Using Topological Statistics to Bias and Accelerate Route Choice: preliminary findings in synthetic and real-world road networks

Fernando Stefanello, Bruno C. da Silva, Ana L. C. Bazzan
Instituto de Informática – Universidade Federal do Rio Grande do Sul – Porto Alegre, RS, Brazil
{fstefanello, bsilva, bazzan}@inf.ufrgs.br.

Abstract

This paper discusses the first steps towards the definition of novel statistics and metrics that characterize networks in terms of the complexity they pose to the traffic assignment problem. Here, we follow an approach in which the assignment emerges from routes selected by learning agents. Specifically, we deal with issues related to how routes are *coupled*. We first define and quantify route coupling, i.e., how much a given route is coupled with other routes that can be used by learning agents. The investigation of route coupling is important in multi-agent reinforcement learning settings since it measures how a change in action selection by one agent interferes with the actions taken by other agents. Our preliminary empirical results indicate that using route coupling to bias the learning process of agents results in faster convergence in the traffic assignment problem.

1 Introduction

Traffic networks can be represented as directed graphs $G = (V, A)$, where V represents the set of nodes (street or road intersections, points of interest, or districts), and A the set of arcs (street or road segments). The topology of these graphs, as well as other characteristics such as how demand is distributed and how latency (cost) functions are defined, have a great influence on the *traffic assignment problem* (TAP), whose goal is to assign the traffic demand (number of trips, vehicles) onto the arcs of a network G.

Our objective is to characterize road networks by designing qualitative metrics and statistics. In particular, given a graph, an origin-destination matrix, and latency functions, we wish to design metrics that numerically represent the degree of difficulty posed to the assignment task. We study the TAP not under the traditional, centralized approach, but via an agent-based variant. Here, each agent (in this context, a driver or vehicle) learns to select a route taking it from its origin to its destination. Given that in such a process a decision made by an agent affects the outcome of other agents, this is a typical multiagent reinforcement learning problem (MARL).

A closely related issue to the one studied in this paper is that of how to characterize a traffic network in terms of how high the price of anarchy (*PoA*); [Koutsoupias and Papadimitriou, 1999; Roughgarden and Tardos, 2002]) is. The PoA measures the loss in performance caused by a situation in which each driver seeks to minimize its travel time independently. Some works refer to this as *selfish routing*. One of our goals is to investigate how the magnitude of the PoA affects the learning task. Knowing this, one could anticipate how complex the assignment will be. As the Braess paradox—a case in which the PoA can be very high—shows, the addition of arcs to a network may end up causing more congestion [Braess, 1968].

This paper discusses the first steps towards the definition of novel statistics and metrics to characterize networks in terms of the complexity they pose to the TAP. In particular, we propose a metric based on topological properties of the network, whose goal is to measure how *coupled* routes are. *Coupling* refers, intuitively, to how many points of interaction a route has with other routes, and how likely it is that agents might decide to switch routes. This serves as a way of estimating how strongly an agent on one route might be affected by other agents who might change their behavior. Measuring this effect is important since strong dependencies make learning in MARL settings harder. We describe a way of using coupling statistics to bias the learning process of agents in a way that empirically counteracts the negative effects of non-stationary in the learning process, and that is conducive to faster convergence to an equilibrium. We evaluate our methods in several road networks with different topologies and demands—both real-world networks and synthetic ones, such as those affected by the Braess paradox. In order to construct challenging networks that are affected by this paradox, we modify an existing method to extend Braess networks to arbitrary sizes.

This paper is organized as follows: Section 2 introduces the classical, optimization-based approach to the TAP, and an alternative MARL-based approach to compute a user equilibrium. Section 3 introduces our main methods and presents a few networks that we use to illustrate them. We present preliminary results in Section 4, related work in Section 5, and present concluding remarks in Section 6.

2 The Traffic Assignment Problem

This section introduces a mathematical formulation for the traffic assignment problem and presents the notation that will

be used throughout the paper. As previously mentioned, a transportation network can be represented as a directed graph $G = (V, A)$. Each arc $a \in A$ has a latency function which is a function of the traffic in that arc—it quantifies the effects of network usage, such as traffic congestion. This function depends on parameters of the arc such as the time τ to transverse it without congestion (this is also known as free flow time), and the nominal capacity ρ of the arc (e.g., in terms of number of vehicles).

In this work we denote the set of incoming arcs to node $v \in V$ by $IN(v)$, and the set of outgoing arcs from node $v \in V$ by $OUT(v)$. In addition, let $\mathcal{C} = \{(o(1), d(1)), \ldots, (o(|\sigma|), d(|\sigma|)\} \subseteq V \times V$ denote the set of commodities, i.e., a set of origin-destination (OD) pairs. Here, $o(\sigma)$ and $d(\sigma)$ represent, respectively, the origin and destination nodes for $\sigma = 1, \ldots, |\mathcal{C}|$. Each commodity σ has an associated demand $r_\sigma = r_{o(\sigma),d(\sigma)}$; i.e., each OD pair $(o(\sigma), d(\sigma))$ has an associated demand r_σ that emanates from node $o(\sigma)$ and terminates in node $d(\sigma)$.

Furthermore, each arc has a latency function that expresses how travel time depends on the traffic flow on that arc. If drivers were to selfishly select routes that minimize their individual travel times, they could simply select the shortest path that satisfies their desired origin and destination nodes. This strategy, however, makes several underlying assumptions which are often not met, or are unrealistic: for example, that the time taken to traverse an arc is constant and independent of other drivers. This is clearly not the case in real traffic networks, where the maximum flow allowed in an arc depends on which routes other drivers take and on how many drivers occupy an arc at a given time.

When simulating traffic conditions on a given network, a designer needs to select a latency function that approximates the real-life costs of navigating in that network. One of the best-known and widely used latency function for real-world networks, often referred to as the BPR function, was introduced by the U. S. Bureau of Public Roads [Bureau of Public Roads, 1964]. This is a non-linear, convex, and strictly increasing function. Linear functions are also frequently used (e.g., in the case of networks affected by the Braess paradox) to represent the latency on each arc. In this work, whenever we refer to networks affected by this paradox, we assume that the latency is represented by $l_a(f_a) = m_a f_a + n_a$, where $m_a \in \mathbb{R}+$ and $n_a \in \mathbb{R}$ are parameters and f_a is the flow on arc a. We also assume that $l_a(f_a) \geq 0$.

In the next section we introduce a mathematical model for assignment problem—this is a classical, optimization-based method to solve the TAP. We then describe an alternative way of solving a version of this problem, namely by searching for a user equilibrium via MARL techniques.

2.1 A Model of Traffic Assignment

In this subsection we present mathematical models describing the two main principles that characterize the traffic assignment: the *system optimum* (SO) and *user equilibrium* (UE) Wardrop [1952]. The latter principle states that "under equilibrium conditions traffic arranges itself in congested networks such that all used routes have equal and minimum costs, while all those routes that were not used have greater or equal costs". The former principle refers to the system as a whole and states that the average trip time is minimum.

Beckmann *et al.* [1956] were the first to propose and solve a mathematical model to compute both the SO and UE solutions. In what follows we present an arc-based mathematical model for SO and for the UE model. A path-based mathematical model may also be used to represent the respective assignment problems. Let x_a^σ be variables indicating the flow on arc a for the commodity σ; let f_a be the total flow on arc a and Φ_a be the associated cost for the arc a. The **SO model** for a multi-commodity network can be written as:

$$\min \ \Phi = \sum_{a \in A} \Phi_a \tag{1}$$

subject to:

$$m_a f_a^2 + n_a f_a \leq \Phi_a \tag{2}$$

$$f_a = \sum_{\sigma \in \mathcal{C}} x_a^\sigma \quad \forall a \in A \tag{3}$$

$$\sum_{a \in IN(v)} x_a^\sigma - \sum_{a \in OUT(v)} x_a^\sigma = \begin{cases} d_\sigma, & \text{if } v = d(\sigma) \\ -d_\sigma, & \text{if } v = o(\sigma) \\ 0, & \text{otherwise} \end{cases} \quad \forall v \in V, \sigma \in \mathcal{C} \tag{4}$$

$$x_a^\sigma \geq 0, \forall a \in A, \forall \sigma \in \mathcal{C} \tag{5}$$

$$f_a \geq 0, \Phi_a \geq 0 \ \forall a \in A. \tag{6}$$

Objective function (1) aims at finding a flow assignment for each arc that minimizes the total cost for the system—resulting in an assignment respecting the SO principle. Constraints (2) associate the cost of each arc a to the variable Φ_a; constraints (3) associate the total flow in arc a to variables f_a; constraints (4) ensure flow conservation, and constraints (5) and (6) define the domain of variables. Note that this model has quadratic constraints, since it contains a product between flow variables in the constraints (2)—in particular, the product of latency costs and arc flows (i.e., $(m_a f_a + n_a)\, f_a$). The SO model can be extended to networks that consider the BPR latency function by changing constraints (2). This formulation uses a set of variables Φ_a and constraints (2) to define the latency cost on each arc; this is especially useful for the case where the latency function is a composition of linear functions (see Case 2, Section 3.1, for more details).

We now consider the **UE model**, whose objective is to minimize the function

$$\Phi = \sum_{a \in A} \int_{\bar{x}_a}^{f_a} l_a(x) dx \tag{7}$$

Since we assume that the latency function is $l_a(x) = m_a x + n_a$, we can simplify this expression. For the case where $n_a \geq 0$, $\bar{x}_a = 0$. For the case where $n_a < 0$, $\bar{x}_a$ should be the max x such that $l_a(x) = 0$; i.e., $\bar{x}_a = -n_a/m_a$. Since we also consider $l_a(x) \geq 0$, the latency function becomes a composition of two line segments, thereby defining a piecewise linear function which is convex and strictly increasing. To completely define the UE model, we now only need to replace the constraints (2) with the following set of constraints:

$$\frac{1}{2}m_a f_a^2 + n_a f_a - \left(\frac{1}{2}m_a \bar{x}_a^2 + n_a \bar{x}_a\right) \leq \Phi_a. \quad (8)$$

If $n_a \geq 0$, then $\bar{x}_a = 0$ and constraints (8) are reduced to

$$\frac{1}{2}m_a f_a^2 + n_a f_a \leq \Phi_a \quad (9)$$

Solving the above-mentioned models involves assigning a traffic flow to each arc in order to obtain a global assignment that is consistent either with the SO or UE hypotheses. The models can be solved via mathematical programming using general-purpose solvers such as CPLEX and MOSEK.

2.2 Multiagent Learning for Route Choice

Mathematical programming-based methods like the ones previously mentioned may have difficulties if non-linear latency functions are used. Furthermore, these methods can typically only solve static assignments. Unlike optimization approaches that use mathematical models suitable only for static assignment with linear or convex latency functions, MARL can be used to compute traffic assignment solutions by considering each individual driver as an autonomous agent, in a microscopic fashion. This strategy can be used to tackle a wide range of problems, such as those involving static or dynamic assignment, and also ones that require the simulation of complex systems.

In this paper we assume that when using MARL to compute traffic assignment solutions, each agent learns to make decisions (i.e., to select routes) by using reinforcement learning. We use the Q-learning algorithm to update the value of each state-action pair of the agent; this value represents the expected long-term utility that the agent hopes to achieve by selecting a given action in a state, and following the current action-selection strategy thereafter. This update is performed based on an experience tuple $\langle s, a, s', rew \rangle$ according to Equation 10, where α is the learning rate and γ is a discount rate applied for future rewards. Details of the use of Q-learning for the TAP are given in Section 4.

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left(rew + \gamma \max_{a'} Q(s',a') - Q(s,a)\right) \quad (10)$$

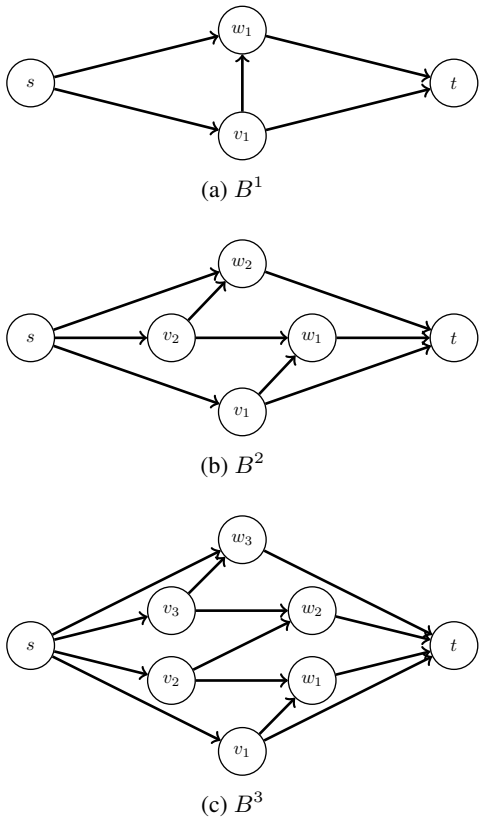
3 Methods

We are interested in characterizing traffic networks of different types—for instance, synthetic (or pictorial) networks such as those used to illustrate the Braess paradox; networks whose demand distributions are closer to real-world cases (versus symmetric-demand distributions such as those in Braess paradox); single versus multicommodity cases; networks with linear versus non-linear latency functions; etc. Hence, prior to discussing the statistics we employ to measure the coupling between routes, we introduce and discuss the nature of a few selected networks: arbitrarily large Braess-paradox networks (in Section 3.1), the OW network (Section 3.2), and the Sioux Falls network (Section 3.3). We then introduce a metric for characterizing some properties of these networks—the *coupling statistic* (Section 3.4).

3.1 Arbitrarily Large Braess-Paradox Networks

The Braess paradox occurs whenever adding resources to a transportation network deteriorates the quality of a UE. Using Beckmann's model, Braess [1968] described situations in which adding a road to a congested traffic network could have a counter-intuitive outcome—namely, the overall travel time could increase. This phenomenon can be interpreted as follows: suppose we close a road or increase its free travel time by decreasing the maximum allowed speed; if the cost (e.g., the total travel time at UE) decreases, then we observe the Braess paradox.

Roughgarden [2001] discusses the problem of designing networks so that the Braess paradox does *not* occur—more specifically, which edges should be removed from a network to obtain the best possible flow at Nash equilibrium. This author also discusses how to create arbitrarily large Braess graphs, whose sizes depend on a factor p; a few examples are shown in Figure 1. Although single-commodity, these networks are interesting since they are associated with a high PoA. Roughgarden's method allows the investigation of the PoA in large graphs, rather than in simple ones like the network in Figure 1a. Being able to produce large networks with this property is useful in the context of our work because we aim at defining novel statistics and metrics to characterize networks (such as Braess networks of different sizes) in terms of the complexity they pose to the TAP—in particular in the context of using MARL algorithms to solve it.



(a) B^1

(b) B^2

(c) B^3

Figure 1: Sample graphs that result in the Braess paradox.

Braess graphs of arbitrary size can be generated as follows: given a size parameter p, the p-th Braess graph B^p is constructed with a set $V^p = \{s, v_1, ..., v_p, w_1, ..., w_p, t\}$ of $2p + 2$ vertices; and a set of arcs A^p defined by $\{(s, v_i), (v_i, w_i), (w_i, t) : 1 \leq i \leq p\} \cup \{(v_i, w_{i-1}) : 2 \leq i \leq p\} \cup \{(v_1, t)\} \cup \{(s, w_p)\}$. Next, we describe how to associate latency functions with each of these arcs.

Latency Functions

A key decision when designing arbitrarily large Braess networks is how to associate latency functions to the arcs. We extend the method described by Roughgarden [2001] in three ways: (i) we allow networks with arbitrary constant arc costs c, instead of unitary costs; (ii) we allow arbitrary demand values[1] r instead of only fractions of a unitary demand; and (iii) we introduce simpler piecewise latency functions which, although resulting in lower PoA values, allow for solutions to be more easily obtained via standard commercial optimization packages.

In what follows, the demand for a given commodity (or origin–destination pair) $\sigma \in \mathcal{C}$ is indicated as r_σ. Let $c > 0$ be the cost associated with constant-cost arcs, and $r \in \mathbb{R}^*+$ be the total demand of the network. Roughgarden [2001] uses $c = 1$ and $r = p$; this constrains the networks that can be generated since p is typically much smaller than the demand. We modify this formulation so that r can be arbitrarily large. By default, we consider $c = 10$ and $r = 4200$.

We now define the latency function $l_a(x)$ associated with each arc a. The value of i of each arc is the same as described in Section 3.1. We omit the subscript p in the latency function to simplify notation. The latency functions are defined as:

- $l_a(x) = 0$ for arcs of form $a = (v_i, w_i)$ $\forall i \in \{1, 2, \ldots, p\}$;
- $l_a(x) = c$ for arcs of form $a = (v_i, w_{i-1})$ $\forall i \in \{2, \ldots, p\}$, (s, w_p) or (v_1, t).

For the remaining arcs of the form $a = (w_i, t)$ or (s, v_{p-i+1}) $\forall i \in \{1, 2, \ldots, p\}$, the latency function is defined as a function of the flow in the arc. We use two strategies to define the latency of these arcs. The first strategy is to use a linear function with $n_a = 0$, and the second one is to use a piecewise function based on the latency function described by Roughgarden [2001]; these are henceforth referred to as *Case 1* and *Case 2* respectively.

Case 1 - Linear function with $n_a = 0$

In this case, the latency function $l_a(x) = m_a x + n_a$ is a simple line segment satisfying $l_a(0) = 0$ and $l_a^p(\frac{r}{p}) = ic$, i.e.:

$$m_a = \frac{icp}{r} \tag{11}$$

$$n_a = 0. \tag{12}$$

Case 2 - Piecewise Linear function with $n_a < 0$

In this case, the latency function $l_a(x) = m_a x + n_a$ is a composition of two line segments satisfying $l_a(0) = 0$, $l_a(\frac{r}{p+1}) = 0$, and $l_a^p(\frac{r}{p}) = ic$, i.e.,

$$m_a = \frac{icp^2 + cip}{r} \tag{13}$$

$$n_a = -cip. \tag{14}$$

Figure 2 depicts fixed-cost arcs in blue, while arcs with cost equal to zero appear in red. The remaining arcs (in black) are the ones with latency functions that depend on the arc's flow.

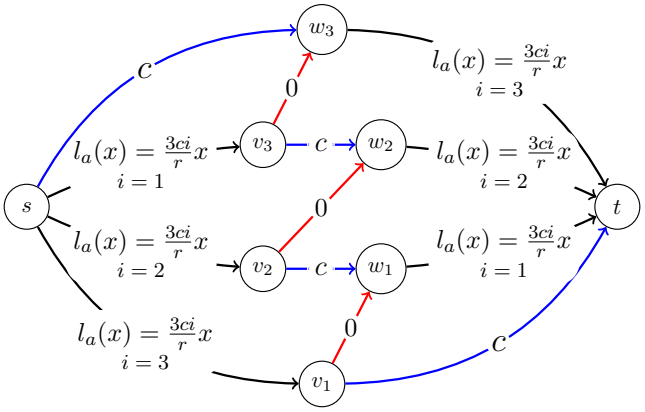


Figure 2: Braess Graph B^3 with latency functions.

Figure 3 shows a few sample latency functions: f_0 is the latency function for the family functions proposed by Roughgarden [2001]; f_1 represents the latency function for Case 1, and f_2 represents the latency function for Case 2.

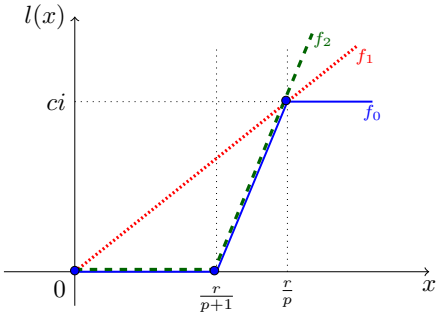


Figure 3: Sample latency functions.

3.2 The OW Network

Besides Braess-paradox networks, another network of interest in this work is the OW network (due to Ortúzar and Willumsen [2001]), depicted in Figure 4 and henceforth referred to simply as OW. Although this not a full reproduction of a real-world network, it contains interesting real-world elements. This network represents two residential areas (nodes A and B in the figure) and two major shopping areas (nodes L and M). The numbers associated with arcs, τ_a, denote travel times in those arcs under free flow (in both ways). The proposed demand for this network corresponds to a total of $r = 1700$ trips, distributed among four commodities: AL,

[1] In this paper we use the term *demand*, rather than *traffic rate*, but keep the symbol r used by Roughgarden [2001].

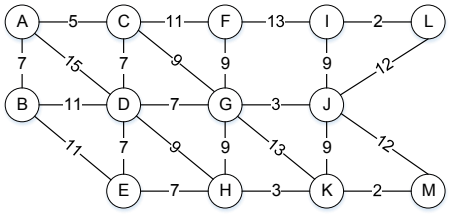


Figure 4: OW Road Network

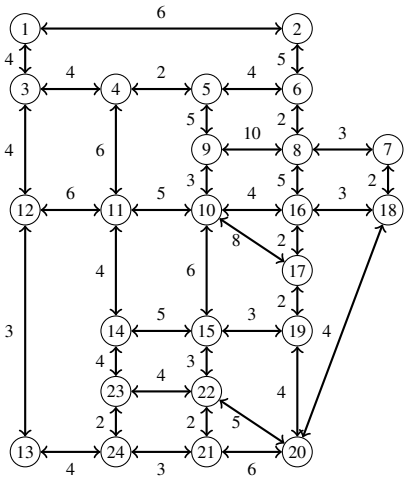


Figure 5: Sioux Falls Road Network

AM, BL, and BM (600, 400, 300, and 400 trips respectively). Furthermore, the network is defined via a latency function that relates the latency $l_a(f_a)$ of an arc a and the flow f_a associated with that arc.

3.3 Sioux Falls Network

The Sioux Falls (SF) network (Figure 5) is based on the real-world city of Sioux Falls, USA. It is frequently used as a testbed for traffic assignment approaches. The SF network uses as a latency function the BPR function (Eq. 15), where f_a is the flow on arc a, τ_a is the free-flow travel time in a, and ρ_a is its nominal capacity. In this paper we use $\tilde{a} = 0.15$ and $\tilde{b} = 4$, as suggested in the literature.

$$l_a(f_a) = \tau_a \left(1 + \tilde{a}\left(\frac{f_a}{\rho}\right)^{\tilde{b}}\right) \tag{15}$$

3.4 A Topological Coupling Statistic

In this section we introduce novel statistics and metrics to characterize networks in terms of the complexity they pose to the traffic assignment problem—in particular to multiagent reinforcement learning algorithms. We do so by measuring how *coupled* routes are. We first define and quantify route coupling, i.e., how much a given route is coupled with other routes that can be used by learning agents. The investigation of route coupling is important in multiagent reinforcement learning settings since it measures how changes in the actions of one agent interfere with the actions taken by other agents.

Assume a graph G and latency functions $l_a \in \mathbb{R}$, as previously defined; suppose that we have a demand r resulting from drivers wishing to traverse this graph. Different drivers may have different origins and may wish to reach different destination nodes. Our goal is to have drivers select paths that allow for an overall cost function to be minimized. To this end, drivers may use a set R of routes, where each route R_i is defined as a sequence of nodes connecting the starting node o_i of a route to a destination node d_i; i.e., $R_i \equiv [o_i, \ldots, d_i]$. Concretely, assume that there are $|R|$ routes defined over the graph G. If drivers were to selfishly select paths that minimize their own travel time, measured with respect to the latency l_a of each arc in a route, we could simply compute the shortest path between o_i and d_i. This strategy, however, makes assumptions that are often unrealistic: for example, that the time taken to traverse an arc is independent of other drivers. This is clearly not the case in real traffic networks.

Alternatively, we can minimize an overall cost function defined over G by using learning methods capable of identifying the optimal allocation of drivers to routes. In this case, one recurring problem is that of *non-stationarity*: if each driver independently observes the state of the network and makes a decision, the effects of that decision (e.g., its impact on total travel time) may seem to change with time in unpredictable ways. The reason for this is that choices made by *other* agents, and which the driver cannot always predict or observe, cause properties of the network to remain as unobservable variables in the learning problem. From each driver's point of view, therefore, the overall learning problem seems to change with time: even if it always selects the same action when observing a same local state, the effective result of that action depends on a series of latent factors. If drivers learn to select routes in this manner, it may take a long time for the aggregate effects of other drivers' decisions to be averaged out and for the learning algorithm to converge to an equilibrium.

We propose a way of analyzing topological properties of the network in order to estimate how likely it is that non-stationarity (caused by partial observation of other drivers' decisions or intent) will negatively affect the learning process. On one one hand, networks with sparse and non-interacting routes will typically not be affected by non-stationarity; after all, each driver's decisions will have almost no effect on the travel time of other agents, therefore resulting in a process in which drivers can independently optimize their route choices. Networks where important routes share many arcs with other routes, on the other hand, will be strongly affected by non-stationarity: if a group of agents decides to change their decisions and switch to a different route, this will directly affect the travel time of many other agents—even if they do not change their own decisions.

The metric that we propose is called *coupling*, and consists in a statistic computed based on topological properties of the network. Our goal with it is to estimate how strong the effect of non-stationarity might be and to describe a way of using it to bias the learning process, in a way that empirically counteracts the negative effects of non-stationary and is conducive to faster convergence to an equilibrium. Intuitively, the coupling statistic measures how many points of interaction a route has with other routes, and how likely it is that other agents might

decide to switch to any given alternative route. If two routes share many arcs, for instance, the effective flow on both of them will be more strongly affected by agents deciding to travel on those routes or deciding to abandon them in favor of other options; these routes are, therefore, highly coupled.

We define the coupling $\Psi(R_i)$ of a route R_i as the expected value of the *interaction* $I(R_i, R_j)$ of that route with other routes R_j in the system. This expectation is defined with respect to a probability distribution P over possible routes: routes that are more likely to be selected by agents (based on their individual preferences for reaching particular places in the network) have higher probability. Specifically, we define:

$$\begin{aligned} \Psi(R_i) &= E_P\big[I(R_i, \cdot)\big] \\ &= \sum_{j=1}^{|R|} P(R_j) I(R_i, R_j) \end{aligned} \tag{16}$$

where $I(R_i, R_j)$ is the normalized number of shared arcs between routes R_i and R_j:

$$I(R_i, R_j) = \frac{1}{|R|} \frac{|R_i \cap R_j|}{|R_i|} \tag{17}$$

Intuitively, I measures how much of the underlying structure and resources of the network are shared by two routes, and P reflects the agents' demands for different routes at some point in time—defined according to their preferences for reaching different regions of the network[2].

In our experiments, we refer to $\Psi(R_i)$ as the *mean coupling* of a route R_i whenever P is assumed to be a uniform distribution. This corresponds to the case where we have no prior information about agents' preferences for reaching particular nodes of the network. When we do have that information, we can encode it in P, which then represents the relative preferences of agents for choosing different routes.

To illustrate the use of the proposed coupling statistics, we start with a simple example—the Braess graph B^1, depicted in Figure 1(a). This network has a single commodity: the entire demand of $r = 4200$ drivers travels from s to t. There are 3 possible routes they can select from: st1, st2, and st3, as shown in Table 1. These form the set R of routes, which was generated by using a k-shortest paths algorithm Yen [1971]. This algorithm returns k shortest paths (when analyzed under free flow) associated with a given commodity. The st1 path is the shortest one to satisfy the single demand in B^1; st2 and st3 are the second and third shortest paths, respectively.

Table 2 shows the normalized number of shared arcs between any two routes, which is the second term in Equation 17. This table should be read row-wise: e.g., st1 shares 33% of its elements with st2. The coupling Ψ of each route (Equation 16) is shown in Table 3 for the case of uniform P. Under this distribution, it is equivalent to the average normalized number of shared arcs with other routes: $\Psi(st2)$, for instance, is $\frac{1}{2}(50.00 + 0.00)$. Note that st1 has the highest coupling—but traditional MARL approaches ignore this information.

We propose to use the coupling statistic to bias the learning process of each agent. We report results that relate to a very simple type of biasing; namely, the Q-table of each agent is initialized with the negative of the coupling for each route. Continuing with our example (the B^1 network), the Q-values associated with action st1 were initialized with -33.33, and the Q-values associated with actions st2 and st3 were initialized with -25.00. This means we are using information about route coupling to bias agents' preferences in a way that leads more agents to prefer routes st2 or st3, rather than st1. Our hope is that this will successfully bias route selections, guiding agents in their exploration of which are the best actions to perform and accelerating convergence to an equilibrium.

Route Name	Arcs
st1	$sv_1 \rightarrow v_1w_1 \rightarrow w_1t$
st2	$sw_1 \rightarrow w_1t$
st3	$sv_1 \rightarrow v_1t$

Table 1: $k = 3$ shortest routes for network B^1

	st1	**st2**	**st3**
st1	100.00	33.33	33.33
st2	50.00	100.00	0.00
st3	50.00	0.00	100.00

Table 2: Normalized number of shared arcs (B^1)

Ψ		
st1	st2	st3
33.33	25.00	25.00

Table 3: Route Coupling Ψ under uniform P (B^1 network)

4 Simulations and Results

In order to evaluate the use of the coupling Ψ as a biasing method in MARL we will use the OW network, since solving the traffic assignment problem in it has been shown to be a complex task. In our experiments, agents use Q-Learning, a standard reinforcement learning algorithm, to learn to select routes and reach a UE. Note that at this point we disregard the fact that the UE may be socially bad; see discussion in Section 5. In a traditional reinforcement learning setting, each agent keeps its own Q-table. Q-values are associated with each action—in this case, one of the k shortest routes available for an agent to travel from its origin to its destination. Note that this formulation resembles a repeated game, where there is just one state Claus and Boutilier [1998]. This means that Equation 10 can be simplified and does not require a discount rate. While this simplifies the learning problem, the large number of concurrently-learning agents in MARL makes the problem inherently more complex. Each agent selects actions according to an ϵ-greedy strategy: with probability $1 - \epsilon$, the action with highest Q-value is selected; with

[2]Route coupling does not take into account interactions of a route with itself. We abuse notation in Equations 16 and 17. In reality, these are defined over the set $R - \{R_i\}$, not R.

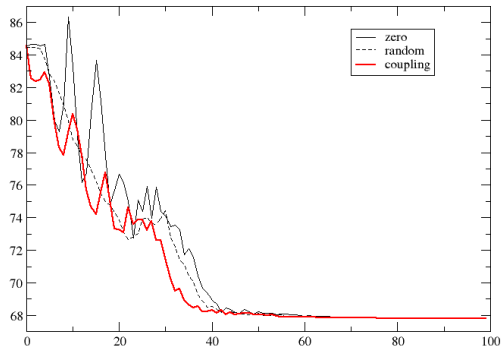


Figure 6: Mean latency as a function of number of episodes.

probability ϵ, a random action is selected. We initialize ϵ with a high value and decay it by $(1-\delta)\%$ at the end of each episode. This allows for high exploration at the beginning of the learning process. In our experiments, the reward is the negative of the individual travel time of an agent, $\alpha = 0.3$, $\epsilon = 1.0$ at the first episode, and $\delta = 0.9$. These values were selected after extensive tests with different ranges of values.

The OW network, used in the following experiments, has $|\mathcal{C}| = 4$ commodities, and we associate with it a total demand of $r = 1700$. We computed $k = 5$ shortest paths per commodity. Table 4 shows the coupling for each of the $|\mathcal{C}| \times k = 20$ shortest routes, computed via Equation 16.

In our experiments, we measure the average latency of the r agents in the system and plot it as a function of time (Figure 6). We do so under three distinct situations: (i) the Q-table of agents is initialized with zeros; (ii) the Q-table is initialized with the negative of the coupling statistic $\Psi(R_i)$ associated with route R_i; (iii) the Q-table is initialized with a random value between 0 and the (negative) maximum value of Ψ. Note that an approximation of the UE for the OW network (which can be computed, e.g., via CPLEX) is of approximately 67 minutes of average latency. In Figure 6 we show that biasing the Q-values of agents with Ψ leads to a faster convergence to the UE.

In the future we plan to perform further experiments; first, one that starts with a lower value of ϵ so that agents can exploit the bias provided at the beginning of the learning process for longer periods of time. Due to lack of space, we omit a table that shows that, at the end of the learning process, the number of agents using each of the k paths is roughly correlated with Ψ of the corresponding path. This suggests that the coupling statistic does serve, indeed, as a proxy to how desirable different routes are—one that considers how agents selecting between them are (as a consequence of the concurrent learning aspect of MARL) negatively impacted by non-stationarity.

5 Related Work

A number of techniques from transportation planning, economics, operations research, and computer science deal with the TAP. Due to lack of space, we omit classical approaches. The reader is referred to Ortúzar and Willumsen [2001]. For

Name	Arcs	Mean Coupling
AL1	AC → CG → GJ → JI → IL	36.84
AL2	AC → CG → GJ → JL	39.47
AL3	AC → CF → FI → IL	23.68
AL4	AC → CD → DG → GJ → JI → IL	30.70
AL5	AC → CD → DG → GJ → JL	31.58
AM1	AC → CD → DH → HK → KM	31.58
AM2	AC → CG → GJ → JK → KM	37.89
AM3	AC → CG → GH → HK → KM	32.63
AM4	AD → DH → HK → KM	22.37
AM5	AC → CG → GJ → JM	35.53
BL1	BD → DG → GJ → JI → IL	27.37
BL2	BD → DG → GJ → JL	27.63
BL3	BA → AC → CG → GJ → JI → IL	32.46
BL4	BA → AC → CG → GJ → JL	33.68
BL5	BA → AC → CF → FI → IL	21.05
BM1	BE → EH → HK → KM	21.05
BM2	BD → DH → HK → KM	27.63
BM3	BD → DE → EH → HK → KM	20.00
BM4	BE → ED → DH → HK → KM	18.95
BM5	BD → DG → GJ → JK → KM	28.42

Table 4: Meaning Coupling statistic of the $k = 5$ routes associated with each of the four OD Pairs in the OW network.

approaches that seek to balance the UE and the SO, we refer the reader to Bazzan and Chira [2015]. We focus on those that seek to approximate the UE by means of reinforcement learning or other agent-based approaches.

To the best of our knowledge, no attempts have been made to bias the learning of the UE by using similar statistics and metrics as the one we propose. Similar metrics such as the Path Size Logit model [Ben-Akiva and Bierlaire, 1999] and the C-Logit model [Cascetta *et al.*, 1996] are used to select routes in a network in route choice models. However, these strategies differ in the formulation metrics since they are based on the length of arcs.

A natural way to tackle the problem of route choice is via agent-based simulation techniques. Examples are MATSim Balmer *et al.* [2004], Klügl and Bazzan [2004] and Dia and Panwai [2014]. A learning-based approach to route choice was proposed by Tumer and Agogino [2006], where agents learn to select from pre-computed routes in a single-commodity network.

Finally, another topic related to our objective is the study of performance degradation caused by the selfish behavior of individual road users—this remains an important research topic, as shown by Koutsoupias and Papadimitriou [1999] regarding the PoA problem.

6 Conclusions and Future Work

Traffic assignment and route choice are difficult learning problems because the routes available for agents may be highly coupled. Furthermore, the price of anarchy might be strongly affected by issues such as the topology of the network, the demand distribution, and the nature of the latency functions, among other factors. In this paper we presented the

first steps towards the definition of statistics and metrics that quantify how difficult the traffic assignment is, in general—and in particular how difficult the computation of an UE is.

In this work we assumed that agents trying to reach an UE learn to select routes independently. As a consequence, their learning processes can be negatively affected by the non-stationarity intrinsic to MARL settings. To counteract this effect, it might be beneficial to bias agents' decisions in order to accelerate convergence towards less coupled routes. We have shown how to define and compute such a coupling metric and experimented with using the coupling statistics as initial Q-values; our objective was to give agents initial incentives to select routes that are less coupled. Preliminary results show that this strategy can lead to faster convergence.

Ongoing work is being developed to improve the biasing strategy. We are also working towards evaluating the proposed approach in more complex networks, such as the Sioux Falls network and the Braess graphs with higher p values. Since the Sioux Falls network has 528 commodities, it may be time consuming to compute couplings for, say, all $k = 4$ routes for each of the commodities. We plan to use previous results from Chudak *et al.* [2007] to pre-select routes that contain the most congested arcs, and concentrate our analyzes on them. Finally, given that we do have prior information about agents' preferences for reaching particular nodes of the network (i.e., we do know r_σ for each commodity), we can also define a variant of the coupling statistic Ψ which is weighted by these preferences. We also plan to extend the learning process to a state-based one, such as that described in Bazzan and Grunitzki [2016], where agents learn to select an *arc* (action) at each *node* (state) of the network that is visited. This contrasts with our current model, where agents learn to select a complete pre-defined route among k options.

Acknowledgments

Fernando Stefanello was supported by CAPES/PNPD.
Ana Bazzan is partially supported by CNPq.

References

Michael Balmer, Nurhan Cetin, Kai Nagel, and Bryan Raney. Towards truly agent-based traffic and mobility simulations. In N.R. 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.

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

Ana L. C. Bazzan and Ricardo Grunitzki. A multiagent reinforcement learning approach to en-route trip building. In *To appear in 2016 International Joint Conference on Neural Networks (IJCNN)*, 2016.

Martin Beckmann, C. B. McGuire, and Christopher B. Winsten. Studies in the economics of transportation. Technical report, Yale University Press, 1956.

Moshe Ben-Akiva and Michel Bierlaire. Discrete choice methods and their applications to short term travel decisions. In *Proceedings of the International Series in Operations Research & Management Science*, pages 5–33. Springer, 1999.

D. Braess. Über ein Paradoxon aus der Verkehrsplanung. *Unternehmensforschung*, 12:258, 1968.

Bureau of Public Roads. *Traffic Assignment Manual*. US Department of Commerce, Urban Planning Division, Washington D.C., 1964.

Ennio Cascetta, Agostino Nuzzolo, Francesco Russo, and Antonino Vitetta. A modified logit route choice model overcoming path overlapping problems. specification and some calibration results for interurban networks. In *Proceedings of the 13th International Symposium on Transportation and Traffic Theory*, pages 697–711, 1996.

Fabian A. Chudak, Vania Dos Santos Eleuterio, and Yurii Nesterov. Static traffic assignment problem. a comparison between Beckmann (1956) and Nesterov & de Palma (1998) models. In *Proceedings of the 7th Swiss Transport Research Conference*, pages 1–22, sep 2007.

Caroline Claus and Craig Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 746–752, 1998.

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

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

Elias Koutsoupias and Christos 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.

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

Tim Roughgarden and Éva Tardos. How bad is selfish routing? *J. ACM*, 49(2):236–259, 2002.

Tim Roughgarden. Designing networks for selfish users is hard. In *Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on*, pages 472–481. IEEE, 2001.

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

John Glen Wardrop. Some theoretical aspects of road traffic research. In *Proceedings of the Institution of Civil Engineers*, volume 1, pages 325–362, 1952.

Jin Y. Yen. Finding the k shortest loopless paths in a network. *Management Science*, 17(11):712–716, 1971.