The Knowledge Engineering Review, Vol. 35, e8, 1 of 23.
doi:10.1017/S0269888920000119

Toll-based reinforcement learning for efficient equilibria in route choice

GABRIEL DE O. RAMOS[1,2], BRUNO C. DA SILVA[3], ROXANA RĂDULESCU[2], ANA L. C. BAZZAN[3], and ANN NOWÉ[2]

[1]*Graduate Program in Applied Computing, Universidade do Vale do Rio dos Sinos, São Leopoldo, Brazil*
e-mail: gdoramos@unisinos.br
[2]*Artificial Intelligence Lab, Vrije Universiteit Brussel, Brussels, Belgium*
e-mails: roxana@ai.vub.ac.be, ann.nowe@ai.vub.ac.be
[3]*Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, Brazil*
e-mails: bsilva@inf.ufrgs.br, bazzan@inf.ufrgs.br

Abstract

The problem of traffic congestion incurs numerous social and economical repercussions and has thus become a central issue in every major city in the world. For this work we look at the transportation domain from a multiagent system perspective, where every driver can be seen as an autonomous decision-making agent. We explore how learning approaches can help achieve an efficient outcome, even when agents interact in a competitive environment for sharing common resources. To this end, we consider the route choice problem, where self-interested drivers need to independently learn which routes minimise their expected travel costs. Such a selfish behaviour results in the so-called user equilibrium, which is inefficient from the system's perspective. In order to mitigate the impact of selfishness, we present Toll-based Q-learning (TQ-learning, for short). TQ-learning employs the idea of marginal-cost tolling (MCT), where each driver is charged according to the cost it imposes on others. The use of MCT leads agents to behave in a socially desirable way such that the is attainable. In contrast to previous works, however, our tolling scheme is distributed (i.e., each agent can compute its own toll), is charged *a posteriori* (i.e., at the end of each trip), and is fairer (i.e., agents pay exactly their marginal costs). Additionally, we provide a general formulation of the toll values for univariate, homogeneous polynomial cost functions. We present a theoretical analysis of TQ-learning, proving that it converges to a system-efficient equilibrium (i.e., an equilibrium aligned to the system optimum) in the limit. Furthermore, we perform an extensive empirical evaluation on realistic road networks to support our theoretical findings, showing that TQ-learning indeed converges to the optimum, which translates into a reduction of the congestion levels by 9.1%, on average.

1 Introduction

Efficient urban mobility plays a major role in modern societies. Notwithstanding, the fast-growing demand for mobility associated with the lack of appropriate investments has compromised the efficiency of traffic systems, as evidenced by the increasing number (and intensity) of traffic congestions. In fact, according to the Centre for Economics and Business Research (2014), the cost imposed by traffic congestions on the economy of the USA was around US$ 124 billion in 2013. In the UK, the amount was of US$ 20.5 billion during the same period. Such values correspond to approximately 0.7% of the GDP of these countries. Furthermore, as suggested by the same report, such costs are expected to increase by 50% and 63% until 2030, respectively.

Traditional approaches for dealing with arising traffic congestions include increasing the physical capacity of existing traffic infrastructure. Nonetheless, such approaches have proven unsustainable from many perspectives and may even deteriorate the traffic performance (Braess, 1968). Hence, against this background, ways of making a more efficient use of the existing infrastructure have been increasingly studied.

In this work, we approach traffic from the drivers perspective. In particular, we consider the route choice problem, which models how commuting drivers choose routes to travel from their origins to their destinations everyday. In such scenarios, agents are self-interested and try to minimise some kind of cost (e.g., travel time) associated with their trips. As a result, the expected outcome corresponds to an equilibrium point where no driver benefits from unilaterally changing its route. This is the so-called User Equilibrium (UE) (Wardrop, 1952), which is equivalent to the Nash equilibrium (Nash, 1950).

Although appealing from the drivers' perspective, the UE does not represent the system at its best operation (i.e., when average travel time is minimum). In fact, the average travel time under UE can be considerably higher than the so-called system optimum (SO). However, the SO is only attainable if some agents take sub-optimal routes to improve the system's performance, which is not realistic given that agents are self-interested. Accordingly, the deterioration in the system's performance due to drivers' selfishness is known as the Price of Anarchy (PoA) (Koutsoupias & Papadimitriou, 1999).

In this sense, different approaches have been proposed to align the UE with the SO, including: charging tolls (Cole *et al*., 2003; Bonifaci *et al*., 2011; Sharon *et al*., 2017), computing difference rewards (Wolpert & Tumer, 1999, 2002; Proper & Tumer, 2012; Colby *et al*., 2016), enforcing altruism (Chen & Kempe, 2008; Hoefer & Skopalik, 2009), etc. Among these fronts, charging tolls stand out for their relatively simplicity and for their less restrictive assumptions. One of the most important such schemes was introduced by Pigou (1920) and is known as marginal-cost tolling (MCT), in which each agent is charged proportionally to the cost (e.g., travel time) it imposes on others. By employing MCT, the UE is biased towards the SO in such a way that they both coincide.

In this article, we approach the toll-based route choice problem from the multiagent reinforcement learning (MARL) perspective and provide theoretical guarantees on the agents' convergence to a system-efficient equilibrium (i.e., aligning the UE to the SO). Learning is a fundamental aspect of route choice because drivers must learn independently how to adapt to the changing traffic conditions. In this sense, we introduce *Toll-based Q-learning* (TQ-learning), in which each driver is represented by a Q-learning agent whose objective is to learn which route minimises its expected cost. TQ-learning deploys marginal-cost tolls and, as such, defines the cost of a link as comprising two terms: the travel time and the toll charged on it. Furthermore, TQ-learning introduces a generalised toll formulation that charges an agent *a posteriori* (i.e., only after it has completed its trip) and that can be computed by the agents themselves. In this sense, as compared to existing approaches, our toll formulation is more general (i.e., it applies to most traffic scenarios), it is fairer (i.e., agents pay exactly their marginal costs), and it is easier to deploy (i.e., it has fewer infrastructure requirements). To the best of our knowledge, this is the first time that learning agents are proven to converge to a system-efficient equilibrium without having full knowledge about the reward functions.

In particular, the main contributions of this work can be enumerated as follows:

- We generalise the marginal-cost toll formulation for univariate, homogeneous polynomial cost functions. We show that the proposed formulation comprises the most commonly-used cost functions in the literature, and that it can be computed locally by the agents themselves (i.e., without knowing the overall traffic situation).
- We devise Toll-based Q-learning (TQ-learning), through which each agent can compute the toll it has to pay *a posteriori* (i.e., whenever it finishes a trip) and can use such information to learn the best route to take. We then show that the proposed *a posteriori* tolling scheme is fairer and simpler than *a priori* schemes.
- We provide theoretical results showing that our method converges to the UE in the limit (as opposed to existing works, which assume that the UE is given) and that, by using MCT, the UE corresponds to the SO. Thus, in the limit, the PoA achieves its best ratio. These results are supported by an extensive

experimental evaluation, showing that our method minimises congestions even in large, realistic road networks available in the literature.

The rest of this article is organised as follows. Sections 2 and 3 discuss the background and related work. Our method is introduced in Section 4, theoretically analysed in Section 5, and empirically validated in Section 6. Finally, conclusions, limitations, and future work are presented in Section 7.

2 Background

This section presents the theoretical background upon which we build our work. We begin by formally introducing the route choice problem (Section 2.1) and related problems (Section 2.1.1). We then describe the basics of reinforcement learning (RL), briefly discussing its challenges in multiagent settings (Section 2.2).

2.1 Route choice

An instance of the toll-based route choice problem is defined as a tuple $P = (G, D, f, \tau)$. Let $G = (N, L)$ represent a road network, where the set of nodes N represents intersections and the set of links L represents roads between intersections. Each driver $i \in D$ (with $|D| = d$) has an OD pair, which corresponds to its origin and destination nodes. A trip is made by means of a route[1]

$$R = \{(n_u, n_v) \in L \mid \forall p \in [0, |R| - 1], n_v^p = n_u^{p+1}\}$$

which is a loop-less[2] sequence of links connecting an OD pair. Such a demand for trips generates a flow of vehicles on the links, where x_l is the flow on link $l \in L$. The cost $c_l : x_l \to \mathbb{R}^+$ associated with crossing link $l \in L$ is given by:

$$c_l(x_l) = f_l(x_l) + \tau_l(x_l) \tag{1}$$

where $f_l : x_l \to \mathbb{R}^+$ represents its travel time and $\tau_l : x_l \to \mathbb{R}^+$ denotes the toll charged for using it. In order to enhance presentation, hereafter we leave x_l implicit and use simply c_l, f_l, and τ_l to represent the cost, travel time, and toll on link l, respectively. The cost of a route R is then computed as:

$$C_R = \sum_{l \in R} c_l \tag{2}$$

Travel times f_l are typically abstracted as volume-delay functions (VDF), which map a flow of vehicles into a travel time (a.k.a. latency). The toll values τ_l, on the other hand, should be defined according to a specific purpose, for example, maximising revenue, minimising link usage, etc. We refer the reader to Ortúzar & Willumsen (2011) for a more detailed overview.

Toll values can be defined according to different objectives. In this work, we consider the case of biasing the user equilibrium (UE) towards the system optimum (SO). According to Pigou (1920), this can be achieved by means of marginal cost tolling (MCT), under which each agent is charged proportionally to the cost it imposes on others. Specifically, the marginal cost toll on link l is the product of its flow and the derivative of its VDF function with respect to the current flow x_l on that link (Beckmann *et al.*, 1956; Pigou, 1920), that is,

$$\tau_l = x_l \cdot (f_l(x_l))' \tag{3}$$

It should be noted, on the other hand, that charging tolls arbitrarily (e.g., charging a constant price on selected links) does not necessarily lead to the SO (Beckmann *et al.*, 1956).

As usual in the literature, we approach route choice using a macroscopic traffic model. This kind of model represents dynamic aspects of traffic as aggregated quantities, such as flow and travel time

[1] We abuse notation here and use n_u^p (n_v^p) to denote the start (end) node of the p^{th} link of route R.

[2] As discussed forward, links' costs are represented by positive reals. Since for every route with a cycle we can obtain an equivalent sequence without cycles, then cycles can be ignored without loss of generality.

(Bazzan & Klügl, 2013). Here, a link's flow can be seen as the total number of vehicles whose route includes the considered link, regardless of when the vehicles effectively traverse that link. This means that costs can be efficiently computed altogether (e.g., when all drivers reach their destinations), which makes the model fast to calculate and run.

2.1.1 *Related problems*

We highlight that the route choice problem shares some similarities with other problems. Next, we describe the most representative ones (at least for our purpose). The interested reader is referred to Ramos (2018) for a more detailed overview.

In traffic engineering, route choice is mainly approached from two perspectives. Discrete choice models try to accurately approximate the behaviour of human travellers (McFadden, 2001). Assignment methods are centralised mechanisms employed to find an allocation of vehicles into routes so as to satisfy a given solution concept (Bar-Gera, 2010; Ramos & Bazzan, 2015, 2016). In common, these works propose centralised approaches to facilitate the work of traffic managers on analysing different traffic patterns, policies, etc. On the other hand, route choice provides a decentralised, driver-centered approach, which allows one to investigate how self-interested drivers learn (with limited knowledge) and adapt to each other while trying to maximise their rewards.

In the game theory literature, route choice has also been approached using congestion games (Rosenthal, 1973) and, in particular, (selfish) routing games (Roughgarden, 2005). These games, however, assume that drivers control a negligible, infinitesimally small amount of traffic, whereas in route choice the flow of vehicles is fundamentally discrete. Again, our approach models traffic from the drivers' perspective, allowing one to precisely investigate how drivers interact while learning and adapting to each other.

Multi-armed bandits (Robbins, 1952) can also be used to model route choice (Auer *et al.*, 2002; Awerbuch & Kleinberg, 2004). In this problem, at each round, the agent selects one among $K \in \mathbb{N}$ available arms (routes) and the environment returns a payoff (negative cost) for the selected arm. This payoff is sampled from a distribution that is unknown to the agent. The agent then needs to decide on which arms to play (and in which order) to maximise its cumulative reward. Despite their similarities, multi-armed bandits and route choice are conceptually different. Whereas in the former the rewards are simply random variables, in the latter they are a function of the choices made by *all* drivers. Such a dependence on what everyone else is doing poses an additional layer of complexity to an agent's decision process, thus making route choice more challenging.

2.2 *Reinforcement learning*

In RL, an agent learns by trial and error how to behave within an environment (Sutton & Barto, 1998). The basic RL cycle can be described as follows. Initially, an RL agent observes the current state of the environment and chooses an action based on its knowledge. Afterwards, the agent executes the chosen action and receives a reward, which is then used to update its knowledge base. An agent's knowledge here refers to its *policy*, that is, a mapping from states to actions. A complete RL cycle is called an episode.

The RL problem is typically formulated as a Markov decision process (MDP), which consists in a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, r)$, where $\mathcal{S}$ represents the set of environment states, $\mathcal{A}$ represents the set of actions, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ defines the transition function, and $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ denotes the reward function. In route choice, drivers know their routes *a priori* (or at least a subset of them) and just need to decide on which one to take everyday. In this sense, an agent's actions represent the possible routes between its origin and destination. The reward[3] for taking action $a \in \mathcal{A}$ can then be denoted as:

$$r(a) = -C_R \tag{4}$$

[3] Observe that although the reward an agent receives is formulated as a function of its route, it actually depends on the flow of vehicles on the links that comprise that route. This is expressed by means of the VDF function, as explained in Section 2.1, whose value is a function of the *flow* of vehicles on its links. Furthermore, since we assume a macroscopic traffic model, we remark that all routes' costs can be computed together, when all drivers complete their trips.

with $a = R$. Whenever a driver takes a route, it will inevitably reach its destination, thus rendering the *state* definition irrelevant here. Thus, in the context of RL, the route choice problem is typically modelled as a stateless MDP.

Solving a stateless MDP involves finding a policy π (e.g., which route to take) that maximises the agent's average reward. To learn such a policy, the agent needs to repeatedly interact with the environment so as to learn its dynamics. A particularly suitable algorithm for this purpose is Q-learning (Watkins & Dayan, 1992), whose stateless version learns the expected return *Q(a)* of selecting each action a while balancing exploration (gain of knowledge) and exploitation (use of knowledge). In particular, after taking action a and receiving reward *r(a)*, the stateless Q-learning algorithm updates *Q(a)* as:

$$Q(a) = (1 - \alpha)Q(a) + \alpha r(a) \tag{5}$$

where the learning rate $\alpha \in (0, 1]$ weights how much of the previous estimate should be retained. As for exploration, a typical strategy is ϵ-greedy, in which the agent chooses a random action with probability ϵ or the best action otherwise. The Q-learning algorithm is guaranteed to converge to an optimal policy if all state-action pairs are experienced an infinite number of times (Watkins & Dayan, 1992).

Although Q-learning is guaranteed to converge to an optimal policy in the single-agent case, it has no guarantees in multiagent settings. In fact, no convergence guarantees exist for the *general* multiagent RL setting (i.e., for an arbitrary number of players and actions). The point is that, when multiple agents need to learn their policies simultaneously in a shared environment, their actions may affect the reward received by others. In other words, an agent's best policy may change as other agents change their own policies. Formally, we say that this kind of situation invalidates the so-called Markov property, thus rendering the environment no longer stationary (Tuyls & Weiss, 2012; Buşoniu *et al.*, 2008; Laurent *et al.*, 2011). In practical terms, we can say that learning an optimal policy becomes a moving target. In spite of the aforementioned challenges, interesting progress has shown possible when more specific (rather than general) multiagent RL scenarios are considered, as discussed next.

Multiagent reinforcement learning (MARL) problems may be approached from different perspectives. Stochastic (or Markov) games (Littman, 1994) represent a straightforward approach, where agents' decisions are represented in a joint action space $\mathcal{A} = A_1 \times A_2 \times \ldots \times A_d$, where A_i denotes the actions available to agent $i \in \{1, \ldots, d\}$. Several algorithms have been proposed in this context for 2-player zero-sum games (Littman, 1994), 2-player general-sum games (Hu & Wellman, 1998, 2003), and coordination games (Littman, 2001; Verbeeck *et al.*, 2007; Vrancx *et al.*, 2010). However, route choice usually involves several (not only two) agents, which rarely cooperate (Sandholm, 2007). Gradient ascent algorithms were also proposed to handle multiagent learning scenarios (Zinkevich, 2003; Bowling, 2005; Abdallah & Lesser, 2006). Nonetheless, their convergence guarantees still only apply to 2-player games.

Another common approach to deal with MARL is to model agents as *independent learners* (Claus & Boutilier, 1998). In this kind of approach, each agent has its own stateless MDP and interprets the behaviour of other agents as the dynamics underlying its environment. In spite of its simplicity, independent learners have obtained promising results (Tan, 1993; Sen *et al.*, 1994; Boyan & Littman, 1994; Tesauro, 1994; Crites & Barto, 1998; Lauer & Riedmiller, 2004; Vrancx *et al.*, 2008; Kaisers & Tuyls, 2010; Matignon *et al.*, 2012; Foerster *et al.*, 2017; Lanctot *et al.*, 2017; Omidshafiei *et al.*, 2017; Ramos *et al.*, 2017). Moreover, we highlight that this approach is particularly suitable in the context of traffic (Bazzan, 2009), given that drivers have a very limited knowledge about what others are doing (not to mention their policies).

Hence, in this work, we follow this direction and model agents as *independent Q-learners*. In particular, we advance the state-of-the-art by introducing Toll-based Q-learning (which allows agents to compute – and pay – tolls for the routes they take), which is guaranteed to converge to a UE that is aligned to the SO. Our algorithm deals with the non-stationarity inherent to MARL by systematically decreasing the learning and exploration rates until agents converge to a fixed point corresponding to a system-efficient equilibrium. We then prove (see Section 5) that, using our algorithm, Q-values converge to their true values in the limit and that exploration does not destabilise the equilibrium.

3 Related work

In this section, we discuss representative literature on system-efficient equilibria in route choice (and related problems). In order to enhance presentation, we categorise existing works into: toll-based approaches (Section 3.1), difference rewards based approaches (Section 3.2), and other approaches (Section 3.3). For a more detailed overview, the interested reader is referred to the works of van Essen *et al*. (2016), Ramos (2018), and Ortúzar & Willumsen (2011).

3.1 Toll-based approaches

The use of tolls to enforce system-efficient behaviour has been widely explored in the literature. There is a plethora of works in this line, considering drivers with heterogeneous utility preferences (Cole *et al*., 2003; Fleischer *et al*., 2004; Meir & Parkes, 2018), toll information mechanisms (Kobayashi & Do, 2005), tolls with bounded values (Bonifaci *et al*., 2011), RL-based tolls (Tavares & Bazzan, 2014; Ramos *et al*., 2018; Chen *et al*., 2018), and so on. We concentrate, however, in the marginal-cost tolling (MCT) scheme (Pigou, 1920). The concept of MCT has been investigated in several works, such as those by Sharon *et al*. (2017), Mirzaei *et al*. (2018), Sharon *et al*. (2019), Ye *et al*. (2015), Yang *et al*. (2004), and Meir & Parkes (2016). As opposed to these works, nonetheless, here we approach the problem from the drivers perspective and investigate how they behave when facing tolls. In particular, we introduce a learning procedure that allows drivers to compute their own toll values (using local information) and to learn the best route to take based on that information. In this sense, we go beyond existing works and show that, using our approach, self-interested drivers converge to a system-efficient equilibrium.

Another drawback of existing tolling schemes is that they charge tolls *a priori*, that is, before agents start their trips. Ideally, however, tolls should only be charged after their real marginal costs are available (i.e., at the end of the trips). *A priori* tolling is indeed appealing from the agents' perspective, since such agents can see in advance the toll associated with each of their possible actions. Nonetheless, these schemes usually define toll values based on historical congestion levels, meaning that the agents may end up paying a toll that is higher than their actual marginal costs. In particular, since MCT is based on the impact an agent causes on others, one cannot assess such impact before it happens (except if one can predict drivers decisions along their trips). Hence, we say that these schemes are unfair (see discussion in Section 5.3). In this work, by contrast, we assume that tolls are charged *a posteriori* and *per route*. We then present a general toll formulation that can be computed directly by the agents.

We highlight that our toll formulation can simplify the infrastructure requirements for deploying MCT schemes by assuming that each vehicle has a navigation device responsible for charging the toll whenever a trip is finished. Additionally, our modelling makes the drivers' decision process easier since they can better understand the costs being charged, as pointed out by the National Surface Transportation Infrastructure Financing Commission (2009). Traditional tolling schemes could also benefit from connected navigation devices. However, such approaches would strongly depend on stable communication (otherwise tolls would not be available *a priori*), whereas our approach remains robust even under precarious communication conditions (since tolls could be computed at any time once the corresponding trip is finished).

3.2 Difference rewards based approaches

Wolpert & Tumer (1999, 2002) introduced the idea of *difference rewards*, which also relates to our approach. In a stateless context, the difference reward that agent i receives after taking action a_i is given by $D_i(a_i) = G(\mathbf{a}) - G(\mathbf{a}_{-i})$, where $\mathbf{a}$ is the joint action of all agents, $\mathbf{a}_{-i}$ is the joint action without action a_i, and $G(\cdot)$ is the global reward signal (which, in traffic scenarios, could represent the system's average travel time). In other words, the difference reward an agent receives can be seen as the amount that the system's performance deteriorates due to its individual action. By using difference rewards, agents' reward signal is aligned with the system's utility, which enforces agents to converge to the SO.

Notwithstanding, difference rewards can only be computed by a central authority with full observability, or by assuming that agents can observe/compute function G. Such assumptions, however, tend to be unrealistic, especially in traffic scenarios.

Methods for approximating the difference reward signals were also proposed (Colby *et al*., 2016; Agogino & Tumer, 2004). Nonetheless, these approaches still depend on some sort of global information (e.g., the value of $G(\mathbf{a})$) and take too long to converge (e.g., hundreds of thousands of episodes even for small, competitive scenarios). In contrast, our approach drops any full observability assumption and can be run distributedly by the agents. Furthermore, by using only local information, our approach converges much faster to the optimum.

3.3 Other approaches

Similarly to charging tolls, some works investigated the SO by explicitly assuming that agents behave altruistically. Chen & Kempe (2008) and Hoefer & Skopalik (2009) investigated altruism in routing games. Levy & Ben-Elia (2016) developed an agent-based model where drivers choose routes based on subjective estimates over their costs. However, whereas tolls can be imposed on agents, altruistic behaviour cannot be assumed or made mandatory (Fehr & Fischbacher, 2003). Furthermore, these works assume that agents know each others' payoff to compute their utilities.

Route guidance mechanisms have also been employed to approximate the SO. These include mechanisms for: negotiating traffic assignment at the intersection level (Lujak *et al*., 2015), biasing trip suggestions (Bazzan & Klügl, 2005), allocating routes into abstract groups that offer more informative cost functions (Malialis *et al*., 2016; Rădulescu *et al*., 2017), etc. Notwithstanding, in general, these works assume the existence of a centralised mechanism.

4 Toll-based Q-learning

This section introduces Toll-based Q-learning (TQ-learning, for short), an RL algorithm through which agents can compute the tolls associated with their routes *using only local information* and use such values to learn their best routes. Specifically, we model the toll-based route choice problem as a stateless Markov Decision Process (MDP) and represent drivers by means of Q-learning agents. At every episode, each agent chooses a route from its origin to its destination and, once the trip is completed, the agent observes its travel time[4]. As for the tolls, we propose a general tolling scheme that allows the agents to compute the tolls by themselves, using only local information (Section 4.1). Together, the travel time and toll value experienced by an agent in a given route compose the cost of that route. Each agent then uses such cost to update the Q-value for the corresponding route. The complete algorithm is presented in Section 4.2.

4.1 Tolling scheme

Our generalised tolling scheme assumes that each agent can observe its travel time and compute its toll *a posteriori*. In practical terms, this is equivalent to coupling each driver with a mobile navigation device, which computes and provides such information (de Palma & Lindsey, 2011). We remark that, by definition, travel times and tolls are defined per link, whereas agents' decisions are based on routes. In this sense, hereafter we refer to a route's travel time (and toll value) as the sum of its links' travel times (and toll values).

Toll values are defined according to the marginal cost of the agents, as defined in Equation (3). Recall that such cost is obtained through the derivative of the link's cost, which depends on the VDF being employed. Sharon *et al*. (2017) have shown that, for the BPR (1964) function, the marginal cost toll can be written as $\tau_l = \beta(f_l - F_l)$, where f_l and F_l represent the *actual* (i.e., as given by the VDF function) and *free flow* (i.e., the lower bound when $x_l = 0$) travel times on link l, and β represents a VDF-specific constant.

[4] We remark that, using a macroscopic traffic model (see Section 2), travel times are computed whenever all drivers complete their trips.

Nonetheless, given that different VDFs are available in the literature, we go beyond and generalise the toll formulation according to the following proposition.

PROPOSITION 1. *The marginal-cost toll value τ_l on any link l with a univariate, homogeneous polynomial VDF function is $\beta(p_1 x_l^\beta)$, where β and p_1 represent VDF-specific constants.*

Proof. First we analyse the case of linear and polynomial functions. Then, we define the general MCT formulation.

Linear functions are in the form $f_l(x_l) = p_1 x_l + p_0$. We consider two such examples from the literature. The OW function (Ortúzar & Willumsen, 2011) is represented as $f_l(x_l) = F_l + 0.02 x_l = p_1 x_l + p_0$, with $p_0 = F_l$ and $p_1 = 0.02$ representing VDF-specific constants. The linear Braess functions can be represented as $f_l(x_l) = \left(\frac{kci_l}{d}\right) x_l = p_1 x_l + p_0$, with $p_0 = 0$ and $p_1 = \frac{kci_l}{d}$ representing VDF-specific constants (Stefanello & Bazzan, 2016).

Polynomial functions can be defined as $f_l(x_l) = \sum_{\beta=0}^{n} p_\beta x^\beta$. In this work, we consider the specific case of univariate (single variable), homogeneous (all terms with the same degree) polynomial functions, which can be written in the simpler form $f_l(x_l) = p_1 x_l^\beta + p_0$. Such a subclass of polynomial functions includes VDFs that are well-known in the transportation literature, such as the one by the Bureau of Public Roads (1964). The so-called BPR function is represented as $f_l(x_l) = F_l \left(1 + \alpha \frac{x_l}{C_l}^\beta\right) = F_l + x_l^\beta \left(\frac{\alpha F_l}{C_l^\beta}\right) = p_1 x_l^\beta + p_0$, with $p_0 = F_l$ and $p_1 = \frac{\alpha F_l}{C_l^\beta}$ representing VDF-specific constants. Note that this polynomial definition generalises over linear and constant functions. Specifically, linear functions correspond to the special case where $\beta = 1$ and constant functions correspond to the special case where $p_1 = 0$.

The MCT of link l is defined as $\tau_l = x_l \cdot (f_l(x_l))'$. By using the definition of univariate, homogeneous polynomial functions above, we have that

$$\begin{aligned} \tau_l &= x_l \left(p_1 x_l^\beta + p_0\right)' \\ &= x_l \left(p_1 \beta x_l^{\beta-1}\right) \\ &= \beta\left(p_1 x_l x_l^{\beta-1}\right) \\ &= \beta\left(p_1 x_l^{\beta-1+1}\right) \\ &= \beta\left(p_1 x_l^\beta\right) \end{aligned} \tag{6}$$

which completes the proof. □

We emphasise that Proposition 1 only holds when the VDF is defined as an univariate (i.e., with a single parameter, such as flow), homogeneous (i.e., all terms with the same degree) polynomial. It should be noted, however, that this assumption is not unrealistic, given that the most commonly-used VDF functions in the literature are in this class. Moreover, the above proposition can be extended to overcome these limitations. Such an extension is left as future work.

From Proposition 1, observe that computing toll values requires some parameters, such as the flow of vehicles. Recall that this information may not be directly available to the agents. Fortunately, however, such information can be obtained by means of the agents' travel times. In this regard, we can combine Proposition 1 with the formulation of Sharon *et al.* (2017), thus obtaining the following corollary.

COROLLARY 1. *The toll value on link l can be rewritten as $\tau_l = \beta(p_1 x_l^\beta) = \beta(p_1 x_l^\beta + p_0 - p_0) = \beta(f_l - F_l)$, considering $F_l = p_0$ and $f_l(x_l) = p_1 x_l^\beta + p_0$. In other words, whenever an agent finishes its trip (i.e.,* a posteriori*), it can compute the toll on the corresponding route based on its actual and free flow travel times.*

As seen, agents can compute the tolls associated with their routes knowing neither the reward of all routes nor the actions taken by the other agents. Having defined the toll values, we can rewrite the

Algorithm 1 Toll-based Q-learning

input: D; A; λ; μ; T; β and F_l (for every link $l \in L$)
$Q(a_i) \leftarrow 0\ \forall i \in D, \forall a_i \in A_i$; `// initialise agents' Q-tables`
for $t \in T$ **do**
  $\alpha \leftarrow \lambda^t$; $\epsilon \leftarrow \mu^t$; `// update learning and exploration rates`
  for $i \in D$ **do**
    $a_i^t \leftarrow$ select action (route) using ϵ-greedy;
  end
  $f \leftarrow$ compute travel time of all links and routes;
  for $i \in D$ **do**
    $f_{a_i^t} \leftarrow$ observe travel time on route a_i^t;
    $\tau_{a_i^t} \leftarrow \beta(f_{a_i^t} - F_{a_i^t})$; `//compute` i`'s toll`
    $r(a_i^t) \leftarrow -(f_{a_i^t} + \tau_{a_i^t})$; `//compute` i`'s reward`
    $Q(a_i^t) \leftarrow (1-\alpha)Q(a_i^t) + \alpha r(a_i^t)$; `//update` i`'s Q-table`
  end
end

routes reward function as in Equation (7), which follows from Proposition 1 and Equations (1), (2), and (4).

$$\begin{aligned} r(a_i^t) &= -C_{a_i^t} \\ &= -\textstyle\sum_{l \in a_i^t} c_l \\ &= -\textstyle\sum_{l \in a_i^t} f_l + \tau_l \\ &= -\textstyle\sum_{l \in a_i^t} f_l + \beta(f_l - F_l) \\ &= -\left(f_{a_i^t} + \beta\left(f_{a_i^t} - F_{a_i^t}\right)\right) \end{aligned} \tag{7}$$

We remark that, by using Equation (7), agents can compute their tolls using only local information. This is an important distinguishing feature of TQ-learning as compared to difference rewards (discussed in Section 3), since it eliminates the need for having a central authority with full knowledge about the complete traffic state. This is particularly suitable in traffic settings, where such information is hardly available.

4.2 Learning process

We can now discuss the learning process in detail. Again, the problem is represented as a stateless MDP and each driver $i \in D$ as a Q-learning agent. The set of routes of agent i is denoted by $A_i = \{a_1, \ldots, a_K\}$. The reward $r(a_i^t)$ that agent i receives for taking route a_i^t at episode t is given by Equation (7). The drivers' objective is to maximise their cumulative reward. An overview of TQ-learning is presented in Algorithm 1.

The learning process is described as follows. At every episode $t \in [1, T]$, each agent $i \in D$ chooses an action $a_i^t \in A_i$ using an ϵ-greedy exploration strategy. The exploration rate ϵ at episode t is given by $\epsilon(t) = \mu^t$. After taking the chosen action, the agent observes its travel time $f_{a_i^t}$ and computes its reward $r(a_i^t)$ following Equation (7). Note that, by computing the toll only after the agent observes its travel time, we ensure that our mechanism charges tolls *a posteriori*. Finally, the agent updates $Q(a_i^t)$ as in Equation (5). The learning rate α at episode t is given by $\alpha(t) = \lambda^t$.

We highlight that, as opposed to the traditional Q-learning algorithm (Watkins & Dayan 1992), our approach computes the toll value for its route and uses that information as part of the reward definition. By using this information to guide the learning process, we ensure that agents' optimal choices are aligned with the social welfare, which promotes convergence to a system efficient equilibrium. In the next section, we prove these points by means of a theoretical analysis.

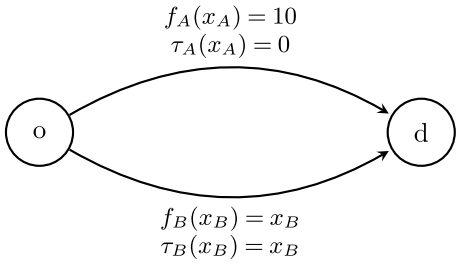

Figure 1 Example network adapted from Pigou (1920), with two routes and 10 agents

5 Theoretical analysis

In this section, we provide a theoretical analysis of our approach. The aim is to show that TQ-learning converges to a system-efficient equilibrium, that is, a UE whose average travel time equals the SO. We begin with the main results of our analyses, namely that TQ-learning converges to a system-efficient equilibrium (Section 5.1). This result is made possible by combining a canonical analysis by Beckmann *et al.* (1956) with the fact that TQ-learning converges to an equilibrium (Section 5.2). Finally, in Section 5.3, we also discuss the advantages of charging tolls *a posteriori* rather than *a priori*.

5.1 Main results

The next theorem, adapted from Beckmann *et al.* (1956), states that the use of MCT is enough to align the UE to the SO, thus obtaining a system-efficient equilibrium.

THEOREM 1 (Beckmann *et al.* (1956)). *Consider a toll-based instance* $P' = (G, D, f, \tau)$ *of the route choice problem, where the cost experienced by any driver* $i \in D$ *after traversing link* l *is given by* $c_l = f_l + \tau_l$*, with* f_l *and* τ_l *representing the travel time and marginal-cost toll charged at that link, respectively. Under these settings, the average travel time under UE for* P' *equals that of the SO for* $P = (G, D, f)$.

Intuitively, Theorem 1 says that given an instance P of the route choice problem, if we apply MCT to it – thus obtaining an instance P' of the toll-based route choice problem – then the UE in P' will be equivalent to the SO in P. In other words, the UE with MCT achieves the same average travel time of the SO of the original problem. We refer the reader to Beckmann *et al.* (1956) for the complete proof. An illustrative example on how this theorem applies to Pigou (1920)'s network is presented in the next example.

Example 1. Consider the network in Figure 1, adapted from Pigou (1920), which is traversed by 10 agents. To traverse the network, each agent must take one out of two possible routes, A or B, whose travel times are given by $f_A(x_A) = 10.0$ and $f_B(x_B) = x_B$, respectively.

By definition, the UE in this network is achieved when all vehicles choose route B, which results in an average travel time of 10.0. The SO, on the other hand, corresponds to the case where each route receives half of the flow, which results in an average travel time of 7.5. Here, the price of anarchy (PoA) is $4/3$.

Now consider the same example, but adopting the MCT scheme. The cost on each link now corresponds to the sum of its travel time (as before) and the toll charged on it, that is, $c_l = f_l + \tau_l$. Specifically, for routes A and B we have that $c_A = 10.0 + 0.0 = 10.0$ and $c_B = x_B + x_B = 2x_B$, respectively. In this case, the UE is achieved when each route receives half of the drivers, which corresponds to an average cost of 10.0 and an average travel time of 7.5. This is precisely the SO. Hence, under MCT, we have that SO = UE and that PoA is 1.

Note that Theorem 1 is about the equivalence of SO and UE under MCT. However, *it does not consider how the UE can be achieved*. In other words, Theorem 1 simply assumes that the UE is given. Indeed, this is a common assumption of other works in the literature, such as in Sharon *et al.* (2017). However, since route choice is a multiagent problem, guaranteeing convergence to the UE is not trivial (as discussed

in Section 2.2). Therefore, in order for Theorem 1 to apply to our approach, we need first to show that TQ-learning indeed achieves the UE. In contrast to other works in the literature, we show that TQ-learning *converges to the UE*, and then we show that such UE is aligned to the SO. This is shown in Theorem 2. The complete proof is presented in Section 5.2.

THEOREM 2. *Consider an instance P of the route choice problem. If all agents use Toll-based Q-learning, then the system converges to the user equilibrium in the limit.*

From Theorem 2, we can conclude that our algorithm can find the UE both in the original problem (P) as well as in the corresponding toll-based version (P'). This means that, by employing MCT, TQ-learning achieves a system-efficient equilibrium (Theorem 1). In other words, TQ-learning reduces the PoA to its best ratio of 1. Therefore, based on Theorems 1 and 2 we can formulate the following corollary.

COROLLARY 2. *Consider an instance P of the route choice problem, where all drivers use Toll-based Q-learning. Since tolls are based on marginal costs, the agents converge to a system-efficient equilibrium in the limit, that is, the user equilibrium is aligned to the system optimum. Thus, the price of anarchy converges to 1 in the limit.*

5.2 Convergence to the UE

In this section, we prove Theorem 2 by showing that TQ-learning converges to the UE. For simplicity and without loss of generality, we assume that the actions' rewards are in the interval [0, 1].

The intuition underlying the proof of Theorem 2 is that, given that learning (α) and exploration (ϵ) rates are decreasing with time (using decays λ and μ, respectively), then the system is becoming more stable (Theorem 3). We say that the environment is stabilising if randomness (due to agents exploration) is decreasing along time. Consequently, we can show that, in the limit, the actions with the highest Q-values are precisely the optimal ones (Lemma 3), which leads the agents to exploit only optimal actions in the limit (Lemma 2), thus achieving the UE (Theorem 2).

Initially, the next proposition defines the probability that best[5] and non-best actions are chosen by a given agent i at episode t.

PROPOSITION 2. *Using ϵ-greedy exploration with $\epsilon(t) = \mu^t$, at episode t agent i chooses its best action $\overset{+}{a}{}_i^t = \arg\max_{a_i^t \in A_i} Q(a_i^t)$ with probability*[6] $\rho(\overset{+}{a}{}_i^t) = 1 - \frac{\mu^t(K-1)}{K}$ *and any other action $\bar{a}_i^t \in A_i \setminus \overset{+}{a}{}_i^t$ with probability $\rho(\bar{a}_i^t) = \frac{\mu^t(K-1)}{K}$.*

Proof. In a given episode t, by definition, ϵ-greedy *exploits* the best action $\overset{+}{a}{}_i^t = \arg\max_{a_i^t \in A_i} Q(a_i^t)$ with probability $1 - \epsilon$ or *explores* any action $\bar{a}_i^t \in A_i$ with probability ϵ. Observe that the best action can also be selected under exploration. In this sense, the best action is selected with probability $(1 - \epsilon) + \frac{\epsilon}{K}$. A non-best action (i.e., ignoring the best action), on the other hand, is selected with probability $\epsilon - \frac{\epsilon}{K}$. Now, considering that the value of ϵ at episode t is given by μ^t, we can rewrite the probability of agent i selecting the best action at that given episode as:

$$\begin{aligned}\rho(\overset{+}{a}{}_i^t) &= (1 - \mu^t) + \frac{\mu^t}{K} \\ &= 1 + \frac{\mu^t - K\mu^t}{K} \\ &= 1 - \frac{\mu^t(K-1)}{K}\end{aligned}$$

[5] Hereafter, we refer to the action with highest Q-value as the *best action* and to the other actions as *non-best*. Observe that the best action is not necessarily optimal.

[6] In order to improve presentation, whenever it is clear from the context, we refer to $\rho(\overset{+}{a}{}_i^t)$ and $\rho(\bar{a}_i^t)$ as $\overset{+}{\rho}{}_i^t$ and $\bar{\rho}_i^t$, respectively. Whenever possible, we also omit t and i.

Similarly, we can rewrite the probability of agent i selecting any non-best action at a given episode t as follows:

$$\begin{aligned}\rho(\bar{a}_i^t) &= \mu^t - \frac{\mu^t}{K} \\ &= \frac{K\mu^t - \mu^t}{K} \\ &= \frac{\mu^t(K-1)}{K}\end{aligned}$$

□

From Proposition 2, observe that $\overset{+}{\rho} \to 1$ and $\bar{\rho} \to 0$ as $t \to \infty$ and $\epsilon \to 0$. To this respect, as time goes to infinity, the values of α and ϵ become so small that the probability of noisy observations changing the Q-table (and, mainly, the best action) goes to zero. When the system behaves in this way, we say it is *stabilising*. Under such circumstances, we can apply Theorem 3, adapted from Ramos *et al*. (2017) (we refer the reader to their work for a complete proof).

THEOREM 3 (Ramos et al. (2017)). *The environment is stabilising as $t \to \infty$. In this scenario, the probability that the Q-values of best actions (of any agent) become non-best after ∇ agents decide to explore a non-best action is bounded by $O(\bar{\rho}^{\nabla}(\overset{+}{\rho} + \bar{\rho}))$, which goes to zero as $t \to \infty$.*

Observe that an agent can, eventually, change its best action given that it *is* learning. However, the agent should be able to prevent its Q-values from reflecting unrealistic observations. Of course, stability does *not* imply that the Q-value estimates are correct and that the agents are under UE. These are shown to be true, however, in Lemma 3 and Theorem 2, respectively.

We can now advance to the main part of the proofs and show that, in the limit, the action with highest estimated Q-value is indeed the optimal action. In this regard, we firstly characterise the agent's behaviour in terms of the UE, as shown in the next lemma.

LEMMA 1. *Under user equilibrium, every agent $i \in D$ using ϵ-greedy exploration exploits its best route $\overset{+}{a}_i = \arg\max_{a_i \in A_i} Q(a_i)$.*

Proof. By definition, under UE, for each pair of routes a' and a'' of the same OD pair, with $x_{a'} > 0$, we have that $r(a') \geq r(a'')$. For the sake of contradiction, assume that the system is under UE and that there exists a pair of routes a' and a'' belonging to the same OD pair for which $x_{a'} > 0$ but $r(a') < r(a'')$. Recall that we model the problem as a stateless MDP and agents as Q-learners with ϵ-greedy exploration. Consequently, Q-values can be seen as estimates of the reward values of their corresponding actions. Therefore, given that the reward on a' is lower than on a'', then all the $x_{a'}$ vehicles using a' would deviate to a'' (i.e., they would exploit a'', not a') as soon as their Q-values are correct (which is the case in the limit, as shown next in Lemma 3). This contradicts the initial assumption, which completes the proof. □

Observe that, in the UE definition, the notion of *best* refers to the value associated with each action (route). In RL-settings, these values correspond to actions' Q-values. Therefore, now we need to show that agents actually choose actions with highest estimated Q-values and that such actions are *indeed the optimal ones*. These are shown in Lemmas 2 and 3, respectively.

LEMMA 2. *In the limit, agents exploit their knowledge most of the time, that is, they tend to choose the actions with highest estimated Q-values.*

Proof. This lemma follows directly from Proposition 2 and Theorem 3, since $\overset{+}{\rho}_i^t \to 1$ and $\bar{\rho}_i^t \to 0$ as $t \to \infty$ and $\epsilon \to 0$. □

LEMMA 3. *In the limit, the action with highest estimated Q-value $\overset{+}{a}_i = \arg\max_{a_i \in A_i} Q(a_i)$ is indeed the optimal action $\overset{*}{a}_i = \arg\max_{a_i \in A_i} r(a_i)$, that is, $\overset{+}{a}_i = \overset{*}{a}_i$ as $t \to \infty$.*

Proof. This lemma can be proved by contradiction. Assume that agent i has an action $\overset{+}{a}_i = \arg\max_{a_i \in A_i} Q(a_i)$ with highest estimated Q-value but that this action is not optimal, that is, $\overset{+}{a}_i \neq \overset{*}{a}_i = \arg\max_{a_i \in A_i} r(a_i)$. In order for that be possible, we need that $r(\overset{+}{a}_i) < r(\overset{*}{a}_i)$ and $Q(\overset{+}{a}_i) > Q(\overset{*}{a}_i)$ hold at the same time. Although counter-intuitive, this behaviour often occurs in the initial episodes, given that the agents' learning process leads travel times to oscillate. In this case, some Q-values may not correspond to the most accurate reward estimate of an action. However, due to exploration, agent i will eventually take route $\overset{*}{a}_i$. In fact, in the limit all actions will be infinitely explored. Thus, as $t \to \infty$, we have that $Q(\overset{*}{a}_i)$ will increase until it eventually becomes the highest one, that is, $Q(\overset{*}{a}_i) \approx r(\overset{*}{a}_i) > Q(\overset{+}{a}_i) \approx r(\overset{+}{a}_i)$, which contradicts the initial assumption. □

Observe that, as agents are adapting to each other, it is possible that an agent's optimal action seem no longer optimal due to other agents' behaviour. Nevertheless, from Theorem 3 and Lemmas 2 and 3, we have that such agent will keep exploring (non-best actions) as well. Hence, this agent will eventually take its true optimal action and update its perception accordingly, so that the optimal action has the highest Q-value again.

We remark that one of the key requirements of Q-learning is that each action should be infinitely explored. However, such *exploration should not lead optimal actions to seem sub-optimal*. This is shown in the next lemma.

LEMMA 4. *In the limit, agents using ϵ-greedy exploration with $\epsilon(t) = \mu^t$ can still explore non-best actions without invalidating the UE, that is, exploration does not destabilise the equilibrium.*

Proof. Suppose that the system has converged to the UE in the limit (after a sufficiently large number of episodes). At this point, all agents are using their best actions, that is, the ones with highest estimated Q-values (Lemmas 1 and 2). Observe that agents can still explore other actions, though less frequently (Proposition 2 and Lemma 2). Thus, in order to prove this lemma, one needs to show that, under UE, exploration will not generate an *abrupt* change in the Q-values. An abrupt change occurs in an agent's Q-table only if it receives a reward that leads the Q-value of a non-best action to become better than that of the best one. However, from Theorem 3, we have that such abrupt changes will not affect the UE and that even if they do, a little amount of additional exploration is enough to lead the Q-values back to their true values (Lemma 3). □

We now have the required tools for proving Theorem 2. Recall that our final objective is to show that our approach converges to a system-efficient equilibria (i.e., the SO) as soon as MCT is employed. From Theorem 1, this is only attainable if TQ-learning is guaranteed to converge to the UE. Therefore, proving Theorem 2 is sufficient to show that, by employing MCT, TQ-learning converges to the SO.

Proof of Theorem 2. According to Theorem 3, the system becomes stable in the limit and abrupt changes do not affect the Q-values (i.e., non-best actions cannot become the best ones). Moreover, from Lemma 2, we know that in the limit all agents keep exploiting most of the time. Remember that exploiting means choosing the action with the highest estimated Q-value, which in the limit corresponds to the optimal one, according to Lemma 3. Finally, from Lemma 4 we have that exploration does not affect the UE. Therefore, TQ-learning can be said to converge to the UE. □

5.3 Fairness

In this section, we analyse the fairness of our approach. We begin with a more precise definition of fairness, which is given as follows.

DEFINITION 1 (MCT fairness). *A marginal-cost tolling scheme is fair if the agents are charged exactly their marginal costs (i.e., the cost they impose on others).*

Observe that tolls can be seen as a mean to penalise undesired (i.e., selfish) behaviour. In this sense, from Definition 1, we can conclude that if toll values do not correspond to marginal costs, then such tolls may end up penalising the wrong agents (i.e., those that are not acting selfishly). In other words, unfair tolling should be avoided.

Table 1. Example comparing *a priori* and *a posteriori* tolling in the road network of Figure 1, with three episodes

	Flow		***A priori tolling***		***A posteriori tolling***	
episode	x_A	x_B	τ_A	τ_B	τ_A	τ_B
1	*4*	*6*	*0.0*	*0.0*	*0.0*	*6.0*
2	*0*	*10*	*0.0*	*6.0*	*0.0*	*10.0*
3	*5*	*5*	*0.0*	*10.0*	*0.0*	*5.0*

In contrast to other works in the literature, TQ-learning charges tolls *a posteriori*. The next theorem shows that charging agents *a posteriori* translates into a fairer tolling scheme, since agents only pay for the cost they are actually imposing on others. A more concrete example comparing *a priori* and *a posteriori* tolling schemes in terms of fairness is presented forward, in Example 2.

THEOREM 4. *Consider a toll-based instance* $P = (G, D, f, \tau)$ *of the route choice problem. Then, charging tolls in* P a posteriori *is fairer than charging* a priori.

Proof. Building upon Definition 1, to show that *a posteriori* toll charging is fairer than *a priori* toll charging, we need to show that the former charges exactly the marginal cost, whereas the latter may not. For simplicity, we perform this analysis from the links perspective (although it easily extends to routes). In general terms, the toll charged on link l is given by $\tau_l = \beta(p_1 x_l^{\beta})$ (as formulated in Proposition 1). Assume, without loss of generality, that $p_1 = \beta = 1$. In this case, we have that $\tau_l = x_l$, which corresponds to one of the cost functions presented in Pigou (1920)'s example. Abusing notation, assume that $\tau_l^t = x_l^t$ corresponds to the toll charged on link l at episode t based on the flow on that link at that episode. Observe that the flow on link l can change from one episode to another. This is especially true at the beginning of the learning process, when the system is not yet stable. Such a difference can be expressed as $\Delta_l^t = |x_l^{t-1} - x_l^t| \geq 0$.

In the case of *a priori* toll charging, τ_l^t is computed based on previous steps. For simplicity, assume that $\tau_l^t = x_l^{t-1}$. On the one hand, if $\Delta_l^t = 0$, then the toll τ_l^t charged on link l is precisely x_l^t, given that $x_l^{t-1} = x_l^t$. On the other hand, if the flow on link l changes from one episode to another, then $x_l^{t-1} \neq x_l^t$ and $\Delta_l^t > 0$. Observe that the marginal cost for taking link l at episode t should be x_l^t, whereas *a priori* toll charging considers x_l^{t-1}. Therefore, whenever $\Delta_l^t > 0$, agents using l would be charged above (or below) the cost they are actually imposing on others. Consequently, *a priori* toll charging is unfair whenever $\Delta_l^t > 0$. This cost can be even higher when τ_l^t is not based on the flow of a *single* previous episodes, but on *many* previous episode (e.g., an average of previous flows).

In contrast, *a posteriori* toll charging defines that $\tau_l^t = x_l^t$, which corresponds precisely to the cost agents are imposing on others. Observe that Δ_l^t does not affect the toll values here. Thus, *a posteriori* toll charging (as used in TQ-learning) can be said fairer than *a priori* toll charging. □

Example 2. Consider again the 10-agent network presented in Example 1 and Figure 1. In this extended example, we consider a hypothetical sequence of three episodes (in which every agent chooses a route). Such a sequence is presented in Table 1. In the table, we present the toll values for both routes (A and B) as generated by *a priori* (as usual in the literature, assuming that tolls are initialised with zero, as in Sharon *et al.* (2017)) and *a posteriori* (as in our approach) tolling schemes.

In the case of *a priori* tolling, assume that toll values are initialised with 0.0, as in Sharon *et al.* (2017). On subsequent episodes, the toll of each route is defined as the marginal cost of such route in the previous episode. The rationale behind such model is that agents can check the tolls that they are going to pay on each route before they actually take any route. However, this leads to outdated toll values. We note that, by definition, MCT schemes should charge each agent according to its marginal cost, which is not achieved by *a priori* tolling schemes. As seen in Table 1, in the second episode, even though all agents are using route B, the toll they are going to pay is only 6.0, which corresponds to 60% of their actual

marginal cost. Later on, in the third episode, half of the agents are using each route, which corresponds to the SO. Nevertheless, agents using route B need to pay a toll of 10.0. Therefore, the prices charged by *a priori* tolling may be (and often are, as shown in this example) unfair.

In the case of *a posteriori* tolling schemes, by contrast, tolls are charged only after a route is taken. At this point, one could argue that our approach prevents agents from analysing the costs of their decisions *a priori*. However, as tolls are incorporated into agents' utility functions, the effects of such *a posteriori* charges are naturally captured by the learned Q-functions. As seen in Table 1, the tolls defined by *a posteriori* tolling schemes always correspond to the actual flow of vehicles (and their marginal costs). Consequently, *a posteriori* tolling schemes can be said to be fairer than *a priori* tolling schemes.

6 Experimental evaluation

In this section, we empirically analyse the performance of our approach to validate our theoretical results. Recall that *learning* in route choice means finding the best route to take, which can be seen as a moving target given the existence of multiple agents with possibly conflicting interests. In this context, the term *convergence* refers to the point at which the agents keep *exploiting* their knowledge most of the time and the system is *stable* (so that agents only observe small fluctuations in their costs). Our aim is to show that, by using our approach, such a stable point corresponds to a system-efficient equilibrium, that is, a user equilibrium (UE) that is aligned to the system optimum (SO).

6.1 Methodology

We simulate our method in several road networks available in the literature[7], described as follows.

- $\boldsymbol{B^1}, \ldots, \boldsymbol{B^7}$: expansions of the synthetic network introduced with the Braess paradox (Braess, 1968; Stefanello & Bazzan, 2016). The B^p graph has $|N| = 2p + 2$ nodes, $|L| = 4p + 1$ links, a single origin-destination (OD) pair, and $d = 4{,}200$ drivers.
- $\boldsymbol{BB^1}, \boldsymbol{BB^3}, \boldsymbol{BB^5}, \boldsymbol{BB^7}$: also expansions of the Braess graphs, but with two OD pairs (Stefanello & Bazzan, 2016). The $\boldsymbol{BB}^p$ graph has $|N| = 2p + 6$ nodes, $|L| = 4p + 4$ links, and $d = 4{,}200$ drivers.
- **OW:** synthetic network (Ortúzar & Willumsen, 2011) with $|N| = 13$ nodes, $|L| = 48$ links, 4 OD pairs, $d = 1{,}700$ drivers, and overlapping routes.
- **Anaheim:** abstraction of the Anaheim city, USA (Jayakrishnan *et al.*, 1993), with $|N| = 416$ nodes, $|L| = 914$ links, 38 OD pairs, $d = 104{,}694$ drivers, and highly overlapping routes.
- **Eastern-Massachusetts:** abstraction of the eastern region of the Massachusetts state, USA (Zhang *et al.*, 2016), with $|N| = 74$ nodes, $|L| = 258$ links, 74 OD pairs, and $d = 65{,}576$ drivers. Again, the routes are highly overlapped.
- **Sioux Falls:** abstraction of the Sioux Falls city, USA (LeBlanc *et al.*, 1975), with $|N| = 24$ nodes, $|L| = 76$ links, 528 OD pairs, $d = 360{,}600$ drivers, and with highly overlapping routes.

The number of routes in the above networks can be overly high. As in the literature, we limit the number of available routes to the K shortest ones[8], which we computed using the KSP algorithm (Yen, 1971).

An experiment corresponds to a complete execution, with 10,000 episodes, of TQ-learning on a single network. We measure the performance of a single execution by computing its proximity to the system optimum (SO), which is defined as in Equation (8), where v denotes the average travel time of the agents at the *last episode* of the simulation, and v^* represents the system-optimal average travel time (as reported in the literature). Since v^* represents the minimum average travel time, then the closer ϕ is to 1.00, the better.

[7] The road networks are publicly available at https://github.com/goramos/transportation_networks.

[8] As for the BB networks, we enforced the route with fewest links among the shortest ones, otherwise the system optimum would not be attainable, as discussed in Ramos (2018).

Table 2. Configuration of parameters that produced the best results for each network

Network	K	λ	μ
B^1	4	0.99	0.99
B^2	8	0.999	0.999
B^3	8	0.999	0.999
B^4	12	0.999	0.999
B^5	12	0.999	0.999
B^6	16	0.999	0.999
B^7	16	0.999	0.999
BB^1	4	0.999	0.999
BB^3	8	0.999	0.999
BB^5	4	0.999	0.999
BB^7	4	0.999	0.999
OW	12	0.999	0.999
Anaheim	16	0.999	0.999
Eastern-Massachusetts	16	0.999	0.999
Sioux Falls	12	0.9997	0.999

$$\phi(v, v^*) = \left(1 - \frac{|v - v^*|}{v^*}\right) \tag{8}$$

We tested different value combinations for our method's parameters (i.e., λ, μ, and K). For each combination, we ran 30 repetitions. The best configurations, as shown in Table 2, were selected for further analyses in the next subsection.

In order to better assess the performance of TQ-learning, we compared it against other approaches available in the literature[9]:

- Difference rewards[10] (Wolpert & Tumer, 1999, 2002). This algorithm was implemented using Q-learning, but using the difference functions as rewards. Here, when computing the difference functions, $G(\mathbf{a})$ corresponds to the average travel time of the system and $G(\mathbf{a}_{-i})$ corresponds to the average travel time of the system as if agent i were *not* using the road network.
- Standard (toll-free) Q-learning (Watkins & Dayan, 1992). The reward associated with a given route corresponds to its negative cost, as in Equation (4).

In what follows, any claim about whether one approach is better than the other is supported by Student's t-tests at the 5% significance level.

6.2 *Results*

The average performance of the algorithms in different networks in terms of proximity to the system optimum (Equation (8)) is presented in Table 3. As seen, TQ-learning indeed approximates the system optimum (SO) in all tested networks. On average, our results are within 99.814% of the SO, with a

[9] We remark that, although other tolling schemes are also available in the literature (see discussion in Section 3.2), these cannot be directly applied to our problem. This is because such approaches do not present a learning scheme and cannot be run distributedly by the drivers.

[10] Since our objective here is to compare the quality of the solutions obtained by the methods, we adopted the traditional version of difference rewards (DR, which assumes full observability in order to compute the difference signals) rather than the most recent, function approximation-based version (FADR) by Colby *et al.* (2016). The reason is that DR obtains better results than FADR (as discussed in Section 3). Hence, we believe this choice promotes a fairer comparison of DR against our method.

Table 3. Average (and standard deviation) proximity to the SO achieved by the algorithms in different networks. Statistically best results are shown in bold

Network	TQ-learning	Difference rewards	Q-learning
B^1	0.99999 (10^{-5})	0.99999 (10^{-5})	0.78856 (10^{-2})
B^2	1.00000 (10^{-6})	**1.00000** (**0.00**)	0.85413 (10^{-2})
B^3	0.99999 (10^{-5})	1.00000 (10^{-5})	0.87778 (10^{-2})
B^4	0.99999 (10^{-5})	0.99999 (10^{-5})	0.90301 (10^{-2})
B^5	**1.00000** (**10^{-5}**)	0.99997 (10^{-5})	0.91957 (10^{-3})
B^6	0.99998 (10^{-5})	**0.99999** (**10^{-5}**)	0.93498 (10^{-3})
B^7	0.99989 (10^{-5})	0.99991 (10^{-5})	0.94448 (10^{-3})
BB^1	1.00000 (0.00)	1.00000 (0.00)	0.66677 (10^{-4})
BB^3	**1.00000** (**10^{-6}**)	0.99997 (10^{-5})	0.86196 (10^{-2})
BB^5	0.99999 (10^{-6})	0.99993 (10^{-4})	0.95033 (10^{-2})
BB^7	**0.99998** (**10^{-5}**)	0.99996 (10^{-5})	0.97718 (10^{-3})
OW	0.99968 (10^{-4})	0.99969 (10^{-4})	0.99635 (10^{-4})
Anaheim	0.99341 (10^{-5})	**0.99678** (**10^{-5}**)	0.98597 (10^{-5})
Eastern-Massachusetts	**0.98429** (**10^{-4}**)	0.97124 (10^{-2})	0.96120 (10^{-4})
Sioux Falls	**0.99497** (**10^{-4}**)	0.99387 (10^{-3})	0.98662 (10^{-4})
Average	**0.99814** (**10^{-5}**)	0.99742 (10^{-3})	0.90726 (10^{-2})

standard deviation of only 0.006%. We highlight that the average travel times achieved by TQ-learning correspond to the system optimum and that, due to the toll values, no agent has incentive to change route. In other words, the achieved solution also corresponds to the UE. Thus, as expected, the experimental results are consistent with the theoretical analysis, showing that TQ-learning converges to a system-efficient equilibrium in the limit.

In the case of difference rewards, it was also possible to approximate the system optimum. In fact, as seen in Table 3, in several networks there is no statistically significant difference between the results obtained by TQ-learning and difference rewards. Nevertheless, we highlight that the overall results obtained by our approach are slightly better than those obtained by difference rewards, outperforming them at the 7% significance levels. Furthermore, our approach has shown to fare better on networks with multiple origin-destination pairs and more drivers, which evidence its robustness in more realistic scenarios. Also important, the standard deviation obtained by TQ-learning was two orders of magnitude lower than that obtained by difference rewards, on average. Such improvements over difference rewards are due to the nature of the reinforcement signals, which in our case are based on marginal costs and, thus, more accurately represent the impact an agent causes on others using the same links of the network. Finally, we also remark that, in spite of the similar results, our approach has less restrictive assumptions than difference rewards. In particular, as opposed to difference rewards, TQ-learning does not assume the existence of a central authority with full knowledge about the cost functions. In fact, our method can run in a distributed fashion, with each agent computing its own reward based exclusively on locally available information. Concerning standard Q-learning, the results are considerably worse than those by other methods, as expected. In fact, our statistical tests show that standard Q-learning was dominated by the other algorithms in all cases. This is due to the fact that, using standard Q-learning, agents do not take the system welfare into account when making their decisions. Consequently, the system optimum becomes unattainable, as detailed in Example 1, and agents end up converging to a point close to the user equilibrium. In contrast, TQ-learning (and also difference rewards) shapes the agents' rewards in order to penalise selfish behaviour, thus enforcing agents to make more altruistic decisions. Therefore, by using our tolling scheme, the trips became 9.1% faster on the tested networks, on average.

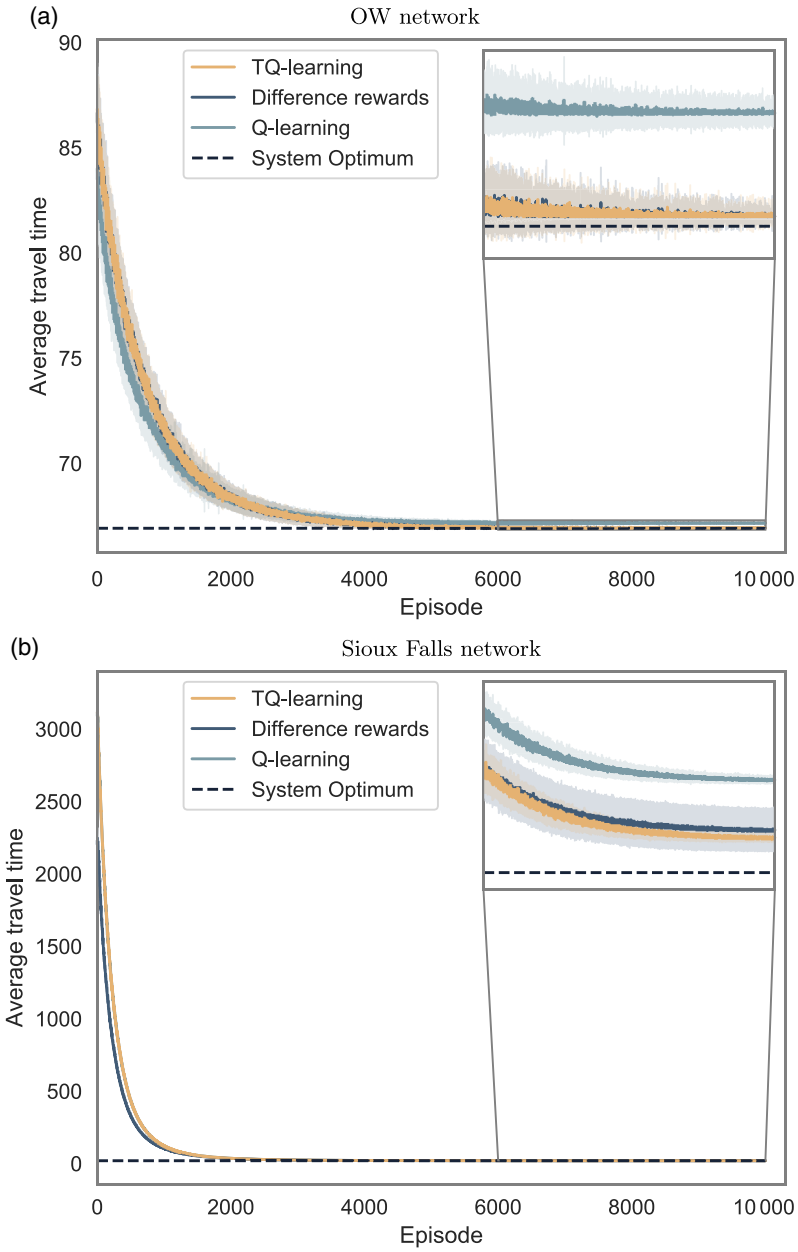

Figure 2 Evolution of average travel time along episodes for TQ-learning, difference rewards, and standard Q-learning, in two representative networks: (a) OW and (b) Sioux Falls. Shaded lines depict the standard deviation and dashed lines present the system optimum. The insets detail the last 4,000 episodes

In order to better understand the learning process, Figure 2 plots the evolution of the average travel time along episodes for the tested algorithms, in two representative networks. As can be seen, TQ-learning and difference rewards approximate the system optimum, as expected. Nonetheless, we can highlight two main differences between them. First, as previously discussed, the standard deviation of our approach is lower, since marginal costs reflect the current situation of traffic better than difference signals. Second, as shown in Figure 2(b), the average travel times obtained by difference rewards seem to decrease sooner than those by TQ-learning. Nevertheless, as seen in the inset, our algorithm ends up achieving results closer to the optimum.

We also highlight that, in the initial episodes, all algorithms remain very far from the optimum on the Sioux Falls network (Figure 2(b)). This is a consequence of the size of the agents population, meaning

that the amount of exploration in the initial episodes has a huge impact on traffic performance. This is not the case of smaller networks, such as the OW (Figure 2(a)). In all cases, however, as agents start to exploit their knowledge, the average travel time (and, thus, congestion levels) decreases steadily.

Lastly, Figure 2 also shows the performance of standard Q-learning. As previously discussed, standard Q-learning converges to the UE, not to the system optimum. The plots confirm such a hypothesis, as evidenced by the insets. We also remark that, in the tested networks, the distance between the two solution concepts is small (around 9%, on average). In real networks, however, such a difference lies around 30% (Youn *et al*., 2008), meaning that the benefits brought by our tolling scheme would be even more visible in practice.

7 Concluding remarks

In this article, we investigated how to minimise traffic congestions using a combination of tolls and reinforcement learning (RL). We considered the route choice problem in particular, which concerns how selfish drivers behave when choosing (commuting) routes everyday. Two challenges arise in this context. First, drivers need to learn independently how to adapt to each others' decisions, since the actions taken by one driver may affect the travel time perceived by others. Second, drivers' selfish behaviour deteriorates the system's performance and, as such, it should be prevented. In this sense, we introduced Toll-based Q-learning (TQ-learning), which charges tolls based on agents' marginal costs and allows agents to learn a behaviour that is also beneficial to the system.

We provided theoretical results on the agents' and system's performance. In particular, we proved that, using TQ-learning, agents converge to a user equilibrium (UE) in the limit whose average travel time corresponds to the system optimum. Moreover, we have shown that, as compared to other tolling schemes, ours is fairer in a sense that agents pay exactly (rather than approximately) their marginal costs. Additionally, we performed an extensive empirical evaluation of TQ-learning on realistic road networks with thousands of agents, whose results support our theoretical findings.

Also important, we remark that TQ-learning features important advantages over existing tolling schemes and learning methods. First, in contrast to other tolling schemes, ours charges agents *a posteriori* (i.e., after they finish their trips) and generalises the toll values formulation for univariate, homogeneous polynomial cost functions (which encompasses the most commonly used cost functions in the literature). Such a toll formulation allows agents to compute the toll associated with their routes using only their own (local) knowledge. Second, as compared to other learning methods, such as difference rewards, TQ-learning drops any full observability assumption and can be run distributedly by the agents. These features not only allow TQ-learning to obtain better average results than difference rewards but also evidence its potential applicability in real traffic settings.

In spite of the promising results, there is space for improvements. One of the main limitations of our approach is convergence speed. In fact, the more routes (and agents) we have, the longer the algorithm needs to run until convergence is achieved. An interesting direction here would be to investigate more efficient exploration strategies that could potentially speed up convergence (Hernandez-Leal *et al*., 2017). Another limitation of our approach is that agents' preferences with respect to time and money are assumed to be homogeneous. In practice, however, drivers' preferences may be heterogeneous. Hence, we would like to investigate the impact of heterogeneous preferences in the convergence properties of our algorithm. Also important, we would like to investigate how to fairly redistribute (part of) the toll values among the agents (Ramos *et al*., 2020). Such a mechanism could be useful to avoid penalising altruistic agents and also to prevent a self-interested traffic authority from strategically setting prices so as to increase its own profit.

Another topic that deserves further investigation refers to the implications associated with MCT deployment. We remark that our approach drops full-knowledge assumptions usually made in the literature, thus representing a further step towards deploying MCT. However, to the best of our knowledge, no existing work investigated (i) how much MCT would impact the tolls currently charged from drivers and (ii) how drivers would behave under such taxation schemes. We believe such points are worthy

investigating, especially in a fully distributed setting, to better understand the best procedures to adopt when deploying MCT.

Acknowledgements

The authors thank the anonymous reviewers for their thorough analysis and valuable suggestions. Ramos, Rădulescu, and Nowé were supported by Flanders Innovation & Entrepreneurship (VLAIO), SBO project 140047: Stable MultI-agent LEarnIng for neTworks (SMILE-IT). Part of this research was also supported by the The Flanders AI Research Impulse Program, Belgium. Ramos and da Silva were partially supported by FAPERGS (grants 19/2551-0001277-2 and 17/2551-000, respectively). Bazzan was partially supported by CNPq (grant 307215/2017-2). This work was also partially supported by CNPq and CAPES scholarships.

References

Abdallah, S. & Lesser, V. 2006. Learning the task allocation game, In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multi-Agent Systems, AAMAS '06,* ACM Press, 850–857.

Agogino, A. K. & Tumer, K. 2004. Unifying temporal and structural credit assignment problems, In *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '04, IEEE, 980–987.

Auer, P., Cesa-Bianchi, N., Freund, Y. & Schapire, R. E. 2002. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing* **32**(1), 48–77.

Awerbuch, B. & Kleinberg, R. D. 2004. Adaptive routing with end-to-end feedback: Distributed learning and geometric approaches. In *Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing, STOC '04*, ACM, 45–53.

Bar-Gera, H. 2010. Traffic assignment by paired alternative segments. *Transportation Research Part B: Methodological* **44**(8–9), 1022–1046.

Bazzan, A. L. C. 2009. Opportunities for multiagent systems and multiagent reinforcement learning in traffic control. *Autonomous Agents and Multiagent Systems* **18**(3), 342–375.

Bazzan, A. L. C. & Klügl, F. 2005. Case studies on the Braess paradox: simulating route recommendation and learning in abstract and microscopic models. *Transportation Research C* **13**(4), 299–319.

Bazzan, A. L. C. & Klügl, F. 2013. Introduction to Intelligent Systems in *Traffic and Transportation*, Vol. **7**. Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan and Claypool.

Beckmann, M., McGuire, C. B. & Winsten, C. B. 1956. *Studies in the Economics of Transportation*, Yale University Press, New Haven.

Bonifaci, V., Salek, M. & Schäfer, G. 2011. Efficiency of restricted tolls in non-atomic network routing games. In *Algorithmic Game Theory: Proceedings of the 4th International Symposium (SAGT 2011)*, Persiano, G. (ed). Springer, 302–313.

Bowling, M. 2005. Convergence and no-regret in multiagent learning. In L. K. Saul, Y. Weiss & L. Bottou, (eds.) *Advances in Neural Information Processing Systems **17**: Proceedings of the 2004 Conference, MIT Press*, 209–216.

Boyan, J. A. & Littman, M. L. 1994. Packet routing in dynamically changing networks: A reinforcement learning approach. *Advances in Neural Information Processing Systems* **6**, 671–678.

Braess, D. 1968. Über ein Paradoxon aus der Verkehrsplanung. *Unternehmensforschung* **12**, 258.

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

Chen, H., An, B., Sharon, G., Hanna, J. P., Stone, P., Miao, C. & Soh, Y. C. 2018. DyETC: Dynamic electronic toll collection for traffic congestion alleviation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, number February, AAAI Press, 757–765.

Chen, P.-A. & Kempe, D. 2008. Altruism, selfishness, and spite in traffic routing. In *Proceedings of the 9th ACM Conference on Electronic Commerce (EC '08)*, Riedl, J. & Sandholm, T. (eds.), ACM Press, 140–149.

Claus, C. & Boutilier, C. 1998. The dynamics of reinforcement learning in cooperative multiagent systems. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, 746–752.

Colby, M., Duchow-Pressley, T., Chung, J. J. & Tumer, K. 2016. Local approximation of difference evaluation functions. In *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016)*, IFAAMAS, Singapore, 521–529.

Cole, R., Dodis, Y. & Roughgarden, T. 2003. Pricing network edges for heterogeneous selfish users. In *Proceedings of the Thirty-fifth Annual ACM Symposium on Theory of Computing*, STOC '03, ACM, 521–530.

Crites, R. H. & Barto, A. G. 1998. Elevator group control using multiple reinforcement learning agents. *Machine Learning* **33**(2), 235–262.

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

Fehr, E. & Fischbacher, U. 2003. The nature of human altruism. *Nature* **425**(6960), 785–791.

Fleischer, L., Jain, K. & Mahdian, M. 2004. Tolls for heterogeneous selfish users in multicommodity networks and generalized congestion games. In *45th Annual IEEE Symposium on Foundations of Computer Science*, IEEE, Rome, Italy, 277–285.

Foerster, J., Nardell, N., Farquhar, G., Afouras, T., Torr, P. H., Kohli, P. & Whiteson, S. 2017. Stabilising experience replay for deep multi-agent reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning*, 70, PMLR, 1146–1155.

Centre for Economics and Business Research 2014. *The Future Economic and Environmental Costs of Gridlock in 2030*, Technical report, Centre for Economics and Business Research, London.

Hernandez-Leal, P., Zhan, Y., Taylor, M. E., Sucar, L. E. & Munoz de Cote, E. 2017. An exploration strategy for non-stationary opponents. *Autonomous Agents and Multi-Agent Systems* **31**(5), 971–1002.

Hoefer, M. & Skopalik, A. 2009. Altruism in atomic congestion games. In *17th Annual European Symposium on Algorithms*, Fiat, A. & Sanders, P. (eds.), Springer, Berlin Heidelberg, 179–189.

Hu, J. & Wellman, M. P. 1998. Multiagent reinforcement learning: Theoretical framework and an algorithm. In *Proceedings of the 15th International Conference on Machine Learning*, Morgan Kaufmann, 242–250.

Hu, J. & Wellman, M. P. 2003. Nash q-learning for general-sum stochastic games. *Journal of Machine Learning Research* **4**, 1039–1069.

Jayakrishnan, R., Cohen, M., Kim, J., Mahmassani, H. S. & Hu, T.-Y. 1993. *A Simulation-Based Framework for the Analysis of Traffic Networks Operating with Real-Time Information*, Technical Report UCB-ITS-PRR-93-25, University of California, Berkeley.

Kaisers, M. & Tuyls, K. 2010. Frequency adjusted multi-agent q-learning. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, International Foundation for Autonomous Agents and Multiagent Systems, 309–316.

Kobayashi, K. & Do, M. 2005. The informational impacts of congestion tolls upon route traffic demands. *Transportation Research A* **39**(7–9), 651–670.

Koutsoupias, E. & Papadimitriou, C. 1999. Worst-case equilibria. In *Proceedings of the 16th Annual Conference on Theoretical Aspects of Computer Science (STACS)*, Springer-Verlag, 404–413.

Lanctot, M., Zambaldi, V., Gruslys, A., Lazaridou, A., Tuyls, K., Perolat, J., Silver, D. & Graepel, T. 2017. A unified game-theoretic approach to multiagent reinforcement learning. In *Advances in Neural Information Processing Systems*, Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S. & Garnett, R. (eds.), *30*, Curran Associates, Inc., 4190–4203.

Lauer, M. & Riedmiller, M. 2004. Reinforcement learning for stochastic cooperative multi-agent-systems. *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2004* 3, 1516–1517.

Laurent, G. J., Matignon, L. & Le Fort-Piat, N. 2011. The world of independent learners is not markovian. *International Journal of Knowledge-based and Intelligent Engineering Systems* **15**(1), 55–64.

LeBlanc, L. J., Morlok, E. K. & Pierskalla, W. P. 1975. An efficient approach to solving the road network equilibrium traffic assignment problem. *Transportation Research* **9**(5), 309–318.

Levy, N. & Ben-Elia, E. 2016. Emergence of system optimum: A fair and altruistic agent-based route-choice model. *Procedia Computer Science* **83**, 928–933.

Littman, M. L. 1994. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the 11th International Conference on Machine Learning, ML*, Morgan Kaufmann, 157–163.

Littman, M. L. 2001. Friend-or-Foe Q-learning in general-sum games. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML'01)*. Morgan Kaufmann, 322–328.

Lujak, M., Giordani, S. & Ossowski, S. 2015. Route guidance: Bridging system and user optimization in traffic assignment. *Neurocomputing* **151**, 449–460.

Malialis, K., Devlin, S. & Kudenko, D. 2016. Resource abstraction for reinforcement learning in multiagent congestion problems. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems, International Foundation for Autonomous Agents and Multiagent Systems*, 503–511.

Matignon, L., Laurent, G. J. & Le Fort-Piat, N. 2012. Independent reinforcement learners in cooperative markov games: a survey regarding coordination problems. *The Knowledge Engineering Review* **27**(1), 1–31.

McFadden, D. 2001. Disaggregate behavioral travel demand's RUM side. In *Travel Behaviour Research: The Leading Edge*, Hensher, D. A. (ed), Elsevier, 17–63.

Meir, R. & Parkes, D. 2018. Playing the wrong game: Bounding externalities in diverse populations of agents. In *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018)*, IFAAMAS, Stockholm, 86–94.

Meir, R. & Parkes, D. C. 2016. When are marginal congestion tolls optimal? In *Proceedings of the Ninth Workshop on Agents in Traffic and Transportation (ATT-2016)*, Bazzan, A. L. C., Klügl, F., Ossowski, S. & Vizzari, G. (eds). CEUR-WS.org, 8.

Mirzaei, H., Sharon, G., Boyles, S., Givargis, T. & Stone, P. 2018. Link-based parameterized micro-tolling scheme for optimal traffic management. In *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems, AAMAS '18*, Dastani, M., Sukthankar, G., André, E. & Koenig, S. (eds). IFAAMAS, 2013–2015.

Nash, J. 1950. *Non-Cooperative Games*, PhD thesis, Princeton University.

National Surface Transportation Infrastructure Financing Commission 2009. *Paying Our Way: A New Framework for Transportation Finance,* Technical report, National Surface Transportation Infrastructure Financing Commission, Washington DC.

Bureau of Public Roads 1964. *Traffic Assignment Manual*, Technical report, US Department of Commerce, Washington, D. C.

Omidshafiei, S., Pazis, J., Amato, C., How, J. P. & Vian, J. 2017. Deep decentralized multi-task multi-agent reinforcement learning under partial observability. In *Proceedings of the 34th International Conference on Machine Learning*, **70**, 4108–4122.

Ortúzar, J. d. D. & Willumsen, L. G. 2011. *Modelling Transport*, 4 edition, John Wiley & Sons.

Pigou, A. 1920. *The Economics of Welfare*, Palgrave Classics in Economics, Palgrave Macmillan.

Proper, S. & Tumer, K. 2012. Modeling difference rewards for multiagent learning. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, V., Winikoff, M., Padgham, L. & van der Hoek, W. (eds). IFAAMAS.

Rădulescu, R., Vrancx, P. & Nowé, A. 2017. Analysing congestion problems in multi-agent reinforcement learning. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, International Foundation for Autonomous Agents and Multiagent Systems*, 1705–1707.

Ramos, G. de O. 2018. *Regret Minimisation and System-Efficiency in Route Choice*, PhD thesis, Universidade Federal do Rio Grande do Sul, Porto Alegre.

Ramos, G. de O. & Bazzan, A. L. C. 2015. Towards the user equilibrium in traffic assignment using GRASP with path relinking. In *Proceedings of the 2015 on Genetic and Evolutionary Computation Conference, GECCO '15*, ACM, 473–480.

Ramos, G. de O. & Bazzan, A. L. C. 2016. Efficient local search in traffic assignment. In *2016 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 1493–1500.

Ramos, G. de O., da Silva, B. C. & Bazzan, A. L. C. 2017. Learning to minimise regret in route choice. In *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017)*, Das, S., Durfee, E., Larson, K. & Winikoff, M. (eds). IFAAMAS, 846–855.

Ramos, G. de O., da Silva, B. C., Rădulescu, R. & Bazzan, A. L. C. 2018. Learning system-efficient equilibria in route choice using tolls. In *Proceedings of the Adaptive Learning Agents Workshop 2018* (ALA-18), Stockholm.

Ramos, G. de O., Rădulescu, R., Nowé, A. & Tavares, A. R. 2020. Toll-based learning for minimising congestion under heterogeneous preferences. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020)*, An, B., Yorke-Smith, N., El Fallah Seghrouchni, A. & Sukthankar, G. (eds). IFAAMAS.

Robbins, H. 1952. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society* **58**(5), 527–535.

Rosenthal, R. W. 1973. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory* **2**, 65–67.

Roughgarden, T. 2005. *Selfish Routing and the Price of Anarchy*, MIT Press.

Sandholm, T. 2007. Perspectives on multiagent learning. *Artificial Intelligence* **171**(7), 382–391.

Sen, S., Sekaran, M. & Hale, J. 1994. Learning to coordinate without sharing information. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, 426–431.

Sharon, G., Boyles, S. D., Alkoby, S. & Stone, P. 2019. Marginal cost pricing with a fixed error factor in traffic networks. In *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019)*, Agmon, N., Taylor, M., Elkind, E. & Veloso, M. (eds). IFAAMAS, Montreal, 1539–1546.

Sharon, G., Hanna, J. P., Rambha, T., Levin, M. W., Albert, M., Boyles, S. D. & Stone, P. 2017. 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)*, Das, S., Durfee, E., Larson, K. & Winikoff, M. (eds). IFAAMAS, 828–836.

Stefanello, F. & Bazzan, A. L. C. 2016. *Traffic Assignment Problem – Extending Braess Paradox*, Technical report, Universidade Federal do Rio Grande do Sul, Porto Alegre, RS.

Sutton, R. & Barto, A. 1998. *Reinforcement Learning: An Introduction*, MIT Press.

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

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

Tesauro, G. 1994. Td-gammon, a self-teaching backgammon program, achieves master-level play. *Neural Computation* **6**(2), 215–219.

Tuyls, K. & Weiss, G. 2012. Multiagent learning: Basics, challenges, and prospects. *AI Magazine* **33**(3), 41–52.

van Essen, M., Thomas, T., van Berkum, E. & Chorus, C. 2016. From user equilibrium to system optimum: a literature review on the role of travel information, bounded rationality and non-selfish behaviour at the network and individual levels. *Transport Reviews* **36**(4), 527–548.

Verbeeck, K., Nowé, A., Parent, J. & Tuyls, K. 2007. Exploring selfish reinforcement learning in repeated games with stochastic rewards. *Autonomous Agents and Multi-Agent Systems* **14**(3), 239–269.

Vrancx, P., Verbeeck, K. & Nowe, A. 2008. Decentralized learning in markov games. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **38**(4), 976–981.

Vrancx, P., Verbeeck, K. & Nowé, A. 2010. Learning to take turns. In *Proceedings of the AAMAS 2010 Workshop on Adaptive Learning Agents and Multi-Agent Systems (ALA 2010)*, 1–7.

Wardrop, J. G. 1952. Some theoretical aspects of road traffic research. *Proceedings of the Institution of Civil Engineers, Part II* 1(36), 325–362.

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

Wolpert, D. H. & Tumer, K. 1999. *An introduction to Collective Intelligence, Technical report* NASA-ARC-IC-99-63, NASA Ames Research Center. arXiv:cs/9908014 [cs.LG].

Wolpert, D. H. & Tumer, K. 2002. Collective intelligence, data routing and Braess' paradox. *Journal of Artificial Intelligence Research* **16**, 359–387.

Yang, H., Meng, Q. & Lee, D.-H. 2004. Trial-and-error implementation of marginal-cost pricing on networks in the absence of demand functions. *Transportation Research Part B: Methodological* **38**(6), 477–493.

Ye, H., Yang, H. & Tan, Z. 2015. Learning marginal-cost pricing via a trial-and-error procedure with day-to-day flow dynamics. *Transportation Research Part B: Methodological* **81**, 794–807.

Yen, J. Y. 1971. Finding the k shortest loopless paths in a network. *Management Science* **17**(11), 712–716.

Youn, H., Gastner, M. T. & Jeong, H. 2008. Price of anarchy in transportation networks: Efficiency and optimality control. *Physical Review Letters* **101**(12), 128701.

Zhang, J., Pourazarm, S., Cassandras, C. G. & Paschalidis, I. C. 2016. The price of anarchy in transportation networks by estimating user cost functions from actual traffic data. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, IEEE, 789–794.

Zinkevich, M. 2003. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the Twentieth International Conference on Machine Learning*, AAAI Press, 928–936.