traffic flow on the link. The most commonly used LPF is from the Bureau of Public Roads (United States Bureau of Public Roads, 1964). Detailed information on dynamic macroscopic models (Chiu et al., 2011) and on static models (Verhoef, 1999) can be found elsewhere. Other works make comparisons between the various models (Hoogendoorn and Bovy, 2001; Storani et al., 2021; van Wageningen-Kessels et al., 2015).

2.6 Objectives and Equilibria

One objective of the network authority is to optimize the performance, measured by some metric, of a traffic network. There are different metrics that can be considered for that purpose. One is the total travel time, the total time needed for all the vehicles to arrive at their destination, another the overall distance traveled by all vehicles, yet another the rate in which the vehicles arrive at their intended destination. The metric is calculated by some objective function J. We assume that the network authority can choose to apply means to influence the traffic. Now assume that every vehicle selects the route that minimizes its cost function c_v. Over time, traffic is then likely to move to a situation in which no vehicle can reduce its cost by choosing another path (Wardrop, 1952). This situation is called a User Equilibrium or Dynamic User Equilibrium (Chiu et al., 2011) in case of dynamic traffic assignment. It may imply that some links are congested, while others are underutilized.

Assume now that the network authority has the objective of minimizing the total cost and has full control over the vehicle route choices. In that case, diverting some vehicles to other paths, non-optimal for themselves, can help alleviate the congestion and improve the traffic in the network resulting in minimization of the total travel time. This situation in which the overall network traffic is optimized is known as the (Dynamic) Social Optimal equilibrium (Wardrop, 1952). However, in most cases, a network authority has not full control and will need to apply other means to influence vehicles to change their paths.

There are various means by which traffic can be controlled, implying vehicles to follow routes that are not always the most optimal for them. One is through hard restrictions, which divert vehicles towards specific links or restrict their speed (traffic signaling, traffic lights or speed limits). For this to materialize, the network authority needs to have a global view of the network and the ability to implement these controls. Once implemented, these controls can either operate decentralized, cooperatively or

not, or centralized. Another way to affect traffic is to let the final decision be made to individual vehicles, but at the same time providing incentives for them to choose routes that will improve overall network traffic, even if for the individual vehicle the route is not the most beneficial. The charging of a toll for road use is an example of such an incentive.

3 Road Pricing

3.1 Why Road pricing?

Road pricing (Tsekeris and Voß, 2009) can be used to manage travel demand, raise revenues for funding transport investment or a combination of both. Just as other demand management tools noted in Section 2, road pricing builds on the assumption that people weigh the cost and benefits of their actions and take their actions to maximize their utility.

3.2 Modeling Road Pricing

To illustrate the workings of road pricing we describe first best pricing, time dependent road pricing and second best pricing. First best road pricing is termed first best because it relies on a highly idealized scenario, assuming, among other things, that tolls are charged on all roads with perfectly differentiated prices. It is usually illustrated with a single road (Verhoef, 2002). All vehicles have the same cost function $c_v(q)$ for this road and the demand function $d(q)$ is also dependent on q. Traffic flow, speed and density are independent of time and uniform along the road. The User Equilibrium which will materialize in this case does not take account of the costs vehicles impose on each other. To mitigate this, a Marginal Cost Toll can be derived and applied, leading to a Social Optimal equilibrium. This toll is also known as a Pigouvian tax (Knight, 1924; Lindsey and Verhoef, 2001; Pigou, 1924). A detailed analysis using variations of the cost curve (for example, when it is extended with a backward bending part because of continuous congestion) is given by Verhoef (1999). Furthermore, marginal cost pricing for a single road, can also be applied to a general road network (Yang and Huang, 1998). We do not cover these cases here.

In a time dependent model, the time-independent model is extended with a time-dependent travel demand function and specifies how the flow changes over space and time. Demand is assumed fixed (price independent), but vehicles are heterogeneous with respect to their trip-timing as well as value of time. The private cost of a trip is (Arnott et al., 1993):

$$c_v(\underline{t}) = \alpha T(\underline{t}) + \beta(\text{time early}) + \gamma(\text{time late}), \tag{6}$$

where α is the *vot*, β is the unit cost for arriving early and γ is the unit cost for arriving late. The cost of arriving early or late is referred to as schedule delay cost and T as travel time cost. We note the desired arrival time as $\underline{t}^*$. Time early is $max[0, \underline{t}^* - \underline{t} - T(\underline{t})]$ and time late is $max[0, \underline{t} + T(\underline{t}) - \underline{t}^*]$. Under certain conditions, vehicles cannot overtake each other and tolls can vary freely over time (Arnott et al.,

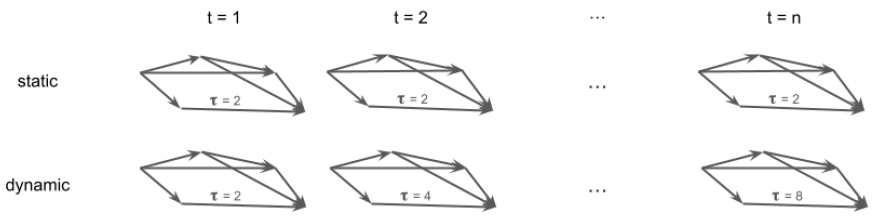


Fig. 3: Static and dynamic tolling of a single road

1993; Lindsey and Verhoef, 2001), a User Equilibrium and Social Optimal equilibrium can be calculated and tolls can be applied to achieve this Social Optimal equilibrium. More variants of this model exist (Lindsey and Verhoef, 2001). First best pricing and time dependent models are often not feasible given practical constraints (not all roads can be tolled), or information limitations (incomplete information for vehicles).

Second best pricing methods (Arnott, 2007; Ekström, 2014; Tsekeris and Voß, 2009; Verhoef, 2000, 2002), are more realistic, as various constraints are taken into account. Examples of second best pricing methods are pay lanes and cordon pricing in contrast to pricing every road, the use of step tolls as opposed to smoothly time varying tolls and a fixed daily tolling schedule rather than a schedule dependent on traffic conditions during the day. In relation to this, we also distinguish static and dynamic road pricing (Fig. 3). In static road pricing, toll prices are determined by analysis of historical and economic data, without taking into account current traffic. In dynamic road pricing, tolls are determined by also taking into account current or anticipated congestion levels, and hence can vary, based on the number of vehicles, their speed, and their location on the network (Cole et al., 2003; Como and Maggistro, 2021; Eliasson, 2017; Genser and Kouvelas, 2019, 2022; Maheshwari et al., 2024; Nohekhan et al., 2021; Paccagnan et al., 2021; Pandey and Boyles, 2019; Vickrey, 1963).

Dynamic road pricing is a challenging task, as driver behavior on the network needs to be continuously monitored and tolls need to be adjusted, with each having an immediate effect on the other. Determining and setting prices is performed by toll agents

3.3 Toll Agents

Toll agents (ta) are charged with observing traffic on the network and determining tolls. A model representing a traffic network with a toll agent is given in Figure 2 (modified from (Boyles et al., 2023)). The figure illustrates that toll agents obtain information from the traffic system as a basis to determine the tolls to charge for link use to vehicles. Toll agents can be given objectives by the network authority, for example with respect to the flow on a link. If provided with a license the toll agents determine their own objective within the constraints of the license. For example to maximize profit on a tolled link in which they invested.

We detail baseline objectives for toll agents. We use the following equation:

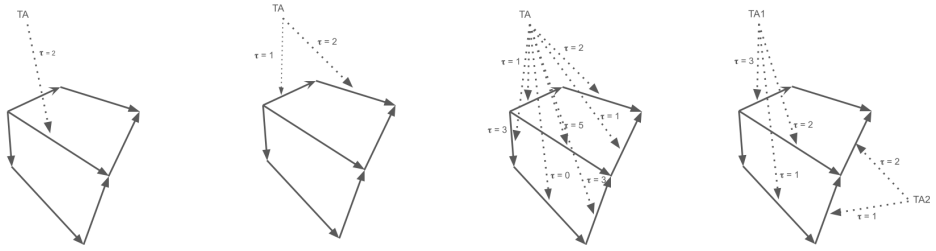
$$J = \sum_{v \in \mathcal{V}} (\phi * tt(v) + \psi * \sum_{l' \in p(v)} \tau_{l'}). \tag{7}$$

Here ϕ is a scaling parameter, often representing the vot, $tt(v)$ the experienced travel time of the vehicle, $p(v)$ is the path which the vehicle has traversed, l' a tolled link and $\tau_{l'}$ the toll incurred when traversing that link. The baseline objectives are given in Table 2. When considering Social Welfare, tolls are not taken into account as tolls are considered transfer payments which remain internal to society (Mirzaei et al. (2018a)). If the average is taken over the number of vehicles $|\mathcal{V}|$ we use ASTT and ASC.

Table 2: Baseline objectives for toll agents

	ϕ	ψ
Total System Travel Time (TSTT)	1	0
Total System Cost (TSC)	vot	1
Social Welfare (SW)	vot	0
Maximum Revenue (MR)	0	1

On a network, one (global) or more (local) toll agents can operate (Fig. 4). In the latter case each one covering a part of the network. Toll agents can work together to achieve the same objective or pursue their own objectives, leading to different levels of cooperation or competition. When starting to set tolls, vehicles may take other



(a) Single agent for one road (b) Single agent for multiple roads (c) Global agent for all roads (d) Multiple agents for multiple roads

Fig. 4: Example toll agent configurations

decisions with respect to their mode choice, departure time and route choice. This causes different traffic patterns which the toll agents may not have anticipated causing them to either adjust their prices or their tolling model.

In Section 3.2 we described how a toll is determined in the case of a first best pricing situation on a single link. To determine tolls for road pricing on networks, different methods and technologies have been proposed (De Palma and Lindsey, 2011)

using various coverage and toll differentiation mechanisms. Joksimovic et al. (2005) identifies a mechanism for optimal toll design in a dynamic network with route and departure time choice. Dynamic pricing for ITS surveyed in (Saharan et al., 2020) describe among other dynamic programming, evolutionary optimization, swarm optimization and game theory based techniques. A survey work (Lombardi et al., 2021) on model-based dynamic toll pricing presents an overview of methods for price definition, simulation techniques and technology applications. Reinforcement Learning, which we cover next, is another mechanism to determine toll prices for toll agents.

4 Reinforcement Learning

Reinforcement learning (RL) is a computational framework wherein an agent learns to take optimal decisions through direct interaction with an environment. The core interactive loop consists of the agent observing the environment's state, selecting an action and receiving a reward signal. For the tolling problem, an action is, e.g., raising or lowering the toll on a given road, while the reward could, e.g., be (inversely) related to the time it takes vehicles to go from origin to destination. The agent's goal is to develop a policy (a mapping from states to actions) that maximizes the long-term cumulative reward.

The standard formalism for this sequential decision-making problem is the Markov Decision Process (MDP), formally defined as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$ that represents the set $\mathcal{S}$ of all states s, the set $\mathcal{A}$ of actions a, a transition probability function for the environment $\mathcal{P}$, and a reward function $\mathcal{R}$.

The central challenge in RL is to find an optimal policy π_* by learning value functions ($V(s)$ or $Q(s,a)$) which estimate expected cumulative rewards, or by directly optimizing the policy itself. Solutions to MDPs are based on the Bellman equations, which provide recursive relationships for value functions under a given policy (Sutton and Barto, 2018). In road pricing contexts specifically, RL enables the modeling of adaptive tolling strategies in dynamic, uncertain traffic environments. For a detailed treatment of these foundational concepts and algorithms, readers are directed to Appendix A.

RL algorithms are typically categorized into three main categories, all three have been used for the tolling problem. Value-based methods (e.g., Q-learning), which focus on learning an accurate value function and are well-suited to problems with discrete action spaces. In contrast, policy-based methods (e.g., policy gradient), directly search a policy which make them suitable for handling continuous action spaces and learning stochastic policies. Actor-Critic methods, which represent a powerful hybrid architecture where an actor (policy) learns which actions to take, and the critic (value function) evaluates the quality of that action. Many state-of-the-art algorithms are based on this structure.

We further distinguish between model-free and model-based approaches. Model-free methods learn directly from experience without ever creating an explicit model of the environment's dynamics ($\mathcal{P}$ and $\mathcal{R}$). In contrast, model-based methods either use a known model of the environment or first learn a model from experience and then use this model for planning and policy optimization.

While traditional tabular methods for the above approaches were feasible in domains with (relatively) small state and action spaces, they failed to scale to larger, complex problems. Deep Reinforcement Learning overcomes this limitation. Deep RL leverages deep neural networks as universal function approximators to represent policies and/or value functions. The integration of deep learning with reinforcement learning has enabled the field to solve larger problems with high-dimensional state- and action spaces. This makes it an essential tool to address modern real-world challenges such as the tolling problem analyzed in this survey.

5 Problem Formulation and Research Challenges

We will now formulate our problem statement and frame this in reinforcement learning terminology.

Problem. *Given* $\langle G, V, \mathbb{D} \rangle$*, a horizon* H *and an objective* J *find a policy* π *for a Toll Agent* ta *to set toll prices on the links* l *so that* J *is optimized.*

5.1 The Road Pricing Problem in RL Terminology

The road pricing problem can be formalized as:

- *Environment*: the Traffic Network $\langle G, V, \mathbb{D} \rangle$.
- *Agent*: the Toll Agent, ta.

In this road pricing problem the ta has an objective J (see Table 2) that it needs to achieve.

The road pricing problem is further formalized as a MDP:

- $\mathcal{S}$: is the set of all states s. This is a representation of the current situation in the environment: speed, travel times, flow, density on the links and so on. It may also include information on nodes and information from other artefacts like speed limits, ramp metering and traffic lights. State information is collected by the agent. If it consists of a single number, without applying some function, we call it direct, otherwise composite. The state s or observation o could also be subject to preprocessing. Preprocessing can involve deep learning mechanisms like Graph Convolutional Networks (GCN) (Kipf and Welling, 2017) or transformers (Vaswani et al., 2017).
- $\mathcal{A}$: is the set of all actions a. On the basis of state s the agent takes an action a. This action a consists of setting new toll prices for one or more links in the Traffic Network respectively. The actions, can either be the toll price chosen from a discrete set of prices, from a certain price range, or adjustments to the toll price for one or more links. These adjustments are made in accordance with the $ta's$ policy π.
- $\mathcal{P}$: is the transition probability function mapping from any state s to a next state s' after taking action a: $\mathcal{P}(s'|s, a)$. If a toll price is set, vehicles v will act on this price (and traffic conditions), propagating the traffic flow by network loading (5). This will lead to a new state. It is unlikely that $\mathcal{P}$ is known in a traffic network so the agent needs to learn from experience.

- $\mathcal{R}$: the reward function provides a reward from the environment to the agent. The reward signal could for example be the number of vehicles arriving periodically at a certain node, the average speed on a link, the density on a link (direct) or some combination of other values (composite). As the reward, and therefore return, are the signals to evaluate for the *ta* if its actions contribute in a positive or negative way to its objective J, it is therefore crucial that the reward signal is properly defined. Reward signals are combined to produce the return, using a discount factor $\gamma \in [0, 1]$, averaging or other mechanism. As the objective J can not always be observed by the *ta*, it needs to design its reward structure so that its return can act as a proxy for J.

The *ta* has a stochastic policy $\pi(a|s)$ or a deterministic policy $\pi(s)$. Based on the state and this policy, the *ta* sets the toll price or toll price adjustment on one or more links. In case of more than one *ta* the problem needs to be generalized to a Markov Game (Section A.3) and, when other elements of the environment (vehicles, traffic signals) become learning agents, as well.

5.2 Research challenges

Common challenges in (Multi Agent) Reinforcement Learning as surveyed in (Albrecht et al., 2024; Du and Ding, 2021; Dulac-Arnold et al., 2021; Gronauer and Diepold, 2022; Patterson et al., 2024; Wong et al., 2022; Yuan et al., 2023) are outlined below with a specific focus on the implications for the road pricing problem.

Partial Observability Partial Observability (Section A.1), can occur if a toll agent has only information on the links for which it needs to provide the toll prices but not on all links of the network. It may also be that state information is distorted because of sensor failures, or simply because information is not available to the toll agent.

Credit Assignment Credit Assignment (Section A.2) in road pricing may be challenging due to spatio-temporal dependencies in the environment and actions of toll agents. Depending on the route and departure time model of vehicles and the resulting traffic dynamics, it can be difficult to specifically identify which action contributed to which (part of the) reward. Note that credit assignment in single agent settings is already challenging due to spatio-temporal dependencies; this is compounded in situations where multiple agents act simultaneously.

Non-stationarity Non-stationarity (Section A.3) may occur if multiple toll agents are learning and updating their policies simultaneously. When the state of the environment is dependent on the joint actions (toll prices) of all toll agents, the toll agents need to adapt to these new policies, which violates the MDP assumption and convergence may no longer be guaranteed (Wong et al., 2022).

Scalability When the problem is extended from a single agent to a multi agent setting this is accompanied by an expansion of the state and action dimension, which may lead to an exponential rise in the joint action dimension (Du and Ding, 2021; Yuan et al., 2023). Solutions with more agents may also increase the credit-assignment and non-stationarity challenges (Albrecht et al., 2024).

Generalizability This challenge addresses whether trained toll agents can perform well in new and unseen situations. In the road pricing problem, this would encompass among other different traffic demand patterns, changes in network topology and disturbances in data provided (for example inaccurate measurements, sensor noise, missing data).

In addition to the challenges outlined above, the definition of the state, action and the identification of a reward function is part of the research in road pricing problems as well. This contrasts to benchmark problems (Bettini et al., 2024; Bellemare et al., 2013; Sutton and Barto, 2018; Todorov et al., 2012) where these are explicitly defined.

The surveyed works in the next Section (Section 6) detail how these challenges are addressed in the road pricing problem and are summarized in Section 7.

6 Road Pricing Approaches using Reinforcement Learning

In this section a detailed analysis is provided of the ways in which the problem and challenges, formulated in Section 5 has been addressed. The works are clustered in subsections by problem domain and RL solution approach. A conscious attempt has been made to align the notation between the various works and the notation introduced in Section 2 and Section A. The traffic model (Section 2) is the basis for describing the most relevant characteristics, specific for the works. The description of the toll mechanism and application of RL builds on the background information (Sections 3 and A). Additional detail is provided where necessary. The most important characteristics of the works are listed in Table 3.

We emphasize that in the models we both have episode steps $t \in 0, 1, \ldots, H$ and time steps $\underline{t} \in 0, 1, ..., T$ where there are Γ time steps in an episode step. This implies that if a vehicle uses the toll price of a link ($\tau_{l'}(\underline{t})$) it uses the toll price set by the agent at the latest episode step t prior to $\underline{t}$. Time can be indicated both by t and $\underline{t}$.

6.1 Policy Gradient for Enhanced Δ-Tolling in Road Networks

6.1.1 Enhanced Δ-Tolling with Finite Difference Policy Gradient

Δ (Delta)-tolling (Sharon et al., 2016, 2017)(Sharon et al., 2017) dynamically tolls all links in a network optimizing Social Welfare using fixed parameters. In Enhanced Δ-tolling (Mirzaei et al., 2018a,b) ($E\Delta$-tolling) the objective is to optimize Social Welfare (See Table 2) by optimizing the parameters of Δ-tolling. They therefore make the strong assumption that for each vehicle the *vot* is known. This is achieved by using a Finite Difference Policy Gradient Reinforcement Learning (FD PGRL) (Kohl and Stone, 2004) algorithm to learn the parameters that determine the toll.

Traffic model set-up: Given $\langle G, V, \mathbb{D} \rangle$: (c_v) For each link in a network the toll is calculated in two steps: first, Δ is calculated every time step $\underline{t}$ based on the difference between the travel time on a link $tt(l, \underline{t})$ and the free flow travel time $T(l)$ on the same link:

$$\Delta(\underline{t}) = tt(l, \underline{t}) - T(l). \tag{8}$$

Table 3: Overview of important characteristics in surveyed works

	6.1.1 (Mirzaei et al. (2018b))	6.2.1 (Chen et al. (2018))	6.2.2 (Qiu et al. (2019))	6.2.3 (Jin et al. (2021))	6.2.4 (Wang et al. (2022))	6.2.5 (Lu et al. (2024))	6.2.6 (He et al. (2024))	6.3.1 (Zhu and Ukkusuri (2015))	6.3.2 (Pandey and Boyles (2018))	6.4.1 Pandey et al. (2020))	6.4.2 (Zhang et al. (2023))	6.5.1 (Sato et al. (2021))	6.5.2 (Sato et al. (2022))	6.6.1 (Tavares and Bazzan (2014))	6.6.2 (Chakravarty et al. (2024))	6.7.1 (Ramos et al. (2018))	6.7.2 (Ramos et al. (2020))
Toll agent objective																	
Minimal Total System Travel Time						✓	✓	✓		✓	✓					✓	✓
Minimal Average System Travel Time					✓						✓					✓	✓
Maximal Social Welfare	✓																✓
Minimal Total System Cost							✓							✓			
Maximum Flow		✓			✓							✓	✓	✓			
Minimal Delay			✓	✓								✓	✓		✓		
Maximum Revenue									✓	✓	✓				✓		
Vehicle departure time and route choice																	
Departure time												✓	✓				
Deterministic route choice	✓				✓	✓			✓		✓			✓	✓	✓	✓
Stochastic route choice		✓	✓	✓			✓	✓		✓			✓				
Dynamic route choice	✓	✓	✓	✓				✓	✓	✓	✓						
Departure time and route choice													✓				
Reinforcement learning approach																	
Value based								✓	✓			✓		✓	✓	✓	✓
Policy based	✓																
Actor-critic		✓	✓	✓	✓	✓	✓			✓	✓		✓				
Architecture if multi agent																	
DTDE								✓					✓	✓		✓	✓
CTCE													✓				
CTDE			✓	✓	✓	✓			✓				✓				
State or observation																	
Direct	✓																
Composite		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓		
Preprocessed			✓		✓	✓											
Action																	
Toll price road		✓	✓	✓		✓		✓	✓	✓	✓			✓	✓	✓	✓
Toll price road adjustment	✓											✓	✓				
Toll price route					✓												
Toll price region							✓										
Reward																	
Direct		✓	✓	✓			✓							✓		✓	✓
Composite					✓	✓		✓	✓	✓	✓	✓	✓		✓		

After that, Δ is multiplied by a proportionality factor β so the product is proportional to the difference between travel time and free flow travel time. Then this and the previous toll τ are smoothed with a weighting factor R:

$$\tau(l, \underline{t}+1) = R(\beta\Delta(\underline{t})) + (1-R)\tau(l, \underline{t}). \tag{9}$$

The calculated toll is then set to the link in real time. The corresponding cost function is:

$$c_v(l, \underline{t}) = tt(l, \underline{t}) * vot + \tau(l, \underline{t}). \tag{10}$$

($\mathcal{R}$) Route choice: A vehicle optimizes its route dynamically at every node for minimal cost, based on actual travel times and actual tolls. ($\mathcal{M}$) Network loading is done using the CTM.

RL set-up: In a FD PGRL algorithm the parameters $\boldsymbol{\theta}$ of the policy π are optimized with gradients as usual. The negative total cost (10) in an episode, exclusive of toll $c_v(l, \underline{t}) = tt(l, \underline{t}) * vot$, is used as the reward signal. However, these gradients are calculated after performing an episode with a number of policies with slightly perturbed parameters (Peters and Schaal, 2006) which lead to different performance. Using the differences in the parameters and the differences in performance, the gradients are estimated. Here, β and R are optimized by a FD PGRL-algorithm. The specific version used here (Kohl and Stone, 2004) updates the policy parameters with an adjustment vector. The variants in this work consist of different Local/Global combinations of β and R in the network. Local means every link has its own parameter and Global means all links have the same parameter. Variants are: Global R, Local β ($E\Delta$-tolling$_\beta$), Local R, Global β ($E\Delta$-tolling$_R$), and Local R, Local β ($E\Delta$-tolling$_{\beta,R}$).

Results: Experiments are conducted on various networks. All variants show improvements up to 45% over no tolls. $E\Delta$-tolling$_\beta$ underperforms the other variants and performs only slightly better than regular Δ-tolling. Another drawback is that $E\Delta$-tolling is limited in its convergence rate; it is optimized for a specific traffic pattern and might be too slow to learn a new traffic pattern.

6.2 Actor-Critic Dynamic Toll Collection for Road Networks

6.2.1 Dynamic Electronic Toll Collection - PG-β

A Dynamic Electronic Toll Collection (DyETC) system in a network, consisting of tolled and non-tolled links, uses Policy Gradient with actor-critic (Chen et al., 2018). The objective is to optimize traffic flow by dynamically adjusting tolls. The policy is approximated by a parameterized function and optimized by a modified Policy Gradient (PG)-algorithm.

Traffic model set-up: Given $\langle G, V, \mathbb{D} \rangle$: ($c_v$) Link costs are based on travel time on a link and toll. The travel time is calculated with a LPF. The toll $\tau(l, \underline{t})$ in this case is equal to the action A_t^l:

$$c_v(l, \underline{t}) = tt(l, \underline{t}) * vot + \tau(l, \underline{t}). \tag{11}$$

Route costs consist of the sum of its link costs. ($\mathcal{R}$) Route choice of vehicles is based on a Stochastic User Equilibrium and route choice for vehicles is updated every time step. ($\mathcal{M}$) Network loading is done using a modification of a macroscopic static model.

RL set-up: A new algorithm, Policy Gradient-β (PG-β) with a separate critic is proposed, with a continuous action space. This improves traditional PG algorithms by using: a) time-dependent value and policy functions, b) use of the β-distribution as an alternative to the normal distribution to obtain a bounded action space, and c) state

abstraction: an assumption that vehicles on the same link have almost equal effects on tolls. The problem is formulated as a time-dependent Markov Decision Process on the traffic network, with time horizon H, and discrete episode steps $t = 0, 1, \ldots, H$. maintaining time-dependent value and policy functions. State is defined as S_t^{lf}, the number of vehicles on link l with destination node f at time t. The state vector of a link as $\mathbf{S}_t^l$ and state matrix of the network G as $\mathbf{S}_t$. An action $\mathbf{A}_t$ is defined as the toll vector which consists of the tolls A_t^l applied to link l. Once an action is applied at time t, the number of vehicles on each link is updated:

$$S_{t+1}^{lf} = S_t^{lf} - S_t^{lf,out} + S_t^{lf,in}. \tag{12}$$

The immediate reward function $R_t(\mathbf{S}_t)$ is defined as the number of vehicles that arrived at their destination during episode step t. Time-dependent policy and value functions imply that there is a set of policies and a set of value functions; for each episode step t a policy and value function. The value function at episode step t is defined as:

$$v_t(\mathbf{S}_t) = \sum_{t'=t}^{H} \gamma^{t'-t} R_{t'}(\mathbf{S}_{t'}). \tag{13}$$

The optimal policy for the toll agent at episode step t is:

$$\pi_{*,t}(\mathbf{A}_t|\mathbf{S}_t) = \arg\max_{\pi_t} v_t(\mathbf{S}_t). \tag{14}$$

A policy function approximator parameterized by $\boldsymbol{\theta}_t, \pi_t(\mathbf{A}_t|\mathbf{S}_t, \boldsymbol{\theta}_t)$ and a value function approximator parameterized by $\boldsymbol{\vartheta}_t, \hat{v}_t(\mathbf{S}_t|\boldsymbol{\vartheta}_t)$ are used in an actor-critic algorithm to train the policies. Different from the regular actor-critic is that each episode step has its own policy and value function for which the parameters are learned. Secondly, where usually in PG algorithms a Normal probability density function (pdf) is used for MDPs with continuous action spaces, here they use a β-pdf to adapt to a bounded action space.

Results: Experiments are on a synthetic traffic network and a real network. PG-β outperforms algorithms where the episode step is incorporated in the state or not considered at all. A similar algorithm using the Normal distribution does not learn an effective policy. Using PG-β with state abstraction is 75% more efficient than without. PG-β-abs is then compared in five scenarios with four other tolling mechanisms: Fixed toll: proportional to average demand, DyState: dynamic toll proportional to state, Δ-tolling (see Section 6.1.1) and no toll. In all scenarios PG-β-abs outperforms the other tolling schemes. In a second experiment on a real network, PG-β-abs outperforms all other mechanisms, and compared to the Fixed tolling scheme, which was second best, the realized flow was 8% higher and TSTT 14.6% lower. Scalability to larger networks is identified in the work as a challenge.

6.2.2 Dynamic Electronic Toll Collection - MARL

An extension of DyETC (Chen et al., 2018) (Section 6.2.1) with MARL, to enable larger networks and overcome the scalability problems, led to DyETC-MARL using

edge-based Graph Convolutional Networks (GCN) (Qiu et al., 2019). The work proposes the use of an edge-based GCN to extract spatio-temporal correlations of the network status and to apply cooperative MARL where each toll agent serves a part of the network. The objective of the toll agents is to maximize the total number of vehicles arriving at their destination. This means that each agent gets the same reward. The agents share a common value network.

Traffic model set-up: Similar to Section 6.2.1. In addition, the network G is split in partitions, that each cover one or more links.

RL set-up: The RL problem is formulated as in Section 6.2.1. The optimal policy for each agent $ta^i, i \in 1, \ldots, N$ covers its (ta^i) partition of the network. The agents adapt the CTDE framework using an actor-critic method for learning. Figure 6 presents the high-level architecture of the proposed system. Each ta learns its own policy (actor),

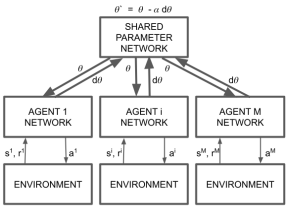


Fig. 5: Asynchronous updating of parameters by multiple agents (Mnih et al., 2016)

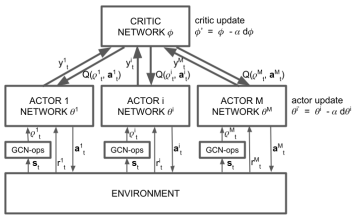


Fig. 6: DyETC MARL with GCN (Qiu et al., 2019)

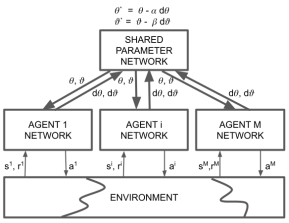


Fig. 7: DADO (Jin et al., 2021)

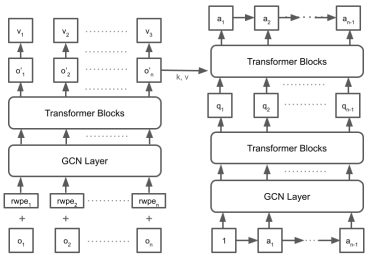


Fig. 8: MAGT-toll (Lu et al., 2024)

and together they use the same value network (critic). By sharing the critic and getting the same rewards, the agents have a common goal; this implies a natural way of cooperating between them. Because of the finite horizon of the MDP, the parameters of the actor and critic are time-dependent. The captured state information is preprocessed using a GCN (Kipf and Welling, 2017). With a GCN the graph nature of G is combined with the state information of each link to generate a richer input to each agent. The GCN's output ρ_i, is specific for each ta, using information from the

partition of that toll agent. So agent ta^i obtains ρ_i and generates the action for every link in its partition.

Results: Experiments are conducted on two real networks. MARL-eGCN, DPG-β-eGCN (not using the multi-agent approach but PG-β with a neural network) and PG-β are compared with each other on both networks. MARL-eGCN and DPG-β-eGCN, significantly outperform PG-β. In addition MARL-eGCN and DPG-β-eGCN converge both faster than PG-β.

In the next set of experiments, various ablation studies and robustness tests were conducted, among other different traffic settings and different pricing schemes. An important finding in this work is that performance decreases when more than 8 agents are used, and best performance is achieved with 6 agents. The use of more agents suffers from feature redundancy and more coordination costs. Fewer agents (than 6) cannot leverage the power of multi-agent learning (Qiu et al., 2019).

The solution demonstrates that decomposing the state and action spaces per agent and cooperation between agents helps solve the DyETC problem. Preprocessing state with a GCN exploits correlations between the links in the network.

6.2.3 Dynamic and Deadline Oriented Road Pricing Mechanism

DADO, considering vehicles' time requirements takes deadlines into account (Jin et al., 2021). For vehicles with a deadline, route cost is appreciated differently if the time interval x between the calculated arrival time and the deadline is below or above a threshold value D. The objective for the agents is to optimize the number of vehicles arriving prior to their deadline, using different reward functions.

Traffic model set-up: Given $\langle G, V, \mathbb{D}\rangle$: (c_v) For vehicles with $x > D$ remaining time is abundant and their cost is based on road tolls and the remaining time, increasing slowly if x decreases. If $x < D$ the time cost will increase dramatically. For vehicles without a deadline the regular way of calculating the cost per link is used (Equation (11)).Each vehicle is provided with a deadline and updates its route at every junction based on Stochastic User Equilibrium. $(\mathcal{R})$ and $(\mathcal{M})$ network loading are similar to the DyETC model (Section (6.2.1)) The objective of toll agents is to maximize the number of vehicles that arrive at the destination before the deadline.

RL set-up: State is defined as S_t^{lfd} where the deadline d is an additional index to the state defined in DyETC; for vehicles with no deadline, $d = 0$. The index d is also used in calculating the fraction of traffic flow in the Stochastic User Equilibrium calculation, as well as in the state transition calculation. Actions (tolls) are as defined in DyETC. Three different reward functions R are defined: R1 to maximize the number of vehicles arriving at their destination before the deadline, R2 to minimize the number of vehicles arriving after the deadline and R3 to minimize the time gap between time of arrival if beyond the deadline and the deadline. The value function is defined as:

$$v_t(\mathbf{S}_t) = \sum_{t'=t}^{H} \gamma^{t'-t} r_{t'}(\mathbf{S}_{t'}). \tag{15}$$

The policy is as in DyETC time-dependent: $\pi_t^i(\mathbf{A}_t|\mathbf{S}_t, \boldsymbol{\theta})$. The optimal policy by training the agent maximizes the value function, which is determined by the critic,

parameterized by $\boldsymbol{\vartheta}$, $v(\mathbf{S}_t, \boldsymbol{\vartheta})$. The multi-agent actor-critic algorithm, using the CTDE concept (Figure 7), is based on three key components: a) The local agents for each tolled link determine the toll prices by the local actor and collect the rewards; b) After completing an episode, the local agent determines the accumulated gradients for the local actor ($\boldsymbol{\theta}$) and critic ($\boldsymbol{\vartheta}$); and c) A global actor-critic is updated asynchronously by the accumulated gradients of the local agents, whose parameters are pulled by the local agents after all updates by local agents are complete.

Results: Experiments are performed on a real network. The results are compared with: a) fixed tolling (fee proportional to average demand); b) Δ-tolling (Sharon et al., 2017) (Section 6.1.1) and c) DyETC (Chen et al., 2018) (Section 6.2.1). When evaluated against the objective of the reward functions R1, R2 and R3, DADO outperforms the other tolling mechanisms. When evaluated without taking account of deadlines, DADO slightly underperforms DyETC. When reward function R1 is used and, respectively the initial state on the links, initial demand, cost sensitivity and maximum toll price are varied, DADO and DyETC outperform fixed tolling and Δ-tolling and DADO is still optimal. Fixed tolling and Δ-tolling have problems with performance under changing traffic conditions.

6.2.4 Toll Pricing with Attention Network and Soft Actor-Critic

Another variation (Cooperative Tolling with Reinforcement Learning, CTRL) uses an attention mechanism and SAC (Section A.5.2) to determine toll prices for routes (Wang et al., 2022). Attention (Vaswani et al., 2017) allows modeling of dependencies without regard to their distance in input or output sequences, in this case the information contained in the route between upstream and downstream links. The quality score of a route (Q-value) is also derived using an attention network. This aims to produce comparable tolls for routes taking both state and action (of other routes) into account.

Traffic model set-up: Given $\langle G, V, \mathbb{D} \rangle$: (c_v) The price (toll) of a route is determined by the toll agent.$(\mathcal{R})$ Route choice: For every OD pair three routes are predefined and route choice by vehicles is performed at the start of their route based on lowest toll. Vehicles can not change their route after departure. $(\mathcal{M})$ Network loading is done with a microscopic simulation model. The objective of the toll agent is to minimize the Average System Travel Time for all vehicles that arrive at their destinations.

RL set-up: First the delays are calculated. Delays are defined as a scaled difference between actual travel time and free flow travel time on all links, denoted as $\mathbf{d}_t^l$. State S_t for route $p^{o\to d}$ is defined as the output of the attention mechanism denoted by Agg and based on the concatenated delays of the links comprising that route:

$$S_t^{p^{o\to d}} = Agg(\mathbf{d}_l^t),\ l \in p^{o\to d}. \tag{16}$$

The action $A_t^{p^{o\to d}}$ is the route price. Reward per link is defined as the cumulative distance covered by vehicles on a link during a time step. Reward per route is defined as the average reward of all links per route. SAC (Haarnoja et al., 2018) is the algorithm to train the agent determining the route prices. In this set-up the actor network calculates the action (toll price) for each path of an OD-pair. The Q values for each

state action pair are calculated by the critic network. Part of this critic network is an attention mechanism to ensure that Q-values are not only based on the state/action per route but also that other routes are taken into account. The parameters of the critic network are updated, after which the policy is updated.

Results: In experiments the set-up is tested on three real networks. Experiments are conducted to compare with other tolling mechanisms: a) No Change: no toll; b) Random: random route choice by vehicles; c) Formula: formula based toll price depending on vehicles on link; d) Δ-tolling: Δ-tolling (Sharon et al., 2017) (see: 6.1.1); e) Indi-SAC: Soft-Actor-Critic per link; f) Share-SAC: shared Soft-Actor-Critic; and g) MARL-eGCN: MARL-eGCN with one partition (Qiu et al., 2019) (see: 6.2.2 hereafter). The comparison is on throughput, Average System Travel Time and return. Its best result is a reduction of 44% in Average System Travel Time of vehicles that arrived at their destination. Overall CTRL outperforms the other toll mechanisms. In only one network Δ-tolling shows comparable results. An ablation study shows that the Q-attention mechanism provides better tolls (defined as differentiated for all three routes and fair) and State-attention increases throughput and decreases Average System Travel Time. Overall CTRL increases throughput and decreases Average System Travel Time.

6.2.5 Dynamic Toll Collection using Transformers and Graph Neural Networks

Further building on the ideas described in Sections 6.2.2 and 6.2.4 leads to a model (Lu et al., 2024) where a Multi Agent Transformer (Wen et al., 2022) architecture is combined with Random Walk Positional Encoding (RWPE) (Dwivedi et al., 2022) with a GCN for structural Position Encoding (PE) (Ye et al., 2023). The objective of the $ta's$ is to reduce the TSTT. For a high-level architecture, see Figure 8.

Traffic model set up: ($\mathcal{R}$) Before departure, vehicles select the lowest cost route, which they do not update after departure. A number of links in the network is tolled. In the graph for the GCN, the links are modeled as vertices, and the junctions as edges. ($\mathcal{M}$) For network loading a microscopic model is used (Zhang et al., 2019).

RL set up: State is defined as the number of vehicles that entered the link since the previous episode step combined with the current number of vehicles on the link. The action is a selection of one of five toll rates per link. Reward is given by the negative average queue length of vehicles on the link during the decision interval, started with $A_t(l)$ which consists of Γ time steps:

$$R_{t+1}(l) = -\frac{1}{\Gamma}\sum_{i=1}^{\Gamma} q_i(l). \tag{17}$$

The approach in this model is to first pre-calculate the PE for the vertices. When taking an episode step, the observations per vertex are concatenated with the PE and fed into the GCN. The output of the GCN is input for the Multi Agent Transformer, which expresses the sequential decision process for each agent (gives the tolls for each link) using the Multi-Agent Advantage Decomposition theorem (Wen et al., 2022).

Results: The model is tested on two city networks using a microscopic traffic model (Zhang et al., 2019). The experiment encompassed three traffic scenarios simulating low, medium and high vehicle demand. The MAGT solution was compared with a MAT solution (without PE/GCN), no toll, adaptive toll (based on Δ-tolling (Sharon et al., 2016)) and EGCN toll (Qiu et al., 2019). The MAGT toll solution outperforms all other mechanisms except in a low demand scenario, where EGCN and MAGT perform almost equally. Its best result was a reduction of 18% in Average System Travel Time. The performance is attributed to the combination of positional encoding and the Multi Agent Transformer.

6.2.6 Dynamic Toll Pricing in Regions

An agent is trained to set tolls for a region-based approach (He et al., 2024) using a similar type of day-to-day and intraday dynamics as in Section 6.5.1. The *ta* has the objective of minimizing the Total System Travel Time. The performance is also evaluated on speed profiles and total system cost.

Traffic model set up: Given $\langle G, V, \mathbb{D} \rangle$: G is divided in a number of regions. The traffic characteristics in the regions are based on a (Macroscopic) Fundamental Diagram (2.5). These characteristics are determined by a trained neural network. Vehicles travel from one region to another region. (c_v) Vehicles take an experienced cost on (day, time) $= (j-1, \underline{t})$ and the predicted cost $(j, \underline{t})$ based on actual travel times and tolls. Their perceived cost $(j, \underline{t})$, is a convex combination of experienced and predicted cost. ($\mathcal{R}$) The perceived cost is input to a softmax function to provide the route choice probability. ($\mathcal{M}$) Network loading is performed using a macroscopic model.

RL model set up: A2C (Section A.5) is used to determine the tolls. State is defined as:

$$S_t = [(n_i(t), \tau_i(t), t, i) | i \in \mathcal{I}], \tag{18}$$

where $n_i(t)$ indicates the number of vehicles in i at time t, and $\mathcal{I}$ is the number of regions. The action $A_t \in [\tau_{min}, \tau_{max}]$ is the adjusted toll price, modeled as a continuous variable. The reward is defined as the negative sum of all vehicles in the region in the toll adjustment interval.

Results: The *ta* is able to improve the speed profile and decrease both Total System Cost and Total System Travel Time with a reduction of 6% as its best result. The model also reaches an equilibrium state. When the system is tested to robustness against input variability, the agent reaches an improved equilibrium state comparable to the non-varied input. The model is further tested with different lengths of Γ. For Γ there is a level (15 minutes) that provides a good balance between accuracy and feasibility with values above (30, 60 minutes) and below (5 minutes) with lower performance.

6.3 Value Based Dynamic Toll Collection for Tolled Lanes

6.3.1 Distance Based Tolling with R-MART

In (Zhu and Ukkusuri, 2015) a traffic network is considered in which some links have both a non-tolled lane and a tolled lane. The objective is to dynamically determine the toll prices for the tolled lanes so that the Total System Travel Time on these links is

minimized. Tabular Q-learning is used to find the optimal policy. One agent operates on each link with a tolled lane, which optimizes the performance only for that link, without cooperation or communication with other agents (DTDE).

Traffic model set-up: Given $\langle G, V, \mathbb{D} \rangle$: (G). The links l are divided in $N(l)$ cells of equal length. The lanes of a link are further indexed by i and, if tolled, by i' like l_i and $l_{i'}$. (c_v) Tolls are calculated based on the remaining distance from cell m to the exit of the tolled lane and actual toll rate $\sigma_e, e \in 1, 2, 3, 4$. The remaining distance is determined as the number of cells until the exit of the lane $(N(l_{i'}) - m)$ multiplied by cell length C.

$$\tau(l_{i'}, m, \sigma_e, \underline{t}) = (N(l_{i'}) - m) * C * \sigma_e \tag{19}$$

The cost for a vehicle on the tolled lane, adding remaining travel time to the end of the lane, therefore is:

$$c_v(l_{i'}, m, \underline{t}) = tt(l_{i'}, m, \underline{t}) + (N(l_{i'}) - m) * C * \sigma_e. \tag{20}$$

And for a non-tolled lane:

$$c_v(l_i, m, \underline{t}) = tt(l_i, m, \underline{t}). \tag{21}$$

A vehicle determines the cost of using the non-tolled lane (based on remaining travel time to the end of the lane) and the tolled lane (based on the remaining travel time to the end of the lane and tolls). $(\mathcal{R})$ Vehicles choose to either stay on a non-tolled lane, or change to a tolled lane. Route choice is implemented using a binomial logit model with lane cost as input. Tolled lanes can be entered at any position on the non-tolled lane; vice versa is not allowed. $(\mathcal{M})$ Network loading is done using the CTM.

RL set-up: The environment is modeled as a MDP. The state S at lane l_i at time t is denoted as $S_t^{l_i}$ and the next state as $\bar{S}_t^{l_i}$, and similar for $l_{i'}$. The states in lane l_i are discretized and based on the comparison between the density k at time t denoted as $k(l_i, t)$ and the jam density k_{j,l_i} as below:

$$S_t^{l_i} = \begin{cases} 1, & \text{if } k(l_i,t) \leq 0.25 k_{j,l_i} \\ 2, & \text{else if } k(l_i,t) \leq 0.50 k_{j,l_i} \\ 3, & \text{else if } k(l_i,t) \leq 0.75 k_{j,l_i} \\ 4, & \text{else if } k(l_i,t) \leq \quad k_{j,l_i} \end{cases} \tag{22}$$

and similar for lane $l_{i'}$. The state on the link, comprising both the tolled and non-tolled lanes, is denoted S_t^l. The actions a in lane $l_{i'}$ depend on the cell m and time t denoted as $A_t^{l_{i'}}(m)$. Action selection is ϵ-greedy on the maximum Q-values. The toll rates corresponding to these actions are σ_b if $A_t^{l_{i'}}(m) = b$ $(b \in 1 \ldots 4)$ with σ_b the threshold values of the toll rate. Reward $R_t^{li'}(S_t^l, A_t^{l_{i'}}(m), \bar{S}_t^l)$ is calculated based on total travel time on a link summing the travel times tt at lanes l_i and $l_{i'}$ from cell m for all vehicles $x(l, m, t)$ to the exit of the lane at time t for all lanes and all cells:

$$R_t^{l_{i'}}(S_t^l, A_t^{l_{i'}}(m), \bar{S}_t^l) = - \sum_{j \in l_i, l_{i'}} \sum_m tt(j, m, t) * x(j, m, t). \tag{23}$$

A R-Markov Average Reward Technique (R-MART (Sutton and Barto, 2018)) is used which applies to continuing problems. In continuing problems, the interaction between agent and environment goes on and on forever. The quality of a policy is defined as the average rate of reward ρ. The Q-values for toll lane i for all cells m on a link l are updated as:

$$Q(S_t^l, A_t^{l_{i'}}(m)) \leftarrow Q(S_t^l, A_t^{l_{i'}}(m)) + \alpha \left[R_t^{l_{i'}}(\cdot) - \rho + \max_a Q(\bar{S}_t^l, a) - Q(S_t^l, A_t^{l_{i'}}(m)) \right]. \tag{24}$$

Results: In an experiment, the Total System Travel Time on the links with both a non-tolled and tolled lane decreases with 25% on a real network. Varying the number of states and actions demonstrates that increasing the number of states and/or actions does not lead per sé to an improvement in travel time; there is even a trend that the algorithm worsens when the number of states and actions are growing. In the work they highlight finding the best combination of states and actions as a problem for further study.

6.3.2 MARL for Distributed Dynamic Pricing of Managed Lanes

A distributed Multi Agent model is used to manage dynamic pricing for managed lanes with multiple entrances and exits (Pandey and Boyles, 2018). To this effect, the sum of each agents' local value is used as an approximation for the total state value. Furthermore, in determining the action (toll price) the agents collaborate by taking the actions of downstream agents into account. Therefore we categorize the training model as CTDE. The objective of the toll agents is to maximize revenue.

Traffic model set-up: Given $\langle G, V, \mathbb{D} \rangle$: (G) Managed lane networks are considered, consisting of one general-purpose lane and one tolled lane, with one or more entries and one or more exits. $(\mathbb{D})$ vehicles are categorized in different classes according to a set of different *vot*. $(\mathcal{R})$ vehicles choose according to minimal cost of the available routes. $(\mathcal{M})$ Network loading is done using CTM.

RL set-up: State $\mathbf{S}_t$ is defined as the vector that contains the number of vehicles of each *vot* class per cell. Action $\mathbf{A}_t$ is the action vector with the actions of all agents (toll prices). Reward $R_t(\mathbf{S}_t, \mathbf{A}_t)$ is the product of the number of vehicles choosing the managed lane times the (toll rate/mile) times the length of of the managed lane. A penalty is applied if the managed lane becomes congested. Because the assumption is that toll agents collaborate and need to coordinate their actions with only a few neighboring agents, the MDP is relaxed. This is done by approximating the value of the state as the sum of the individual value functions of the agents' state:

$$V_*(\mathbf{S}_t) = \sum_{n \in N} V_*^n(\mathbf{S}_t). \tag{25}$$

For each toll agent the Bellman equation (A8) at optimality is:

$$V_*^n(\mathbf{S}_t^n) = \max_{\mathbf{A}_t} R^n(\mathbf{S}_t, \mathbf{A}_t) + V_*^n(\mathbf{S}^n(t+1)) \, \forall \, n \in N, \tag{26}$$

where also the reward is decomposed into the rewards for each agent. To solve this relaxed MDP some modifications are applied to Q-learning. The first modification is that the actions of downstream agents are taken into account in action selection. The second modification is that the value function is used based on cooperative Q-learning (Kok and Vlassis, 2006) as opposed to the action-value (Q) function. The resulting algorithm is called SparseV.

Results: SparseV is compared with three other algorithms using heuristics: a) Density: if the density downstream in the managed lane differs from the required density, toll price is adjusted up or down; b) Ratio: the ratio, between the density in the cells downstream in the managed lane and the density in the cells in the general-purpose lane, determines adjustment of the toll price, and c) Random: a number of policies with random action is simulated, the best performing random policy is chosen. Experiments are performed on a synthetic and real network.

SparseVs outperforms the Density and Ratio heuristic which produce a revenue that is 70-75% lower. The Random policy outperforms SparseV by approximately 9 % in the first network and is outperformed by approximately 24 % in the other network. The better performance of the SparseV and Random policies is explained by the 'jam-and-harvest' nature of these policies; by setting initial tolls such that the General Purpose Lane becomes congested and subsequently charging higher tolls when demand increases. The work highlights convergence to sub-optimal values, attributed to exploration in relation to the aggregation level of the state space and the jam-and-harvest nature, unwanted in practice, as weaknesses.

6.4 Actor-Critic Dynamic Toll Collection for Tolled Lanes

6.4.1 Deep RL for Dynamic Pricing of Express lanes

Deep RL methods are tested in a wide range of use cases (access locations, OD-pairs, VOT heterogeneity, partial observability) where links have both a toll lane and a General Purpose Lane (GPL) (Pandey et al., 2020). A2C, PPO and SAC (Section A) are compared against a feedback control heuristic.The agents are trained for two different objectives (not in the same experiment): Maximum Revenue and minimum Total System Travel Time.

Traffic model set-up: Given $\langle G, V, \mathbb{D} \rangle$: (G). All links are denoted as $l \in L$, links which are tolled are denoted as $l' \in L$. (c_v) Toll lanes have multiple entrances and exits, where tolls are charged. $(\mathbb{D})$. Vehicles are divided in different classes based on their *vot* and on their destination. $(\mathcal{R})$ Lane choice: Two models for lane choice are used, a multi-class binary logit model (two routes, stochastic choice, multiple *vot*) if there are only two routes, and a multi-class decision route model (multiple routes, deterministic choice, multiple *vot*) if more routes are available. $(\mathcal{M})$ Network loading is done using CTM.

RL set-up: As the state is not fully observable the problem is defined as a POMDP for which the policy $\pi(\mathbf{a}|\mathbf{o}(s))$ is learned. The work uses a finite horizon H and incorporates time (the toll update step number) in state observations. This is because of the temporal dependence of the congestion pattern. The observation vector $\mathbf{o}(s)$, in which they also model noise, for state s comprises the number of vehicles on each link,

using $\mathbf{1}_l(v) = 1$ if v is on link l and 0 if not:

$$\mathbf{o}(s) = \left\{ \sum_{i=1}^{I} \mathbf{1}_l(d_i) | l \in L \right\}. \tag{27}$$

The action $\mathbf{a}$ in state s is the vector of tolls $\tau(l')$ which are being applied on the tolled links l'. Tolls are modeled as a continuous variable. Two types of reward are analysed: rewards based on revenue maximization $R_{MR}(s, \mathbf{a})$ which is the sum of the product of toll rate and vehicles entering a toll lane during a toll update time period. In the other type rewards are based on travel time minimization $R_{TSTT}(s, \mathbf{a})$ which counts the number of vehicles every toll update time step. The algorithms to find the policy are A2C, PPO and SAC.

Results: A series of experiments is conducted on synthetic and real networks, for various objectives, variations of the observation space and variations on demand, route choice and *vot*. All three algorithms with a single objective converge for all networks. SAC outperforms PPO and A2C, which is attributed to the entropy regularization used by SAC. In an experiment on the real network it appears that reducing the observation space of the network (reducing 15 links to 4 links to 1 link under observation) does not reduce the convergence speed. As this was an unexpected result it was speculated that this happens due to spatial correlation of congestion patterns on a corridor. When conducting experiments on the same network, with changed demand and *vot* to test generalization and transferability, the key observations are that a) the algorithms learn from the new input situation at equal speed as the original situation, and b) that the transferred policies from earlier experiments (PPO, A2C), perform within a bandwidth of 5 - 12% of the newly trained policies in their experiments. This is not the case if the lane choice model is changed; in that case the transferred policy performs badly, attributed to the fact that the lane choice model has a large impact on the evolution of congestion. Comparison with an industry-based heuristic shows that for all networks for both objectives, DRL outperforms this heuristic.

6.4.2 A Priori Link Selection and Dynamic Tolling of Expressways

In case of a high-speed road network, where each link is tolled, the distance difference between routes can be very large, and route adjustment is relatively difficult. Building on the work in Section 6.4.1 another model is designed (Zhang et al., 2023) in which vehicles weigh distance, travel time and cost and apply multi-objective optimization. Each *ta* is responsible for one toll lane.

In experiments, the model demonstrates its effectiveness. However they note that it does not take account of temporal correlation of traffic flow between different road sections in the network.

6.5 Dynamic Tolling to Manage Departure Time Choice

6.5.1 Dynamic Congestion Pricing for Departure Time Choice

Another model is proposed for dynamic toll pricing for departure time choice (Sato et al., 2021). The objective is to minimize the total delay by influencing the departure time of the vehicles. Q-learning is applied to learn the optimal policy, using a Q-value function approximator.

Traffic model set-up: Given $\langle G, V, \mathbb{D} \rangle$: (G). All links in the network have one bottleneck. (c_v) The cost of a link with a bottleneck is based on the day j, the time $\underline{t}$, the delay w, the required arrival time $\underline{t}^*$, a vehicle's vot, toll and a penalty for arriving early or late at the destination.

$$c_v(l, j, \underline{t}) = \tau(l, j, \underline{t}) + vot * w(l, j, \underline{t}) + \begin{cases} \beta(\underline{t}^* - \underline{t})(\underline{t} < \underline{t}^*), \\ \gamma(\underline{t} - \underline{t}^*)(\text{otherwise}). \end{cases} \tag{28}$$

In this equation, β is the early arrival penalty factor and γ the late arrival penalty factor (see Section 3). Tolls $\tau(l, j, t)$ consist of a stabilization toll $\tau^s(l, j, t)$, a parameterized function based on departure rate (Seo, 2020), and a congestion toll $\tau^c(l, j, t)$ based on the action of the policy. $(\mathcal{R})$ The model studies the morning commute. The day-to-day dynamics for departure flow in the morning commute are modeled using replicator dynamics (Schuster and Sigmund, 1983). This mechanism gives for every OD-pair the time $\underline{t}$ and day j dependent departure rate $\varsigma(p, j, \underline{t})$ for all paths $p \in p^{o \to d}$. All paths $p^{o \to d}$ going through link l are denoted by $P_l^{o \to d}$. The departure rates per route per OD pair at time $\underline{t}$ are given by a logit model. Both have travel cost per route per departure time as input. The departure time of the vehicle is therefore not fixed. $(\mathcal{M})$ Network loading is performed with an extension of the bottleneck model. The bottleneck model can be seen as a special case of the LWR model (Section 2). In a bottleneck model (Arnott et al., 1990; Seo, 2020; Vickrey, 1969) the bottleneck has a fixed capacity (service rate) and if the arrival rate exceeds this capacity, a queue begins to form. In the model used here, the queue occupies no space.

A vehicle arrives at the bottleneck of a link after the free flow travel time from the begin of the link to the bottleneck of the link, and departs the link immediately after leaving the bottleneck. The time spent at the bottleneck is the waiting time w (also: delay).

RL set-up: Q-learning is used to determine the congestion tolls. The effect of tolls on departure time should lead to a reduction in waiting time at the bottleneck. A spatially shared reward structure is to take account of tolls on other links. State $\mathbf{S}$ is a vector of the departure rate from an origin in an OD-pair, waiting time at the bottleneck and the stabilisation toll (T for transposed):

$$\mathbf{S}_t^l = \left(\sum_{p' \in P_l^{o \to d}} \varsigma(p', t), w(l, t), \tau^s(l, t) \right)^T. \tag{29}$$

Action A_t^l is the update to the congestion toll, therefore:

$$\tau^c(l, t+1) = \tau^c(l, t) + A_t^l. \tag{30}$$

Reward R is the weighted sum (by a factor K) of the inverse difference between the bottleneck capacity and traffic flow at the bottleneck in link l lower bounded by a constant E (denoted ξ_t^l) and the bottleneck capacity and traffic flow for all other links l' with bottlenecks collectively also lower bounded by a constant E (denoted by $\xi_t^{l'}$):

$$R_t^l = (\xi_t^l)^{-1} + K(\xi_t^{l'})^{-1}. \tag{31}$$

More reward is given if departure rates are close to bottleneck capacity and it takes account of this for the other bottlenecks. This mechanism implements the spatially shared reward structure. The policy finds the adjustment tolls (the actions), by choosing actions ϵ-greedy, leading to minimal total waiting time. The parameter updates are learned by Q-value function approximation for which Radial Basis Functions are used (Sutton and Barto, 2018).

Results: Experiments are performed on two synthetic networks: one where they apply the method to a single link single bottleneck model, the other with a single OD-pair and three links with three bottlenecks. After training, the agent succeeds in reducing the waiting time to almost zero and outperforms a trial and error congestion pricing scheme (Seo, 2020). Performance in the three bottleneck model was inefficient compared to the single bottleneck model. This is attributed to the reward function in the sense that the spatially shared reward function did not provide sufficient coordination between the bottlenecks.

6.5.2 Pricing for Departure Time Choice with DDPG

Extending Sato et al. (2021) outlined in the departure choice model above (Section 6.5.1) with DRL, and temporally switching learning leads to Distributed Pricing-Deep Deterministic Policy Gradient (DP-DDPG) (Sato et al., 2022). DDPG (Lillicrap et al., 2015), (Section A), is used as RL mechanism. A modification is that an action is fixed at zero if the moving average of the waiting time is less then a threshold (temporally switching learning). This leads to more efficiency and less excessive increases in tolls. Distributed control, using spatially shared rewards (Section 6.5.1), is implemented to enable cooperation.

Traffic model set-up: Given $\langle G, V, \mathbb{D} \rangle$: (G) and (c_v) are as per (Sato et al., 2021) detailed above (Section 6.5.1). $(\mathcal{R})$ Different from (Section 6.5.1), the departure time and route choice are based on a multinomial logit model. Furthermore, vehicles have bounded rationality, meaning that departure time and route will not be changed if the difference between expected cost of current choices compared with the expected costs of alternatives remain below a certain threshold. $(\mathcal{M})$ For network loading the bottleneck model is used. The objective of the toll agents is to reduce delay on the link under its responsibility.

RL set-up: State is defined in a similar way as above (Section 6.5.1), adapted for input to a neural network. Action is an adjustment to the toll price for a bottleneck.

Action A_t^l is the update to the congestion toll:

$$\tau(l, j, t) = \tau(l, j, t) + A_t^l. \tag{32}$$

The spatially shared reward R is a weighted average of the waiting time at bottleneck i of agent ta^i and the bottlenecks of the other agents ta^{-i}, slightly modified from (Sato et al., 2021) above. With this reward, each agent ta^i takes into account the rewards of the other agents ta^{-i} and therefore their objectives are aligned. Each agent ta^i uses its own instance of the DDPG algorithm to observe state, leading to an action and obtaining a reward and observation of the next state. This data $(\mathbf{S}_t^l, A_t^l, R_t^l, \mathbf{S}_{t+1}^l)$ for each agent is stored in an experience replay memory, from which data is sampled to update the actor and critic networks of the agents. If the temporally switching learning criterion has been met, no action is taken (so no toll updates) and no data is added to the experience replay buffer, so this data is not used for learning.

Results: Two experiments are performed, one on a network with three links for one OD-pair with a bottleneck on every link. Another experiment involved a real network to evaluate the performance where multiple OD pairs and multiple routes exist. In the first experiment, the deadline for all vehicles is the same and DP-DDPG is compared with centralized DDPG, fully distributed DDPG and Q-Learning (Sato et al., 2021). DP-DDPG outperforms centralized DDPG and fully distributed DDPG as these could not decrease waiting time. Q-Learning however succeeded in decreasing waiting time. In a simplified real network, toll is charged on four links with a bottleneck and there are three links with a bottleneck without charging tolls. The same arrival time is set for all vehicles. The results demonstrate that the waiting time reduces, meaning that with a trained agent, the vehicles also adapt their behavior leading to lower waiting times.

6.6 Value Based Dynamic Tolling Including Vehicles as Learning Agents

6.6.1 Multi Agent Road Pricing With Learning vehicles And Toll Agents

A traffic network is modeled as a Multi Agent system with two types of agents (Tavares and Bazzan, 2014). For the demand side, vehicles that are traversing the network, with the objective to optimize route choice. For the supply side, toll agents that are setting tolls on links in the network, with the objective to maximize flow. For both types of agents DTDE applies.

Traffic model set-up: Given $\langle G, V, \mathbb{D} \rangle$: (c_v). Vehicles have knowledge of the network based on experience, which means they only know the toll τ and travel time tt for a link if they travelled it in the past and recorded it as $\langle t, v, l, tt, \tau \rangle$ If vehicles have never used a link, the travel time is based on the free flow travel time $T(l)$ and for the toll, the maximum toll $\tau'(l)$ divided by 2: $\tau = \tau'(l)/2$. The cost for a vehicle is the sum of link costs. Link costs $c_v(l, tt, \tau)$ are vehicle specific, and based on a weighting factor $\eta(v) \in [0, 1]$:

$$c_v(l, tt, \tau) = \eta(v) * tt(l, v) + (1 - \eta(v)) * \tau(l, v), \tag{33}$$

where travel times and tolls are based on prior recorded experiences or alternatively the default values. ($\mathcal{R}$) Prior to each episode, vehicles calculate their optimal (lowest cost) route based on calculated link costs and do not update their route while traversing the network. ($\mathcal{M}$) Network loading is done using a microscopic traffic model.

RL set-up: The toll agents are trained as independent learners to limit the problem size (DTDE). The toll agents' actions consist of setting a toll price $\tau(l)$ as a fraction of the maximum toll price $\tau'(l)$ for a link. The model is stateless and only a reward R_l is observed after an action; the reward R_l is the number of vehicles ν_l which have entered a link during the episode. The action values $Q(\tau(l))$ are stored in a Q-table and are learned according to:

$$Q_l(\tau(l)) \leftarrow (1-\alpha)Q_l(\tau(l)) + \alpha R_l. \tag{34}$$

where α is the learning rate. The objective is to find the ϵ-greedy policy which maximizes the flow, where ϵ is decreasing over time. Vehicles initially only know the values of free flow travel time for all links and maximum price for all links. Vehicles can update these values by acquiring local knowledge, that is when they travel a route learning the actual travel times of the links and tolls they pay.

Results: Under different vehicle preference scenarios, the performance of the *ta* is compared with a fixed toll scenario and a Dynamic User Equilibrium scenario which assumes global knowledge for the vehicles. The DUE solution outperforms the MARL solution, which again outperforms the fixed toll scenario. While the average performance of the $ta's$ increases, large differences exist between best and worst performers. In summary, the case where both vehicles and toll agents learn outperforms the fixed-toll solution in these experiments. Global performance increased (more vehicles completed trips in the network) and vehicle costs decreased. These outcomes were achieved without an explicit coordination mechanism between the Toll Agents. With global information DUE led to better results.

6.6.2 Deep Multi Agent Road Pricing with Learning Vehicles and Toll Agent

A variation on 6.6.1 uses DQN (Mnih et al., 2015) both for a *ta* and for vehicles making their lane choice (Chakravarty et al., 2024). The objective of the *ta* is to maximize revenue, generated by the tolled links (l') defined as $rev(t) = k_{l'}(t) \cdot v_{l'}(t) \cdot \tau_{l'}(t)$, the product of density, speed and toll price. The objective of the vehicles is to be in the fast lane, but taking into account the toll price.

Traffic model set-up: Given $\langle G, V, \mathbb{D} \rangle$: In the network are two parallel links, one tolled (l'), one non-tolled (l) where the difference in speed is noted as $\Delta v = v_{l'} - v_l$. Shallow neural networks are pre-trained using real-world data to relate observed density to speed ($f_1(k) = u$) and toll price and speed to provide the fraction ξ choosing the tolled lane ($f_2(\tau, k) = \xi$) and the non-tolled lane ($1 - \xi$). (c_v) The vehicle cost function is not made explicit. ($\mathcal{R}$) Route choice is made by an agent using DQN. ($\mathcal{M}$) We categorize this as a macroscopic model.

RL Set up: The state for vehicles is defined as:

$$S_{\underline{t}} = [\tau, \Delta v, \int \Delta v dt], \tag{35}$$

and the reward function, with $F = 0, 1$ for being in the slower or faster lane as:

$$R_{\underline{t}} = -a_1 \tau^2 + a_2 F. \tag{36}$$

Vehicle characteristics are modeled by a_1, a_2 to indicate how much they value speed over toll price. The action a_t consists of choosing the tolled or non-tolled lane.

For the toll agent state is defined as:

$$S_t = [v_T, \Delta v, \int \Delta v dt, k_{in}, rev(t)], \tag{37}$$

with k_{in} total incoming density. The action space is a set of toll prices. Finally, the *ta* reward combines revenue and Δv:

$$R_t = b_1 * rev(t)^2 - 2 * B, \tag{38}$$

where $B = 0$ if $\Delta v \geq 0$ and $B = 1$ else.

Results: The driver agent succeeded in improving its reward. Training the agent ceased when the average reward was above a certain threshold for five episodes. Another outcome was that vehicles were never willing to pay more than a certain toll price, although the speed in the tolled lane was higher. The *ta* succeeded in increasing its reward as well. Training for this agent was stopped if the average reward was above a certain threshold.

The authors note that both the *ta* and the vehicle agent were able to improve their performance. However they indicate the vehicle agent could be improved by a more complex reward function and that their set-up of the environment showed some weaknesses in calculating the density fraction.

6.7 Value Based Marginal Cost Tolling

6.7.1 Multi Agent Reinforcement Learning - Only Vehicles

In Ramos et al. (2018, 2020) vehicles are represented by a Q-learning agent whose objectives are to find the routes with the optimal Q-value. The tolling scheme, based on Marginal Cost Tolling (MCT), is distributed. The objective of the tolling scheme is to bring the User Equilibrium closer to the System Optimum (Section 2).

Traffic model set-up: Given $\langle G, V, \mathbb{D} \rangle$: (c_v). The toll on a link for a vehicle is the difference between the free flow travel time $T(l)$ and the actual driving time on the link $tt(l)$, multiplied by a constant β which is specific for the Link Performance Function (Section 2).

$$\tau(v, l, tt(l, \underline{t}), \underline{t}) = \beta(tt(l, \underline{t}) - T(l). \tag{39}$$

The cost for a link is the sum of the toll and actual driving time:

$$c_v(l, tt(l, \underline{t}), \underline{t}) = tt(l, \underline{t}) + \tau(v, l, tt(l, \underline{t}), \underline{t}). \tag{40}$$

The self-calculated cost by the vehicles for a path $p^{o\to d}$ is the sum of the cost of the links $l \in p^{o\to d}$ and is denoted as $c_v(p^{o\to d})$. ($\mathcal{R}$) Vehicles know a subset of their routes a priori, and the only action is to select the route every day, based on the Q-value of the routes in the subset. ($\mathcal{M}$) Network loading is done by a microscopic model.

RL set-up: This problem is modeled as a stateless MDP, as the vehicle takes an action A, based on the highest Q-value of that action and then obtains a reward $R(A)$ based on that action. The reward $R(A)$ is defined as:

$$R(A) = -c_v(p^{o\to d}). \tag{41}$$

The vehicles want to take a route $p^{o\to d}$ which minimizes their costs. Therefore, this entails finding the policy π, which provides the route to take and maximises the agent's average reward. The algorithm used is Q-learning (Watkins and Dayan, 1992) with independent Q-Learners (Claus and Boutilier, 1998) thereby implementing a DTDE framework. The Q-learning algorithm is modified to a stateless version:

$$Q(A) = (1 - \alpha)Q(A) + \alpha R(A). \tag{42}$$

The learning rate α is exponentially decaying per episode. Every episode an agent takes an action (chooses a route), using an ϵ-greedy exploration strategy with exponentially decreasing exploration rate per episode. Then it updates its Q-value.

Results: The algorithm is tested on 13 different (synthetic and real) networks and compared against three other algorithms namely Δ-tolling (Sharon et al., 2017) (see 6.1.1), toll-free Q-Learning where they apply the same model without tolls and one other mechanism. Their method provides similar results as Δ-tolling and better performance than toll-free Q-learning to approach SO. In comparison to Δ-tolling this method has a learning scheme, a fairer tolling scheme and convergence guarantees. The work however highlights that speed of convergence is a major area for improvement. A similar work is (Ramos et al., 2020) in which it is also highlighted that speed of convergence is a major area for improvement.

6.7.2 Multi Agent Reinforcement Learning - Heterogeneous Agents

An algorithm capable of realigning agents' heterogeneous preferences over travel time and monetary expenses to obtain a system efficient equilibrium (SO) is Generalised Toll-based Q-Learning (Ramos et al., 2020). It also includes a mechanism to enforce agents to truthfully report their preferences. The key difference with (Ramos et al., 2018, 2020) (see 6.7.1) is the calculation of the cost function. The objective of the tolling scheme is to bring the User Equilibrium closer to the System Optimum.

Traffic model set-up: Given $\langle G, V, \mathbb{D}\rangle$: ($c_v$) In this model with heterogeneous preferences for a vehicle v, travel time and toll are weighted by $(1 - \eta(v))$ and $\eta(v)$

respectively. Furthermore the cost includes side payments indicated by $\rho_{\psi(v)}$:

$$c_v(p^{o\to d}) = -\rho_{\psi(v)} + \sum_{l\in p^{o\to d}} ((1-\eta(v)) * tt(l,\underline{t}) + \eta(v) * \tau(v,l,tt(l,\underline{t}),\underline{t}). \quad (43)$$

The side payments are a redistribution of a part of the collected tolls to the vehicles as not all collected tolls are required to operate the toll system. $(\mathcal{R},\mathcal{M})$ as in Section 6.7.1

RL set-up: The reward R is as Equation (41) and Q-learning algorithm as Equation (42). The difference is that the reward also includes the side payments, described above. To enforce truthful reporting (to prevent misuse of the side payments) the number of times an agent has not chosen the least-cost action based on their preferences is logged. If the number of inconsistent actions, which means reporting action a and executing action a', exceeds a threshold the agent is expected to be cheating which leads to a penalty.

Results: The algorithm is tested on 15 different (synthetic and real) networks. The results are compared with the algorithm in (Ramos et al., 2018, 2020) (see 6.7.1) and with Δ-tolling from (Sharon et al., 2017) (see Section 6.1.1). In general, GTQ-learning was able to converge to a SO efficient equilibrium, leading to a reduction of 30% in Average System Travel Time. Furthermore, when vehicle preferences were applied, GTQ-learning outperformed in almost all cases the other two algorithms. It was also experimentally demonstrated that side payments did not deteriorate the equilibrium. However the other algorithms also achieved reasonable results. The mechanism to prevent misreporting, neutralized agent misreporting, restoring system optimality. Although the mechanisms (Ramos et al., 2018, 2020; Sharon et al., 2017) produced similar results, results in this study were obtained under more realistic assumptions. The work compares to (Ramos et al., 2018, 2020) which does not take heterogeneous preferences into account and to (Sharon et al., 2017) which does not work in a decentralized way. All three do ignore misreporting and do not provide for tax returns.

6.8 Other Works

One work (Nisha et al., 2024) describes an approach to deploy RL for road pricing using a comprehensive state, action and reward set. In their experiments, they compare their solution with three traditional models (Decision Tree, SVM and logistic regression) which are outperformed. The main part of the work describes the design process and the components of a complete solution. Although it highlights the potential of RL for road pricing, we exclude it from further analysis as it does not provide details on the (RL-) design itself.

Another work (Chiou, 2024) combines a stochastic mathematical program with equilibrium constraints with a Q-learning performance index (KAQPI). This value function takes a combination of the rate of delay, the number of stops in the network (caused by traffic signals) and the sum of link tolls which it strives to minimize. The main focus in this work is stochastic optimization and modeling, which we exclude in this survey. We refer to the work for details.

6.9 An assessment of the potential of RL for road pricing

From the works analyzed in this Section, it is a challenge to specifically identify what contributes the most to the demonstrated performance improvements. As can be inferred from Table 3, there is a wide variety in toll agent objectives, RL design choices, and vehicle characteristics. In addition, most of the works use different simulators (see Table B1) and perform their experiments on different road networks (Table B2), which also may affect the results. This further complicates drawing general conclusions whether the performance is attributable to the agent, the specific traffic network settings, or the choice of suitable (hyper) parameters.

Nevertheless, across all works in this Section, seven provide a percentage for the reduction of Average/Total System Travel time (see Table 5). We observe that the bandwidth of reduction ranges from 6% (Section 6.2.6) to 44%, 45% (Sections 6.1.1 and 6.2.4). The other works had results between these maxima and minimum. While these results suggest that using RL for road pricing can lead to meaningful reductions of Average/Total System Travel time we caution against interpreting them as directly comparable. The works differ significantly in terms of traffic models, simulation environments, road networks, demand patterns, agent designs and RL algorithms. These variations make it difficult, maybe even impossible, to attribute these performance gains to specific factors or draw generalizable conclusions. However, the diversity across the works demonstrates that various combinations of environment, agent design and objectives can achieve beneficial outcomes. We consider even a 6% improvement in Average/Total System Travel time to be a significant achievement, underscoring RL as a relevant contribution to solving the road pricing problem.

Table 4: Scaling parameter softmax

Section	θ
6.2.1	0.5
6.2.3	0.5
6.2.6	0.03
6.3.1	1.0
6.5.1	0.015

Table 5: Reduction in Travel Time

Section	Reduction
6.1.1	45%
6.2.1	15%
6.2.4	44%
6.2.5	18%
6.2.6	6%
6.3.1	25%
6.7.2	30%

7 Challenges and Future Research Directions

The works covered in Section 6 of this survey demonstrate the usefulness of Reinforcement Learning for various road pricing scenarios, ranging from tolled lanes to larger road real world networks and corresponding demand data.

7.1 How Identified Challenges are Addressed

Either explicit or implicit, the challenges related to partial observability, credit assignment and non-stationarity have been addressed in most of the surveyed works.

However, given the limited number of works, we cannot conclude which factors contribute the most to driving performance and effectively addressing these challenges. On the other hand, a CTDE architecture in multi agent cases and the use of preprocessing networks provide positive indicators of their usefulness. Scalability and generalizability have received little attention in the surveyed works in Section 6.

Partial observability In some of the approaches (Section 6.4.1), partial observability was explicitly noted, in which case the observation was provided to the learning algorithm. It also modeled noise effects of sensors. In the other single agent approaches, the choice (and implicit simplification) to frame the problem as an MDP enabled the use of simpler algorithms and demonstrate their feasibility. In the solution approaches involving multiple agents, the CTDE architecture (Sections 6.2.2, 6.2.3, 6.2.4, 6.2.5, 6.3.2 and 6.5.2) and the use of communication between agents (Section 6.2.2), address this challenge. In the single case where DTDE is compared with CTDE (Section 6.5.2) the CTDE architecture outperforms DTDE. We observe that in the other DTDE cases, the toll agents were not using state (Sections 6.6.2, 6.7.1, 6.7.2) or no comparison with a CTDE architecture was performed (Section 6.3.1).

Credit Assignment The approaches in Sections 6.2.2, 6.2.4, 6.2.5 address the spatio-temporal dependencies by using a GCN (Kipf and Welling, 2017) and transformer (Vaswani et al., 2017). The multi agent credit assignment challenge has various kinds of solution mechanisms surveyed in the works (Du and Ding, 2021; Gronauer and Diepold, 2022; Wong et al., 2022). In Section 6.3.2 one of these mechanisms (value decomposition) is used, in other works this challenge is not explicitly addressed. We can not conclude that Credit Assignment, for RL applied to road pricing, is solved. Credit Assignment in this domain is complicated because routing decisions, as observed from the works, also depend on effects from other traffic. A clear approach to link the toll price, taken into account by individual vehicles, and its effect on future traffic on that road has not emerged from the works.

Non-stationarity A CTDE architecture and communication (Sections 6.2.2, 6.2.3, 6.2.4, 6.2.5, 6.3.2 and 6.5.2) partially address this challenge. However, the challenges when more complex vehicle behavior is modeled (Section 6.6.1), and when assumptions about vehicle behavior may not hold in the real world (Section 6.1.1) remain. If in addition aspects like mode choice behavior (other means of traffic) and modeling (temporary) disruptions of roads in the network are considered, the non-stationarity challenge is further exacerbated.

Scalability In most of the works, learning is first demonstrated on a synthetic network followed by experiments on real world networks. As noted (Sections 6.1.1, 6.2.2 and 6.2.3) a larger state and action space is used than for example in classic RL problems like cartpole or mountain car (Sutton and Barto, 2018) but overall the state and action space remain small. The impact of state and action space on performance (Section 6.3.1) and of the traffic model used (Section 6.4.1) was mentioned as related challenges in the analyzed works. The works (Sections 6.2.1, 6.5.2) specifically highlighted scalability as an issue.

Generalizability This aspect was partially highlighted (Section 6.4.1) to consider how algorithms could transfer across different traffic models. In the works, different topologies or origin-destination pairs were not considered. As generalizability would

increase acceptance in real-world implementations, this challenge has in our view only minimally been addressed and therefore also remains. We do note that existing, known approaches for the above challenges, surveyed by, among others, (Du and Ding, 2021; Gronauer and Diepold, 2022; Wong et al., 2022) are only to a limited extent utilized.

With respect to the design of state, action, and reward, in each of the works, tailor-made state/observations were created for each environment. The actions varied from discrete to continuous toll prices, for roads and routes. Reward functions also showed a great variety.

7.1.1 Additional Findings

In the simulated environments, covering both synthetic and real-world networks, agent learning has been demonstrated. Nevertheless, specific learning issues noted in the works remain. These were finding the balance between exploration and exploitation (Sections 6.3.1, 6.3.2 and 6.6.1), sub-optimal performance because of the aggregation level of the state space (Section 6.3.2) and convergence challenges (Section 6.1.1) related to changing traffic patterns.

Furthermore, experiments demonstrated that in a multi-agent setting an optimal number of agents exist (Section 6.2.2) and that more fine-grained observation and action spaces not necessarily lead to better performance (Section 6.3.1). However, a detailed analysis of these findings was left as future work by the authors. Another observation concerns the choice of the scaling parameter θ used in stochastic route choice models (see Equation 4 and Table 4). The value of θ varies considerably across the surveyed works, affecting the resulting traffic distributions. A detailed analysis of the selection or calibration of θ, and its impact on agent performance, was generally not provided. In contrast to learning in simulated environments, none of the works covered off-line learning based on real data, or learning in the real world. Unfortunately, despite being present and of major importance, existing literature on RL for road pricing has not elaborated enough on deployment barriers. There are technical challenges, like legacy infrastructures, data acquisition, integration, quality and exploration. Financial obstacles are definitely present, like deployment, operational and maintenance cost. Furthermore, there are societal issues, like acceptance and use by the general population, or diversion of traffic through narrow roads or quiet neighborhoods. Last, but not least, there may be physical bottlenecks, like the ability to install high tech equipment or communication coverage issues. To the best of our knowledge, no cities or transport agencies have yet implemented full-fledged reinforcement learning-based road pricing schemes.

Although all solution approaches demonstrated convergence in learning, the variability in selection of among other the environment (including the use of different traffic simulators and different networks), multi-agent architectures, agent algorithms and evaluation mechanisms, makes comparison of solutions hard. The inherent variability in RL experiments, where evaluation presents significant challenges (Patterson et al., 2024), is aggravated by differences in setups in the works described above. This points to the need for a common benchmark.

7.2 Future Research Directions

In addition to the challenges which remain, as noted above, we give two future research directions based on what we think has not, or only to a limited extent, been covered.

First, in general, we observe that an accepted benchmark, to compare the solutions against, is lacking. In the works we observe that the features of state and observation differ, as well as the reward signals and actions. Furthermore, pre-processing of data, the (Reinforcement Learning) algorithms, and also the means of network loading vary per surveyed work. This makes it hard to specifically identify what causes the changes in behavior and performance of the models.

A significant next step for road pricing using Reinforcement Learning lies in establishing standardized benchmarks and fostering shared datasets. These are essential to enable fair comparison of diverse solutions and accelerate progress in the field. We envision two primary objectives for such benchmarks: First, to rigorously test and compare the performance of various toll agents and their algorithms, a dedicated simulation environment would be highly beneficial. Such an environment, ideally constructed using frameworks like Gymnasium (Towers et al. (2024)), PettingZoo (Terry et al. (2021)), RL4CO (Berto et al. (2023)) or BenchMARL (Bettini et al. (2024)), would need to encompass diverse network topologies, realistic demand patterns, and standardized settings for route- and departure time choice, alongside consistent network loading models. Second, for demonstrating scalability and the efficacy of agents in more realistic, complex traffic settings, future efforts should focus on meticulously detailing the specific simulators used, their full range of traffic settings, and comprehensive interface specifications between the agents and simulators. Crucially, this must be coupled with initiatives to facilitate the sharing of generated datasets (like demand and traffic patterns), which is paramount for enhancing reproducibility and comparability across studies.

By pursuing these avenues, the community can collectively move towards more robust, reproducible, and impactful research in RL-based road pricing. To expand further on this, performing real world experiments is also needed to test the validity of outcomes from simulations.

Next, road pricing is only one of a range of ITS mechanisms. As exemplified by other surveys (Section 1.1) mechanisms like traffic signal controls, variable speed limits and ramp metering are modeled using an MDP formulation as well and involve agents which take actions. Understanding the interaction between agents from other ITS and road pricing agents is essential. First by observing and analysing emergent behaviors. Second by creating and studying agents which can cope with these situations. Obviously the same could be said about the vehicles; in most of the works these are modeled as probabilistic actors who are not learning.

Moving beyond the strict scope of the works covered in this survey, future research could explore connections with adjacent fields such as economics, sociology, and issues of equity and fairness. There is also significant potential in multidisciplinary approaches, where theoretical solutions are tested and evaluated under real-world conditions. This would include technical integration with existing traffic management systems.

8 Conclusions

Road pricing is an effective and convenient method to address traffic congestion. It poses various challenges given its traffic dynamics, the complex behavioral patterns of vehicles and the need to periodically adjust the tolls in the traffic network by one or more toll agents. Reinforcement Learning is a promising solution to tackle these challenges as it lends itself well to model the interaction between traffic network and toll agents. It provides algorithms, both for single and multi agent cases, to optimize the performance of the traffic network to the benefit of vehicles and society. In this work we presented a traffic model, relevant elements of Reinforcement Learning and the different ways Reinforcement Learning can be used for effective toll pricing. While the current solutions in the works make the potential of RL abundantly clear, research challenges remain before solutions are ready for implementation.

Based on our analysis, partial observability, non-stationarity and credit assignment were partially addressed in the surveyed works. While the existing literature in Reinforcement Learning on addressing partial observability, non-stationarity, and credit assignment challenges provides valuable insights, opportunities remain to more effectively leverage these in the context of road pricing. Scalability and generalizability, critical challenges in Reinforcement Learning, and real-world experiments received limited attention in the surveyed works. We also emphasized the need for a benchmark to enable comparison between solution approaches. Finally, specific challenges related to interaction between heterogeneous agents would benefit from more research in this domain.

Declarations

Funding

No funding was received for conducting this study.

Competing interests

The authors have no competing interests to declare that are relevant to the content of this article.

Acknowledgements

We would like to thank the editors and four anonymous reviewers for providing their constructive feedback. Without their suggestions this manuscript would not have looked as in this final version.

References

Albrecht, S.V., Christianos, F., Schäfer, L.: Multi-Agent Reinforcement Learning: Foundations and Modern Approaches. MIT Press, Cambridge (2024). https://www.marl-book.com

Arnott, R., De Palma, A., Lindsey, R.: Economics of a bottleneck. Journal of Urban Economics **27**(1), 111–130 (1990)

Arnott, R., De Palma, A., Lindsey, R.: A structural model of peak-period congestion: A traffic bottleneck with elastic demand. The American Economic Review, 161–179 (1993)

Ahmed, H.U., Huang, Y., Lu, P.: A review of car-following models and modeling tools for human and autonomous-ready driving behaviors in micro-simulation. Smart Cities **4**(1), 314–335 (2021)

Arnott, R.: Congestion tolling with agglomeration externalities. Journal of Urban Economics **62**(2), 187–203 (2007)

Busoniu, L., Babuska, R., De Schutter, B.: A comprehensive survey of multiagent reinforcement learning. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) **38**(2), 156–172 (2008)

Bellemare, M.G., Dabney, W., Munos, R.: A distributional perspective on reinforcement learning. In: International Conference on Machine Learning, pp. 449–458 (2017). PMLR

Bono, G., Dibangoye, J.S., Matignon, L., Pereyron, F., Simonin, O.: Cooperative multi-agent policy gradient. In: Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2018, Dublin, Ireland, September 10–14, 2018, Proceedings, Part I 18, pp. 459–476 (2019). Springer

Berto, F., Hua, C., Park, J., Kim, M., Kim, H., Son, J., Kim, H., Kim, J., Park, J.: RL4CO: a unified reinforcement learning for combinatorial optimization library. In: NeurIPS 2023 Workshop: New Frontiers in Graph Learning (2023). https://openreview.net/forum?id=YXSJxi8dOV

Boyles, S.D., Lownes, N.E., Unnikrishnan, A.: Transportation Network Analysis vol. 1. University of Texas, Austin (2023). edition 0.91

Barth-Maron, G., Hoffman, M.W., Budden, D., Dabney, W., Horgan, D., Dhruva, T., Muldal, A., Heess, N., Lillicrap, T.: Distributed distributional deterministic policy gradients. In: International Conference on Learning Representations (2018)

Bogyrbayeva, A., Meraliyev, M., Mustakhov, T., Dauletbayev, B.: Machine learning to solve vehicle routing problems: A survey. IEEE Transactions on Intelligent Transportation Systems (2024)

Beckmann, M., McGuire, C.B., Winsten, C.B.: Studies in the economics of transportation. Technical report, Cowles Commission for Research in Economics, New Haven (1956)

Bellemare, M.G., Naddaf, Y., Veness, J., Bowling, M.: The arcade learning environment: An evaluation platform for general agents. Journal of Artificial Intelligence Research **47**, 253–279 (2013)

Bettini, M., Prorok, A., Moens, V.: Benchmarl: Benchmarking multi-agent reinforcement learning. Journal of Machine Learning Research **25**(217), 1–10 (2024)

Chen, H., An, B., Sharon, G., Hanna, J., Stone, P., Miao, C., Soh, Y.: Dyetc: Dynamic electronic toll collection for traffic congestion alleviation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)

Claus, C., Boutilier, C.: The dynamics of reinforcement learning in cooperative multiagent systems. AAAI/IAAI **1998**(746-752), 2 (1998)

Chiu, Y.-C., Bottom, J., Mahut, M., Paz, A., Balakrishna, R., Waller, T., Hicks, J.: Dynamic traffic assignment: A primer. Transportation Research Circular (E-C153) (2011)

Cole, R., Dodis, Y., Roughgarden, T.: Pricing network edges for heterogeneous selfish users. In: Proceedings of the Thirty-fifth Annual ACM Symposium on Theory of Computing, pp. 521–530 (2003)

Chiou, S.-W.: A knowledge-assisted reinforcement learning optimization for road network design problems under uncertainty. Knowledge-Based Systems **292**, 111614 (2024)

Como, G., Maggistro, R.: Distributed dynamic pricing of multiscale transportation networks. IEEE Transactions on Automatic Control **67**(4), 1625–1638 (2021)

Chowdhury, D., Santen, L., Schadschneider, A.: Statistical physics of vehicular traffic and some related systems. Physics Reports **329**(4-6), 199–329 (2000)

Chakravarty, S., Tanveer, M.H., Voicu, R.C., Banerjee, M.: Optimal-tolling using reinforcement learning. In: SoutheastCon 2024, pp. 1317–1321 (2024). IEEE

Daganzo, C.F.: The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. Transportation Research Part B: Methodological **28**(4), 269–287 (1994)

Daganzo, C.F.: The cell transmission model, part ii: network traffic. Transportation Research Part B: Methodological **29**(2), 79–93 (1995)

Dulac-Arnold, G., Levine, N., Mankowitz, D.J., Li, J., Paduraru, C., Gowal, S., Hester, T.: Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. Machine Learning **110**(9), 2419–2468 (2021)

Du, W., Ding, S.: A survey on multi-agent deep reinforcement learning: from the perspective of challenges and applications. Artificial Intelligence Review **54**(5),

3215–3238 (2021)

Diallo, A.O., Lozenguez, G., Doniec, A., Mandiau, R.: Comparative evaluation of road traffic simulators based on modeler's specifications: An application to intermodal mobility behaviors. In: ICAART (1), pp. 265–272 (2021)

Dwivedi, V.P., Luu, A.T., Laurent, T., Bengio, Y., Bresson, X.: Graph neural networks with learnable structural and positional representations. In: International Conference on Learning Representations (2022). https://openreview.net/forum?id=wTTjnvGphYj

De Palma, A., Lindsey, R.: Traffic congestion pricing methodologies and technologies. Transportation Research Part C: Emerging Technologies **19**(6), 1377–1399 (2011)

Degris, T., Pilarski, P.M., Sutton, R.S.: Model-free reinforcement learning with continuous action in practice. In: 2012 American Control Conference (ACC), pp. 2177–2182 (2012). IEEE

Dafermos, S.C., Sparrow, F.T.: The traffic assignment problem for a general network. Journal of Research of the National Bureau of Standards B **73**(2), 91–118 (1969)

Degris, T., White, M., Sutton, R.S.: Off-policy actor-critic. Proceedings of the 29$^{\text{th}}$ International Conference on Machine Learning (2012)

Ekström, J.: Finding second-best toll locations and levels by relaxing the set of first-best feasible toll vectors. European Journal of Transport and Infrastructure Research **14**(1) (2014)

Eliasson, J.: Congestion pricing. In: The Routledge Handbook of Transport Economics, pp. 209–226. Routledge, New York and London (2017)

Foerster, J.N.: Deep multi-agent reinforcement learning. PhD thesis, University of Oxford (2018)

Farazi, N.P., Zou, B., Ahamed, T., Barua, L.: Deep reinforcement learning in transportation research: A review. Transportation research interdisciplinary perspectives **11**, 100425 (2021)

Farias, A.V., Zhu, S., Mardan, A.: An overview of dynamic pricing toll roads in the united states: Pricing algorithms, operation strategies, equity concerns, and funding mechanism. Case Studies on Transport Policy, 101226 (2024)

Gronauer, S., Diepold, K.: Multi-agent deep reinforcement learning: a survey. Artificial Intelligence Review **55**(2), 895–943 (2022)

Gupta, J.K., Egorov, M., Kochenderfer, M.: Cooperative multi-agent control using deep reinforcement learning. In: Autonomous Agents and Multiagent Systems: AAMAS 2017 Workshops, Best Papers, São Paulo, Brazil, May 8-12, 2017, Revised

Selected Papers 16, pp. 66–83 (2017). Springer

Genser, A., Kouvelas, A.: Dynamic congestion pricing for multi-region networks: A traffic equilibria approach. In: 19th Swiss Transport Research Conference (STRC 2019) (2019). STRC

Genser, A., Kouvelas, A.: Dynamic optimal congestion pricing in multi-region urban networks by application of a multi-layer-neural network. Transportation Research Part C: Emerging Technologies **134**, 103485 (2022)

Hoogendoorn, S.P., Bovy, P.H.: State-of-the-art of vehicular traffic flow modelling. Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering **215**(4), 283–303 (2001)

Hernandez-Leal, P., Kaisers, M., Baarslag, T., De Cote, E.M.: A survey of learning in multiagent environments: Dealing with non-stationarity. arXiv preprint arXiv:1707.09183 (2017)

Hernandez-Leal, P., Kartal, B., Taylor, M.E.: A survey and critique of multiagent deep reinforcement learning. Autonomous Agents and Multi-Agent Systems **33**(6), 750–797 (2019)

He, Q., Ma, M., Li, C., Liu, W.: Learning and managing stochastic network traffic dynamics: an iterative and interactive approach. Transportmetrica B: transport dynamics **12**(1), 2303050 (2024)

Haarnoja, T., Tang, H., Abbeel, P., Levine, S.: Reinforcement learning with deep energy-based policies. In: International Conference on Machine Learning, pp. 1352–1361 (2017). PMLR

Han, Y., Wang, M., Leclercq, L.: Leveraging reinforcement learning for dynamic traffic control: A survey and challenges for field implementation. Communications in Transportation Research **3**, 100104 (2023)

Haydari, A., Yilmaz, Y.: Deep reinforcement learning for intelligent transportation systems: A survey. IEEE Transactions on Intelligent Transportation Systems (2020)

Haarnoja, T., Zhou, A., Abbeel, P., Levine, S.: Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: International Conference on Machine Learning, pp. 1861–1870 (2018). PMLR

Joksimovic, D., Bliemer, M.C., Bovy, P.H.: Optimal toll design problem in dynamic traffic networks with joint route and departure time choice. Transportation Research Record **1923**(1), 61–72 (2005)

Jin, J., Zhu, X., Wu, B., Zhang, J., Wang, Y.: A dynamic and deadline-oriented road pricing mechanism for urban traffic management. Tsinghua Science and Technology

27(1), 91–102 (2021)

Kakade, S.M.: A natural policy gradient. Advances in neural information processing systems **14** (2001)

Kaelbling, L.P., Littman, M.L., Cassandra, A.R.: Planning and acting in partially observable stochastic domains. Artificial intelligence **101**(1-2), 99–134 (1998)

Kaelbling, L.P., Littman, M.L., Moore, A.W.: Reinforcement learning: A survey. Journal of artificial intelligence research **4**, 237–285 (1996)

Knight, F.H.: Some fallacies in the interpretation of social cost. The Quarterly Journal of Economics **38**(4), 582–606 (1924)

Knorr, F.: Applicability and application of microscopic traffic simulations. PhD thesis, Universitätsbibliothek Duisburg-Essen (2013)

Kumar, N., Raubal, M.: Applications of deep learning in congestion detection, prediction and alleviation: A survey. Transportation Research Part C: Emerging Technologies **133**, 103432 (2021)

Kohl, N., Stone, P.: Policy gradient reinforcement learning for fast quadrupedal locomotion. In: IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004, vol. 3, pp. 2619–2624 (2004). IEEE

Kok, J.R., Vlassis, N.: Collaborative multiagent reinforcement learning by payoff propagation. Journal of Machine Learning Research **7**, 1789–1828 (2006)

Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (2017). https://openreview.net/forum?id=SJU4ayYgl

Lopez, P.A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., Wießner, E.: Microscopic traffic simulation using sumo. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), pp. 2575–2582 (2018). IEEE

Liang, C., Huang, Z., Liu, Y., Liu, Z., Zheng, G., Shi, H., Du, Y., Li, F., Li, Z.: Cblab: Scalable traffic simulation with enriched data supporting. arXiv preprint arXiv:2210.00896 (2022)

Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D.: Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971 (2015)

Lu, J., Hong, C., Wang, R.: Magt-toll: A multi-agent reinforcement learning approach to dynamic traffic congestion pricing. PloS one **19**(11), 0313828 (2024)

Li, Y.: Deep reinforcement learning: An overview. arXiv preprint arXiv:1701.07274 (2017)

Littman, M.L.: Markov games as a framework for multi-agent reinforcement learning. In: Machine Learning Proceedings 1994, pp. 157–163. Elsevier, Amsterdam (1994)

Levine, S., Kumar, A., Tucker, G., Fu, J.: Offline reinforcement learning: Tutorial, review, and perspectives on open problems. arXiv preprint arXiv:2005.01643 (2020)

Lombardi, C., Picado-Santos, L., Annaswamy, A.M.: Model-based dynamic toll pricing: An overview. Applied Sciences **11**(11), 4778 (2021)

Lindsey, R., Verhoef, E.: Traffic congestion and congestion pricing. In: Handbook of Transport Systems and Traffic Control. Emerald Group Publishing Limited, United Kingdom (2001)

Lighthill, M.J., Whitham, G.B.: On kinematic waves ii. a theory of traffic flow on long crowded roads. Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences **229**(1178), 317–345 (1955)

Mnih, V., Badia, A.P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., Kavukcuoglu, K.: Asynchronous methods for deep reinforcement learning. In: International Conference on Machine Learning, pp. 1928–1937 (2016). PMLR

Maheshwari, C., Kulkarni, K., Pai, D., Yang, J., Wu, M., Sastry, S.: Congestion pricing for efficiency and equity: Theory and applications to the san francisco bay area. arXiv preprint arXiv:2401.16844 (2024)

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., *et al.*: Human-level control through deep reinforcement learning. nature **518**(7540), 529–533 (2015)

Mirzaei, H., Sharon, G., Boyles, S., Givargis, T., Stone, P.: Enhanced delta-tolling: Traffic optimization via policy gradient reinforcement learning. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), pp. 47–52 (2018). IEEE

Mirzaei, H., Sharon, G., Boyles, S., Givargis, T., Stone, P.: Link-based parameterized micro-tolling scheme for optimal traffic management. In: Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, pp. 2013–2015 (2018)

Mokbel, M., Sakr, M., Xiong, L., Züfle, A., Almeida, J., Anderson, T., Aref, W., Andrienko, G., Andrienko, N., Cao, Y., et al.: Mobility data science: Perspectives and challenges. ACM Transactions on Spatial Algorithms and Systems (2024)

Marbach, P., Tsitsiklis, J.N.: Simulation-based optimization of markov reward processes. IEEE Transactions on Automatic Control **46**(2), 191–209 (2001)

Nash Jr, J.F.: Equilibrium points in n-person games. Proceedings of the National Academy of Sciences **36**(1), 48–49 (1950)

Nguyen, J., Powers, S.T., Urquhart, N., Farrenkopf, T., Guckert, M.: An overview of agent-based traffic simulators. Transportation research interdisciplinary perspectives **12**, 100486 (2021)

Nisha, A.S.A., Rao, N.V., Venkatesh, G., Murugan, S., Meenakshi, B., *et al.*: Efficient congestion management through iot-driven road user charging systems with reinforcement learning. In: 2024 Second International Conference on Intelligent Cyber Physical Systems and Internet of Things (ICoICI), pp. 431–436 (2024). IEEE

Nohekhan, A., Zahedian, S., Sadabadi, K.F.: Investigating the impacts of i-66 inner beltway dynamic tolling system. Transportation Engineering **4**, 100059 (2021)

Padakandla, S.: A survey of reinforcement learning algorithms for dynamically varying environments. ACM Computing Surveys (CSUR) **54**(6), 1–25 (2021)

Pandey, V., Boyles, S.D.: Multiagent reinforcement learning algorithm for distributed dynamic pricing of managed lanes. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), pp. 2346–2351 (2018). IEEE

Pandey, V., Boyles, S.D.: Comparing route choice models for managed lane networks with multiple entrances and exits. Transportation Research Record **2673**(10), 381–393 (2019)

Paccagnan, D., Chandan, R., Ferguson, B.L., Marden, J.R.: Optimal taxes in atomic congestion games. ACM Transactions on Economics and Computation (TEAC) **9**(3), 1–33 (2021)

Prieto Curiel, R., González Ramírez, H., Quiñones Domínguez, M., Orjuela Mendoza, J.P.: A paradox of traffic and extra cars in a city as a collective behaviour. Royal Society open science **8**(6), 201808 (2021)

Pigou, A.C.: The Economics of Welfare. Macmillan, London (1924)

Patterson, A., Neumann, S., White, M., White, A.: Empirical design in reinforcement learning. Journal of Machine Learning Research **25**(318), 1–63 (2024)

Peters, J., Schaal, S.: Policy gradient methods for robotics. In: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2219–2225 (2006). IEEE

Pandey, V., Wang, E., Boyles, S.D.: Deep reinforcement learning algorithm for dynamic pricing of express lanes with multiple access locations. Transportation

Research Part C: Emerging Technologies **119**, 102715 (2020)

Qiu, W., Chen, H., An, B.: Dynamic electronic toll collection via multi-agent deep reinforcement learning with edge-based graph convolutional networks. In: IJCAI, pp. 4568–4574 (2019)

Qin, Z.T., Zhu, H., Ye, J.: Reinforcement learning for ridesharing: An extended survey. Transportation Research Part C: Emerging Technologies **144**, 103852 (2022)

Ramos, G.d.O., Da Silva, B.C., Rădulescu, R., Bazzan, A.L., Nowé, A.: Toll-based reinforcement learning for efficient equilibria in route choice. The Knowledge Engineering Review **35** (2020)

Ramos, G.d.O., Silva, B.C., Rădulescu, R., Bazzan, A.L.: Learning system-efficient equilibria in route choice using tolls. In: Proceedings of the Adaptive Learning Agents Workshop, vol. 2018 (2018)

Richards, P.I.: Shock waves on the highway. Operations research **4**(1), 42–51 (1956)

Ramos, G.d.O., Rădulescu, R., Nowé, A., Tavares, A.R.: Toll-based learning for minimising congestion under heterogeneous preferences. In: Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), pp. 1098–1106 (2020). IFAAMAS

Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (2018)

Saharan, S., Bawa, S., Kumar, N.: Dynamic pricing techniques for intelligent transportation system in smart cities: A systematic review. Computer Communications **150**, 603–625 (2020)

Schmidt, L.M., Brosig, J., Plinge, A., Eskofier, B.M., Mutschler, C.: An introduction to multi-agent reinforcement learning and review of its application to autonomous mobility. In: 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC), pp. 1342–1349 (2022). IEEE

Storani, F., Di Pace, R., Bruno, F., Fiori, C.: Analysis and comparison of traffic flow models: a new hybrid traffic flow model vs benchmark models. European Transport Research Review **13**(1), 1–16 (2021)

Seo, T.: Trial-and-error congestion pricing scheme for morning commute problem with day-to-day dynamics. Transportation Research Procedia **47**, 561–568 (2020)

Sharon, G., Hanna, J., Rambha, T., Albert, M., Stone, P., Boyles, S.D.: Delta-tolling: Adaptive tolling for optimizing traffic throughput. In: ATT@ IJCAI (2016)

Sharon, G., Hanna, J.P., Rambha, T., Levin, M.W., Albert, M., Boyles, S.D., Stone, P.: Real-time adaptive tolling scheme for optimized social welfare in traffic networks.

In: Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS-2017) (2017)

Singh, S., Kearns, M.J., Mansour, Y.: Nash convergence of gradient dynamics in general-sum games. In: UAI, pp. 541–548 (2000)

Shoham, Y., Leyton-Brown, K.: Multiagent Systems: Algorithmic, Game-theoretic, and Logical Foundations. Cambridge University Press, Cambridge (2008)

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., Riedmiller, M.: Deterministic policy gradient algorithms. In: International Conference on Machine Learning, pp. 387–395 (2014). PMLR

Sharon, G., Levin, M.W., Hanna, J.P., Rambha, T., Boyles, S.D., Stone, P.: Network-wide adaptive tolling for connected and automated vehicles. Transportation Research Part C: Emerging Technologies **84**, 142–157 (2017)

Sutton, R.S., McAllester, D., Singh, S., Mansour, Y.: Policy gradient methods for reinforcement learning with function approximation. Advances in neural information processing systems **12** (1999)

Schuster, P., Sigmund, K.: Replicator dynamics. Journal of theoretical biology **100**(3), 533–538 (1983)

Sato, K., Seo, T., Fuse, T.: A reinforcement learning-based dynamic congestion pricing method for the morning commute problems. Transportation Research Procedia **52**, 347–355 (2021)

Sato, K., Seo, T., Fuse, T.: Dynamic network congestion pricing based on deep reinforcement learning. arXiv preprint arXiv:2206.12188 (2022)

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)

Tan, M.: Multi-agent reinforcement learning: Independent vs. cooperative agents. In: Proceedings of the Tenth International Conference on Machine Learning, pp. 330–337 (1993)

Tavares, A.R., Bazzan, A.L.: An agent-based approach for road pricing: system-level performance and implications for drivers. Journal of the Brazilian Computer Society **20**(1), 1–15 (2014)

Terry, J., Black, B., Grammel, N., Jayakumar, M., Hari, A., Sullivan, R., Santos, L.S., Dieffendahl, C., Horsch, C., Perez-Vicente, R., *et al.*: Pettingzoo: Gym for multi-agent reinforcement learning. Advances in Neural Information Processing Systems **34**, 15032–15043 (2021)

Tampère, C.M., Corthout, R., Cattrysse, D., Immers, L.H.: A generic class of first order

node models for dynamic macroscopic simulation of traffic flows. Transportation Research Part B: Methodological **45**(1), 289–309 (2011)

Todorov, E., Erez, T., Tassa, Y.: Mujoco: A physics engine for model-based control. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5026–5033 (2012). IEEE

Treiber, M., Hennecke, A., Helbing, D.: Congested traffic states in empirical observations and microscopic simulations. Phys. Rev. E **62**, 1805–1824 (2000) https://doi.org/10.1103/PhysRevE.62.1805

Towers, M., Kwiatkowski, A., Terry, J., Balis, J.U., De Cola, G., Deleu, T., Goulão, M., Kallinteris, A., Krimmel, M., KG, A., et al.: Gymnasium: A standard interface for reinforcement learning environments. arXiv preprint arXiv:2407.17032 (2024)

Tsekeris, T., Voß, S.: Design and evaluation of road pricing: state-of-the-art and methodological advances. NETNOMICS: Economic Research and Electronic Networking **10**, 5–52 (2009)

United States Bureau of Public Roads: Traffic Assignment Manual for Application with a Large, High Speed Computer vol. 2. US Department of Commerce, Bureau of Public Roads, Office of Planning, Urban Planning Division, Washington D.C. (1964)

Verhoef, E.T.: Time, speeds, flows and densities in static models of road traffic congestion and congestion pricing. Regional Science and Urban Economics **29**(3), 341–369 (1999)

Verhoef, E.T.: Second-best congestion pricing in general networks-algorithms for finding second-best optimal toll levels and toll points. Technical report, Tinbergen Institute Discussion Paper (2000)

Verhoef, E.T.: Second-best congestion pricing in general static transportation networks with elastic demands. Regional Science and Urban Economics **32**(3), 281–310 (2002)

Vickrey, W.S.: Pricing in urban and suburban transport. The American Economic Review **53**(2), 452–465 (1963)

Vickrey, W.S.: Congestion theory and transport investment. The American Economic Review **59**(2), 251–260 (1969)

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)

Wageningen-Kessels, F., Van Lint, H., Vuik, K., Hoogendoorn, S.: Genealogy of traffic flow models. EURO Journal on Transportation and Logistics **4**(4), 445–473 (2015)

Wardrop, J.G.: Road paper. some theoretical aspects of road traffic research. Proceedings of the institution of civil engineers **1**(3), 325–362 (1952)

Watkins, C.J.C.H.: Learning from delayed rewards. PhD thesis, King's College, Cambridge United Kingdom (1989)

Wong, A., Bäck, T., Kononova, A.V., Plaat, A.: Deep multiagent reinforcement learning: Challenges and directions. Artificial Intelligence Review, 1–34 (2022)

Watkins, C.J., Dayan, P.: Q-learning. Machine learning **8**(3), 279–292 (1992)

Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine learning **8**(3), 229–256 (1992)

Wang, Y., Jin, H., Zheng, G.: Ctrl: Cooperative traffic tolling via reinforcement learning. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management, pp. 3545–3554 (2022)

Wen, M., Kuba, J., Lin, R., Zhang, W., Wen, Y., Wang, J., Yang, Y.: Multi-agent reinforcement learning is a sequence modeling problem. Advances in Neural Information Processing Systems **35**, 16509–16521 (2022)

Wei, H., Zheng, G., Gayah, V., Li, Z.: Recent advances in reinforcement learning for traffic signal control: A survey of models and evaluation. ACM SIGKDD Explorations Newsletter **22**(2), 12–18 (2021)

Yan, Y., Chow, A.H., Ho, C.P., Kuo, Y.-H., Wu, Q., Ying, C.: Reinforcement learning for logistics and supply chain management: Methodologies, state of the art, and future opportunities. Transportation Research Part E: Logistics and Transportation Review **162**, 102712 (2022)

Yang, H., Huang, H.-J.: Principle of marginal-cost pricing: how does it work in a general road network? Transportation Research Part A: Policy and Practice **32**(1), 45–54 (1998)

Ye, G., Song, J., Feng, M., Zhu, G., Shen, P., Zhang, L., Shah, S.A.A., Bennamoun, M.: Position and structure-aware graph learning. Neurocomputing **556**, 126581 (2023)

Yuan, L., Zhang, Z., Li, L., Guan, C., Yu, Y.: A survey of progress on cooperative multi-agent reinforcement learning in open environment. arXiv preprint arXiv:2312.01058 (2023)

Zhang, H., Feng, S., Liu, C., Ding, Y., Zhu, Y., Zhou, Z., Zhang, W., Yu, Y., Jin, H., Li, Z.: Cityflow: A multi-agent reinforcement learning environment for large scale city traffic scenario. In: The World Wide Web Conference, pp. 3620–3624 (2019)

Zhu, F., Ukkusuri, S.V.: A reinforcement learning approach for distance-based dynamic tolling in the stochastic network environment. Journal of Advanced

Transportation **49**(2), 247–266 (2015)

Zhang, X., Wang, W., Chen, J.: A priori lane selection strategy for reinforcement learning of dynamic expressway tolling. In: 2023 International Conference on Pattern Recognition, Machine Vision and Intelligent Algorithms (PRMVIA), pp. 143–154 (2023). IEEE

Appendix A Concise introduction to Reinforcement Learning

A.1 Reinforcement Learning

In RL (Sutton and Barto, 2018), an agent interacts dynamically with an environment at each of a sequence of discrete episode steps $t = 0, 1, 2, 3, \ldots$. RL is about learning from these interactions to achieve a goal. The agent is the learner and decision maker, the environment is everything else. Framing the problem of learning from interaction to achieve a goal can be done with a Markov Decision Process. A Markov Decision Process (MDP) is formally defined as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$ where $\mathcal{S}$ is the set of all states s, a complete description of the environment. $\mathcal{A}$ is the set of all actions a, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition probability function mapping from any state $s \in \mathcal{S}$ to any next state $s' \in \mathcal{S}$ after execution of $a \in \mathcal{A}$, also denoted by $\mathcal{P}(s'|s, a)$. $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the reward function that provides the reward r, also denoted by $\mathcal{R}(s, a, s')$. The agent seeks to maximize the total amount of reward over time through its choice of actions using a policy π. Policy $\pi(a|s)$ gives the probability of taking action a given state s. Policies can be stochastic ($\pi(a|s)$) or deterministic ($\pi(s)$). The MDP and agent, following the policy π, give rise to a sequence of states $S_t = s$, actions $A_t = a$ and rewards $R_t = r$ which is called a trajectory T: $T = S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, \ldots$. An episode is a trajectory T that has reached a final episode step H. See Figure A1 that illustrates agent environment interaction. The probability of a trajectory T,

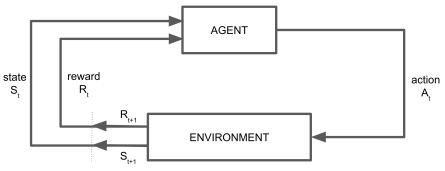


Fig. A1: Agent environment interaction in a MDP (Sutton and Barto, 2018)

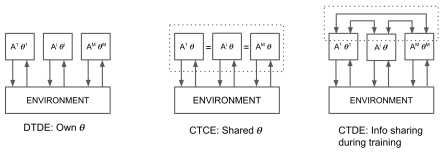


Fig. A2: Approaches to cooperative MARL (Gronauer and Diepold, 2022; Wong et al., 2022)

following policy π and S_0 sampled from the start state distribution, is:

$$p(T|\pi) = p(S_0) \prod_{t=0}^{t=H-1} \pi(A_t|S_t) p(S_{t+1}|S_t, A_t). \tag{A1}$$

The return G_t after episode step t is defined as the cumulative discounted reward over time:

$$G_t = \sum_{i=t+1}^{H} \gamma^{i-t-1} R_i = R_{t+1} + \gamma G_{t+1}, \tag{A2}$$

where $\gamma \in [0, 1]$ is the discount factor, reflecting the importance of future rewards. The goal of the agent is defined to maximize the total expected discounted return per episode:

$$J = \mathbb{E}_{T\sim\ p(T)} G_0(T). \tag{A3}$$

Any policy can be used by an agent for execution in an environment. RL algorithms specify how to find an optimal policy π_* using the experience gained by the agent. In such an algorithm, the agent's experience consists of one or more episodes. On the basis of this experience, the agent can learn (an estimate of) the optimal policy. Where the agent generates its own data we refer to this as online learning, while off-line learning refers to the situation where the data is collected by another policy (Levine et al., 2020). Experiences in Partially Observable MDPs, where the agent does not have perfect state information, lead to partial observations (Kaelbling et al., 1998, 1996). In that case we need and use a set of observations $\mathcal{O}$ and an observation probability function $\mathcal{Z} : \mathcal{S} \times \mathcal{A} \to \mathcal{O}$.

Value Functions. We measure the value of a policy, given a starting state, by a value function to compare the performance of different policies. The value function, $v_\pi : \mathcal{S} \to \mathbb{R}$ is defined as the expected return when starting in state s, at any episode step t and following policy π thereafter. Similarly, the action value function, $q_\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, also known as q-value function gives the value when taking action a when in state s, following policy π thereafter.

For MDPs this can be defined as:

$$v_\pi(s) = \mathbb{E}_\pi[G_t | S_t = s]. \tag{A4}$$

Similarly, the action value function, $q_\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, also known as q-value function or q-function, is defined as:

$$q_\pi(s, a) = \mathbb{E}_\pi[G_t | S_t = s, A_t = a], \tag{A5}$$

which gives the value when taking action a when in state s, following policy π thereafter.

The estimated value of a state S_t is denoted as $V(S_t)$ and for q-value functions the estimated value is denoted as $Q(S_t, A_t)$. Approximate values, based on a function parameterized by weights $\boldsymbol{\theta}$, are denoted as $\hat{v}(s, \boldsymbol{\theta})$ or $\hat{q}(s, a, \boldsymbol{\theta})$. The relation between the value function $v_\pi(s)$ and action-value function $q_\pi(s, a)$ is that, given the same state, the value function is the expectation of the action-value function with respect to the actions:

$$v_\pi(s) = \sum_a \pi(a|s) q_\pi(s, a). \tag{A6}$$

The difference between the action-value function and value function for a state and action can be considered as an evaluation of the policy compared with the average

policy and is called advantage:

$$A(S_t, A_t) = Q(S_t, A_t) - V(S_t). \tag{A7}$$

To solve an RL problem usually involves estimating the value function which describes, in terms of expected return, how good it is to be in a certain state s. We then use the value function to find the optimal policy. If we substitute the return (A2) in the value function (A4) we can derive:

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a) \left[r + \gamma v_\pi(s') \right], \; \forall \, s \in \mathcal{S}. \tag{A8}$$

Equation (A8) is called the Bellman equation for v_π and provides the recursive relation between the value of a state and its successor states. For the action-value function the Bellman equation reads:

$$q_\pi(s, a) = \sum_{s',r} p(s', r|s, a) \left[r + \gamma \sum_{a'} \pi(a'|s') q_\pi(s', a') \right]. \tag{A9}$$

If we consider a policy π, for which the expected return is larger than from any other policy π' for all states s, we call this the optimal policy π_*. The optimal value function and optimal action-value function are defined as: $v_*(s) = \max_\pi v_\pi(s)$ and $q_*(s, a) = \max_\pi q_\pi(s, a)$. If both the transition function $\mathcal{P}$ and reward function $\mathcal{R}$ are known, we can find the optimal policy by planning, using dynamic programming. These are called model-based methods. However, when $\mathcal{P}$ and $\mathcal{R}$ are unknown, we need to solve them using a model-free RL algorithm. These are based on learning; interacting with the environment, obtaining rewards and states and improving the policy based on this.

Apart from strictly adhering to the policy $\pi(a|s)$ by taking action a when in state s, greedy and ϵ-greedy policies can be distinguished. A policy π is called greedy if it always selects the action with maximum estimated action value, in which case the agent is exploiting the current knowledge of the value of its actions. A policy π is called ϵ-greedy if the agent acts greedily most of the time, and with probability ϵ, it selects an action randomly. If one of the non-greedy actions is chosen, we call this exploring because this exposes the agent potentially to new experiences. This enables improvement of the estimate of the non-greedy action's value.

A.2 Solution Methods to Find the Policy

We distinguish three main categories: methods based on value functions, methods directly updating the policy and methods based on a combination of these, actor-critic.

A.2.1 Value Based Methods

Value based methods search the optimal policy based on the (action-)value function. This means that the value is estimated or approximated first. Based on these values the

policy is optimized. Model free Monte Carlo (MC) methods learn value functions and optimal policies from experience by sample episodes. MC methods update estimates of values of a state $V(S_t)$ by waiting for the sample return G_t of an episode and use that as target for $V(S_t)$.

$$V(S_t) \leftarrow V(S_t) + \alpha \left[G_t - V(S_t) \right]. \tag{A10}$$

where $\alpha \in (0, 1]$ is a constant step-size parameter. Model free Temporal Difference (TD) methods update estimates, based on other learned estimates. For example, where MC needs to obtain the sample return (A2), the one-step TD method, TD(0), uses the reward R_{t+1} and the discounted estimate of the value of the next state $\gamma V(S_{t+1})$, to update the estimate of $V(S_t)$:

$$V(S_t) \leftarrow V(S_t) + \alpha \left[R_{t+1} + \gamma V(S_{t+1}) - V(S_t) \right]. \tag{A11}$$

in which we call $R_{t+1} + \gamma V(S_{t+1})$ the TD(0)-target and δ_t the TD-error:

$$\delta_t = R_{t+1} + \gamma V(S_{t+1}) - V(S_t). \tag{A12}$$

In RL, δ_t occurs in various forms depending on the number of steps (N-step returns) taken into account to determine the target. An illustration of a value based method is the Q-learning (Watkins and Dayan, 1992) algorithm. Q-learning, is an off-policy RL algorithm. Off-policy means that the selection of actions is based on another policy than the actions used to estimate the Q-value. The update function for Q-learning is:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t) \right]. \tag{A13}$$

The learned Q-function then directly approximates the optimal q-function $q_*(s, a)$ (Watkins, 1989; Watkins and Dayan, 1992).

A.2.2 Policy Based Methods

A policy based method (Sutton et al., 1999), directly optimizes the parameterized policy of an agent with respect to the expected return by gradient descent. The stochastic policy $\pi(a|s, \boldsymbol{\theta})$ (hereafter denoted as $\pi_{\boldsymbol{\theta}}$) is parameterized by $\boldsymbol{\theta}$ and differentiable with respect to its parameters. Policy gradient methods learn the policy parameters based on the gradient of some performance measure $J(\pi_{\boldsymbol{\theta}})$ (hereafter noted as $J(\boldsymbol{\theta})$) with respect to $\boldsymbol{\theta}$. The parameter update is expressed as:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha \hat{\nabla} J(\boldsymbol{\theta}_t), \tag{A14}$$

where $\hat{\nabla} J(\boldsymbol{\theta}_t)$ is an estimate of the gradient of the performance measure, with respect to $\boldsymbol{\theta}_t$ (Sutton and Barto, 2018). If we define, in an episodic case, the performance measure as the value at the start of the episode:

$$J(\boldsymbol{\theta}) = v_{\pi_{\boldsymbol{\theta}}}(s_0), \tag{A15}$$

then, with $\mu(s)$ as the on-policy distribution of states under π it can be derived that:

$$\nabla J(\boldsymbol{\theta}) \propto \sum_s \mu(s) \sum_a q_\pi(s,a) \nabla \pi(a|s,\boldsymbol{\theta}). \tag{A16}$$

Equation (A16) is called the policy gradient theorem (Sutton et al., 1999)(Marbach and Tsitsiklis, 2001) for stochastic policies. A deterministic case leading to $\pi(s,\boldsymbol{\theta})$ can be derived as well (Silver et al., 2014). A classical example of a policy gradient algorithm is REINFORCE (Williams, 1992). From the policy gradient theorem (A16), it can be derived that:

$$\nabla J(\boldsymbol{\theta}) \propto \mathbb{E}_\pi \left[G_t \frac{\nabla \pi(A_t|S_t,\boldsymbol{\theta})}{\pi(A_t|S_t,\boldsymbol{\theta})} \right], \tag{A17}$$

where the expectation is taken with respect to the policy π. The expression between the brackets can be sampled on each episode step and be used to update $\boldsymbol{\theta}$:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha G_t \frac{\nabla \pi(A_t|S_t,\boldsymbol{\theta}_t)}{\pi(A_t|S_t,\boldsymbol{\theta}_t)}. \tag{A18}$$

REINFORCE uses samples of the full return, and therefore may have a high variance which slows learning. To address this, the policy gradient theorem can be generalized to include a baseline function. As long as the baseline function does not vary with actions a any function $b(s)$ can be used:

$$\nabla J(\boldsymbol{\theta}) \propto \sum_s \mu(s) \sum_a \left(q_\pi(s,a) - b(s) \right) \nabla \pi(a|s,\boldsymbol{\theta}). \tag{A19}$$

In addition to the Policy Gradient method discussed above there are Natural Policy Gradient (Kakade, 2001) and Finite Difference Policy Gradient (Kohl and Stone, 2004) methods as different approaches, also directly optimizing the policy.

A.2.3 Actor-critic Methods.

A frequently used architecture, actor-critic (Degris et al., 2012)(Degris et al., 2012), combines the value-based methods and policy-based methods described above. Actor-critic uses the policy gradient theorem (A16) and value functions (A4, A5). In actor-critic algorithms the actor is the learned policy $\pi(a|s,\boldsymbol{\theta})$ and the critic the learned value function $\hat{v}(s,\boldsymbol{\vartheta})$, or the action value function $\hat{q}(s,a,\boldsymbol{\vartheta})$. In the example below a one-step actor-critic method uses the TD-error δ_t:

$$\delta_t = r' + \gamma \hat{v}(s',\boldsymbol{\vartheta}) - \hat{v}(s,\boldsymbol{\vartheta}). \tag{A20}$$

Replacing $q_\pi(s,a)$ in (A16), with the TD-error (A20) leads to:

$$\nabla J(\boldsymbol{\theta}) \propto \sum_s \mu(s) \sum_a \left(r' + \gamma \hat{v}(s',\boldsymbol{\vartheta}) - \hat{v}(s,\boldsymbol{\vartheta}) \right) \nabla \pi(a|s;\boldsymbol{\theta}). \tag{A21}$$

The learning method to update the critic's parameters $\boldsymbol{\vartheta}$ can be performed using semi-gradient TD(0). Semi-gradient TD(0) takes the effect of changing the parameters $\boldsymbol{\vartheta}_t$ on the learned state value ($\hat{v}(S_t, \boldsymbol{\vartheta})$ into account, but not on the target in minimizing δ_t.

A.2.4 Issues and Challenges.

Common issues in RL problems are credit assignment (which action leads to which outcome), sparse rewards (not delivering non-zero rewards frequently enough) and sample efficiency (Li, 2017). Important criteria, to evaluate the performance of algorithms, are whether they are stable (do they converge), their efficiency (how long does it take to converge), generalization (if they converge, can they be generalized) and scalability (can the solution scale efficiently with growing state and action space). Practical challenges exist as well. Due to the large number of design decisions in Reinforcement Learning experiments, much can go wrong illustrated by a listing of 20 common errors (Patterson et al., 2024).

A.3 Multi Agent Reinforcement Learning

A Multi Agent system describes a system in which multiple agents take decisions based on their own policy and interact with the environment (Gronauer and Diepold, 2022; Wong et al., 2022). If this is done in the context of RL this is called Multi Agent Reinforcement Learning (MARL). Depending on the situation and objectives, the agents may compete, cooperate or demonstrate mixed behavior and may or may not communicate with each other.

A MARL system can be formalized as a Markov Game which is a framework that generalizes MDPs from one agent to Multiple Agents (Gronauer and Diepold, 2022; Littman, 1994; Shoham and Leyton-Brown, 2008). A Markov Game is defined as a tuple $(\mathcal{N}, \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$ where: $\mathcal{N}$ is the set of $N > 1$ interacting agents n, $\mathcal{S}$ is the set of states observed by the agents s, $\mathcal{A}$ is the joint action space $\mathcal{A}^1 \times \cdots \times \mathcal{A}^N$ where $\mathcal{A}^n$ is the action set of agent n, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition probability function mapping from any state $s \in \mathcal{S}$ to any next state $s' \in \mathcal{S}$ after execution of $\boldsymbol{a} \in \mathcal{A}$, also expressed as $\mathcal{P}(s'|s, \boldsymbol{a})$, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the reward function that provides the agent specific reward r^n, also expressed as $\mathcal{R}(s, \boldsymbol{a}, s', n)$.

The policy π taken by agent n is denoted as π^n. A joint policy $\boldsymbol{\pi}$ where $\boldsymbol{\pi} = \{\pi^1, \ldots, \pi^N\}$, is a mapping from states s to probabilities of selecting $\boldsymbol{a}$: $\boldsymbol{\pi}(\boldsymbol{a}|s)$ is the probability of $\boldsymbol{A}_t = \boldsymbol{a}$ given $S_t = s$. If all agents n follow a policy $\pi^n \in \Pi^n$ at time t in state s, then joint action $\boldsymbol{a}$ leads, with respect to $\mathcal{P}$, to a new state s' and each agent gets individual reward r^n from $\mathcal{R}$. As per convention $-n$ means all agents except n so $\boldsymbol{\pi}^{-n} = \{\pi^1, \ldots, \pi^{n-1}, \pi^{n+1}, \ldots, \pi^N\}$.

As in the Single Agent case outlined before, the objective of each agent is to optimize its return. A (Nash) equilibrium (Nash Jr, 1950) exists in the situation that the set of policies $\boldsymbol{\pi}_*$ consisting of one policy π^n_* for each agent, maximizes the return for each agent:

$$J^n(\pi^n_*, \boldsymbol{\pi}^{-\boldsymbol{n}}_*) \geq J^n(\pi^n, \boldsymbol{\pi}^{-\boldsymbol{n}}_*) \, \forall \, \pi^n \in \Pi^n. \tag{A22}$$

This means, informally, that no agent, given policies of other agents fixed, can improve by unilaterally deviating from $\boldsymbol{\pi}_*$. We note that multiple Nash equilibria may exist with different returns, making evaluation of Multi Agent learning harder than Single Agent learning. If in MARL all agents receive the same reward R this is a fully cooperative setting; if they are encouraged to cooperate but don't get an equal reward it is called cooperative. If the sum of rewards of the agents equals zero, it is a competitive setting; each agent in this case tries to maximize its own reward and minimize the rewards of others. Mixed settings are neither cooperative nor competitive. These are also called general sum games.

A.4 Solution methods

MARL algorithms encompass similar solution methods as Single Agent RL; value based (e.g. (Tan, 1993)), policy based (e.g. (Singh et al., 2000)) or actor-critic (e.g. (Foerster, 2018)), which are combined with concepts from game theory (Busoniu et al., 2008; Hernandez-Leal et al., 2019). The three approaches below address different training and execution methods for agents. Architectures and more detail of the three approaches can be found in (Albrecht et al., 2024; Gronauer and Diepold, 2022; Wong et al., 2022).

A.4.1 Decentralized Training Decentralized Execution (DTDE)

An approach to find the policy for each of the agents in the Multi Agent setting would be to use the same approach for each individual agent as in the Single Agent setting. Each of the agents takes its input from the environment, take its own actions and get its own rewards. The agents do not share information with other agents (Bono et al., 2019; Gronauer and Diepold, 2022). An example is the Independent Learners approach (Claus and Boutilier, 1998; Kok and Vlassis, 2006) and is based on Q-learning. Each agent has an individual table for Q-values and the global Q-function is defined as a linear combination of individual contributions. The local Q-function is then updated as:

$$Q^n(S_t, A_t^n) \leftarrow Q^n(S_t, A_t^n) + \alpha \left[R_{t+1}^n(S_t, \mathbf{A}_t) + \gamma \max_{a^n} Q^n(S_{t+1}, a^n) - Q^n(S_t, A_t^n) \right]. \tag{A23}$$

Note that the global state S_t is used to determine the agents' Q-values as well as to determine the individual reward $R_{t+1}^n(S_t, \mathbf{A}_t)$. A disadvantage of this approach is that the environment appears non-stationary (Padakandla, 2021) to the agents. By non-stationarity we mean that the dynamics of the environment change over time. The agents do not have access to the other agents' actions, nor can they perceive the joint action. Finally, each agent is learning, that is updating its own policy independently, impacting the other agents' decision making.

A.4.2 Centralized Training Centralized Execution (CTCE)

In another approach, agents learn their policies based on common or shared information, for example local observations or policies. The collection of observations from

each of the agents is used to train a central policy. The central policy sends the actions to all agents for execution. In centralized Q-learning, the update equation (Albrecht et al., 2024) becomes:

$$Q(S_t, \mathbf{A}_t) \leftarrow Q(S_t, \mathbf{A}_t) + \alpha \left[R_{t+1}(S_t, \mathbf{A}_t) + \gamma \max_{\mathbf{a}} Q(S_{t+1}, \mathbf{a}) - Q(S_t, \mathbf{A}_t) \right]. \quad \text{(A24)}$$

A disadvantage of this approach is that the state-action spaces grow exponentially with the number of agents. This could be mitigated by policy factorization using the combined observations, but also by creating individual policies, leading to individual actions. Although this would significantly reduce the size of the action space, the exponential growth of the observation space would still be a problem (Gupta et al., 2017).

A.4.3 Centralized Training Decentralized Execution (CTDE)

Yet another approach would have homogeneous agents having an individual policy, using local observations to a distribution over individual actions but enable them to share information and/or resources during training. Agents execute their policies based on local observations. With this approach, non-stationarity and partial observability are mitigated even if other agents' policies are changing (Wong et al., 2022).

A.4.4 Issues and challenges.

MARL have additional challenges when compared to Single Agent RL. In addition to credit assignment, sparse rewards, stability and generalization these are partial observability, non-stationarity and the effectiveness of different training schemes (Wong et al., 2022). Except for the cooperative case, it is unlikely that agents share observations, policies or parameters. Additional challenges in that case are observing, analyzing and acting on the other agents without sharing the information as described in the cooperative approaches (Busoniu et al., 2008; Hernandez-Leal et al., 2017).

A.5 Deep Reinforcement Learning

Deep Reinforcement Learning algorithms use neural networks as function approximators to solve problems which have continuous or high dimensional state and/or action spaces. We provide relevant examples below, which are selected as these or modifications are used as part of the solution architecture in road pricing. For details on the cited algorithms we refer to the original works.

A.5.1 Value Based Algorithms.

DQN (Deep Q Networks)(Mnih et al., 2015) is an adaptation of Q-Learning. A separate current Q-network $Q(s, a, \boldsymbol{\theta})$ and a target Q-network $Q(s, a, \boldsymbol{\theta}')$ are used to determine the optimal policy. The experiences gathered and stored by taking actions based on $Q(s, a, \boldsymbol{\theta})$ are used to learn the parameters $\boldsymbol{\theta}$ by minimizing a loss function using the current Q-network and target Q-network. Periodically the target parameters $\boldsymbol{\theta}'$ are updated using $\boldsymbol{\theta}$

A.5.2 Policy Based and Actor-Critic Algorithms.

In Proximal Policy Optimization (PPO) (Schulman et al., 2017) an update to a policy is maximized subject to a constraint on the size of the update. Maximization is performed using clipped probability ratios, which compare the new policy $\pi_{\boldsymbol{\theta}}$ and old policy $\pi_{\boldsymbol{\theta}_{old}}$. In another solution approach, Soft Actor Critic (SAC) (Haarnoja et al., 2018, 2017), the reward is augmented with an entropy term. Adding the entropy term improves both exploration by acquiring diverse behaviors and in addition, the algorithm is more robust for model and estimation errors. Yet another approach, Deep Deterministic Policy Gradient (DDPG) (Lillicrap et al., 2015), combines ideas from DPG (see Section A.2 and (Silver et al., 2014)) and DQN (see above). It is a model free, off-policy actor-critic algorithm with neural networks as function approximators.

A.5.3 Multiple Actor/Learner for Value and Policy Based Algorithms.

These algorithms parallelize part of the learning experience. Each actor interacts with its own version of the environment and experiences are shared. Mnih et al. (2016) proposed a framework Asynchronous Advantage Actor Critic (A3C) that uses asynchronous gradient descent for optimization. Multiple agents operate asynchronously, in parallel, on multiple instances of the environment. Periodically the target network of the actor is updated and an asynchronous update of the global parameters is performed by sending the gradients. Multiple updates from the various learners are applied to the central parameters and because of this, are likely to be less correlated, stabilizing learning. See Figure 5 for a case with M agents (both actor and learner), indexed by i, asynchronously updating their local parameters to the global shared parameter network. In case of a single agent it is referred to as A2C. Approaches to enhance DDPG are D3PG and D4PG (Barth-Maron et al. (2018); Bellemare et al. (2017)).

A.5.4 Multi Agent Reinforcement Learning.

A range of Multi Agent RL algorithms using neural networks exist (Albrecht et al., 2024; Gronauer and Diepold, 2022; Wong et al., 2022). The surveyed works below mostly use an actor-critic variant using a CTDE architecture, specifically designed for their use case and will be discussed there.

Appendix B Additional Tables

Table B1: Overview of simulators used in surveyed works

	6.1.1 (Mirzaei et al. (2018b))	6.2.1 (Chen et al. (2018))	6.2.2 (Qiu et al. (2019))	6.2.3 (Jin et al. (2021))	6.2.4 (Wang et al. (2022))	6.2.5 (Lu et al. (2024))	6.2.6 (He et al. (2024))	6.3.1 (Zhu and Ukkusuri (2015))	6.3.2 (Pandey and Boyles (2018))	6.4.1 Pandey et al. (2020))	6.4.2 (Zhang et al. (2023))	6.5.1 (Sato et al. (2021))	6.5.2 (Sato et al. (2022))	6.6.1 (Tavares and Bazzan (2014))	6.6.2 (Chakravarty et al. (2024))	6.7.1 (Ramos et al. (2018))	6.7.2 (Ramos et al. (2020))
Simulation models																	
Custom DTA Simulator (CTM)	✓																
Custom Simulator (BPR)		✓	✓														
Custom Simulator (BPR)				✓													
CityFlow					✓	✓											
Custom Simulator							✓										
Custom Simulator (CTM)								✓									
Custom Simulator (CTM)									✓	✓							
Custom Highway RL Environment											✓						
Custom Simulator												✓	✓				
SUMO														✓			
MATLAB/Simulink															✓		
Custom Simulator																✓	✓

This table is meant to illustrate the diversity in simulation models used in the surveyed works.

Table B2: Overview of road networks in surveyed works

	6.1.1 (Mirzaei et al. (2018b))	6.2.1 (Chen et al. (2018))	6.2.2 (Qiu et al. (2019))	6.2.3 (Jin et al. (2021))	6.2.4 (Wang et al. (2022))	6.2.5 (Lu et al. (2024))	6.2.6 (He et al. (2024))	6.3.1 (Zhu and Ukkusuri (2015))	6.3.2 (Pandey and Boyles (2018))	6.4.1 Pandey et al. (2020))	6.4.2 (Zhang et al. (2023))	6.5.1 (Sato et al. (2021))	6.5.2 (Sato et al. (2022))	6.6.1 (Tavares and Bazzan (2014))	6.6.2 (Chakravarty et al. (2024))	6.7.1 (Ramos et al. (2018))	6.7.2 (Ramos et al. (2020))
Road networks																	
Sioux-Falls	✓																✓
Simplified Sioux-Falls								✓									
Downtown Austin	✓																
Uptown San Antonio	✓																
Singapore Central Region		✓	✓														
Singapore Whole Net			✓														
Qinhuai ext. network				✓													
Hangzhou					✓	✓											
Manhattan					✓												
Porto Roadnet					✓												
Jinan City						✓											
Network of regions							✓										
DESE Network									✓	✓							
LBJ Texpress Toll Segment									✓	✓							
Single Entry Single Exit										✓							
MoPac Express Lane Network										✓							
Simple Expressway Network											✓						
GYExpress											✓						
Single Bottleneck												✓					
Three Bottleneck												✓					
Parallel Bottleneck Network													✓				
Test Network														✓			
Toll Lane, Regular Lane															✓		
B1-B7																✓	✓
BB1, BB3, BB5, BB7																✓	
OW																✓	✓
Anaheim																	✓
Eastern Massachusets																	✓

This table is meant to illustrate the diversity in road networks used in the surveyed works.