Learning System-Efficient Equilibria in Route Choice Using Tolls

Gabriel de O. Ramos
Vrije Universiteit Brussel
Brussels, Belgium
goramos@ai.vub.ac.be

Bruno C. da Silva
Instituto de Informática, UFRGS
Porto Alegre, Brazil
bsilva@inf.ufrgs.br

Roxana Rădulescu
Vrije Universiteit Brussel
Brussels, Belgium
roxana@ai.vub.ac.be

Ana L. C. Bazzan
Instituto de Informática, UFRGS
Porto Alegre, Brazil
bazzan@inf.ufrgs.br

ABSTRACT

We consider the route choice problem using multiagent reinforcement learning. In this problem, agents individually learn which routes minimise their expected travel costs. Such a selfish behaviour results in the so-called User Equilibrium (UE), which is inefficient from the system's perspective. In order to reduce the impact of selfishness, we develop a toll-based Q-learning algorithm. In particular, we employ 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 so that the system optimum (SO) 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. Furthermore, we deliver theoretical results showing that our approach converges to a system-efficient equilibrium (i.e., an UE aligned to the SO) in the limit.

KEYWORDS

multiagent reinforcement learning; route choice; tolling; user equilibrium; system optimum; price of anarchy

1 INTRODUCTION

Traffic issues are faced everyday in modern society. We consider the route choice problem, which models how commuting driver-agents choose their routes to travel from their origins to their destinations. 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) [43], which is equivalent to the Nash equilibrium [25].

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) [29].

In this sense, different approaches have been proposed to align the UE towards the SO, including: charging tolls [6, 11, 35], computing difference rewards [45, 46], enforcing altruism [9, 14], etc. Among these fronts, charging tolls stand out for their relatively simplicity and for their less restrictive assumptions. One of the most important such scheme was introduced by Pigou [30] 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 paper, we approach the toll-based route choice problem from the reinforcement learning (RL) 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 our approach, each driver is represented by a Q-learning agent whose objective is to learn which route minimises its expected cost. Additionally, we design tolls using the MCT scheme, where the cost of a link comprises two terms: the travel time and the toll charged on it. We then propose 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 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 RL 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 toll values 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 overall traffic situation).
- We devise a toll-based Q-learning algorithm 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. We also validate these results in different road networks available in the literature.

2 BACKGROUND

2.1 Route Choice

An instance of the toll-based route choice problem is defined as $P = (G, D, f, \tau)$. 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 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), whereas toll values τ_l should be defined according to a specific purpose (e.g., maximising revenue, minimising link usage). We refer the reader to [28] for a more detailed overview.

Toll values can be defined according to different objectives. In this work, we consider the case of biasing the UE towards the SO. According to Pigou [30], this can be achieved by means of marginal cost tolling (MCT). Under MCT, 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 [5, 30], i.e.,

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

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 [5].

2.2 Reinforcement Learning

In reinforcement learning (RL), an agent learns by trial and error how to behave within an environment [37]. 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*, i.e., 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[2] for taking action $a \in \mathcal{A}$ can then be denoted as:

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

with $a = R$. Whenever a driver takes a route, it will inevitably reach its destination, thus rendering the *state* definition irrelevant here. Therefore, this 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 [44], which 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 [44].

Although Q-learning is guaranteed to converge in the single-agent case, it has no guarantees in multiagent settings. In fact, no general convergence guarantees exist for multiagent RL. The point is that, when multiple agents share a common environment, their actions may affect the reward received by others, which invalidates the so-called Markov property, i.e., the environment is no longer stationary [8, 39]. Notwithstanding, interesting progress has been achieved in more specific scenarios, as discussed next.

Multiagent RL (MARL) problems may be approached under different perspectives. Stochastic (or Markov) games [20] represent a straightforward approach, where agents' decisions are represented in a joint action space. Several algorithms have been proposed in this context for 2-player zero-sum games [20], 2-player general-sum games [15, 16], and coordination games [21, 41, 42]. However, route choice usually consists in several (not only two) agents, which rarely cooperate [34]. Gradient ascent algorithms were also proposed to handle multiagent learning scenarios [1, 7, 50]. Nonetheless, their convergence guarantees still only apply to 2-player games.

In this paper, we model agents as *independent Q-learners* [10], where each agent has its own stateless MDP and interprets the

[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] Observe that, although the reward an agent receives is formulated as a function of its single 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.

behaviour of other agents as the dynamics underlying its environment. In spite of its simplicity, this modelling properly represents real traffic settings, given that drivers have a very limited knowledge about what others are doing (not to mention their policies). We advance the state-of-the-art by proposing a modified version of Q-learning (where agents also need to compute—and pay—a toll for the routes they take) that is guaranteed to converge both to the UE and to the SO.

3 RELATED WORK

In this section we discuss representative literature on system-efficient equilibria in route choice (and related problems). The reader is referred to [28, 40] for a more detailed overview.

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 [11], toll information mechanisms [17], tolls with bounded values [6], RL-based tolls [38], and so on. We concentrate, however, in the marginal-cost tolling (MCT) scheme [30]. The concept of MCT has been investigated in several works, such as [24, 35, 47, 48]. As opposed to our approach, nonetheless, these works neither investigate how drivers react to tolls nor how to ensure convergence to the user equilibrium (UE)—which, by using MCT, shall be equivalent to the system optimum (SO). Furthermore, these tolling schemes charge tolls *a priori*, i.e., before the 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 the prices based on historical congestion levels, meaning that the agents may end up paying a toll that is higher than their 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 5.2).

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. In this way, we can simplify the infrastructure requirements for deploying the tolling scheme 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 [26]. 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).

Similarly to charging tolls, some works investigated the SO by explicitly assuming that agents behave altruistically. Chen and Kempe [9] and Hoefer and Skopalik [14] investigated altruism in routing games. Levy and Ben-Elia [19] 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 [13]. 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 [22], biasing trip suggestions [4], allocating routes into abstract groups that offer more informative cost functions [23, 31], etc. Notwithstanding, in general, these works assume the existence of a centralised mechanism.

Wolpert and Tumer [45, 46] introduced the idea of *difference rewards*, which also relates to our approach. Basically, the difference reward an agent receives for taking an action corresponds to the amount the system's performance deteriorates considering his action. Precisely, it is measured as the difference between the system's performance with and without it. Using difference rewards, the agents' reward signal is aligned with the system's utility so that they converge to the SO. However, difference rewards can only be computed upon strong, full observability assumptions. Later on, methods for approximating the difference reward signals were proposed [2]. Nonetheless, this kind of approach still depends on some sort of global information.

4 OUR APPROACH

This section presents our reinforcement learning method through which agents can compute the tolls associated with their routes and use that information to learn their best routes. We model the problem as a stateless Markov decision process (MDP) and represent drivers by means of Q-learning agents. At every episode, each such agent chooses a route from its origin to its destination and, once the trip is completed, the agent observes its travel time. Building upon such observations, we propose a general tolling scheme through which the toll values can be computed a posteriori by the agents themselves. Together, the travel time and toll value an agent experiences in a given route compose the cost of such route. Using this cost, each agent then computes the regret associated with the chosen route and uses such information to update its Q-table.

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 [12]. 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 volume-delay function (VDF) being employed. Sharon et al. [35] have shown that, for the BPR function [27], 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. Nonetheless, given that different VDFs are available in the literature, we go beyond and generalise the toll formulation according to the following proposition.

Proposition 4.1. *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 [28] 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 [36] 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.

Polynomial functions can be defined as $f_l(x_l) = \sum_{\beta=0}^{n} p_\beta x^\beta$. In this paper 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 [27]. 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 $\tau_l = x_l(p_1 x_l^\beta + p_0)' = x_l(p_1 \beta x_l^{\beta-1}) = \beta(p_1 x_l^\beta)$, as required. □

We emphasise that Proposition 4.1 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 4.1, 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 4.1 with the formulation of Sharon et al. [35], thus obtaining the following corollary.

Corollary 4.2. *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 routes reward function as in Equation (6), which follows from Proposition 4.1 and Equations (2) and (4).

$$\begin{aligned} r(a_i^t) &= -\textstyle\sum_{l \in a_i^t} c_l \\ &= -\textstyle\sum_{l \in a_i^t} f_l + \beta(f_l - F_l) \\ &= -(f_{a_i^t} + \beta(f_{a_i^t} - F_{a_i^t})). \end{aligned} \tag{6}$$

Algorithm 1: Toll-based Q-learning (for agent i)

input: A_i, λ, μ, T, β, and F_l (for every link $l \in L$)

initialise Q-table: $Q(a_i) \leftarrow 0 \ \forall a_i \in A_i$;
for $t \in T$ **do**
  $\alpha \leftarrow \lambda^t$; $\epsilon \leftarrow \mu^t$;
  $a_i^t \leftarrow$ choose (and take) action using ϵ-greedy;
  $f_{a_i^t} \leftarrow$ observe travel time on a_i^t;
  $r(a_i^t) \leftarrow -(f_{a_i^t} + \beta(f_{a_i^t} - F_{a_i^t}))$;
  $Q(a_i^t) \leftarrow (1-\alpha)Q(a_i^t) + \alpha r(a_i^t)$;
end

We can now present our RL algorithm. 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 (6). The drivers' objective is to maximise their cumulative reward. An overview of our method 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 observe its travel time $f_{a_i^t}$ and computes its reward $r(a_i^t)$ following Equation (6). 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 $\alpha(t) = \lambda^t$.

5 THEORETICAL ANALYSIS

In this section, we provide a theoretical analysis of our approach. The aim is to show that our approach converges to a system-efficient equilibrium (i.e., an user equilibrium—UE—corresponding to the system optimum—SO), as formulated in Theorem 5.1, adapted from [5].

Theorem 5.1 (Beckmann et al. [5]). *Consider a toll-based instance $P' = (G, D, f, \tau)$ of the route choice problem, where driver $i \in D$ experiences a cost $c_l = f_l + \tau_l$ after traversing link l, with f_l and τ_l representing the travel time and toll charged at that link, respectively. Under these settings, the average travel time under user equilibrium for P' equals that of the system optimum for $P = (G, D, f)$.*

Intuitively, Theorem 5.1 says that given an instance P of the route choice problem, if we apply marginal-cost tolling (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. [5] for the complete proof. An illustrative example on how this theorem applies to Pigou [30]'s network is presented in Example 5.2.

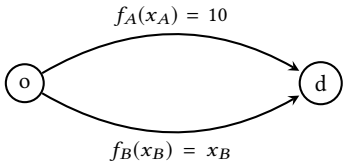

Figure 1: Example network adapted from Pigou [30].

Example 5.2. Consider the network in Figure 1, adapted from Pigou [30], 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, i.e., $c_l = f_l + \tau_l$. 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 5.1 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 5.1 simply assumes that the UE is given. Indeed, this is a common assumption of other works in the literature, such as in [35]. However, since route choice is a multiagent problem, guaranteeing convergence to the UE is not trivial (as discussed in Section 2.2). Hence, in order for Theorem 5.1 to apply to our approach, we need first to show that our approach indeed achieves the UE. In contrast to other works in the literature, we show that our method *converges to the UE*, and then we show that such UE is aligned to the SO. This is shown in Theorem 5.3. The complete proof is presented in the next subsection.

Theorem 5.3. *Consider an instance P of the route choice problem. If all drivers use the Q-learning algorithm with learning rate $\alpha(t) = \lambda^t$ and exploration rate $\epsilon(t) = \mu^t$, then the system converges to the user equilibrium in the limit.*

From Theorem 5.3, 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, our algorithm achieves a system-efficient equilibrium (Theorem 5.1). In other words, our approach reduces the PoA to its best ratio of 1. Therefore, based on Theorems 5.1 and 5.3 we can formulate the following corollary.

Corollary 5.4. *Consider an instance P of the route choice problem, where all drivers use the Q-learning algorithm with learning rate $\alpha(t) = \lambda^t$ and exploration rate $\epsilon(t) = \mu^t$. By employing marginal-cost tolling, the agents converge to a system-efficient equilibrium in the limit, i.e., the user equilibrium is aligned to the system optimum. Thus, the price of anarchy converges to 1 in the limit.*

5.1 Convergence to the UE

In this section, we prove Theorem 5.3 by showing that our approach converges to the UE. Hereafter, by *our approach* we mean the settings presented in Section 4, i.e., a stateless MDP with Q-learning agents using ϵ-greedy exploration, where $\alpha(t) = \lambda^t$ and $\epsilon(t) = \mu^t$. 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 5.3 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 5.6). 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 5.9), which leads the agents to exploit only optimal actions in the limit (Lemma 5.8), thus achieving the UE (Theorem 5.3).

Initially, the next proposition defines the probability that best[3] and non-best actions are chosen by a given agent i at episode t.

Proposition 5.5. *Using ϵ-greedy exploration with $\epsilon(t) = \mu^t$, at episode t agent i chooses its best action $\dot{a}_i^t = \arg\max_{a_i^t \in A_i} Q(a_i^t)$ with probability[4] $\rho(\dot{a}_i^t) = 1 - \frac{\mu^t(K-1)}{K}$ and any other action $\bar{a}_i^t \in A_i \setminus \dot{a}_i^t$ 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 $\dot{a}_i^t = \arg\max_{a_i^t \in A_i} Q(a_i^t)$ 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 $\rho(\dot{a}_i^t) = (1 - \mu^t) + \frac{\mu^t}{K} = 1 + \frac{\mu^t - K\mu^t}{K} = 1 - \frac{\mu^t(K-1)}{K}$. Similarly, we can rewrite the probability of agent i selecting any non-best action at a given episode t as $\rho(\bar{a}_i^t) = \mu^t - \frac{\mu^t}{K} = \frac{K\mu^t - \mu^t}{K} = \frac{\mu^t(K-1)}{K}$. □

From Proposition 5.5, observe that $\dot{\rho} \to 1$ 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 5.6, adapted from Ramos et al. [33] (we refer the reader to their work for a complete proof).

Theorem 5.6 (Ramos et al. [33]). *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 $O(\bar{\rho}^{\nabla}(\dot{\rho} + \bar{\rho}))$, 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

[3] 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.

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

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 5.9 and Theorem 5.3, 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. To this regard, we firstly characterise the agent's behaviour in terms of the UE, as shown in the next lemma.

Lemma 5.7. *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 5.9). 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 5.8 and 5.9, respectively.

Lemma 5.8. *In the limit, agents exploit their knowledge most of the time, i.e., they tend to choose the actions with highest estimated Q-values.*

Proof. It follows from Proposition 5.5 and Theorem 5.6, since $\bar{\rho}_i^t \to 1$ and $\bar{\rho}_i^t \to 0$ as $t \to \infty$ and $\epsilon \to 0$. □

Lemma 5.9. *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 $\overset{*}{a}_i = \arg\max_{a_i \in A_i} r(a_i)$, i.e., $\overset{+}{a}_i = \overset{*}{a}_i$ 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, i.e., $\overset{+}{a}_i \neq \overset{*}{a}_i = \arg\max_{a_i \in A_i} r(a_i)$. 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 $\overset{*}{a}_i$. Moreover, 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, i.e., $Q(\overset{*}{a}_i) \approx r(\overset{*}{a}_i) > Q(\overset{+}{a}_i) \approx r(\overset{+}{a}_i)$, which contradicts the initial assumption. □

We highlight 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 5.10. *In the limit, agents using ϵ-greedy exploration with $\epsilon(t) = \mu^t$ can still explore non-best actions without invalidating the user equilibrium, i.e., exploration does not destabilise the equilibrium.*

Proof (sketch). Suppose 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, i.e., the ones with highest estimated Q-values (Lemmas 5.7 and 5.8). Observe that agents can still explore other actions, though less frequently (Proposition 5.5 and Lemma 5.8). 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 5.6, 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 5.9). □

We now have the required tools for proving Theorem 5.3. 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 5.1, this is only attainable if our approach is guaranteed to converge to the UE. Therefore, proving Theorem 5.3 is sufficient to show that, by employing MCT, our approach converges to the SO.

Proof of Theorem 5.3. According to Theorem 5.6, 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 5.8, 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 5.9. Finally, from Lemma 5.10 we have that exploration does not affect the UE. Thus, our algorithm can be said to converge to the UE. □

5.2 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 5.11 (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 5.11, 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.

In contrast to other works in the literature, our approach 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 5.13.

Theorem 5.12. *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 5.11, 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 4.1). 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 [30]'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 our approach) can be said fairer than a priori toll charging. □

Example 5.13. Consider again the 10-agent network presented in Example 5.2 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. [35]) 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. [35]. 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.

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

	flow		a priori		a posteriori	
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

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, i.e., the user equilibrium (UE) is aligned to the system optimum (SO).

6.1 Methodology

We simulate our method in several road networks available in the literature[5], described as follows.

- $\boldsymbol{B^1}, \ldots, \boldsymbol{B^7}$: expansions of the network introduced with the Braess paradox [36]. 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 [36]. The BB^p graph has $|N| = 2p + 6$ nodes, $|L| = 4p + 4$ links, and $d = 4{,}200$ drivers.
- **OW**: synthetic network [28] with $|N| = 13$ nodes, $|L| = 48$ links, 4 OD pairs, $d = 1{,}700$ drivers, and overlapping routes.
- **SF**: abstraction of the Sioux Falls city, USA [18], with $|N| = 24$ nodes, $|L| = 76$ links, 528 OD pairs, $d = 360{,}600$ drivers, and highly overlapping routes.

[5]The road networks are available at https://github.com/maslab-ufrgs/network-files.

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[6], which we computed using the KSP algorithm [49].

An experiment corresponds to a complete execution, with 10,000 episodes, of our method on a single network. We evaluate the performance of an execution by measuring how close the average travel time obtained by it is to that of the SO; the closer the value is to 100, the better.

We tested different value combinations for our method's parameters (i.e., λ, μ, and K). For each combination, we ran 30 repetitions and selected the best configurations for further analyses in the next subsection. We omit the parameters' analysis due to the lack of space. In order to better evaluate our method, we compared it against other approaches available in the literature, namely difference rewards [45, 46], Δ-tolling [35], and standard (toll-free) Q-learning [44], to which we refer hereafter as DR, ΔT, and SQ, respectively. In all cases, we employed our own algorithm, but replacing the reward functions as appropriate. 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 all algorithms in different networks in terms of proximity to the system optimum (SO) is shown in Table 2. As seen, our approach indeed approximates the SO in the tested networks. On average, our results are within 99.957% of the SO, with a standard deviation of 0.006%. We remark that the average travel times achieved here correspond to the SO and that, due to the toll values, agents have no incentive to deviate (i.e., they reached an equilibrium). Thus, as expected, the experimental results are consistent with the theoretical analysis, showing that our approach converges to a system-efficient equilibrium in the limit.

As for the other algorithms, it can be seen that Δ-tolling (ΔT) and difference rewards (DR) were also able to approximate the SO. In fact, in most cases, there are no statistically significant differences between the results obtained by the three SO-oriented algorithms (i.e., Ours, ΔT, and DR) tested here. We highlight, however, that our approach achieved slightly better average results, especially on instances with multiple origin-destination pairs. This is due to the fact that our approach charges tolls a posteriori, meaning that our rewards reflect the current dynamics of the system more accurately than the other methods do. We also remark that, in spite of the similar results, our approach has less restrictive assumptions than the other methods. In particular, as opposed to ΔT, our approach (i) provides a learning scheme, (ii) defines a fairer tolling scheme that can be computed by the agents themselves, and (iii) has convergence guarantees. Furthermore, in contrast to DR, our method does not assume the existence of a central authority (which, in case of DR, also requires full knowledge about the cost functions).

Finally, observe that standard Q-learning (SQ) obtained the worst results among the tested algorithms, as expected. This is due to the fact that, using SQ, agents do not take the system welfare into account when making their decisions. Consequently, the SO becomes unattainable, as detailed in Example 5.2. In contrast, the SO-based approaches change the agents' reward in order to penalise selfish behaviour, thus enforcing agents to make more altruistic decisions.

[6] As for the BB networks, we enforced the route with fewest links among the shortest ones, otherwise the SO would not be attainable, as discussed in [32].

Table 2: Average (and standard deviation) proximity to the SO achieved by the algorithms in different networks.

Net.	Ours	ΔT	DR	SQ
B^1	99.999 (10^{-3})	99.999 (10^{-4})	99.999 (10^{-3})	78.856 (5.29)
B^2	99.999 (10^{-4})	99.999 (10^{-4})	100.00 (0.00)	85.413 (3.81)
B^3	99.999 (10^{-3})	99.999 (10^{-3})	99.999 (10^{-3})	87.778 (2.12)
B^4	99.999 (10^{-3})	99.999 (10^{-3})	99.999 (10^{-3})	90.301 (1.68)
B^5	99.999 (10^{-3})	99.998 (10^{-3})	99.997 (10^{-3})	91.957 (10^{-1})
B^6	99.998 (10^{-3})	99.997 (10^{-3})	99.999 (10^{-3})	93.498 (10^{-1})
B^7	99.989 (10^{-3})	99.990 (10^{-3})	99.991 (10^{-3})	94.448 (10^{-1})
BB^1	100.00 (0.00)	100.00 (0.00)	100.00 (0.00)	66.677 (10^{-2})
BB^3	99.999 (10^{-4})	99.999 (10^{-4})	99.997 (10^{-3})	86.196 (1.85)
BB^5	99.999 (10^{-4})	99.998 (10^{-3})	99.993 (10^{-2})	95.033 (1.61)
BB^7	99.998 (10^{-3})	99.999 (10^{-3})	99.996 (10^{-3})	97.718 (10^{-1})
OW	99.968 (10^{-2})	99.968 (10^{-2})	99.969 (10^{-2})	99.635 (10^{-2})
SF	99.497 (10^{-2})	99.542 (10^{-2})	99.387 (10^{-1})	98.662 (10^{-2})
Avg.	99.957 (10^{-3})	99.961 (10^{-3})	99.948 (10^{-2})	89.706 (1.40)

7 CONCLUDING REMARKS

In this paper, we investigated how to achieve system-efficient equilibria in route choice by employing marginal-cost tolling (MCT) and reinforcement learning (RL). Learning plays a role in route choice because drivers must learn independently how to adapt to each others' decisions. We defined an MCT scheme that 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 used cost functions in the literature). Our toll formulation allows the agents to compute the toll associated with their routes using only their own knowledge.

We provided theoretical results on the agents and system performance. In particular, we proved that agents converge to a user equilibrium (UE) in the limit whose average travel time corresponds to the system optimum (SO). 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.

As future work we would like to extend our approach to the dynamic route choice problem [3], where routes are not known a priori. This problem is more challenging because agents must learn their routes by exploring the entire network. Another interesting direction would be investigating how to fairly redistribute the toll values (or part of them) among the agents. Such a mechanism could be useful to avoid penalising altruistic agents. Finally, we also would like to incorporate the notion of *regret* into the agents' utility [33], thus improving the convergence analysis.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their helpful suggestions. Gabriel and Roxana were supported by Flanders Innovation & Entrepreneurship (VLAIO), SBO project 140047: Stable MultI-agent LEarnIng for neTworks (SMILE-IT). Bruno was partially supported by FAPERGS (project 17/2551-000). This work was also partially supported by CNPq and CAPES.

REFERENCES

[1] Sherief Abdallah and Victor Lesser. 2006. Learning the Task Allocation Game. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS06)*. ACM Press, Hakodate, 850–857.

[2] Adrian K. Agogino and Kagan Tumer. 2004. Unifying Temporal and Structural Credit Assignment Problems. In *Proc. of the 3rd Intl. Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '04)*. IEEE, New York, 980–987.

[3] Baruch Awerbuch and Robert D. Kleinberg. 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, New York, 45–53. https://doi.org/10.1145/1007352.1007367

[4] Ana L. C. Bazzan and Franziska Klügl. 2005. Case Studies on the Braess Paradox: simulating route recommendation and learning in abstract and microscopic models. *Transportation Research C* 13, 4 (August 2005), 299—319.

[5] Martin Beckmann, C. B. McGuire, and Christopher B. Winsten. 1956. *Studies in the Economics of Transportation*. Yale University Press, New Haven.

[6] Vincenzo Bonifaci, Mahyar Salek, and Guido Schäfer. 2011. Efficiency of Restricted Tolls in Non-atomic Network Routing Games. In *Algorithmic Game Theory: Proceedings of the 4th International Symposium (SAGT 2011)*, G. Persiano (Ed.). Springer, Amalfi, 302–313. https://doi.org/10.1007/978-3-642-24829-0_27

[7] Michael Bowling. 2005. Convergence and No-Regret in Multiagent Learning. In *Advances in Neural Information Processing Systems 17: Proceedings of the 2004 Conference*, L. K. Saul, Y. Weiss, and L. Bottou (Eds.). MIT Press, 209–216.

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

[9] Po-An Chen and David Kempe. 2008. Altruism, selfishness, and spite in traffic routing. In *Proceedings of the 9th ACM conference on Electronic commerce (EC '08)*, J. Riedl and T. Sandholm (Eds.). ACM Press, New York, 140–149. https://doi.org/10.1145/1386790.1386816

[10] Caroline Claus and Craig Boutilier. 1998. The Dynamics of Reinforcement Learning in Cooperative Multiagent Systems. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*. 746–752.

[11] Richard Cole, Yevgeniy Dodis, and Tim Roughgarden. 2003. Pricing Network Edges for Heterogeneous Selfish Users. In *Proceedings of the Thirty-fifth Annual ACM Symposium on Theory of Computing (STOC '03)*. ACM, New York, 521–530. https://doi.org/10.1145/780542.780618

[12] André de Palma and Robin Lindsey. 2011. Traffic congestion pricing methodologies and technologies. *Transportation Research Part C: Emerging Technologies* 19, 6 (2011), 1377–1399. https://doi.org/10.1016/j.trc.2011.02.010

[13] Ernst Fehr and Urs Fischbacher. 2003. The nature of human altruism. *Nature* 425, 6960 (oct 2003), 785–791. https://doi.org/10.1038/nature02043

[14] Martin Hoefer and Alexander Skopalik. 2009. Altruism in Atomic Congestion Games. In *17th Annual European Symposium on Algorithms*, Amos Fiat and Peter Sanders (Eds.). Springer Berlin Heidelberg, Copenhagen, 179–189. https://doi.org/10.1007/978-3-642-04128-0_16

[15] Junling Hu and Michael P. Wellman. 1998. Multiagent Reinforcement Learning: Theoretical Framework and an Algorithm. In *Proc. 15th International Conf. on Machine Learning*. Morgan Kaufmann, 242–250.

[16] Junling Hu and Michael P. Wellman. 2003. Nash Q-learning for General-sum Stochastic Games. *J. Mach. Learn. Res.* 4 (2003), 1039–1069.

[17] Kiyoshi Kobayashi and Myungsik Do. 2005. The Informational Impacts of Congestion Tolls upon Route Traffic Demands. 39, 7–9 (Aug-Nov 2005), 651–670.

[18] Larry J LeBlanc, Edward K Morlok, and William P Pierskalla. 1975. An efficient approach to solving the road network equilibrium traffic assignment problem. *Transportation Research* 9, 5 (1975), 309–318.

[19] Nadav Levy and Eran Ben-Elia. 2016. Emergence of System Optimum: A Fair and Altruistic Agent-based Route-choice Model. *Procedia Computer Science* 83 (2016), 928–933. https://doi.org/10.1016/j.procs.2016.04.187

[20] Michael L. Littman. 1994. Markov Games as a Framework for Multi-Agent Reinforcement Learning. In *Proceedings of the 11th International Conference on Machine Learning, ML*. Morgan Kaufmann, New Brunswick, NJ, 157–163.

[21] Michael L. Littman. 2001. Friend-or-Foe Q-learning in General-Sum Games. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML01)*. Morgan Kaufmann, San Francisco, CA, USA, 322–328.

[22] Marin Lujak, Stefano Giordani, and Sascha Ossowski. 2015. Route guidance: Bridging system and user optimization in traffic assignment. *Neurocomputing* 151 (mar 2015), 449–460. https://doi.org/10.1016/j.neucom.2014.08.071

[23] Kleanthis Malialis, Sam Devlin, and Daniel Kudenko. 2016. Resource abstraction for reinforcement learning in multiagent congestion problems. In *Proc. of the 2016 International Conference on Autonomous Agents & Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 503–511.

[24] Reshef Meir and David C. Parkes. 2016. When are Marginal Congestion Tolls Optimal?. In *Proceedings of the Ninth Workshop on Agents in Traffic and Transportation (ATT-2016)*, Ana L. C. Bazzan, Franziska Klügl, Sascha Ossowski, and Giuseppe Vizzari (Eds.). CEUR-WS.org, New York, 8. http://ceur-ws.org/Vol-1678/paper3.pdf

[25] John Nash. 1950. *Non-Cooperative Games*. Ph.D. Dissertation. Priceton University.

[26] 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. https://itif.org/publications/2009/02/24/paying-our-way-new-framework-transportation-finance

[27] Bureau of Public Roads. 1964. *Traffic Assignment Manual*. Technical Report. US Department of Commerce, Washington, D.C.

[28] Juan de Dios Ortúzar and Luis G. Willumsen. 2011. *Modelling transport* (4 ed.). John Wiley & Sons, Chichester, UK.

[29] Christos Papadimitriou and John N. Tsitsiklis. 1987. The complexity of Markov decision processes. *Mathematics of Operations Research* 12, 3 (August 1987), 441–450.

[30] A. Pigou. 1920. *The Economics of Welfare*. Palgrave Macmillan, London.

[31] Roxana Rădulescu, Peter Vrancx, and Ann Nowé. 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.

[32] Gabriel de Oliveira Ramos. 2018. *Regret Minimisation and System-Efficiency in Route Choice*. Ph.D. Dissertation. Universidade Federal do Rio Grande do Sul, Porto Alegre. http://hdl.handle.net/10183/178665

[33] Gabriel de O. Ramos, Bruno C. da Silva, and Ana L. C. Bazzan. 2017. Learning to Minimise Regret in Route Choice. In *Proc. of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017)*, S. Das, E. Durfee, K. Larson, and M. Winikoff (Eds.). IFAAMAS, São Paulo, 846–855. http://ifaamas.org/Proceedings/aamas2017/pdfs/p846.pdf

[34] Tuomas Sandholm. 2007. Perspectives on Multiagent Learning. *Artificial Intelligence* 171, 7 (May 2007), 382–391. http://www.sciencedirect.com/science/article/B6TYF-4NCJCWW-4/2/aa883aefdf4da4ca3869354e33081bca

[35] Guni Sharon, Josiah P Hanna, Tarun Rambha, Michael W Levin, Michael Albert, Stephen D Boyles, and Peter Stone. 2017. Real-time Adaptive Tolling Scheme for Optimized Social Welfare in Traffic Networks. In *Proc. of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017)*, S. Das, E. Durfee, K. Larson, and M. Winikoff (Eds.). IFAAMAS, São Paulo, 828–836.

[36] Fernando Stefanello and Ana L. C. Bazzan. 2016. *Traffic Assignment Problem - Extending Braess Paradox*. Technical Report. Universidade Federal do Rio Grande do Sul, Porto Alegre, RS. 24 pages. www-usr.inf.ufsm.br/~stefanello/publications/Stefanello2016Braess.pdf

[37] R.S. Sutton and A.G. Barto. 1998. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA.

[38] Anderson Rocha Tavares and Ana LC Bazzan. 2014. An agent-based approach for road pricing: system-level performance and implications for drivers. *Journal of the Brazilian Computer Society* 20, 1 (2014), 15. http://dx.doi.org/10.1186/1678-4804-20-15

[39] K. Tuyls and G. Weiss. 2012. Multiagent Learning: Basics, Challenges, and Prospects. *AI Magazine* 33, 3 (2012), 41–52.

[40] Mariska van Essen, Tom Thomas, Eric van Berkum, and Caspar Chorus. 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 (jul 2016), 527–548.

[41] Katja Verbeeck, Ann Nowé, Johan Parent, and Karl Tuyls. 2007. Exploring selfish reinforcement learning in repeated games with stochastic rewards. *Autonomous Agents and Multi-Agent Systems* 14, 3 (Apr 2007), 239–269. https://doi.org/10.1007/s10458-006-9007-0

[42] Peter Vrancx, Katja Verbeeck, and Ann Nowé. 2010. Learning to Take Turns. In *Proceedings of the AAMAS 2010 Workshop on Adaptive Learning Agents and Multi-Agent Systems (ALA 2010)*. 1–7.

[43] John Glen Wardrop. 1952. Some theoretical aspects of road traffic research. *Proceedings of the Institution of Civil Engineers, Part II* 1, 36 (1952), 325–362.

[44] Christopher J. C. H. Watkins and Peter Dayan. 1992. Q-learning. *Machine Learning* 8, 3 (1992), 279–292.

[45] David H. Wolpert and Kagan Tumer. 1999. *An Introduction to Collective Intelligence*. Technical Report NASA-ARC-IC-99-63. NASA Ames Research Center. 88 pages. arXiv:cs/9908014 [cs.LG].

[46] David H Wolpert and Kagan Tumer. 2002. Collective intelligence, data routing and braess' paradox. *Journal of Artificial Intelligence Research* 16 (2002), 359–387.

[47] Hai Yang, Qiang Meng, and Der-Horng Lee. 2004. Trial-and-error implementation of marginal-cost pricing on networks in the absence of demand functions. *Transportation Research Part B: Methodological* 38, 6 (jul 2004), 477–493. https://doi.org/10.1016/S0191-2615(03)00077-8

[48] Hongbo Ye, Hai Yang, and Zhijia Tan. 2015. Learning marginal-cost pricing via a trial-and-error procedure with day-to-day flow dynamics. *Transportation Research Part B: Methodological* 81 (nov 2015), 794–807. https://doi.org/10.1016/j.trb.2015.08.001

[49] Jin Y. Yen. 1971. Finding the K Shortest Loopless Paths in a Network. *Management Science* 17, 11 (1971), 712–716. https://doi.org/10.1287/mnsc.17.11.712

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