Independent learners in an abstract traffic scenario

Abstract. Traffic movement in a commuting scenario is a phenomena that emerges from individual, uncoordinated and, most of the times, selfish route choice made by drivers. This work presents a reinforcement learning algorithm for route choice which relies solely on drivers experience to guide their decisions. There is no coordinated learning mechanism, so that driver agents are independent learners. Our algorithm is compared to other approaches and experimental results shows good performance regarding travel times and road network load balance. The simplicity, realistic assumptions and performance of the proposed algorithm makes it feasible of being implemented on navigation systems.

1 Introduction

The subject of traffic and mobility presents challenging issues to authorities, traffic engineers and researchers. To deal with the increasing demand, techniques and methods to optimize the existing road traffic network are attractive since they do not include expensive and environmental-impacting changes on the infrastructure.

In a commuting scenario, it is reasonable to assume that drivers choose their routes independently and, most of the time, uninformed about real-time road traffic condition, thus relying on their own experience. Daily commuters usually have an expectation on the time needed to arrive on their destinations and, if a driver reaches its destination within expectation, his travel time can be considered reasonable. From a global point of view, it is desired that vehicles gets distributed on the road network proportionally to the capacity of each road. Multiagent systems like this commuting scenario, where each agent tries to maximize its own utility function, while at the same time there is an global utility which rates the whole system's behavior are called collectives. The local and global goals can be highly conflicting and there is no general approach to tackle this complex question of collectives, as shown by [11].

Traffic assignment deals with route choice between origin-destination pairs in transportation networks. In this work, traffic assignment will be modeled as a reinforcement learning problem so that agents make decisions using only their own experience which is gained through interaction with the environment. The environment is a road network that abstracts some real-world characteristics such as vehicle movement along the roads, allowing us to focus on the main subject which is the choice of one route among the several available for each driver.

The remainder of this document is organized as follows: Section 2 presents basic traffic engineering, single and multiagent reinforcement learning concepts that will be used throughout this paper. Section 3 presents and discusses related work done in this field. Section 4 presents the reinforcement learning for route choice algorithm whose results are discussed in Section 5. Finally, Section 6 concludes the paper and presents opportunities for further study.

2 Background

2.1 Commuting and traffic flow

A road network can be modeled as a graph, where there is a set of nodes, which represent the intersections, and links among these nodes, which represent the roads. The weight of a link represents a form of cost associated with the link. For instance, the cost can be the travel time, fuel spent or length.

A subset of the nodes contains the origins of the road network, where drivers start their trips, and another subset represents the destinations, where drivers finish their trips. In a commuting scenario, a driver's trip consists on a set of links, forming a route between his origin and destination (OD pair) among the available routes.

Traffic flow is defined by the number of entities that use a network link in a given period of time. Capacity is understood as the number of traffic units that a link supports in a given instant of time. Load is understood as the demand generated on a link at a given moment. When demand reaches the link's maximum capacity, the congestion is formed.

One of the most common cost function that relates link's attributes (capacity, free-flow travel time) and traffic flow is shown on Eq. (1) [10].

$$t_j(v) = f_j[1 + \tau \left(\frac{v}{c_j}\right)^{\beta}] \tag{1}$$

In this function, t_j is the travel time on link j, c_j is the link's capacity, f_j is the free-flow travel time on link j and τ and β are calibration parameters. This will be the travel time function used throughout the present work.

2.2 Reinforcement Learning

Reinforcement learning (RL) deals with the problem of making an agent learn a behavior by interaction with the environment. Usually, a reinforcement learning problem is modeled as a Markov Decision Process (MDP), which consists of a discrete set of environment states (S), a discrete set of actions (A), a state transition function ($T : S \times A \rightarrow \mathit{\Pi}(S)$), where $\mathit{\Pi}(S)$ is a probability distribution over S) and a reward function ($R : S \times A \rightarrow \mathbb{R}$).

The agent interacts with the environment following a policy π and tries to make it converge to the optimal policy π^* that maps the current environment state $s \in S$ to an action $a \in A$ in a way that it maximizes the future reward. At each state, the agent must select an action a according to a strategy that balances exploration (gain knowledge) and exploitation (use knowledge). One of the strategies is ϵ-decreasing: choose a random action (exploration) with probability ϵ or choose the best action (exploitation) with probability $1 - \epsilon$. In the beginning, ϵ starts with a higher value (high exploration) and decreases with time, leading to high exploitation at the end.

Q-learning is an algorithm that reaches the optimal policy, given certain conditions [7]. Its update rule is shown on Eq. (2), where $< s, a, s', R >$ is an experience tuple, meaning that the agent performed action a in state s, reaching state s', receiving reward R. Action a' is one that can be taken on s', α is the learning rate and γ is the discount factor.

$$Q(s, a) = (1 - \alpha)Q(s, a) + \alpha(R + \gamma \max_{a'}(Q(s', a'))) \tag{2}$$

For a complete description of Q-learning, the reader may refer to [12].

2.3 Independent learners

Traffic scenarios are non-cooperative multi-agent systems, as there are several agents interacting with the environment and coordinating themselves to use the traffic network resources. Multi-agent reinforcement learning (MARL) tasks can be divided in two forms: independent learners (ILs), that ignore the existance of other agents and joint action learners (JALs), that learn the value of their own actions combined with other agents' actions via integration of RL with coordination learning methods [5]. ILs understand other agents learning and changing their behavior as a change of environment dynamics. In the present work, agents will be modeled as ILs.

Modeling agents as JALs leads to scalability problems, as agents must consider every other agents' actions. In complex scenarios like transportation systems, there is a large number of agents, making the modeling of JALs unfeasible. On the other hand, when agents are modeled as ILs, the convergence properties of Q-learning becomes invalid, as the environment is nonstationary. Also, it is remarked by [9] that training adaptive agents without considering other agents adaptation is not mathematically justified and it is prone to reaching a local maximum where agents quickly stop learning. Even so, some researchers achieved amazing results with this approach.

3 Related work

Traffic assignment problems are tackled with several approaches, many of them considering abstract traffic scenarios. Two-route scenarios are studied in [1, 8]. The former analyses the effect of different strategies on minority game for binary route choice. The latter includes a forecast phase for letting agents know the decision of the others and then let they change their original decision or not. Each one of these works assessed relevant aspects of agents decision-making process, even though only binary route choice scenarios were studied. The interest on the present work is to evaluate a route choice approach in a more complex scenario, with several available routes.

This kind of complex scenario was investigated by [2]. On their work, Bazzan and Klügl assessed the effect of real time information on drivers' route replanning, including studies with adaptive traffic lights. In the most successful route replanning strategy presented on that work, the authors assume that the entire network occupancy is known by the drivers. This assumption was needed for assessing the effects of re-routing, although the availability of real time information of the entire network for all the drivers is an unrealistic assumption.

More recently, the minority game algorithm was modified for use in a complex scenario with several available routes [6]. Using the proposed algorithm, drivers achieve reasonable (within expectation) travel times and distribute themselves over the road network in a way that few links get overused. The modified minority game algorithm uses historic usage data of all links to choose the next one on the route. Having historical information of the links used by the driver is a reasonable assumption, but having historic information of all links on the network is unrealistic. The algorithm proposed on the present work will be compared with the modified minority game.

Learning-based models for route choice are studied in [4, 3]. In the former, the authors present a reinforcement algorithm and in the latter, a reinforcement-based model of route-choice behavior with availability of real-time information is presented. Both works try to reproduce human decision-making in corresponding experimental studies, based on two-route scenarios. The works are focused on reproducing human decision-making rather than proposing a new approach for improving the route choice process in a scenario with several available routes.

In the literature review performed for the present work, a reinforcement learning based approach for the route choice process itself was not found. Either researchers present other approaches for the problem or the reinforcement-based schemes are used to try to reproduce human behavior.

4 Approach and scenario

4.1 Reinforcement learning for route choice

The MDP for this problem is modeled as follows: the states are the nodes of the road network. The set of actions comprises the selection of the outbound links from the nodes of the network. Not every link will be available for the agents to choose, as it depends on which node of the network it is and whether the link belongs to an possible route to the agent's destination. The reward function is given by Eq. (3), where t_j is the travel time function (Eq. (1)) applied to the number of vehicles (v) on the link the agent traversed.

$$R = -t_j(v) \tag{3}$$

The reward decreases as travel time increases, so drivers will strive to minimize their individual travel times.

4.2 Building the route:

Each learning episode is a trip that drivers do departing from their origins and building a route to their destinations.

Initialization: At the beginning of execution, OD pairs are randomly distributed among drivers. Then, each driver calculates the shortest route P_i^* for his OD pair. In this phase, the shortest route is considered as the one with less links between origin and destination. Drivers also initialize the exploration coefficient (ϵ) to 1. This means that drivers will explore in the first episodes to gain knowledge about the road network and exploit it in the final episodes.

Execution: In each episode of this reinforcement learning for route choice algorithm, each driver follows the steps shown in Figure 1.

At episode start, all drivers are placed in their origins. At step 1, the driver chooses an outbound link to traverse according to the ϵ-greedy action selection strategy. At step 2, the destination node of the chosen link is reached. At step 3, the driver tests whether the node reached is its final destination. If so, the trip ends, otherwise steps 1 to 3 are repeated. At step 4, each driver i calculates its travel time a_i experienced on it's route P_i, given by Eq. (4), where v is the number of vehicles on link j.

$$a_i = \sum_{j \in P_i} t_j(v) \tag{4}$$

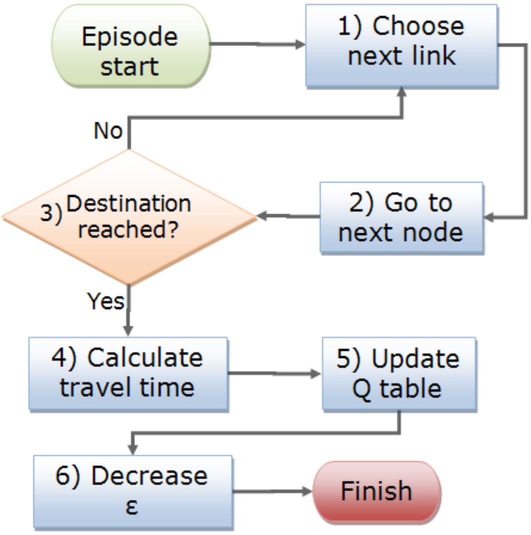


Fig. 1. RL for route choice algorithm flowchart

Then, at step 5, drivers update their Q-tables according to Eq. (2). Finally, at step 6, the exploration factor (ϵ) is decreased by a multiplicative factor.

4.3 Evaluation metrics

Reasonable travel times: In real world, drivers have an expectation on the trip travel time on the route length, the expected number of drivers in it and the links' capacities. In the present work, for each driver i, the expected travel time e_i on his shortest route P_i^* is given by Eq. (5), where t_j is the travel time function defined in Eq. (1) applied to the estimated number of vehicles on the same route (v_e). This estimation is given by the number of vehicles in driver i OD pair, plus a random number in the range $[-0,05d : 0,05d]$, where d is the total number of drivers on the scenario. This "noise" is to simulate the effect of each driver "guessing" the number of vehicles going to the same destination.

$$e_i = \sum_{j \in P_i^*} t_j(v_e) \tag{5}$$

In order to assess how reasonable are the travel time obtained by drivers using the proposed algorithm, a metric called AED (actual and expected travel time difference) was created. It is given by the average of the difference between actual and expected travel times of the drivers on a given OD pair. For this metric, negative values are desired as this means that actual travel times are lower than the expected by the drivers.

Road network load balance: From a global point of view, it is desired that vehicles get distributed proportionally to the capacity of each link on the network. Road network load balance will be measured in two forms: number of congested links (n) and average overload (o). Considering C as the set of congested links, these metrics will be measured according to Eq. (6), where v_j and c_j are the number of vehicles and the capacity of link j, respectively.

$$n = |C| \qquad\qquad o = \frac{\sum_{j \in C} \left(\frac{v_j}{c_j} - 1 \right)}{|C|} \tag{6}$$

Both metrics are needed because n alone does not measures how heavy are the links congestions and o does not shows how many links are overloaded. For both (n) and (o), smaller values means better performance, as less links are congested and the severity of congestions is lower.

4.4 Studied scenario

The abstract road network used in the present work is the same used by [6]. It consists on 10 nodes and 24 links, as depicted in Figure 2. All nodes have 3 outbound links, except nodes 8, 9 and 10 which have 2, 1 and 0 outbound links, respectively. Nodes 1, 2 and 3 are the possible origins and nodes 8, 9 and 10 are the possible destinations, resulting in nine possible OD pairs. The network links have the same weights, representing no differences on their lengths.

The proposed algorithm will be compared with three different approaches:

- Random: At each step, drivers choose one of the possible outbound links with random probability.

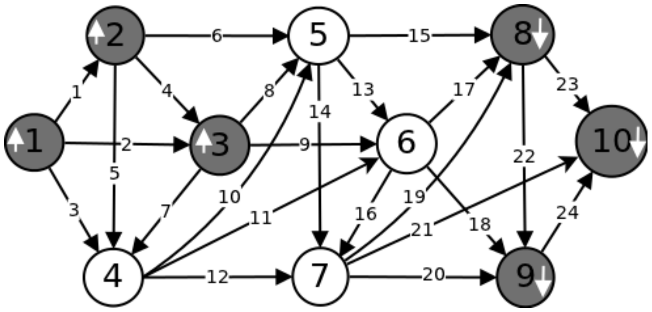


Fig. 2. Road network, the same used by [6]. Labels on links are identification numbers, nodes with upward arrows are the origins and downward arrows represent the destinations

- Greedy: At initialization, the shortest paths[1] for each driver are calculated. At each step, drivers choose one of the possible outbound links with a probability proportional to its capacity. The possible outbound links are the ones that belongs to a shortest path.
- Minority Game: In this approach, drivers use a modified minority game algorithm to build their routes. For a complete description of the algorithm, it is suggested that readers refer to [6].

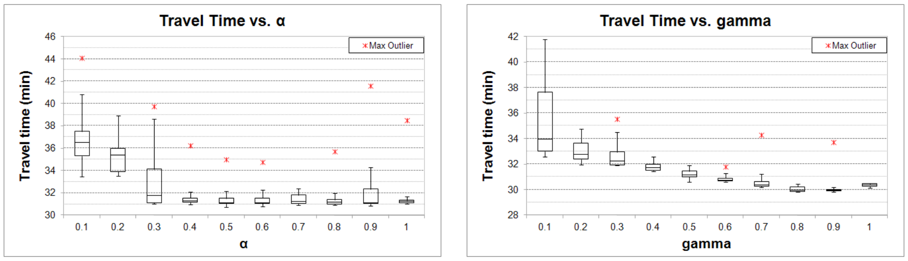
5 Results and discussion

In the experiments, link's capacities were randomly assigned within the range [130:250] prior to the simulations. The values are persisted from one simulation to another to ensure a correct comparison of different algorithms. The same is done for the amount of drivers on each OD pair. There are 1001 drivers on the road network. For Eq. (1), f_j is set as 5 minutes for all links, $\tau = 1$ and $\beta = 2$ making travel time increase quadratically with the number of drivers. Each run consists of 100 episodes. The desired value for ϵ is 0.01 at the end of the run, so the multiplicative decrease factor is set as $10^{log 0.01/100} = 0.95499$.

[1] It is possible to have more than one shortest path on the road network. Given that link's weights are equal, the shortest paths will be the ones with less links.

5.1 Influence of Q-learning parameters

Several simulations were run to assess the effect of Q-learning parameters, namely the learning rate (α) and the discount rate for future rewards (γ).



(a)

(b)

Fig. 3. Box-and-whisker charts showing the effect α and γ on travel time. Each parameter value was tested 10 times.

Figure 3 shows that for α within the range of 0.4 to 0.8, travel time does not change in a significant way. On Fig. 3(b), a decrease on travel time is observed with the increase of γ. These plots shows that a good choice for the parameters are $\alpha = 0.5$ and $\gamma = 0.9$, as lower travel times are achieved with these values. They will be used for the experiments that follows.

5.2 Comparison with other algorithms

For each algorithm, the AED metric is shown on Figure 4.

In this metric, the RL approach outperforms the others, achieving reasonable travel times in all OD Pairs. The minority game algorithm achieves travel times within expectation in 4 OD pairs. On the other OD pairs, the performance is still good, as actual travel times are no longer than 5 minutes beyond the expected, showing a degree of fairness of the algorithm. With the greedy approach, drivers from 5 OD pairs experience travel times within expectation, but especially on OD pairs 1-10 and 2-8, actual travel times are far beyond the expected. The worst approach is the random, in which no drivers achieve reasonable travel times.

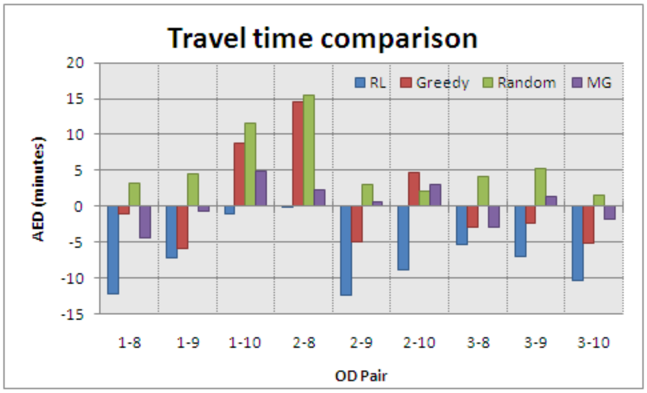


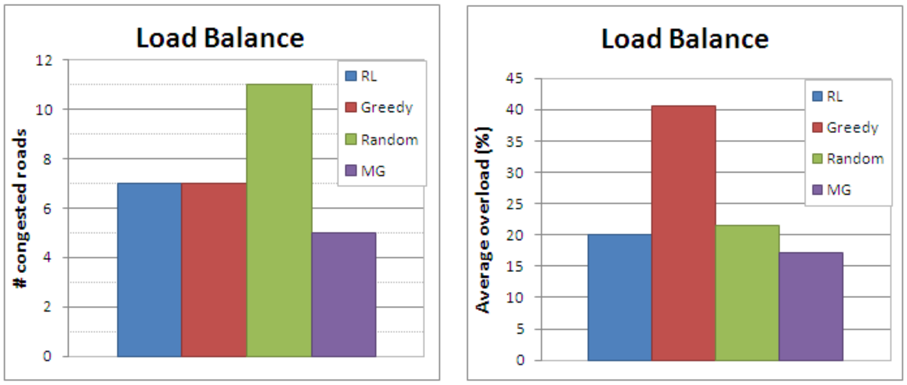
Fig. 4. AED metric comparison for each algorithm. RL and MG are the reinforcement learning and minority game approaches, respectively

On the defined load balance evaluation metrics, the minority game based approach achieved the best results. Both the number of congested links and the average overload were the lowest. This means that vehicles were distributed over the network in a proportion that is closer to the links capacity. The RL based algorithm achieves good results as the average overload shows that none of the seven congested links (out of 24 links) were heavily congested. Both the random and the greedy approach performed poorly, as many links were congested in the random approach and links were heavily congested in the greedy approach.

6 Conclusions and future work

In this work we applied the concept of independent learners in a complex, non-cooperative scenario. Our reinforcement learning for route choice algorithm is helpful for either individual and global point of view, as drivers achieve reasonable travel times on average, and traffic is distributed over the network, as links does not get heavily congested.

The proposed approach has the advantage of making realistic assumptions as it only relies on drivers own experience about the road network (i.e. the experienced travel time), dismissing the use of real-time information and historic data of links. This makes our



(a) (b)

Fig. 5. Load balance evaluation in terms of number of congested links (a) and the average overload (b).

algorithm an attractive and feasible alternative to be used on existing navigation systems, as no new technologies are required.

Further investigation can be conducted to assess how the algorithm performs in heterogeneous scenarios. It would be interesting to investigate whether the RL drivers can adapt themselves to the greedy drivers or even the ones using the minority game algorithm proposed by [6]. Future work can also attempt to assess how good it would be for agents when they consider other agents, that is, how good it would be to learn joint actions in this competitive environment.

7 Acknowledgments

Authors would like to thank Mr. Syed Galib for clarifying questions on the minority game for route choice algorithm [6] and for providing data for comparison. Both authors are partially supported by CNPq and FAPERGS.

References

1. Bazzan, A.L.C., Bordini, R.H., Andriotti, G.K., Viccari, R., Wahle, J.: Wayward agents in a commuting scenario (personalities in the minority game). In: Proc. of the Int. Conf. on Multi-Agent Systems (ICMAS). pp. 55–62. IEEE Computer Science (July 2000), `www.inf.ufrgs.br/%7Emas/emotions`

2. Bazzan, A.L.C., Klügl, F.: Re-routing agents in an abstract traffic scenario. In: Zaverucha, G., da Costa, A.L. (eds.) Advances in artificial intelligence. pp. 63–72. No. 5249 in Lecture Notes in Artificial Intelligence, Springer-Verlag, Berlin (2008), `www.inf.ufrgs.br/maslab/pergamus/pubs/BazzanKluegl.pdf.zip`
3. Ben-Elia, E., Shiftan, Y.: Which road do i take? a learning-based model of route-choice behavior with real-time information. Transportation Research Part A: Policy and Practice 44(4), 249–264 (2010)
4. Chmura, T., Pitz, T.: An extended reinforcement algorithm for estimation of human behavior in congestion games. Journal of Artificial Societies and Social Simulation 10(2) (2007)
5. Claus, C., Boutilier, C.: The dynamics of reinforcement learning in cooperative multiagent systems. In: Proceedings of the Fifteenth National Conference on Artificial Intelligence. pp. 746–752 (1998)
6. Galib, S.M., Moser, I.: Road traffic optimisation using an evolutionary game. In: Proceedings of the 13th annual conference companion on Genetic and evolutionary computation. pp. 519–526. GECCO '11, ACM, New York, NY, USA (2011), `http://doi.acm.org/10.1145/2001858.2002043`
7. Kaelbling, L.P., Littman, M., Moore, A.: Reinforcement learning: A survey. Journal of Artificial Intelligence Research 4, 237–285 (1996)
8. Klügl, F., Bazzan, A.L.C.: Simulated route decision behaviour: Simple heuristics and adaptation. In: Selten, R., Schreckenberg, M. (eds.) Human Behaviour and Traffic Networks, pp. 285–304. Springer (2004)
9. Littman, M.L.: Markov games as a framework for multi-agent reinforcement learning. In: Proceedings of the 11th International Conference on Machine Learning, ML. pp. 157–163. Morgan Kaufmann, New Brunswick, NJ (1994)
10. Ortúzar, J., Willumsen, L.G.: Modelling Transport. John Wiley & Sons, 3rd edn. (2001)
11. Tumer, K., Wolpert, D.: A survey of collectives. In: Tumer, K., Wolpert, D. (eds.) Collectives and the Design of Complex Systems, pp. 1–42. Springer (2004)
12. Watkins, C.J.C.H., Dayan, P.: Q-learning. Machine Learning 8(3), 279–292 (1992)