Experience Sharing in a Traffic Scenario

Ana L. C. Bazzan[1] and **Franziska Klügl**[2]

Abstract. Travel apps become more and more popular giving information about the current traffic state to drivers who then adapt their route choice. In commuting scenarios, where people repeatedly travel between a particular origin and destination, learning effects add to this information. In this paper, we analyse the effects on the overall network, if adaptive driver agents share their aggregated experience about route choice in a reinforcement learning (Q-learning) setup. Drivers share what they have learnt about the system, not just information about their current travel times. We can show in a standard scenario that experience sharing can improve convergence times for adaptive driver agents.

1 Introduction

Which route to choose for travelling from origin to destination is a central question a traffic participant – in our case, a driver – faces. The route choice of a driver however is not independent of the choices of others, as the load on a link determines the travel time on it and as a consequence the travel time on a route. This is the well-known problem of traffic assignment. Drivers adapt to their experience and choose routes which they expect to be the best choice – mostly with respect to shortest travel time. In traffic analysis this idea is manifested in the concept of user equilibrium, which is a situation in which no user can reduce travel time by changing its route.

There many approaches for driving the overall system into overall system optimum making individual drivers with incentives or information. Information just about recent travel times may be seen as miss-leading, as it just may give a one-shot impression. A consequence are ineffective oscillations based on too fast and overbearing reactions. A way to address this problem is by introducing a kind of inertia into the system, assuming that people are hesitant in following new information.

More and more drivers use travel apps that also show the traffic state collected from real-time speed observations, such as Google Maps or Waze. Drivers fuse the information that they receive from those apps with their individual experience to actually make their routing decision.

In this paper, we want to analyse whether sharing of experience among drivers rather than using recent travel time information enables the overall system to develop into the user (or Nash) equilibrium (UE).

Our analysis is inspired by a hypothetical app with which drivers share what they have learnt related to their repeated route choice between some origin an destination in a traffic system – like friends chatting about which route they prefer or avoid.

As discussed later in Section 5, there are some approaches that are based on sharing or transferring knowledge to improve or accelerate the learning process. However, the majority of them deal with cooperative environments. In non-cooperative tasks, such as the route choice, we observed that the transfer of a good solution from one agent to others may produce sub-optimal outcomes. Therefore, there is a need for approaches that address transfer of knowledge in other contexts. With the present paper, we provide the first steps in this direction.

The other sections of this paper are organised as follows. First, we introduce some background on route choice, traffic assignment analysis concepts and the usage of reinforcement learning. Section 3 then describes the proposed approach, while Section 4 discusses experiments performed with a standard scenario and results gained from that. Section 6 then presents the concluding remarks and points to future research.

2 Background: Traffic Assignment, Route Choice and Reinforcement Learning

2.1 The Traffic Assignment Problem

The traffic assignment problem (TAP) aims at assigning a route to each driver that wants to travel from its origin to its destination. In traffic analysis and simulation, traffic assignment is the prominent step of connecting the demand (how many drivers want to travel between two nodes in a traffic net?) and the supply (roads in a traffic net with particular capacities, speed, etc.). The TAP can be solved in different ways. Agent-based approaches aim at finding the UE [28], in which every driver perceives that all possible routes between its origin and destination (its OD pair) have similar, minimal costs resulting in that the agent has no incentive to change routes. This means that the UE corresponds to each driver selfishly computing a route *individually*.

A traditional algorithm to solve the TAP, (i.e. to find the UE) is the method of successive averages (MSA). Yet, this method is performed in a centralised way[3].

A learning based approach, on the other hand, can be done in a decentralised way, with each agent learning individually by means of RL (see next section). We remark however that, since their actions are highly coupled with other agents actions, this is not a trivial task. Moreover, it is a non-cooperative learning task.

2.2 Reinforcement Learning

In reinforcement learning (RL), an agent's goal is to learn a mapping between a given state to a given action, by means of a value function. A popular algorithm to compute such value functions is Q-learning (QL) [29]. Value functions take the instantaneous reward received (a numerical signal) and compute the expected, discounted value of the

[1] UFRGS, Porto Alegre, Brazil, email: bazzan@inf.ufrgs.br
[2] Örebro University, Sweden, email: franziska.klugl@oru.se

[3] For details see, e.g., Chapter 10 in [19].

corresponding action. Once such mapping is learned, an agent can decide which action to select in order to maximize its rewards.

We can model RL as a Markov decision process (MDP) composed by a tuple (S, A, T, R), where S is a set of states; A is a set of actions; T is the transition function that models the probability of the system moving from a state $s \in S$ to state $s' \in S$, upon performing action $a \in A$; and R is the reward function that yields a real number associated with performing an action $a \in A$ when one is in state $s \in S$.

Q-learning (QL) computes a Q-value $Q(s, a)$ for each action-state pair. This value represents a reward estimate for executing the action a at state s. The updating of $Q(s, a)$ is done through Eq. 1, where $\alpha \in [0, 1]$ is the learning rate and $\gamma \in [0, 1]$ is the discount factor.

$$Q(s, a) = Q(s, a) + \alpha(r + \gamma * max_a(Q(s', a')) - Q(s, a)) \quad (1)$$

Each agent is a learner that maintains a Q-table that is updated at each subsequent episode. In order to address the exploration–exploitation dilemma, the *ε-greedy* exploration strategy can be used to choose actions: the action with the highest Q value is selected with a probability of $1-\varepsilon$ and a random action is selected with probability ε.

Finally, even if agents may be learning independently (as, e.g., in [7]), this is a non-trivial instance of multiagent RL or MARL.

3 Sharing Experiences

3.1 Q-Learning in a Route Choice Scenario

Given is a road network G in which nodes represent point of interest, which can be junctions, neighborhoods, or areas; links represent road segments with a free-flow travel time depending on the distance and capacity and an additional cost function that relates number of vehicles on it to its costs (travel time). Some of the nodes serve as origins, some as destinations. Between those origin and destination nodes, we consider sequences of links as possible routes. Different routes connecting origin and destination are not independent from another, they may contain shared links. Such dependencies exists between routes of the same origin-destination combination as well as with routes from others.

Drivers use Q-learning to individually determine which is the best route they can take from their individual origin and destination pair. Hereby, S is the particular origin-node of the individual driver. Each driver learns independently. Each agent state is determined by its origin, i.e., this is a stateless formulation of QL, as in [7].

The set of actions A contains all k shortest routes between origin and destination. Therefore the number k needs to be sufficiently large to give sufficient flexibility in route choice. The transition function maps the origin to the destination for all actions. Reward is a function of the experienced travel time on the route (we use the negative of travel time, for maximization purposes).

It is important to actually understand the multi-agent characteristics of this learning task. Because links have travel times that depends on the number of agents using them, learning is challenging. The desirable, shortest path under free-flow, may end up producing a high travel time, if too many agents want to use it. Agents need to learn how to distribute themselves in a way that an optimal number of agents use each edge, minimizing their individual travel time on the selected routes. Coordination is desirable not just between agents with the same origin-destination pair, but needs to happen between all agents, even if their routes would share only a single link.

3.2 Information Sharing via App

With the dissemination of various traffic apps, it is no longer realistic to assume that drivers only learn from their own experience when repeatedly travelling from origin to destination just observing their own travel time. Carrying the idea of traffic apps beyond current traffic state, more information produced by others can be used for decision making. While there has been a number of works in which the effect of information about current travel times on route or on parts of the route was tested (see Section 5), the question emerged whether explicitly sharing *accumulated* information about particular routes could improve the learning process. We envision a travel app in which participating drivers can share experience about their favourite or most disappointing routes. The setup is visualised in Figure 1.

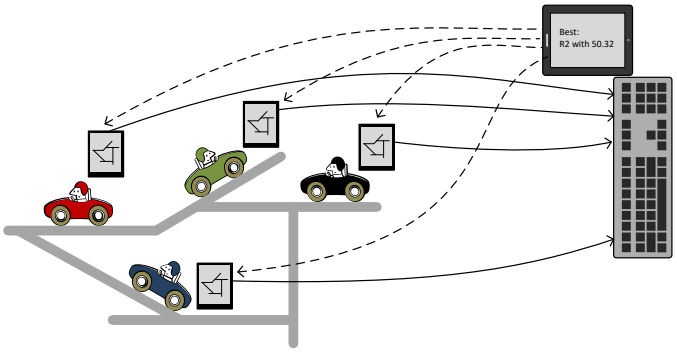

Figure 1. Setup of the system: Drivers decide about sharing their experience via an app with a server. The server processes the collected information and publishes the aggregated experience. Drivers may access this information depending on their information strategy.

The overall decision making results in three steps:

1. Agents select an action from their set of possible actions. Each agent experiences the time that it took to perform the travel using the route that corresponds to the selected action. The agent updates its Q-table according to the QL rules. At the end of the day, they share an evaluation of a route with the app. The shared information does not need to be about the route they took most recently, it may be their best or their worst or about a randomly route. The app can be seen as a platform for chatting "neighbours" who talk about what their experiences when travelling to a particular destination. Drivers may also decide to share no information at all. More precisely: sharing their accumulated experience means that they share the Q-value assigned to a particular action.
   The agents do not reason about the potential effect of sharing information on the reward they can receive. They do not assume that the publication of the information influences others' decision making to an extend that the evaluation of the published route changes dramatically.
2. The information server behind the app collects all information from the information sharing drivers and determines the shared information that the server then publishes. This information contains, per origin-destination pair, an action and the Q-value that belongs to the action aggregated from all handed-in information.
3. Driver agents decide whether to access the published information. Accessing has the consequence, that they incorporate the pub-

lished information in their Q-table: they replace the Q-value of the concerned action with the published Q-value for the given action. There is no reasoning about the credibility of the published value in relation to the existing Q-value. The decision of accessing the shared information is based on a budget acquired by sharing - the more one shares the more one may access the shared value. In the current version, the agents do not reason about whether the investment into acquiring published information may pay off, they rather access the information with a given frequency/probability, provided they have a budget.

This can be seen as a rudimentary form of transfer learning, with transfer happening during the learning process not from one task to another. Due to the way we modelled actions, transfer between different problems – here different origin-destination pairs makes no sense.

This is a multiagent reinforcement learning setup in which agents have to learn anti-coordination. Sharing happens in groups: only agent within the same group, that means with the same origin and destination pair share partial information from their Q-tables.

3.3 Sharing Issues in Detail

More details need to be considered to provide full information about the analysed scenarios:

The overall dynamics are so, that agents select their route before the actual travelling happens. Then, all agents travel through the network. Using cost functions for determining travel time on a link from its load, we abstract away from detailed temporal dynamics in which departure time, etc. matters. As we want to focus on the effect of sharing experiences and different aspects related to that, we did not use a microscopic traffic simulation to determine how an individual driver moves through the network. As a consequence, all agents taking the same route encounter the same travel times. Reward from travelling is sent to drivers after all have finished. The agents update their Q-tables, send information to the app. In the next episode, the drivers who want to access the published information, look into the service and again update their Q-tables based on the published value. So, there are a number of decisions involved:

3.3.1 When does a driver access the shared information?

Assuming that the driver needs to pay for the service of getting such information, the decision about whether and when it is best to incorporate information from other agents into ones Q-table may be a strategic one.

1. the simplest strategy could be to simply access information in every round.
2. the agent could collect some budget – e.g. from sharing information – that it could invest into accessing information. So whenever the agent has collected enough budget, it accesses. The relation between gain per sharing in relation to costs of accessing is the relevant parameter here.
3. The agent could access the information in the beginning with decreasing frequency the longer the learning proceeds.
4. One can image more advanced strategies based on dynamics that the agent observes. For example, if the action with the best Q-value was “disappointing” for a number of times, the agent may trigger a look-up on the app.
5. An alternative could be to have a look, if the Q-values are not sufficiently distinct to have a clear favourite route.

We decided to focus on the simpler alternatives and leave the more sophisticated approaches to future work, as they involve a lot of individual parameters to fully specify such strategies. Hence, in this work we follow alternative 2, varying the budget agent has, starting with unlimited budget and then decreasing it so that agents only access the information in some episodes. Although we consider that all have the same budget, the access is not synchronous, i.e., not everybody spend their budgets in the same episode (except in case budget is unlimited and thus every agent access at every episode).

3.3.2 Whether, when and with whom to share information?

The agents learn about good actions/routes for their origin-destination combination, individually. In principle, they do not need to share – yet without a sufficiently large subset of the driver population sharing, the information that the app publishes may not possess any value. So, it is in the interest of the app providers that as many agents as possible hand-in their experience.

There might be some required consent for basic service usage to send information about the agents’ experience to the app. This has the consequence that basically everybody would share experiences much like GPS traces are collected without full awareness of the (careless) traveller.

As an alternative, one can image that there is a small compensation for actively sharing experiences. The budget system indicated above would need a corresponding side for earning what will be spend later. For every time sending information, the agent could increase the budget for retrieving information.

The compensation for sharing could be also independent from the information usage. It could be financial or give social credits turning the individual agent to somebody more compliant with societal values.

For this first analysis, we just analysed situations in which every agent is basically forced to contribute with its experience, that means agents share

A variant of this question is related to with whom an agent may share. The idea of an app that publishes a value aggregated from all handed-in experiences, is one extreme: the agent shares with everybody that travel between the same origin and destination. Whether the information that the agent contributes is actually valuable is a decision of the aggregation mechanism used by the app. As an alternative, one can image to restrict the information dissemination to only a group of agents, denoting a group of friends or neighbours. This could bring more heterogeneity into the information transfer. As an extreme case, agents could pair and just share information with one other agent, this thwarts the app idea. Additionally, we did not expect a large impact and therefore just tested without restrictions. In the future work, we need to test this assumption.

3.3.3 What information shall be shared and how the information shall be aggregated?

This is the central question for the setup of the app. What information shall be shared and how does the app process all the information collected?

The first question relates to what information the agents do send to the app. It is obvious that this information can be information about the route with the highest Q-value - basically telling others that this route is the best for the agents’ origin-destination pair according to the agents’ experience. Yet, also other information may be interesting

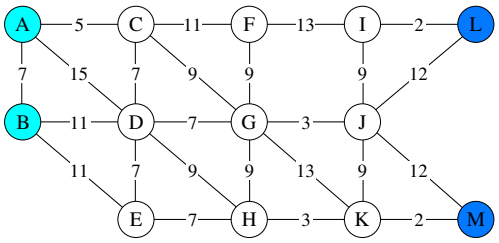

Figure 2. OW Road Network (as proposed by Ortuzar and Willumsen) [19]

to know: There are situations in which it is better to avoid choices. So, we foresee the following alternatives to be relevant:

1. Share the last chosen action including its Q-value
2. Share a randomly selected action with its Q-value
3. Share the action with the best Q-value
4. Share the action with the worst Q-value

As the first alternative in most cases is the same as the third one and otherwise a random selection, we deemed this alternative to be not interesting. We tested all others. The agents hereby share the information. If they access the aggregated information, they integrate it into their own individual Q-table replacing the Q-value of the concerned route/action. So, they do not automatically select the route that they were told about, but only if there is no better one and continue with the usual learning and decision making process.

As written above, the app collects and aggregates information from all sharing drivers. Also for this overall process step, there are different alternatives:

1. the app publishes a random value
2. the app publishes the overall best q-value and corresponding action
3. the app publishes the overall lowest q-value with corresponding action
4. the app averages the q-values for each possible action and then give best or worst of these average

Together with the alternatives for which values to share, this produces a number of interesting alternative combinations: it might be interesting to share randomly selected information from all best or all worst q-values, yet we tested the "pure" combinations: The app provides a fully randomly selected information, the best of all best and the worst of all worst q-values.

4 Experiments and Results

4.1 Scenario

We performed a number of experiments to understand how the sharing actually impacts the adaptation and eventually the performance of the drivers using the app.

The OW network (proposed in Chapter 10 in [19]) seen in Figure 2, represents two residential areas (nodes A and B) and two major shopping areas (nodes L and M). So, there are four OD pairs, connecting residential areas with shopping areas. The numbers in the edges are their travel times between two nodes under free flow (in both ways). We assume no additional delays when passing nodes, e.g. due to traffic lights. We computed $k = 8$ shortest paths per OD pair, following the algorithm by [30].

The volume-delay (or cost) function is $t_e = t_e^f + 0.02 \times q_e$, where t_e is the travel time on edge e, t_e^f is the edge's travel time per unit of time under *free flow* conditions, and q_e is the flow using edge e. This means that the travel time in each edge increases by 0.02 of a minute for each vehicle/hour of flow.

For the sake of getting a sense of what would be optimal, the free-flow travel times (FFTT) for each OD pair are shown in Table 1. We note however that these times cannot be achieved because the free-flow condition is not realistic, and when each agent tries to use its route with the lowest travel time, jams occur and the times increase. In addition the table shows the number of agents that travel between the four particular origin and destination combinations and the travel time in UE. This is averaged as drivers take different routes between their OD pairs. Also, the fact that 1700 agents learn simultaneously, makes the overall learning complex.

Table 1. OW network: some characteristics

OD Pair	Agents	shortest path	FFTT	Avg. travel time (in UE)
AL	600	ACGJIL	28	71.0
AM	400	ACDHKM	26	64.94
BL	300	BDGJIL	32	68.78
BM	400	BEHKM	23	62.3
	1700			67.13

For the QL, Q-tables are initialised with a random value around -90. All experiments were conducted with $\alpha = 0.5$ (learning rate) and $\varepsilon = 0.05$ (for ε-greedy action selection), following previous works (e.g., [2, 1]) that have extensively tried other values. As mentioned above, $k = 8$ is the number of shortest paths, that means number of possible actions.

We measure overall performance of the different approaches with the average travel time over all 1700 agents.

Each experiment was repeated 30 times. Standard deviations are not shown to keep the plots cleaner. We thus remark that they are of the order of 1% between the results of different runs in terms of average travel times. There are partially dramatic oscillations between episodes.

4.2 Experimental Results

4.2.1 Sharing Best Experience

We start by looking into cases in which the agents share their best experience.

We tested different settings of the budget strategy. Agents need different numbers of share-actions for being able to afford looking into what the app publishes. Figure 3 shows the results of testing scenarios in which an agent can access the information twice, 4, 6 or 8 times within 10 rounds. Agents decide individually, not all access at the same time, but in average over the episodes with the given relation. For comparison, we also display the learning curve if pure Q-learning is applied, that means in which agents learn without sharing Q-values via the app.

Figure 3 shows average travel time over all agents as well as over 30 experiments. It allows to compare the learning curves of different budget limitations for accessing the information that the app published. We can see that after 500 episodes pure QL shows convergence towards the UE, which is approximately 67 in the case of the OW network (see Table 1).

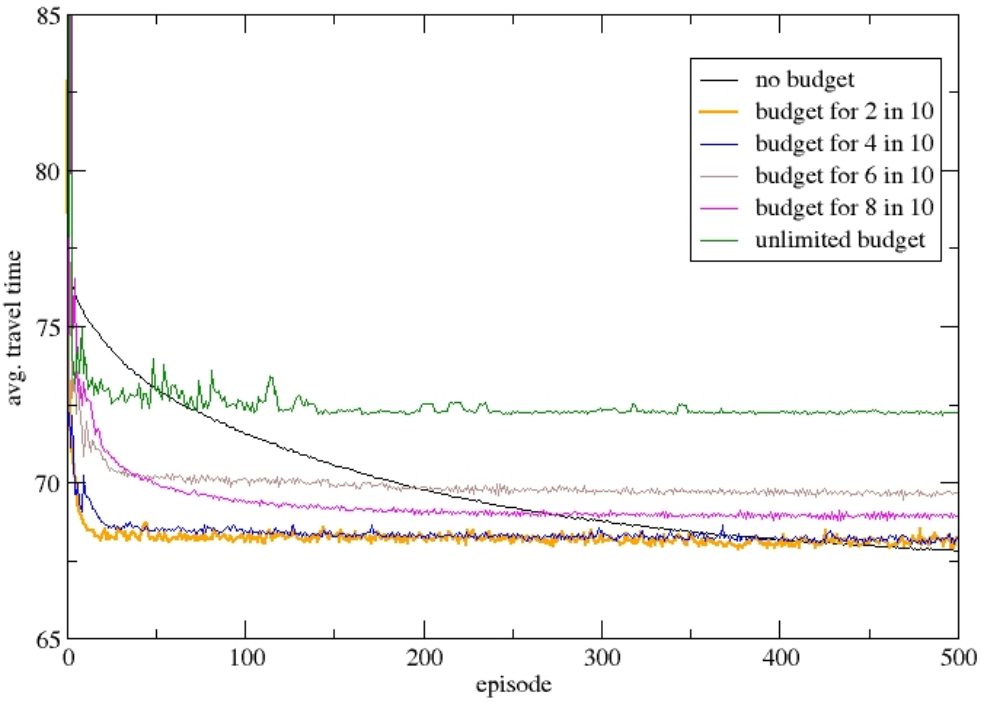

Figure 3. OW network: Development of overall averaged travel times over episodes for different sharing setup. Unlimited budget means that agents access the shared experience every round; no budget means no sharing.

As for the cases in which there is sharing of experiences, the following observations can be made. First, no matter how frequently the agents access the shared experience, we see an acceleration in the learning process in the initial episodes. This is clear in Figure 3, and is due to the fact that agents do not have to try out some actions given that they share better options. Unfortunately, depending on the frequency of access to this shared experience, this could lead to sub-optimal learning causing the collective of agents to not converge to the UE.

Let us now discuss individual cases. When agents have unlimited budget to access the shared experience (and fully use this budget), then the performance is bad (see green line in Figure 3). The reason is that too many agents are informed about the best performance in the previous episode and hence try to do the same action. Thus, the supposedly best action is followed by virtually all agents and they then end up selecting the same route (for each OD pair). This can lead to bad performance, not to mention a lot of oscillations (that are seen in the plot). This is a well known phenomenon and could be confirmed in our experiments.

When the budget is limited so that it allows the access to the shared experiences in 6 or 8 out of 10 episodes, then there is an increase in performance, if compared to the case in which agents have unlimited access. We recall that such accessing is not synchronous, i.e., some agents may access while others are not doing so. Still, the performance is sub-optimal (brown and magenta curves in the figure), with the collective of agents missing the UE in the end. It must be said, though, that the performance in the initial episodes is not bad, i.e., there is an acceleration in the learning due to the fact that some actions need not to be tried, as aforementioned.

When the agents have a budget that allows then to only access the shared experiences from time to time (e.g., 2 or 4 in 10 episodes, see blue and orange curves in Figure 3), then the overall performance is much improved, especially if the driver-vehicle units cannot afford to access the shared experiences in more than 2 out of 10 episodes. Not only there is an acceleration in the learning process already at the initial episodes, but also there is a convergence towards the UE. We assume that the best setting of the budget limits is depending on the scenario details, especially on the particular OD pair. It will be interesting to analyse the situation deeper. A budget strategy in which agents share in the beginning, but then gradually give up sharing over time, may lead to the results that we want to achieve with fast convergence to the UE.

4.2.2 Sharing Worst or Random Experience

We also tested the effect of what happens if agents share their worst experience which the app aggregates to the worst action/route for each particular OD pair. The results are almost identical to vanilla QL. So, sharing has apparently no effect when sharing the worst experience. The reason is that when exploiting - which is according to

our settings in 95% of the episodes - the agents select the best route. Which route has a bad Q-value does not matter hereby. So, the only situation in which such an update of the Q-table would change the decision of an agent is, if the route with previously the best Q-value, is the one that the app informs about. One could image that this happens if the previously best route is heavily overloaded, but this was not the case here. Maybe, the number of considered possible routes is too high for such a setup. Nevertheless, reducing the set of possible routes makes no sense in such a learning setting.

Also, randomly selecting a route from the Q-table and sharing it with the app, which then publishes a randomly selected experience from all handed-in values, generates a similar outcome as the sharing the worst experience. There is no difference to vanilla QL - that means a QL without sharing experiences. The explanation is similar.

5 Related Work

Besides the classical methods to address the TAP, there has been some works – based on other AI paradigms – with the goal of finding the UE. The main motivation is to solve the TAP from the perspective of the individual agent (driver, driver-vehicle unit), thus relaxing the assumption that a central entity is in charge of assigning routes for those agents. In such decentralized approaches, the agent itself has to collect experiences in order to reach the UE condition. Thus, these approaches are suitable for commuting scenarios, where each agent can collect experiences regarding the same kind of task (in this case, driving from a given origin to a given destination).

One popular approach to tackle such decentralized decision-making is via RL (more specifically MARL). However, other are also mentioned in the literature. We start with these and later discuss those based on RL.

In the work of [9], a neural network is used to predict drivers' route choices, as well as compliance to such predictions, under the influence of messages containing traffic information. However, the authors focus on the impact of the message on the agents rather than the impact on system-level traffic distribution and travel time.

The work of [10] uses the Inverted Ant Colony Optimization (IACO). Ants (vehicles) deposit their pheromones in the routes they are using, and the pheromone is used to repel them. Consequently, they avoid congested routes. Also [8] applied ant colony optimisation to the TAP. However, in both cases, the pheromone needs to be centrally stored, thus this approach is not fully suitable for a the decentralised modelling.

One game theoretic approach to the route choice problem appears in [12]. This approach uses only past experiences for the route choice. The choice itself is made at each intersection of the network. However, it assumes that historical information is available to all drivers.

The work of [24] uses adaptive tolls to optimise drivers' routes as tolls change. Differently from our purpose, they are concerned with alignment of choices towards the system optimum, which can only be achieved by imposing costs on drivers (in their case, tolls). In the same line, [5] deal with the problem from a centralised perspective to find an assignment that aligns users and system utilities by imposing tolls in some links.

In the frontier of aligning the optimum of the system with the UE, [1] has introduced an approach that seeks a balance in which not only the central authority benefits but also the individual agents.

RL-based approaches to compute the UE are becoming increasingly popular. Here, each agent seeks to learn to select routes (these are the actions) based on the rewards obtained in each daily commuting, where the reward is normally based on the experienced travel time.

Two main lines of approaches can be distinguished: One, less popular due to its complexity, follows a traditional RL recipe, where besides a set of actions, there is also a set of states, these being the vertices in which the agent finds itself. Works in this category include [4]. However, the majority of the papers in the literature follow a so-called stateless approach in which there is a single state (the OD pair the agent belongs to), while the agent merely has to select actions, which are normally associated with a set of pre-computed routes that can be recalculated en-route, or not. This literature includes approaches based both on Learning Automata ([17], [23]) and QL.

QL is increasingly being used for the task of route choice. Of particular interest are those that compute the regret associated with the greedy nature of selecting routes in a selfish way ([21, 20]), as well as those that combine selfish route choice with some sort of biasing from a centralized entity ([1, 6, 22]).

There are works that, as we propose in the current paper, also deal with some forms of communication between agents. [14, 3, 13, 16]. Information is here not always truthful, but can be manipulated for driving agents towards overall intended outcome.

The idea of sharing either learned policies or Q-values is not new. In fact, as early as in 1993, Tan [25] suggested that communication of some kind of knowledge could help, especially in cooperative environments. In particular, sharing Q-values may reduce the time needed to explore the space-action space.

Some researchers have dealt with aspects such as what and when to share. In a abstract view of sharing, [15] deal with agents that keep a list of states in which they need to coordinate. This idea also appears in the context of traffic management in [18], where traffic signal agents keep joint Q-tables based on coupled states and actions.

Some works – for instance, [26] – deal with more fine grained views of sharing knowledge, as for instance the transfer learning community, in which the quest regarding what and when to transfer gets more precise. Transfer of reward values or policies is also explored in the literature. For instance, [27, 11, 31] show that the learning can be accelerated if a teacher shares experiences with a student. We remind that virtually all these works deal with cooperative environments, where it makes sense to transfer knowledge. In non-cooperative tasks, such as the route choice, we have seen that transfer of a good solution from one agent to others may produce highly sub-optimal outcomes. Coordination here is actually anti-coordination, appropriately distributing agents between different alternatives.

6 Conclusion and Ideas for the Future

The idea of this paper is to discuss the effects of a possible next type of travel app, in which users intentionally share their experiences, as opposed to conventional travel apps, which are based on collecting drivers position, in order to be able to display current average speed at a particular segment. The app that we are considering here, can be thought as a device that replaces direct interaction between colleagues or neighbours chatting about habits and experiences regarding route choice.

Assuming that humans continuously adapt to what they experience when performing (commuting) tasks, we analysed the potential effect of such an app. So, agents can not just learn based on their own experience, but also use others' experience on the same task for decision making. As we have shown - integrating others' experiences from time to time - speeds up the learning progress.

In previous works, drivers were informed about travel times that were collected by a central authority (as, e.g., Waze). This is effective only if some inertia is introduced. We can observe a similar effect here: the agents who do not excessively use integrate others' experience, learn the best choice for their route. Agents that update their Q-table with additional information from the app in every round perform worst.

These findings are quite preliminary. More investigation is necessary to be able to claim general conclusions. We will test other interesting setups: combining different sharing and aggregation strategies. For example the agents may share their best experience, but instead of aggregating the information, the app publishes a randomly selected value from those sent. Another interesting idea could be the organisation of access into groups: instead of aggregating from all received and publishing to all who want to access, agents could be organised into smaller groups of friends who share/aggregate/publish only within the respective group.

As we have indicated in the previous sections, we also plan to test more dynamic and adaptive strategies: Agents may have a high initial budget and decide to use the budget in different strategies: a first such strategy could be that the agent uses its budget in the beginning of the learning process, when it is exploring more. A second alternative is that agents could spare their budgets and use them when they notice that the Q-Value of their best route is decreasing for a given number of rounds.

More innovatively, the app becomes an agent and actively adjusts the parameters of the budget strategies to improve the overall learning process. The app collects a lot of information from the driver agents, that can be used to drive the system towards the desired state, e.g. by telling the agents how to acquire more or less credits.

ACKNOWLEDGEMENTS

Ana Bazzan was partially supported by CNPq under grant no. 307215/2017-2.

REFERENCES

[1] Ana L. C. Bazzan, 'Aligning individual and collective welfare in complex socio-technical systems by combining metaheuristics and reinforcement learning', *Eng. Appl. of AI*, **79**, 23–33, (2019).

[2] Ana L. C. Bazzan and Camelia Chira, 'Hybrid evolutionary and reinforcement learning approach to accelerate traffic assignment (extended abstract)', in *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015)*, eds., R. Bordini, E. Elkind, G. Weiss, and P. Yolum, pp. 1723–1724. IFAAMAS, (May 2015).

[3] Ana L. C. Bazzan, M. Fehler, and F. Klügl, '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)*, eds., Karl Tuyls, Pieter Jan't Hoen, Katja Verbeeck, and Sandip Sen, number 3898 in Lecture Notes in Artificial Intelligence, pp. 115–128, (2006).

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

[5] Luciana S. Buriol, Michael J. Hirsh, Panos M. Pardalos, Tania Querido, Mauricio G.C. Resende, and Marcus Ritt, 'A biased random-key genetic algorithm for road congestion minimization', *Optimization Letters*, **4**, 619–633, (2010).

[6] Daniel Cagara, Bjorn Scheuermann, and Ana L.C. Bazzan, 'Traffic optimization on Islands', in *7th IEEE Vehicular Networking Conference (VNC 2015)*, pp. 175–182, Kyoto, Japan, (December 2015). IEEE.

[7] Caroline Claus and Craig Boutilier, 'The dynamics of reinforcement learning in cooperative multiagent systems', in *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, AAAI '98/IAAI '98, pp. 746–752, Menlo Park, CA, USA, (1998). American Association for Artificial Intelligence.

[8] Luca D'Acierno, Bruno Montella, and Fortuna De Lucia, 'A stochastic traffic assignment algorithm based on ant colony optimisation', in *Ant Colony Optimization and Swarm Intelligence, 5th International Workshop, ANTS 2006*, eds., M. Dorigo, L.M. Gambardella, M. Birattari, A. Martinoli, R. Poli, and T. Stützle, volume 4150 of *Lecture Notes in Computer Science*, pp. 25–36, Berlin, (2006). Springer-Verlag.

[9] H. Dia and S. Panwai, *Intelligent Transport Systems: Neural Agent (Neugent) Models of Driver Behaviour*, LAP Lambert Academic Publishing, 2014.

[10] José Capela Dias, Penousal Machado, Daniel Castro Silva, and Pedro Henriques Abreu, 'An inverted ant colony optimization approach to traffic', *Engineering Applications of Artificial Intelligence*, **36**(0), 122–133, (July 2014).

[11] Anestis Fachantidis, Matthew E. Taylor, and Ioannis P. Vlahavas, 'Learning to teach reinforcement learning agents', *Machine Learning and Knowledge Extraction*, **1**(1), 21–42, (2019).

[12] Syed Md. Galib and Irene Moser, 'Road traffic optimisation using an evolutionary game', in *Proceedings of the 13th annual conference companion on Genetic and evolutionary computation*, GECCO '11, pp. 519–526, New York, NY, USA, (2011). ACM.

[13] Ricardo Grunitzki and Ana L. C. Bazzan, '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)*, eds., Ana L. C. Bazzan, Franziska Klügl, Sascha Ossowski, and Giuseppe Vizzari, New York, (July 2016). CEUR-WS.org.

[14] F. Klügl and Ana L. C. Bazzan, 'Simulation studies on adaptative route decision and the influence of information on commuter scenarios', *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, **8**(4), 223–232, (October/December 2004).

[15] Jelle R. Kok and Nikos Vlassis, 'Sparse cooperative q-learning', in *Proceedings of the 21st. International Conference on Machine Learning (ICML)*, pp. 481–488, New York, USA, (July 2004). ACM Press.

[16] Andrew Koster, Andrea Tettamanzi, Ana L. C. Bazzan, and Célia da Costa Pereira, '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)*, pp. 2355–2360, The Hague, The Netherlands, (2013). IEEE.

[17] Kumpati S. Narendra and Mandayam A. L. Thathachar, *Learning Automata: An Introduction*, Prentice-Hall, Upper Saddle River, NJ, USA, 1989.

[18] Denise de Oliveira and Ana L. C. Bazzan, 'Multiagent learning on traffic lights control: effects of using shared information', in *Multi-Agent Systems for Traffic and Transportation*, eds., Ana L. C. Bazzan and Franziska Klügl, 307–321, IGI Global, Hershey, PA, (2009).

[19] Juan de Dios Ortúzar and Luis G. Willumsen, *Modelling Transport*, John Wiley & Sons, 3rd edn., 2001.

[20] Gabriel de O. Ramos, Ana L. C. Bazzan, and Bruno C. da Silva, 'Analysing the impact of travel information for minimising the regret of route choice', *Transportation Research Part C: Emerging Technologies*, **88**, 257–271, (Mar 2018).

[21] Gabriel de O. Ramos, Bruno C. da Silva, and Ana L. C. Bazzan, 'Learning to minimise regret in route choice', in *Proc. of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017)*, eds., S. Das, E. Durfee, K. Larson, and M. Winikoff, pp. 846–855, São Paulo, (May 2017). IFAAMAS.

[22] Gabriel de O. Ramos, Bruno C. da Silva, Roxana Rădulescu, and Ana L. C. Bazzan, 'Learning system-efficient equilibria in route choice using tolls', in *Proceedings of the Adaptive Learning Agents Workshop 2018 (ALA-18)*, Stockholm, (Jul 2018).

[23] Gabriel de O. Ramos and Ricardo Grunitzki, 'An improved learning automata approach for the route choice problem', in *Agent Technology for Intelligent Mobile Services and Smart Societies*, eds., Fernando Koch, Felipe Meneguzzi, and Kiran Lakkaraju, volume 498 of *Communications in Computer and Information Science*, 56–67, Springer Berlin Heidelberg, (2015).

[24] Guni Sharon, Josiah P Hanna, Tarun Rambha, Michael W Levin, Michael Albert, Stephen D Boyles, and Peter Stone, '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)*, eds., S. Das, E. Durfee, K. Larson, and M. Winikoff, pp. 828–836, São Paulo, (May 2017). IFAAMAS.

[25] Ming Tan, 'Multi-agent reinforcement learning: Independent vs. cooperative agents', in *Proceedings of the Tenth International Conference on Machine Learning (ICML 1993)*, pp. 330–337. Morgan Kaufmann, (June 1993).
[26] Adam Taylor, Ivana Dusparic, Edgar Galván López, Siobhán Clarke, and Vinny Cahill, 'Accelerating learning in multi-objective systems through transfer learning', in *2014 International Joint Conference on Neural Networks (IJCNN)*, pp. 2298–2305, Beijing, China, (2014). IEEE.
[27] Lisa Torrey and Matthew E. Taylor, 'Teaching on a budget: Agents advising agents in reinforcement learning', in *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems*, St. Paul, MN, USA, (May 2013). IFAAMAS.
[28] John Glen Wardrop, 'Some theoretical aspects of road traffic research', *Proceedings of the Institution of Civil Engineers, Part II*, **1**(36), 325–362, (1952).
[29] Christopher J. C. H. Watkins and Peter Dayan, 'Q-learning', *Machine Learning*, **8**(3), 279–292, (1992).
[30] Jin Y. Yen, 'Finding the k shortest loopless paths in a network', *Management Science*, **17**(11), 712–716, (1971).
[31] Matthieu Zimmer, Paolo Viappiani, and Paul Weng, 'Teacher-student framework: a reinforcement learning approach', in *AAMAS Workshop Autonomous Robots and Multirobot Systems*, Paris, France, (May 2014).