Accelerating Learning of Route Choices With C2I: A Preliminary Investigation

Guilherme D. dos Santos and Ana L. C. Bazzan

Universidade Federal do Rio Grande do Sul
{gdsantos,bazzan}@inf.ufrgs.br

Abstract. How to choose a route that takes you from A to B? This is an issue that is turning more and more important in modern societies. One way to address this agenda is through the use of communication between the infrastructure (network), and the demand (vehicles). In this paper, we use car-to-infrastructure (C2I) communication to investigate whether the road users (agents) can accelerate their learning process regarding route choice problem, via reinforcement learning (RL). We employ a microscopic simulator in order to compare our method with two others: RL without communication and an iterative method. Experimental results show that our method outperforms both methods in terms of effectiveness and efficiency.

CCS Concepts: • **Computing methodologies** → **Multi-agent reinforcement learning**.

Keywords: Multi-Agent Reinforcement Learning, Route Choice, Car-to-Infrastructure Communication

1. INTRODUCTION

How to choose a route that takes you from A to B? This is an issue that is turning more and more important in modern societies, impacting the quality of life. To address this, traffic authorities and experts try to distribute the flow among existing routes in order to minimize the overall travel time. This task involves some form of communication with the drivers. Traditional approaches such as variable message panels or radio broadcast are now being replaced by directed (and potentially personalized) communication, via new kinds of communication devices. Hence, while the current pattern is that each individual driver selects a route based on his/her own experience, this is changing as new technologies allow all sorts of information exchange. Examples of these technologies are not only based on broadcast (e.g., GPS or cellphone information) but also a two-way communication channel, where drivers not only receive traffic information but also provide them). Hence, currently, many traffic models deal with the idea of a central authority in charge of assigning routes for drivers, as an attempt to find a feasible solution. Examples are Waze, Google apps, etc. However, these platforms do not handle locally collected and processed data, thus being centralized. This leads to them being ineffective when the penetration of their services is low (see, e.g., `https://link.estadao.com.br/noticias/empresas,por-que-apps-como-waze-e-google-maps-tem-problemas-em-dias-de-enchente,70003192968`). A way to mitigate this could be to decentralize the handling of information, as proposed here, and passing it to drivers to make their route choices.

One way to study route choice is through the use of multi-agent reinforcement learning (MARL), where it is possible to simulate how drivers (or agents) choose their preferable route based on their own learning experiences.

In the present paper, we connect MARL to new technologies such as car-to-infrastructure communication (C2I). We do so in order to investigate how C2I communication could act to augment the information drivers use in their route choices. A key difference between existing approaches (e.g., the aforementioned Waze) is that, here, we do not recommend a whole route to drivers, but rather, give them local information about the most updated state of the links that happen to be near their current location. This way, drivers can change their route on-the-fly (the so-called en route trip building).

Our approach assumes that the infrastructure is able to communicate with the vehicles, both collecting information about their most recent travel times (on given links), as well as providing them with information that was collected from other vehicles. One advantage of our approach is that it does not suggest or impose whole routes to drivers. In fact, although the infrastructure is not able to force the agents to take the best routes, it might influence their choices by providing updated information.

As a result of our approach, we are able to show that the MARL technique combined with a C2I model can accelerate the learning process, meaning it will take less time for the system to reach the user equilibrium. Moreover, we deal with a microscopic, agent-based approach where agents can potentially use different pieces of information in order to perform en route choice.

This paper is organized as follows. The next section briefly presents some concepts on traffic assignment and reinforcement learning. Then, Section 3 discusses the related work. Section 4 presents how the problem is modeled, and how the solution is built. The results are discussed in Section 5. Conclusions and future works are stated in Section 6.

2. BACKGROUND

2.1 The Traffic Assignment Problem

In transportation, the traffic assignment problem (TAP) refers to how to associate a supply (traffic infrastructure) to its demand in the best way possible, i.e., how to reduce the travel time of vehicles driving within a network. This network can be seen as a graph $G = (N, E)$, where N is the set of nodes that operate as junctions/intersections, and E is a set of directed links (or edges, as both terms are used interchangeably) that connect the nodes. Hence the goal is then to assign vehicles to routes so that the travel time is minimized. For more details, the reader is referred to Chapter 10 in [Ortúzar and Willumsen 2011]. For our purposes it suffices to mention that classical approaches aim at planning tasks, are centralized (i.e., trips are *assigned* by a central authority, not *selected* by individual drivers). Also, the main approaches are based on iterative methods that seek convergence to the user equilibrium (see next).

2.2 User Equilibrium

When it comes to reaching a solution to the TAP, one can take into account two perspectives: one that considers the system as a whole, and one that considers each user's point of view. In the system perspective, the best solution refers to the system reaching the best average travel time possible; this is the so called system optimum (SO), or Wardrop's second principle [Wardrop 1952]. We stress that the SO is a desirable property, but hardly achievable given that it comes at the cost of some users, who are not able to select a route leading to their personal best travel times. On the other hand, at the user's perspective, the system reaches the user (or Nash) equilibrium (UE) when there is no advantage for any individual to change its routes in order to minimize their travel time, as stated in the first Wardrop's principle [Wardrop 1952].

2.3 Reinforcement Learning

Reinforcement learning (RL) is a machine learning method whose main objective is to make agents learn how to map a given state to a given action, by means of a value function. RL can be modeled as a Markov decision process (MDP), where there is a set of states S, a set of actions A, a reward function $R : S \times A \to \mathbb{R}$, and a probabilistic state transition function $T(s, a, s') \to [0, 1]$, where $s \in S$ is a state the agent is currently in, $a \in A$ is the action the agent takes, and $s' \in S$ is a state the agent might end up, taking action a in state s, so the tuple (s, a, s', r) states that an agent was in state s, then took action a, ended up in state s' and received a reward r. The key idea of RL is to find an optimal policy π^*, which maps states to actions in a way that maximizes future reward.

RL methods fall within two main categories: model-based and model-free. While in the model-based approaches the reward function and the state transition are known, in the model-free case, the agents learn R and T by interacting with an environment. One method that is frequently used in many applications is Q-Learning (QL), which is a model-free approach. In QL, the agent keeps a table of Q-values that estimate how good it is for it to take an action a in state s, in other words, a Q-value $Q(s, a)$ holds the maximum discounted value of going from state s, taking an action a and keep going through an optimal policy. In each learning episode, the agents update their Q-values using the Equation 1, where α and γ are, respectively, the learning rate and the discounting factor for future values.

$$Q(s, a) = Q(s, a) + \alpha(r + \gamma max_a[Q(s', a') - Q(s, a)]) \tag{1}$$

In a RL task, it is also important to define how the agent selects actions, while also exploring the environment. A common action selection strategy is the ϵ-greedy, in which the agent chooses to follow the optimal values with a probability $1 - \epsilon$, and takes a random action with a probability ϵ.

3. RELATED WORK

Solving the TAP is not a new problem; there have been several works that aim at solving it. Besides classical methods (see Chapter 10 in [Ortúzar and Willumsen 2011]), which mostly deal with planning tasks, RL is turning popular.

When we refer to RL methods to solve the TAP, these usually fall into two categories: a traditional RL method, and a stateless one. Contrarily to the traditional approach, in the stateless case, the agents actually have only one state that is associated with its origin-destination pair, and they choose which actions to take. Actions here correspond to the selection of one among k pre-computed routes. Works in this category are [Ramos and Grunitzki 2015] (using a learning automata approach), and [Grunitzki and Bazzan 2017] (using QL). In [Zhou et al. 2020] the authors used a learning automata approach combined with a congestion game to reach the UE. [Tumer et al. 2008] adds a reward shaping component (difference utilities) to QL, aiming at aligning the UE to a socially efficient solution.

Apart from the stateless formulation, in the traditional case, agents may found themselves in multiple states, which are normally the nodes (intersections) of the network. Actions then correspond to the selection of one particular link (edge) that leaves that node. In [Bazzan and Grunitzki 2016] this is used to allow agents to learn how to build routes. However, they use a macroscopic perspective by means of cost functions that compute the abstract travel time. In the present paper, the actual travel time is computed by means of a microscopic simulator (details ahead).

As aforementioned, our approach also includes C2I communication, as these kinds of new technologies may lead agents to benefit from sharing their experiences (in terms of travel times), thus reducing the time needed to explore. The use communication in transportation systems, as proposed in the present paper, has also been studied previously ([Grunitzki and Bazzan 2016], [Bazzan et al. 2006], [Koster et al. 2013], [Auld et al. 2019]). In some cases, the information is manipulated to bias the agents to reach an expected outcome. In a different perspective, works like [Yu et al. 2020] evaluate the impact of incomplete information sharing in the TAP.

4. PROPOSED METHOD

Our approach is based on using communication to augment the information each agent[1] has and, hence, the learning performance. Section 4.1 gives a brief explanation on representation of the infrastructure. Section 4.2 focuses on communication. Section 4.3 presents the details of the RL algorithm.

[1]Henceforth, the term agent is used to refer to a vehicle agent.

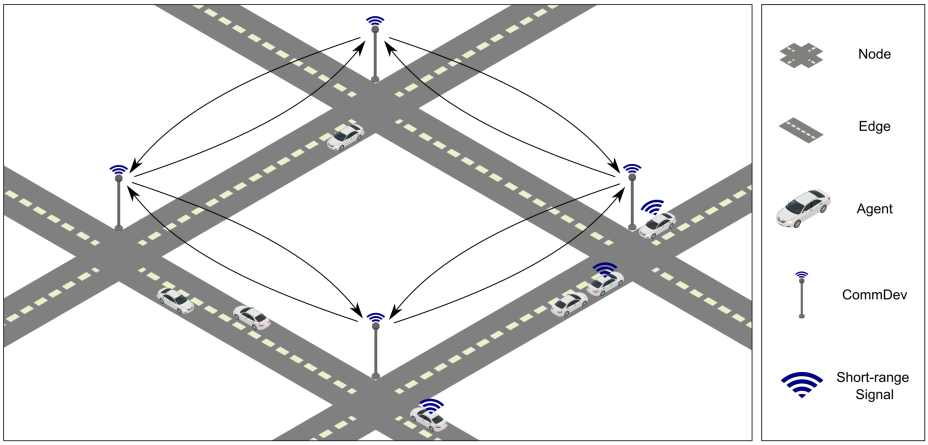

Fig. 1: Scheme of the communication infrastructure

4.1 Representing the Infrastructure

We assume that every node $n \in N$ present in the network G is equipped with a communication device (henceforth, CommDev) that is able to send and receive messages in a short range signal (e.g., with vehicles around the intersection). Figure 1 shows a scheme that represents G and the CommDevs. Using the short-range signal, the CommDevs are able to communicate with vehicles that are close enough, and are able to exchange information related to local traffic data (refer to Section 4.2 for details). Moreover, these CommDevs are able to store the data exchanged with the agents in order to propagate this information to other agents that cross the intersection in the near future. The arrows that connect CommDevs in Figure 1 represent a planar graph, meaning that every CommDev is connected and can communicate to its neighboring devices. This is necessary for CommDevs to get information about the traffic situation in neighboring edges, which is then passed to agents.

4.2 How Communication Works

Every time an agent reaches an intersection, prior to choosing an action (the next intersection to visit), it communicates with the intersection's CommDev (see Figure 1) to exchange information. The actual piece of information sent from agents to CommDevs is about the rewards received by the agents (refer to Section 4.3 for details on the rewards definition) regarding their last action performed. Conversely, the infrastructure communicates to the agent an aggregation of rewards informed by other agents regarding the actions available in the current state. The agent then perceives this information as expected rewards for the actions available to it. More details are given next.

4.2.1 *Information Hold by Infrastructure.* The infrastructure uses queue based data structures to hold the rewards (for each agent), as informed by the agents. These queues have a length (in the experiments, this was set to 30), and when new information arrives after the queue is full, the oldest information is discarded to make room to the most recent one. Since the information provided to agents consists of an aggregation of these values stored (e.g., an average of them), the length of these queues have an impact on how updated the expected reward will be. The smaller the queue length, the higher the influence of the newest reward values.

4.2.2 *Information Used by the Agent.* In a standard QL algorithm, the agents update their Q-values based on the feedback from the action they have just taken. However, in our case agents also update their Q-values based on the expected rewards received by the infrastructure. This means that every time they reach an intersection, they update their Q-values with the information provided by

the CommDevs. It is worth noting that the information received from the CommDev only concerns actions that can be selected in his current state, and since we do not want the bonus reward to be overwritten, we consider that agents discard information regarding edges that lead to their destination node.

4.3 Algorithm

Given a network G, every agent (vehicle) $v \in V$ has a pair $(o, d) \in N \times N$, that defines its origin-destination pair (OD-pair). Nodes $n \in N$ are seen as states the agents might be in, and the outgoing edges of a node are the possible actions for that given state, thus the agents build their routes on-the-fly by visiting nodes and edges. Upon choosing an action (edge) e, v perceives its reward (being a microscopic model, this reward is actually computed by the simulator). Assuming that the simulator reports a travel time of t_e^j for agent v traveling on edge e, the reward is $-t_e^v$, as we want to make sure the agents prefer to take edges that minimize travel times.

This alone does not guarantee that the agents will reach their destination, as they might end up running in loops throughout the network. Hence a positive bonus B is given to each agent that reaches its destination, giving them incentives to end their trips rather than wander around.

We deal with a commuting scenario, where each agent performs day-to-day experiments in order to reach an equilibrium situation, in which no agent can reduce its travel time by changing routes. Because agents belong to different OD pairs and/or select different routes, their trips take a different number of simulation steps. These steps represent elapsed seconds in simulation time. Hence, this means that not every agent finishes its trip simultaneously and, therefore, the standard notion of a learning episode cannot be used here. Rather, each agent has its own learning episode that will take as many simulation steps as necessary to reach its destination. Next, we explain the main parts of our approach, which can be seen in Algorithm 1.

Line 1 list the inputs of Algorithm 1: G is the topology of the network, D is the demand (flow rate) that will be running constantly, P is the set of OD-pairs, and M is the maximum number of steps to simulate. It is also necessary α, γ, bonus B, and ϵ for QL. The main loop is presented between lines 3 – 17, where the learning and the communication actually take place. The first *if* statement shown in line 5 takes care of all agents that finished their trips in the current step: they receive their reward

Algorithm 1 QL with C2I

```
1:  Input: G, D, P, M, α, γ, ϵ, B
2:  s ← 0
3:  while s < M do
4:    for v in V do
5:      if v.finished_trip() then
6:        v.update_Q_table(B − v.last_edge_travel_time)
7:        G.commDev[v.curr_node].update_queue(v.last_reward, v.last_edge)
8:        v.start_new_commuting_trip()
9:      else if v.has_reached_a_node() then
10:       v.update_Q_table(−v.last_edge_travel_time)
11:       G.commDev[v.curr_node].update_queue(v.last_reward, v.last_edge)
12:       v.update_Q_values(G.commDev[v.curr_node].info)
13:       v.choose_action()
14:     end if
15:   end for
16:   s ← s + 1
17: end while
```

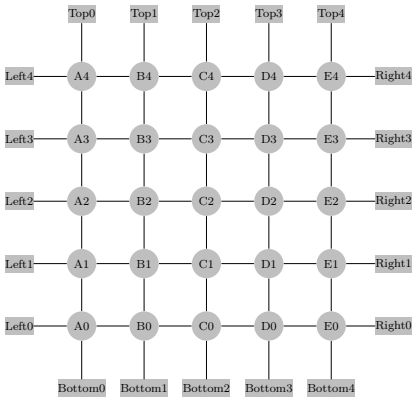

Fig. 2: Network used as scenario

Origin	Destination	Flow
Bottom0	Top4	102
Bottom1	Top3	86
Bottom3	Top1	86
Bottom4	Top0	102
Left0	Right4	102
Left1	Right3	86
Left3	Right1	86
Left4	Right0	102

Table I: Distribution table of the demand over the OD-pairs

plus the bonus for finishing the trip. In line 7, the agent informs CommDevs the rewards, and since its trip has ended, it gets reinserted at the origin node. The second *if* statement (Line 9) represents the intermediary nodes, where agents also receive their rewards and inform the CommDevs (line 11), so that these can update their queue structures. CommDevs also inform agents about the expected rewards from the actions agents might take next (line 12). Agent then updates their Q-values and choose action.

5. EXPERIMENTS, RESULTS, AND ANALYSIS

5.1 Scenario: Network and Demand

Simulations were performed using a microscopic tool called Simulation of Urban Mobility (SUMO [Lopez et al. 2018]). SUMO's API was used to allow vehicle agents to interact with the simulator "en route", i.e., during simulation time. The scenario chosen is a 5x5 grid depicted in Figure 2; each line in the figure represents bi-directed edges containing two lanes, $200m$ long.

The demand was set to maintain the network populated at around 30% of its maximum capacity, (considering a vehicle occupies $5m$), which is considered a high occupation. This demand was then distributed between the OD-pairs as represented in Table I. The last column represents the flow per OD-pair on any instant of the simulation. Those values were selected so that the shorter the path, the smaller the demand, which seems to be a more realistic assumption than a uniform distribution of the demand.

5.2 QL Parameters

A study conducted by [Bazzan and Grunitzki 2016] shows that, in an en route trip building approach, the learning rate α does not play a big role, and so a value of $\alpha = 0.5$ suits our needs. As for the discount factor γ, we have performed extensive tests and found that a value of $\gamma = 0.9$ performs best. For the epsilon-greedy action selection, empirical analysis led using a fixed value of $\epsilon = 0.05$. These values guarantee that the agents will mostly take a greedy option (as they only have a 5% chance to make a non-greedy choice), and also take into account that the future rewards have a considerable amount of influence in the agent's current choice, since γ has a high value. For the bonus part at the end of each trip, after tests, a value of $B = 1000$ was used, as this value manages to compensate possible jams close to the agents destination. We remark that trips take an average of roughly 450 time steps thus this value of B fits the magnitude of the rewards.

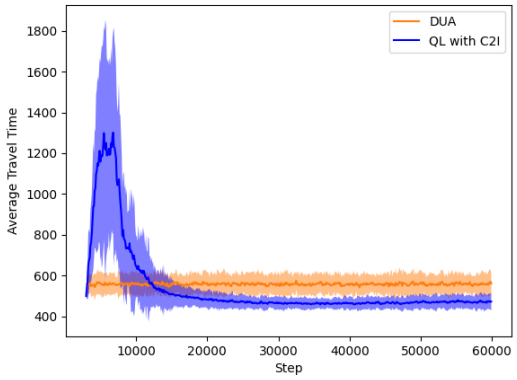

Fig. 3: QL w/ C2I vs DUA

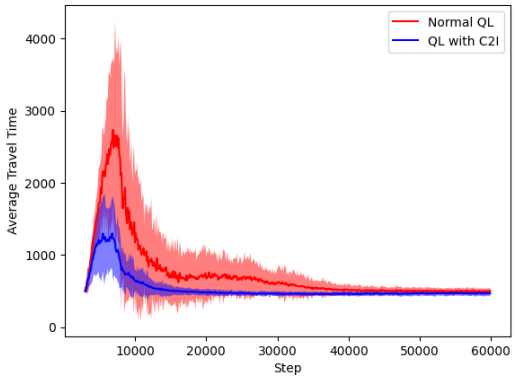

Fig. 4: QL w/ C2I vs Normal QL

5.3 Performance Metric and Results

To measure the performance, a moving average of the travel times is used.

Given the probabilistic nature of the process, it is necessary to run repetitions of simulations. Thus, 30 runs were performed. Plots show a comparison between the QL with C2I and two other approaches.

Figure 3 shows a comparison between our QL with C2I algorithm and a method called Dynamic User Assignment (DUA), which is an interactive method implemented by the SUMO developers. We remark that DUA is a centralized approach. This tool performs iterative assignment of routes to the given OD-pairs in order to find the UE [2]. In our tests, DUA was run for 100 iterations. The output of DUA is a route that is then followed by each vehicle, without en route changes. Since DUA has also a stochastic nature, we show results for 30 repetitions of the simulation using DUA.

Our approach outperforms DUA. The figure shows that, obviously, at the beginning, the performance of our approach reflects the fact that the agents are still exploring. However, after a certain time, the agents have learned a policy to map states to action and, by using it, they are able to reduce their travel times. We remark that, even after step $20,000$, agents still explore with probability ϵ thus there is room for improvements if other forms of action selection are used.

In the other experiment (seen in Figure 4), our approach is compared to a traditional QL algorithm, i.e., with no communication involved. This way, the learning approach follows basically the methods discussed in Section 2.3. This means that the agents learn their routes only by their own previous experiences, without any knowledge regarding the traffic situation and the experiences of other agents.

We can divide the learning process in both cases shown in Figure 4 in two distinct phases: the exploration phase, where the agents have yet no information about the network and explore it to find their destination, that is when the spikes in the learning curves can be seen; and the exploitation phase, when agents know the best actions to take in order to have the lowest travel time possible. Both approaches converge to the same average travel times in the exploitation phase. However, the advantage of our approach comes in the exploration phase. As we see in Figure 4, the exploration phase in the QL with C2I algorithm is reduced by a considerable amount when compared to the traditional QL algorithm, meaning that in our case the equilibrium is reached earlier.

[2]For details on how the DUA method is made the reader may refer to `https://sumo.dlr.de/docs/Demand/Dynamic_User_Assignment.html`

6. CONCLUSIONS AND FUTURE WORK

A wise route choice is turning more and more important when the demand is increasing and the networks are not being expanded in the same proportion. MARL is an attractive method for route choice, as it mimics the way drivers perform experimentation in their daily commuting. The present paper presented a method that combines MARL with C2I communication. Vehicles interact with the infrastructure every time they reach an intersection, exchanging traffic information regarding the values of their last action and the expected values regarding their next possible ones.

We compared our approach with two others and ours outperformed both of them. When compared to the DUA method, our method seems to be a closer approximation of the UE. When compared to the traditional QL, with no communication, our method accelerated the learning process by a considerable amount.

This work focused in using the QL algorithm combined with C2I in a way to accelerate the process of reaching the UE. Nevertheless in a real world scenario, it is quite possible that not all vehicles will have a way to communicate with the infrastructure, and this limitation is something that should be addressed in a future work. Another possible investigation can be the addition of a biased information provided by the infrastructure in order to reach a different outcome, namely, to reach the system optimum (socially efficient distribution of routes to vehicles).

REFERENCES

Auld, J., Verbas, O., and Stinson, M. Agent-based dynamic traffic assignment with information mixing. *Procedia Computer Science* vol. 151, pp. 864 – 869, 2019.

Bazzan, A. L. C., Fehler, M., and Klügl, F. Learning to coordinate in a network of social drivers: The role of information. In *Proceedings of the International Workshop on Learning and Adaptation in MAS (LAMAS 2005)*, K. Tuyls, P. J. Hoen, K. Verbeeck, and S. Sen (Eds.). Number 3898 in Lecture Notes in Artificial Intelligence. Utrecht, pp. 115–128, 2006.

Bazzan, A. L. C. and Grunitzki, R. A multiagent reinforcement learning approach to en-route trip building. In *2016 International Joint Conference on Neural Networks (IJCNN)*. pp. 5288–5295, 2016.

Grunitzki, R. and Bazzan, A. L. C. Combining car-to-infrastructure communication and multi-agent reinforcement learning in route choice. In *Proceedings of the Ninth Workshop on Agents in Traffic and Transportation (ATT-2016)*, A. L. C. Bazzan, F. Klügl, S. Ossowski, and G. Vizzari (Eds.). CEUR-WS.org, New York, 2016.

Grunitzki, R. and Bazzan, A. L. C. Comparing two multiagent reinforcement learning approaches for the traffic assignment problem. In *Intelligent Systems (BRACIS), 2017 Brazilian Conference on*, 2017.

Koster, A., Tettamanzi, A., Bazzan, A. L. C., and Pereira, C. d. C. Using trust and possibilistic reasoning to deal with untrustworthy communication in VANETs. In *Proceedings of the 16th IEEE Annual Conference on Intelligent Transport Systems (IEEE-ITSC)*. IEEE, The Hague, The Netherlands, pp. 2355–2360, 2013.

Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., and Wiessner, E. Microscopic traffic simulation using sumo. In *The 21st IEEE International Conference on Intelligent Transportation Systems*, 2018.

Ortúzar, J. d. D. and Willumsen, L. G. *Modelling transport*. John Wiley & Sons, Chichester, UK, 2011.

Ramos, G. de. O. and Grunitzki, R. An improved learning automata approach for the route choice problem. In *Agent Technology for Intelligent Mobile Services and Smart Societies*, F. Koch, F. Meneguzzi, and K. Lakkaraju (Eds.). Communications in Computer and Information Science, vol. 498. Springer Berlin Heidelberg, pp. 56–67, 2015.

Tumer, K., Welch, Z. T., and Agogino, A. Aligning social welfare and agent preferences to alleviate traffic congestion. In *Proceedings of the 7th Int. Conference on Autonomous Agents and Multiagent Systems*, L. Padgham, D. Parkes, J. Müller, and S. Parsons (Eds.). IFAAMAS, Estoril, pp. 655–662, 2008.

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

Yu, Y., Han, K., and Ochieng, W. Day-to-day dynamic traffic assignment with imperfect information, bounded rationality and information sharing. *Transportation Research Part C: Emerging Technologies* vol. 114, pp. 59 – 83, 2020.

Zhou, B., Song, Q., Zhao, Z., and Liu, T. A reinforcement learning scheme for the equilibrium of the in-vehicle route choice problem based on congestion game. *Applied Mathematics and Computation* vol. 371, pp. 124895, 2020.