Learning-based traffic assignment: how heterogeneity in route choices pays off

Ana L. C. Bazzan
Instituto de Informática – UFRGS
Caixa Postal 15064, 91.501-970 Porto Alegre, RS, Brazil
bazzan@inf.ufrgs.br

ABSTRACT

An important stage in traffic modeling and planning is traffic assignment. For this, mainly an aggregate perspective has been taken, in which zonal data is considered. In contrast, if individuals are considered as active and autonomous agents, instead of having a central component assigning trips to links, agents do their actual route choices. This disaggregate perspective yields choices that are more heterogeneous because there is no batch assignment. A consequence is that the agents are able to distribute themselves in the network, thus using it in a better way. In this paper, a disaggregate, agent-based perspective is taken in which agents learn to select routes by selecting links at each node of the network, thus also addressing en-route changes in the known routes. To illustrate this approach, a non-trivial network is used and the results are compared to iterative methods that approximate the user equilibrium.

1. INTRODUCTION

Traffic assignment is an important stage in the task of modeling and simulating a transportation system. It connects the physical infrastructure and the demand that is going to use it, i.e., it assign trips to each link of the road network. Thus it appears as one of the stages in the so-called "four-stage models" of traffic modeling. Specifically, it is the fourth stage, the previous three being: trip generation, trip distribution, and modal split. The present paper deals with that last stage, hence how trips are generated (a function of attractiveness of certain zones of the network), their distribution (how many trips per zone), and modal split are not addressed (in fact, this paper only deals with vehicular traffic so that other modes are not relevant).

Classical methods for traffic modeling – including trip assignment – normally adopt an aggregate perspective, i.e., zone-based instead of individual-based. The reason is that it is simpler to get zonal data (how many trips originate or terminate there) than individual data, which may also include which intermediate activities each road user does during the trip, which knowledge it has, as well as its preferences for routes. Aggregate modeling assumes a centralized entity that controls those four stages. Hence, trips are *generated*, *distributed*, *split*, and *assigned*. In contrast, in a disaggregate perspective, one talks about trip *choice*, destination *choice*, mode *choice*, and route *choice*, in opposition to generation, distribution, split, and assignment respectively.

The disaggregate perspective naturally fits an agent-based approach and it is the one followed here. In it, agents do the actual choice (instead of being told which trip to make, which destination to go, etc.). Specifically, for the assignment stage, this means that each agent will choose its route based on *local*, partial knowledge. This may look trivial but makes a difference in terms of which knowledge must be available when one decides for an aggregate versus disaggregate approach. Moreover, it also means that the choices are as heterogeneous as possible. Ultimately, each agent can decide which route to take based on its individual behavioral rule. As this is a very complex approach (it is questionable if such behaviors can be collected at all, at least with the kind of technology and sensors that we have at this stage), the perspective in the present paper is that agents are heterogeneous only regarding the information they have, but not yet regarding completely heterogeneous behavioral rules. Of course, an intermediate situation could be that classes of agents with different behavioral rules could be modeled, as in discrete choice modeling [5]. However, since even this kind of data is not always available (e.g., how many percent of the agents are greedy, etc.), this paper assumes an homogeneous population w.r.t. behavior. Also, having classes of agent in the model would mean that this should be validated against some real-world situation for which the data is not available.

As mentioned, this paper takes an agent-based, disaggregate perspective for trip assignment. The selection of route is made by each agent, based on a reinforcement learning (RL) method. This means that agents have knowledge about the travel time for the shortest path between their origins and destinations, given an uncongested state of the network. However, they may explore other alternatives as well. Given that congestion may arise, this exploration is likely to make they exploit other routes. With this, a huge number of combinations of route or link choices arise in some links. The problem is not only complex due to this fact, but also because each agent is trying to learn in this environment. This is a multiagent learning problem, for which we know there is no guarantee of convergence to the optimum choice of the users. This user equilibrium can be found only for very simple networks, and they consider aggregate flows. Section 2 discusses this and approximate methods. However, they are not efficient and cannot handle fully individual choices, thus they miss a significant portion of the space of combinations

Appears in: *Proceedings of the 8th International Workshop on Agents in Traffic and Transportation (ATT-2014), Klügl, Vokrinek, Vizzari (eds.), held at the 13th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2014), May 5-6, 2014, Paris, France*

of route or link choices, eventually missing the optimal solution. Furthermore, classical methods do not handle en-route replanning, i.e., changes in the initially planned route during the actual trip.

In short, in this paper it is argued that a RL-based method at individual agent level, though not guaranteed finds the optimum solution for the trip assignment, has advantages over some classical methods. Perhaps the most important is that it allows a higher degree of heterogeneity in the choices of routes, without assuming a central authority that has global information, as it is the case with some classical trip assignment methods. The consequences of this is a better distribution of the road users in the road network. To illustrate this, the present paper uses a non-trivial scenario, in which some links are highly attractive to all agents, but produce severe congestion if all of them use those links in their trips.

This problem has not been adequately addressed in the literature. The traffic engineering literature mainly takes the aggregate perspective (see Section 2). When dealing with disaggregate modeling, it is not individual-based in the sense that each individual can make its own choice on link basis, as in the present paper. Rather, *portions* of the individuals make the same decision about which route to use. A coarse discretization has severe implications as discussed later. In the autonomous agents and multiagent systems literature, scenarios dealing with more than two or three routes, and those in which agents can change their routes on the fly are just beginning to be investigated. It is unclear what happens when drivers can adapt to traffic patterns in complex traffic networks. From the point of view of the whole system, the goal is to ensure reasonable travel times for all users, which can be conflicting with some individual utilities. Some of these works are discussed in Section 3.

Apart from background concepts, methods, and related work on trip assignment, which are discussed in the next two sections, Section 4 describes the proposed approach and the scenario used to illustrate it. Results are shown and analyzed in Section 5, while Section 6 presents the concluding remarks and points to future research.

2. TRAFFIC ASSIGNMENT METHODS

In this section, basic concepts about traffic (or trip) assignment methods are given. For an extensive explanation, please refer to Chapter 10 in [12] or to Chapter 4 in [3].

A road network can be represented as a graph $G = (V, E)$, where V is the set of vertices that represent the intersections of the network, and E is a set of directed arcs, describing the existing road segments as directed connections between pairs of vertices. Each link $l_k \in L$ has a cost c_k, which is given by a function that takes as input attributes such as length, toll, free-flow speed (and hence, free-flow travel time), capacity, current volume, etc. A route r_p is defined by a set of connected nodes $(n_0, n_1, n_2, ...)$. The length of each r_p is the sum of the lengths of all links l_k that connect these nodes.

Another relevant concept that needs to be introduced here is the one of volume-delay functions (VDFs) or cost-flow relationship. These are used in macroscopic modeling and aim at accounting for congestion effects, i.e., how over-capacity of a given volume or flow in a link affects the speed and travel times (costs of delays). These functions account for the flows in the whole network, i.e., they consider the interactions between flows that use the network at the same time, and the corresponding delays that may occur. As a simple example of a VDF, one can consider the following: $t_k = t_{k_0} + 0.02 \times q_k$. Here, t_k is the travel time on link k, t_{k_0} is the travel time per unit of time under free flow conditions, and q_k is the flow using link k. This means that the travel time in each link increases by 0.02 of a minute for each vehicle/hour of flow.

Given a demand T_{ij} for trips between origin i and destination j, there are several schemes to assign these trips to the links of a road network. Such schemes can be classified over two main dimensions: (1) are capacity constraints included?, and (2) are stochastic effects included? The classical scheme for situations in which there are no congestion effects and no stochasticity in route choices is the all-or-nothing scheme (discussed later). Stochasticity is handled by simulation-based methods. Assignment under congestion is of course a hot research topic and many approaches exist in this category. If one ignores the stochastic effects and focus on capacity constraints, the aforementioned concept of VDFs play a major role. For example, given VDFs for each link in the network, a goal of these approaches is to approximate the equilibrium conditions as stated by Wardrop [19]: "under equilibrium conditions traffic arranges itself in congested networks such that all used routes between an OD pair have equal and minimum costs while all those routes that were not used have greater or equal costs". This is Wardrop's first principle, also known as Wardrop's equilibrium or user equilibrium.

Thus, given a traffic network, the assignment from the point of view of the user equilibrium can be analytically stated as an optimization problem: find all flows from each OD pair s.t. only paths with minimal costs have a nonzero flow assigned to them, which corresponds to Wardrop's first principle. For a mathematical formulation of this problem, the reader is referred to Chapter 2 in [7], as well as to [14].

One problem with this scheme is that it is not possible to solve the equilibrium flows algebraically, except for very simple cases (e.g., two or three links connecting a single OD pair). Thus, approximate solutions to the Wardrop's equilibrium were proposed. To evaluate their quality, relevant issues are solution stability and convergence, as well as computational requirements.

Such approximate solutions are discussed later in this section. Before, it is important to introduce a general procedure that underlies any of the assignment schemes. Indeed, each assignment scheme discussed before has several steps that must be treated in turn: (i) to identify a set of routes that might be considered attractive to drivers; (ii) to assign suitable proportions of the trip matrix to these routes; this results in flows on the links of the network; (iii) to search for convergence: many techniques follow an iterative pattern of successive approximations to an ideal solution (e.g., Wardrop's equilibrium).

The first step can be accomplished with any variant of the Dijkstra algorithm for shortest paths. This step is also known as tree-building step. However, normally these paths are generated based on a first-approximation or an estimated cost function (e.g., one that considers no congestion, i.e., only free-flow travel times are considered) because the real cost is not known, given that it depends on the route choices of all users. Therefore, in a non-free-flow regime (i.e., under congestion), the second aforementioned step must be performed iteratively, until some sort of convergence is reached.

Next, some classical trip assignment approaches are discussed. The typical approach to trip assignment under no congestion is to assign all trips to the route with minimum cost, on the basis that these are the routes travelers would rationally select. That is as in Eq. 1, where T_{ij} is the given demand between origin i and destination j. This procedure is referred as "all-or-nothing" assignment. It is possible to see that this scheme assigns all trips between nodes i and j to the same links (because, as mentioned, this scheme assumes no congestion).

$$\left.\begin{array}{ll} T_{ijr^\star} = T_{ij} & \text{for the minimum cost route } r^\star \\ T_{ijr} = 0 & \text{for all other routes} \end{array}\right\} \quad \forall_{i,j} \quad (1)$$

For route assignment under congestion (i.e., the capacity of a link k can be surpassed and, as such, a VDF is necessary to account for the effects of the over-capacity), mainly two iterative methods can be used. The first is to load the network incrementally in n stages, e.g., assigning a given fraction p_n (e.g., 10%, 20%, etc.) of the total demand (for each OD pair) at each stage. Further fractions are then assigned based on the newly computed link costs. This procedure continues until 100% of the demand is assigned. Typical values for fractions p_n are 0.4, 0.3, 0.2, and 0.1. An algorithm for this is the following (adapted from [12]):

1. select an initial set of current link costs (usually the free-flow travel times); initialize flows at all links k: $V_k = 0$; select a fraction p_n of the trip matrix T such that $\sum_n p_n = 1$; make $n = 0$.

2. build the set of minimum cost trees (one for each origin) using the current costs; $n \leftarrow n + 1$.

3. load $T_n = p_n T$ all-or-nothing trips to these trees, obtaining a set of auxiliary flows F_k; accumulate flows on each link: $V_k^n = V_k^{n-1} + F_k$.

4. calculate a new set of current link costs based on flows V_k^n; if not all fractions of T have been assigned, proceed to step 2.

It must be remarked that there is no guarantee that this algorithm converges to the Wardrop's equilibrium, no matter how small each p_n is. This procedure has the drawback that once a flow has been assigned to a link, due to the accumulated nature (see step 3), it is never removed. Thus, in case an arbitrarily low over-capacity is assigned to a link, then it prevents the convergence to the optimum solution. However, it is very easy to program.

The other approach is to start from some initial values for the link costs and find the minimum cost routes. Trips are then assigned to these routes. New costs are computed and this cycle is repeated until there is no significant change in link or route volumes. For instance, in the method of successive averages, the flow at the n-th iteration is calculated as a linear combination of the flow on the previous iteration and an auxiliary flow resulting from an all-or-nothing assignment in the n-th iteration. This can be formalized as the following procedure (again, adapted from [12]):

1. select an initial set of current link costs (usually the free-flow travel times); initialize flows at all links k: $V_k = 0$; make $n = 0$.

2. build the set of minimum cost trees (one for each origin) using the current costs; $n \leftarrow n + 1$.

3. load the whole of the matrix T all-or-nothing to these trees obtaining a set of auxiliary flows F_k.

4. calculate the current flows as: $V_k^n \leftarrow (1 - \phi)V_k^{n-1} + \phi F_k$, with $0 \leq \phi \leq 1$.

5. calculate a new set of current link costs based on V_k^n; if no V_k^n has changed significantly in two consecutive iterations, stop; otherwise proceed to step 2 (or, alternatively, use a maximum number of iteration).

The last step of the method admits several ways to fix the value of ϕ. A useful one is to make $\phi = 1/n$. There is a proof that this produces solutions convergent to the Wardrop's equilibrium but this may be very inefficient.

Note that both iterative methods to approximate Wardrop's equilibrium are based on the all-or-nothing scheme (applied in each iteration). Thus, even for fine discretization levels, a number of trips is assigned to the same links.

3. RELATED WORK

A number of works from transportation planning and economics, as well as from mathematics and operations research, physics, and computer science deal with this problem. Computer science plays a role when it comes to solving large-scale road network problems. The most relevant and close to the approach proposed here are discussed next.

In [11], the author makes the point that trip assignment schemes that are based on steady-state flow conditions of the road network are adequate only for analysis of long-term strategic planning horizons, but not for tactical measures that are of interest in applications around intelligent transportation systems. However, the author also recognizes the challenges of finding an analytic representation that satisfies the laws of physics and traffic sciences, while also being mathematically tractable. Therefore they propose a simulation-based approach for the dynamical traffic assignment (DTA) problem. In DTA one goal is to describe how flows develop not only spatially but also temporally in the network. This means that DTA considers road users that depart from an origin to a destination at different times. Hence, they experience different travel times and, as such, the user equilibrium condition applies only to travelers who are assumed to depart at the same time between the same OD pair. In the present work, departure at different times is not considered, but the approach is not purely simulation-based given that the agents learn by interacting with the environment. DTA is also the focus of [16] in which the authors propose a predictive DTA model, also based on simulation and combined with the method of successive averages. Henn ([8]) proposes a fuzzy-based method to take the imprecisions and the uncertainties of the road users into account. These predict costs for each path based on a fuzzy subset that can represent imprecision on network knowledge, as well as uncertainty on traffic conditions.

As mentioned, a natural way to represent the problem of route choice (in opposition to trip assignment) is to model it using an agent-based modeling and simulation approach. Hence, there has been some works in this direction. An example is MATSim [1, 2], which deals with activity-based simulation of route choice.

Route choice under various levels of information is turning a hot research topic due to the increasing use of navigation devices. Agent-based route choice simulation has been applied to research concerning the effects of intelligent traveler information systems. Main questions here are what happens to the overall demand, if a certain share of drivers is informed and adapt. What kind of information is the best one to be given? Examples for such research line can be found in [9] for a two-route scenario, or in [6] where a neural net-based agent model for route choice is presented regarding a three route scenario. In [13], a simple network for fuzzy-rule based routing (including qualitative decisions) is used.

One problem with these approaches is that their application in networks with more than a couple of routes between a few locations is not trivial. The first problem is that a set of reasonable route alternatives has to be generated. A n-shortest path algorithm can be used but it may output routes that differ only marginally. Additionally, all approaches, including agent-based ones, consider one route as one complete option to choose. On-the-fly re-routing has hardly been a topic for research. Even more sophisticated agent architectures such as the one proposed by [15] do not include the possibility of re-routing during the trip.

To address this issue, re-routing in a scenario with multiple origins and destinations was studied in [4]. Besides route choice by the driver agents, the authors also consider traffic lights as adaptive agents in order to test whether such a form of co-adaptation may result in interferences or positive cumulative effects. This was one of the first works in the agent-based community that has dealt with agents computing new routes on the fly. This is important because en-route modifications cannot be ignored in a realistic simulation of decision making in traffic. An abstract route choice scenario was used, having some features of real world networks. However, in this work no comparison is made to methods that approximate the user equilibrium thus, it is not possible to fully assess the quality of those results.

Degradation in performance caused by the selfish behavior of individual road users remains an important research topic. [10] have proposed the so-called price of anarchy to measure this degradation. They show results for small networks such as the one used to illustrate the Braess paradox.

A learning-based approach was used by [17] where agents learn to select routes; thus there is no en-route changes in the routes. The size and topology of the network is not mentioned but it seems to be a single origin and destination.

4. APPROACH AND CASE STUDY

One of the problems with the methods discussed in Section 2 is that the set of routes that are considered in each step of the iterative process is reduced in order to gain in terms of computing time. However, this set can be far from the set that would be used by real world drivers, even if considering their informational constraints regarding the status of the traffic at the moment they make decisions. In other words, the granularity of the route selections is very coarse.

The approach proposed here can handle much finer granularities; actually, there is no limitation or restriction on the number and kinds of routes that users can select. This means that all possible routes can be combined (one for each agent), contrarily to schemes discussed in Section 2. This occurs because the granularity of those schemes is coarse per se. For example, the all-or-nothing approach is the extreme case where the whole volume for each OD pair is assigned to the same route. However, even less coarse methods as for instance the incremental method, still assigns the same route to a given fraction of road users. Of course these fractions can be small but the efficiency of this method decreases with the discretization (number of incremental steps). Similarly, in the successive averages method, the computational cost of the method depends on a good choice of the parameter ϕ. If it is a function of the parameter n (see Section 2), then the efficiency may be compromised.

In the iterative methods it is not possible to actually assign a different route to each road user at each iteration, as the method proposed here does. For this, this method pays a cost (more iterations are necessary as agents are learning while selecting routes) but it is still a tractable method given that each iteration typically takes just a few minutes. As it will be discussed further, the proposed method and the iterative methods present basically similar running times, but the former is heterogeneous in terms of combinations of individual route choices, thus exploring the possible search space in ways that are not possible with the iterative methods (without incurring in much higher running times).

The approach proposed here is based on RL. Agents learn the value of their actions by interacting with an environment that gives a feedback signal to each agent, based on which state the agent is in, and the action this agent decides to make while in that state. RL problems can be modeled as Markov decision processes (MDPs). An experience tuple $\langle s, a, s', r\rangle$ denotes the fact that the agent was in state s, performed action a and ended up in s' with reward r. Here, a popular model-free algorithm for RL is used, namely Q-learning. The update rule for each experience tuple $\langle s, a, s', r\rangle$ is given in Equation 2, where α is the learning rate and γ is the discount for future rewards.

$$Q(s,a) \leftarrow Q(s,a) + \alpha\ \left(r + \gamma\, max_{a'}\, Q(s',a') - Q(s,a)\right) \quad (2)$$

Considering a high number of agents in multi-agent RL turns the problem inherently more complex. This complexity has many causes and consequences, one being that mathematical convergence guarantees no longer hold.

The learning agents are the road users; the environment is a road network, where nodes form the set of states an agent may be, and the links departing from each node form the set of actions an agent may take. Each agent has an origin and a destination. Routes connecting these two are represented as a set of consecutive nodes. Of course there are many ways in which a destination node can be reached. Because links have a travel time that depends on the number of agents using them in their route choices, this problem is complex. Mostly, the desirable, shortest path under free-flow, may end up producing a high travel time if too many agents want to use it. Agents then need to learn how go from their origins to their destinations by finding routes that allows them to distribute themselves in such a way that the optimal number of agents use each link, minimizing their travel times.

In order to address a non trivial network, the one suggested in [12] (Exercise 10.1) is used, as depicted in Figure 1. All links are two-way. This network represents two residential areas (nodes A and B) and two major shopping areas (nodes L and M). The numbers in the links are their travel times under free flow (in both ways). These also appear in the second column of Table 3. For the shortest path algo-

Table 1: Shortest Paths and Free-Flow Travel Times (FFTT) for the Four OD Pairs (original and modified networks).

OD	original		modified	
pair	sh.st path	FFTT	sh.st path	FFTT
AL	ACGJIL	28	ACDGJIL	23
AM	ACDHKM	26	ACDGJKM	23
BL	BDGJIL	32	BDGJIL	22
BM	BEHKM	23	BDGJKM	22

rithms, these can be seen as their costs so henceforth both terms are used indistinctly.

Further, in order to make the assignment more complex, two modifications in the fixed costs were made: to make an arterial more attractive to all road users (and hence the learning effort more difficult as there is more competition for cheap resources), the fixed costs of links DG and GJ (and their opposite directions as well) were reduced from, respectively, 7 and 3 to zero. These modifications are indicated in Table 3 by shadowed cells. This way the shortest paths, for each OD pair, and their travel times (under free-flow) are shown in Table 1, both for the original network and for the modified network. Note that the proposed approach (as well as the iterative methods) were run for both the original and for the modified versions but since the latter is more challenging, only results steaming from the experiments using the latter are mentioned. Notice, however, that the general conclusions are valid for both, i.e., the RL-based approach outperforms the other methods. Henceforth this network is referred as OW network.

In the experiments, 1700 driver agents were used as this is the proposed demand for the OW network during a Saturday morning peak (see exercise 10.1 in the book) and an estimated demand from A and B to L and M as depicted in Table 2. Further, the exercise proposes a VDF that relates cost $c(q_k)$ at link k to its flow q_k. Specifically, it is proposed that the travel time in each link is increased by 0.02 for each vehicle/hour of flow ($t_k = t_{k_0} + 0.02 \times q_k$, as discussed in Section 2).

This simple scenario goes far beyond simple two-route (binary) choice scenario. It captures properties of real-world scenarios, like interdependence of routes with shared links and heterogeneous capacities and demand throughout the complete network. Moreover, the number of possible routes between two locations is high and/or it may involve loops as links are two-way, and it has more than a single OD pair. Hence, it is hardly possible to compute the Wardrop's equilibrium algebraically.

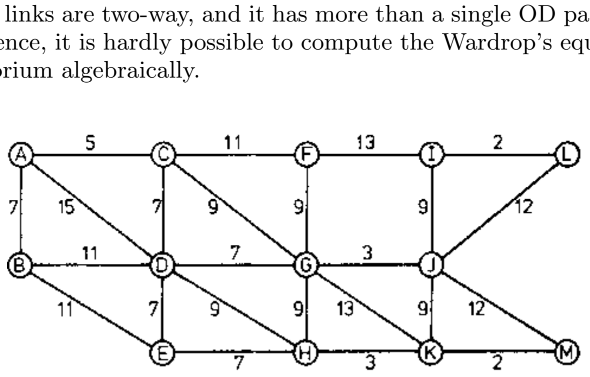

Figure 1: Original Road Network (as proposed by Ortuzar and Willumsen)

Table 2: Average Travel Time per OD Pair: iterative methods

OD Pair	Trips	Incremental	Succ. Avgs.
AL	600	69.00	68.04
AM	400	63.00	62.58
BL	300	69.60	64.50
BM	400	63.00	58.42
	1700	66.28	63.87

5. EXPERIMENTS AND RESULTS

Experiments were conducted using both the iterative methods discussed in Section 2, and the RL-based approach proposed here. As mentioned, the main aim is to show that a disaggregate, decentralized, agent-based approach in which agents learn by interacting with the environment, is able to find solutions that have at least the same travel times as the iterative methods, with little computational effort. Moreover, it is possible to show that the routes selected using the proposed approach are sometimes different from those found by the centralized, iterative approaches used as comparison. The fact that the RL-based approach was able to find lower travel times shows that the routes found by the iterative approaches could still be improved if more iterations were used, but this is unlikely to happen due to the coarse nature of discretization that underlies these methods.

Results for incremental and successive averages methods steam from the implementation of these algorithms provided by the publisher of Ortuzar and Willumsen's book. The shortest paths under free-flow (Table 1) were found algebraically. For the RL-based approach, simulations results reported here are averaged over 10 repetitions (for each condition). To render some tables cleaner, standard deviations are not always shown but they are of the order of 5% at most. Running times are in the order of few seconds for the iterative methods. For the RL-based approach, running times greatly depend on the number of episodes and on the value of the discount rate γ. For the cases shown next, simulations take at most a few minutes. Note however that the number of episodes can be greatly reduced, as indicated in the plots. Experiments were run in a standard PC (8 GB RAM), running Linux (for the RL-based approach) and Windows XP (for the algorithms provided by Ortuzar and Willumsen).

For evaluation, the performance measure is the same used in [12]: travel times averaged over all agents and also for each OD pair. Also, the number of trips using each link is shown, highlighting some differences found among the methods.

5.1 Results from Iterative Methods

Results for the iterative methods discussed before are shown in Tables 2 and 3; these were obtained using $p_n = 0.4, 0.3, 0.2, 0.1$ for the incremental method, and $\phi = 1/n$ and 100 iterations as stop criterion.

In Table 3, it is possible to see that both methods yield very different values for some links. Take links CD, JI, JK, JM, GK, KJ, DG, AD and BA as examples. Later, these numbers can be compared to the RL-based approach.

Table 2 summarizes the average travel times per OD pair for both methods, as well as the average travel times over all trips.

Table 3: Travel Time Each Link: incremental and successive average methods.

	Fixed	Incremental		Succ. Avg.s	
Link	Cost	Flow	Cost	Flow	Cost
AB	7	0	7	4	7.08
AC	5	800	21	655	18.10
AD	15	200	19	348	21.96
BA	7	0	7	7	7.14
BD	11	370	18.40	374	18.48
BE	11	330	17.60	323	17.46
CA	5	0	5	0	5
CD	7	400	15	10	7.20
CF	11	240	15.80	372	18.44
CG	9	160	12.20	276	14.52
DA	15	0	15	0	15
DB	11	0	11	0	11
DC	7	0	7	3	7.06
DE	7	0	7	0	7
DG	0	770	15.40	551	11.02
DH	9	200	13	178	12.56
EB	11	0	11	0	11
ED	7	0	7	0	7
EH	7	330	13.60	323	13.46
FC	11	0	11	0	11
FG	9	0	9	0	9
FI	13	240	17.80	375	20.50
GC	9	0	9	0	9
GD	0	0	0	0	0
GF	9	0	9	3	9.06
GH	9	0	9	0	9
GJ	0	890	17.80	700	14
GK	13	40	13.80	124	15.48
HD	9	0	9	0	9
HE	7	0	7	0	7
HG	9	0	9	0	9
HK	3	530	13.60	501	13.02
IF	13	0	13	0	13
IJ	9	0	9	0	9
IL	2	600	14	450	11
JG	0	0	0	0	0
JI	9	360	16.20	75	10.50
JL	12	300	18	450	21
JK	9	320	15.40	8	9.16
JM	12	0	12	176	15.52
KG	13	0	13	0	13
KH	3	0	3	0	3
KJ	9	90	10.80	9	9.18
KM	2	800	18	624	14.48
LI	2	0	2	0	2
LJ	12	0	12	0	12
MJ	12	0	12	0	12
MK	2	0	2	0	2
sum			543.4		522.38

5.2 Results of the RL based Approach

The approach presented in Section 4 has some parameters that refer basically to the Q-learning. These are the learning rate α, the discount rate γ, and the exploration rate ϵ. In the present paper, ϵ starts at $\epsilon_0 = 1$ and is multiplied by a factor of 0.995 at each episode in order to allow agents to explore the environment for a certain time. The value of this multiplicative factor must be set to fit the simulation horizon. As a general rule, 1000 episodes were run, so that that after 1000 episodes $\epsilon \approx 10^{-3}$. Notice that not all combinations of values for α and γ require 1000 episodes. In some cases convergence to a given route choice pattern is reached much earlier, but for uniformity, the same number of episodes (1000) was used in all cases.

Next the results obtained when this approach was employed in the OW network are presented. Tables 4 and 5 show different measures that are of interest. First, Table 4 shows the average travel time over all 1700 trips, at the last episode, for different combinations of values for α and γ. To render it more clear, standard deviations are omitted and the numbers were rounded to integers. It is clear that the discount factor γ plays a major role in the learning, while the learning rate α is less selective. This can be explained by the fact that choices that can be made at states that can be reached from a given state, are very important in this problem since the agent is trying to make a series of decisions in order to minimize travel time at the whole route. Therefore the discount rate must be high. It needs to be remarked that, in some cases, the number of trips over these links are much higher than the number of users. This is due to the fact that loops are possible and some users perform these loops. This is mainly the case when the discount rate is low and agents do not consider the future.

If one takes travel times given in Table 4, for different values of α and for the highest value of γ, it is is possible to see that these values (between 51 and 52) are lower than those shown in the last line of Table 2. Thus the RL-based approach yields travel times that are lower than the iterative methods, with roughly the same order of running times. Moreover, and perhaps more interesting, the choice made by the agents is based purely on local knowledge, whereas the iterative methods assume global knowledge of the links' costs.

Apart from values averaged over all links, it is interesting to check what happens in each link. As remarked before, the number of trips using some links differ much in the incremental and in the successive averages methods. Thus, a direct comparison with the RL-based approach is interesting. Table 5 shows the number of trips in selected links (to facilitate the comparison, numbers for the iterative methods were copied from Table 3), for $\alpha = 0.5$ and $\gamma = 0.99$. The criterion for inclusion in this table was that either the result achieved by the RL-based approach was different from both iterative methods, or it is close to one of these while these differ among them. For instance, for the link JM, the incremental and successive average methods assign zero and 176 trips respectively. The RL-based approach assigns 185 trips (with standard deviation – given the 10 runs – of about 8), thus being closer than the value obtained using the successive averages method. For cases in which the method proposed here differs from both, take for instance links AB and BA.

In this paper, an important point is that this difference, far from being bad, is what makes the RL-based approach more efficient. Links AB and BA were barely used in the trips assigned using the iterative methods. However, the learning agents found out that they can distribute themselves in ways that use the resources (links) in more efficient ways.

Table 4: Average Travel Time (over all 1700 trips)

γ	α				
	0.1	0.3	0.5	0.7	0.9
0.99	52	52	51	51	51
0.8	50	50	50	50	50
0.6	58	56	55	55	57
0.5	84	80	81	76	80
0.4	114	111	100	102	107
0.2	329	225	183	152	181

Table 5: Number of Trips Over Selected Links, for $\alpha = 0.5$ and $\gamma = 0.99$

			RL-based:
Link	Incremental	Succ. Avg.s	Avg. (Std. Dev.)
AB	0	4	213 (8)
BA	0	7	168 (4)
CD	400	10	103 (5)
JM	0	176	185 (8)
KJ	90	9	5 (1)

That travel times were efficient at user level was already discussed. A final comparison that can be made regards the sum of all costs, a measure of how efficient the method is at global level. The last line of Table 3 shows that the sum of costs over all links is over 500 for both iterative methods. When this sum is made considering the costs of links resulting from the RL-based approach, this value reaches only 462.94, with standard deviation of 0.22.

So far tables have shown the results of the assignment after 1000 learning episodes. The inset plot in Figure 2 depicts how the sum of links' costs change along time. The main plot shows how the number of trips changes with time, for three selected links: AB, BA and CD. These were selected because they show the greatest variation regarding the iterative methods, as shown in Table 5. Note that for $\alpha = 0.5$ and $\gamma = 0.99$, it would not be necessary to run 1000 episodes to reach convergence.

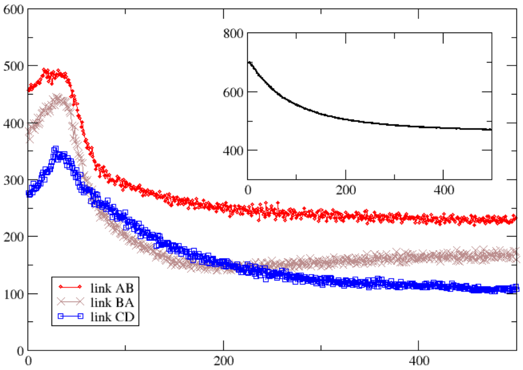

Figure 2: Performance x time: sum of costs over all links (inset) and number of agents in selected links.

6. CONCLUSIONS AND FUTURE WORK

Traffic assignment is an important step in modeling a transportation system. Classical approaches assume some degree of centralization, in which trips are assigned to links or routes. In this paper the perspective of the road user is taken: these users are modeled as agents that autonomously select their routes in an adaptive way. A similar perspective is taken in simulation-based works mentioned in Section 3, but there are two main differences to the present paper. First, here, agents do not anticipate traffic states (e.g., using fuzzy sets) but rather learn these states while interacting with the environment. This is a hard multiagent learning problem given the number of agents (here, thousands) trying to learn simultaneously in a competitive environment (links are shared by many agents). Second, agents form their routes while taking actions at nodes of the network, thus addressing the issue of en-route planning (as this task is known in traffic engineering, even if it is not a planning task from the AI point of view). In most previous simulation scenarios route adaptation was only allowed before and after the actual driving.

Results are twofold. First, the routes that are learned using the proposed approach are sometimes different from those found by the centralized, iterative approaches used as comparison. Second, the learning-based approach is more efficient than the iterative methods: exactly because the agents distribute themselves in different ways in the links of the road network (as compared to these approaches), the overall travel time is approximately 15% less than when iterative methods are used to assign trips to links. Also, the average travel time is lower for each of the OD pairs. This suggests that there is room for further improvements when the iterative methods are used. However, only few works reported in the literature show how far their results are from the optimum (only those dealing with simple networks).

A future direction of this work is to investigate the mathematical properties of the mathematical properties of the multiagent learning approach in order to provide insights about the bound to the optimum assignment. As this is a complex problem, one possibility is to use domain-dependent knowledge and/or properties of the domain. Also, a kind of reward shaping scheme as proposed in [18] can prove useful.

Acknowledgments

Ana L. C. Bazzan is/was partially supported by CNPq and FAPERGS (grants 305062 and 10/0049-7 respectively). Gabriel Ramos has implemented the RL scheme for the original network design.

7. REFERENCES

[1] M. Balmer, N. Cetin, K. Nagel, and B. Raney. Towards truly agent-based traffic and mobility simulations. In N. Jennings, C. Sierra, L. Sonenberg, and M. Tambe, editors, *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multi Agent Systems, AAMAS*, volume 1, pages 60–67, New York, USA, July 2004. New York, IEEE Computer Society.

[2] M. Balmer, M. Rieser, K. Meister, D. Charypar, N. Lefebre, and K. Nagel. MATSim-T: Architecture and simulation times. In A. L. Bazzan and F. Klügl, editors, *Multi-Agent Systems for Traffic and*

Transportation Engineering, pages 57–78. IGI Global, Hershey, US, 2009.

[3] A. L. Bazzan and F. Klügl. Introduction to intelligent systems in traffic and transportation. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 7(3):1–137, 2013.

[4] A. L. C. Bazzan and F. Klügl. Re-routing agents in an abstract traffic scenario. In G. Zaverucha and A. L. da Costa, editors, *Advances in artificial intelligence*, number 5249 in Lecture Notes in Artificial Intelligence, pages 63–72, Berlin, 2008. Springer-Verlag.

[5] M. Ben-Akiva and M. Bierlaire. Discrete choice methods and their applications to short term travel decisions. In *Handbook of transportation science*, pages 5–33. Springer US, 1999.

[6] H. Dia and S. Panwai. Modelling drivers' compliance and route choice behaviour in response to travel information. *Special issue on Modelling and Control of Intelligent Transportation Systems, Journal of Nonlinear Dynamics*, 49(4):493–509, 2007.

[7] C. Gawron. *Simulation-based traffic assignment.* PhD thesis, University of Cologne, Cologne, Germany, 1998.

[8] V. Henn. Fuzzy route choice model for traffic assignment. *Fuzzy Sets and Systems*, 116(1):77–101, 2000.

[9] F. Klügl and A. L. C. Bazzan. Route decision behaviour in a commuting scenario. *Journal of Artificial Societies and Social Simulation*, 7(1), 2004.

[10] E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. In *Proceedings of the 16th annual conference on Theoretical aspects of computer science (STACS)*, pages 404–413, Berlin, Heidelberg, 1999. Springer-Verlag.

[11] H. S. Mahmassani. Dynamic network traffic assignment and simulation methodology for advanced system management applications. *Networks and Spatial Economics*, 1(3-4):267–292, 2001.

[12] J. Ortúzar and L. G. Willumsen. *Modelling Transport.* John Wiley & Sons, 3rd edition, 2001.

[13] S. Peeta and J. W. Yu. A hybrid model for driver route choice incorporating en-route attributes and real-time information effects. *Networks and Spatial Economics*, 5:21–40, 2005.

[14] B. Ran and D. E. Boyce. *Modeling dynamic transportation networks: an intelligent transportation system oriented approach.* Springer, 1996.

[15] R. Rossetti and R. Liu. A dynamic network simulation model based on multi-agent systems. In F. Klügl, A. L. C. Bazzan, and S. Ossowski, editors, *Applications of Agent Technology in Traffic and Transportation*, pages 88–93. Birkhäser, 2005.

[16] C. Tong and S. Wong. A predictive dynamic traffic assignment model in congested capacity-constrained road networks. *Transportation Research Part B: Methodological*, 34(8):625 – 644, 2000.

[17] K. Tumer and A. Agogino. Agent reward shaping for alleviating traffic congestion. In *Workshop on Agents in Traffic and Transportation*, Hakodate, Japan, 2006.

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

[19] J. G. Wardrop. Some theoretical aspects of road traffic research. *Proceedings of the Institute of Civil Engineers*, 1(3):325–362, 1952.