## RESEARCH OUTPUTS / RÉSULTATS DE RECHERCHE

**A dynamic behavioural traffic assignment model with strategic agents**

Barthelemy, Johan; Carletti, Timoteo

Link to publication

# A dynamic behavioural traffic assignment model with strategic agents ☆

Johan Barthélemy [a,*], Timoteo Carletti [b]

[a] SMART Infrastructure Facility, University of Wollongong, Australia
[b] Namur Research Center for Complex Systems, University of Namur, Belgium

## ARTICLE INFO

## ABSTRACT

Foresee traffic conditions and demand is a major issue nowadays that is very often approached using simulation tools. The aim of this work is to propose an innovative strategy to tackle such problem, relying on the presentation and analysis of a behavioural dynamic traffic assignment.

The proposal relies on the assumption that travellers take routing policies rather than paths, leading us to introduce the possibility for each simulated agent to apply, in real time, a strategy allowing him to possibly re-route his path depending on the perceived local traffic conditions, jam and/or time already spent in his journey.

The re-routing process allows the agents to directly react to any change in the road network. For the sake of simplicity, the agents' strategy is modelled with a simple neural network whose parameters are determined during a preliminary training stage. The inputs of such neural network read the local information about the route network and the output gives the action to undertake: stay on the same path or modify it. As the agents use only local information, the overall network topology does not really matter, thus the strategy is able to cope with large and not previously explored networks.

Numerical experiments are performed on various scenarios containing different proportions of trained strategic agents, agents with random strategies and non strategic agents, to test the robustness and adaptability to new environments and varying network conditions. The methodology is also compared against existing approaches and real world data. The outcome of the experiments suggest that this work-in-progress already produces encouraging results in terms of accuracy and computational efficiency. This indicates that the proposed approach has the potential to provide better tools to investigate and forecast drivers' choice behaviours. Eventually these tools can improve the delivery and efficiency of traffic information to the drivers.

© 2017 Elsevier Ltd. All rights reserved.

## 1. Introduction

In the recent decades researchers from socio-economical sciences have increasingly tighten the ties with colleagues from other disciplines such as physics, mathematics and computer science just to mention few of them. Hence methods and tools own to the latter have been successfully improved and then applied to understand some socio-economical phenomena.

---

However differences are still present between such domains, in particular concerning the experimental method and its reproducibility. In social sciences experiments are highly time consuming as many repetitions are needed to cover the human heterogeneity and unravel possible "universal social laws" not yet determined or limited to describe aggregate data, quoting Asimov "In studying society, we place human beings in the place of subatomic particles, but now there is the added factor of the human mind. Particles move mindlessly; human beings do not. To take into account the various attitudes and impulses of mind adds so much complexity that there lacks time to take care of all of it" (Asimov, 1988). Moreover there is a clear difficulty to scale up results of experiments done on small group to larger sizes (cities, countries).

For all these reasons, scientists resort to numerical micro-simulations to tackle relevant societal research questions (Helbing, 2016), in particular individual based models, also called micro-simulations, are very appealing, once properly calibrated on real data, because they allow to fully take care of the heterogeneity of the involved agents. This approach has been already used and proved its validity in transportation problems.

Traffic flows simulation represents a central part of traffic micro-simulators such as POLARIS (Auld et al., 2016), MATSim (Meister et al., 2010), DynaMIT (Ben-Akiva et al., 1998), DynaSmart (Mahmassani et al., 1992), AIMSUN (Barceló and Casas, 2005), DynusT (Chiu et al., 2011b) and DTALite (Zhou et al., 2014) as well as the traffic modelling part of UrbanSim (Waddell, 2002) and ILUTE (Salvini and Miller, 2005) integrated simulators. This component is in charge of executing the daily plans of simulated individuals in a physical environment, i.e. representing the traffic flows dynamics on a road network.

In recent decades, dynamics traffic assignment (DTA) models have emerged for solving this problem (see Chiu et al. (2011a) for an extensive description of these techniques), which either aim at reaching a steady-state of the considered system or at simulating the agents route choice behaviours. A steady-state of the system is achieved when it reaches either one of the following.

**User equilibrium**
the journey times on all the routes actually used are equal, and less than those which would be experienced by a single vehicle (or user) on any unused route (Wardrop's first principle, Wardrop, 1952);

**Stochastic user equilibrium**
no user believes he/she can improve his/her travel time by unilaterally changing routes (Daganzo and Sheffi, 1977).

DTA techniques can also be distinguished by their analytical or simulation-based nature (Peeta and Ziliaskopoulos, 2001). Analytical methods formulate the traffic assignment as non-linear programming and optimisation problems or variational inequalities instead of focusing on the agents' behaviours. Examples of such works include Friesz et al. (1993), Merchant and Nemhauser (1978a,b). Even though they have demonstrated their usefulness and are grounded on sound mathematical theories, their complexity and computational cost make their application to large-scale scenarios difficult (Peeta and Ziliaskopoulos, 2001).

Hence simulation-based methods, which explicitly model the individuals' mobility behaviours and allow to account for the heterogeneity of the involved agents, have recently gained more attention in the literature (Nagel and Flötteröd, 2009; Bazghandi, 2012; Ben-Akiva et al., 2012). The underlying idea is to try reaching an (stochastic) user equilibrium by means of iterative simulations. These successive steps generate traffic flows until the travel time of every agent becomes stationary, i.e. reaches a (stochastic) user equilibrium. This class of models is more suited to an agent-based approach than the analytical ones by focusing on agents' mobility behaviour rather than optimising a complex objective function. Nevertheless, due to their iterative nature those methods can be endowed with computational issues. Indeed if the road network and the number of agents involved are large, the algorithms of this type may converge slowly to an equilibrium state (Pan et al., 2012).

We can observe that both categories of DTA methods for steady-state solutions are not suitable to dynamic networks as the agents lack of real-time response to network modifications. For instance if an accident occurs at some point of an agent's trip, if the number of agents in the network changes, or if the network is modified by adding/removing streets the whole optimisation/iterative stages must be repeated to compute a new equilibrium. Moreover these steady-states approaches rely on strong assumptions and have several limitations that have been long well identified. We refer the reader to Dehoux and Toint (1991) for a discussion of these limitations and why these models should be avoided in favour of purely behavioural models.

These behavioural models, focusing on the travellers decision (in terms of path and/or departure time choices) have been widely investigated since the seminal work of Mahmassani and Chang (1987). For instance behavioural models are proposed in the FREESIM (Rathi and Nemeth, 1986), CARSIM (Benekohal and Treiterer, 1988), PACSIM (Cornélis and Toint, 1998) and AgBM-DTALite (Xiong et al., 2017) simulation packages. Interested readers may find a review of these schemes in Pel et al. (2012). Investigating these models leads to the development of new tools able to better investigate and forecast drivers' behaviours, a necessary step for the design of efficient traffic information delivery systems.

The aim of this work is to present a first step development of a behavioural DTA. The proposal relies on the assumption that travellers can take routing policies rather than paths. This concept, originally used for transit (Nguyen and Pallotino, 1988; Spiess and Florian, 1989; Nguyen et al., 1998) has been adapted to model drivers' adaptive route behavior (Gao et al., 2010; Gao, 2012; Xiong et al., 2016). The existing works typically assume that a driver has access to real time information that can be provided via a proper device (e.g. a radio, a connected navigation system or dynamic message signs.

Recently, Kim et al. (2014) proposed to use the extended desire-belief-intention framework to simulate the drivers' en-route behaviour. This approach seems promising but only relies on drivers' past experience to trigger the rerouting process. Due to its computational complexity, it is also not suited to large scale scenarios.

These observations lead us to introducing the possibility for each simulated agent to apply a simple strategy that allows it to possibly dynamically re-route his path depending on perceived local traffic conditions, i.e. without the need of an external information provider (Barthélemy and Carletti, 2017). Let us also observe that the introduction of a global information system, for instance GPS or radio, is not so straightforward because it requires to study and incorporate into the model the way agents decide to cope with such information (Ma et al., 2016). For instance the total solar eclipse of August 2017 caused disastrous traffic jams in various states across the United States of America, even though road traffic information was provided to the drivers through various means of communications (Wright, 2017).

Due to the exploratory nature of this work, we decide to model the agents' strategy with a simple neural network whose parameters are determined during a preliminary learning stage. As every agent can have a different strategy, this approach has the potential to simulate heterogeneous driver behaviours obtained by multi-user class dynamic traffic assignment (Lam and Huang, 1992; Peeta and Mahmassani, 1995). This strategic approach has been applied on several networks, characterised by different congestion levels and sizes to assess its effectiveness and robustness to investigate its performance.

The paper is organised as follows. Section 2 formally details the design of the agents' strategies and their optimisation process. The resulting mobility behaviour is then illustrated under various scenarios, testing the robustness of the strategies, in Section 3. Finally concluding remarks and perspectives are discussed in Section 4.

## 2. Methodology

This Section details a dynamic traffic assignment model relying on strategic agents perceiving local traffic conditions and thus adapting their behaviour to reduce their trip duration. We represent each agent as a *neural network* whose inputs are the local information about the route network and whose output is the action to undertake: stay on the same path or modify it (keeping unchanged the destination). Because we are interested in modelling adaptive agents able to well perform in many different scenarios, we decide to introduce a learning phase instead of fine tuning by hand each agent's behaviour (defined by the parameters of the neural network), in this way the emerging strategies would be able to have good performances across different environments, possibly changing in time.

We firstly describe formally the agents' strategy based on a neural network, and its application on a road network. The remaining part of the Section is then devoted to the learning process of the agents, optimising their strategies via a genetic algorithm.

### 2.1. A neural-network based strategy for dynamic traffic assignment

#### 2.1.1. Traffic simulation

Assume a road network represented by a directed graph $G = (N, L)$, where $N$ and $L$ correspond respectively to the sets of nodes and links which can be though respectively as junctions and roads and a set $\mathscr{A}$ composed by $n_{car}$ agents willing to move in the network.

The resulting traffic on the road network is simulated using a *queue* model where the links are modelled as simple *first-in-first-out* (FIFO) queues. This mesoscopic model, also retained by MATSim, is vehicle-based and has been shown to offer a satisfactory representation of the travel times as well as excellent computational efficiency (Charypar et al., 2015). Consequently it reaches an suitable compromise to more detailed and computationally costly car-following models.

Every link $l \in L$ is a queue characterized by the following properties:

- a free flow travel time $l_t$;
- a length $l_l$ and number of lanes $l_n$ defining the queue storage capacity;
- a flow capacity $l_c$, i.e. the maximum number of vehicles per km per hour on the link;
- the number of agents currently in the queue $l_v$

The traffic dynamic is then performed by updating the state of each queue at every time step: for each $l \in L$ such that $l_v > 0$, if the following conditions yield for the first agent of the queue.

- it has spent at least a duration of $l_t$ on the link;
- $l_c$ has not been exceeded during the current time step;
- and the next link on its route has free storage capacity

then the agent is removed from $l$ and put in the next link of its route. It must be noted that the time step retained in this work is one second.

This approach ensures that the specifications of the network are accounted for as the flow capacity and the storage capacity constraints may cause congestion on the links (which can spill-back). Additionally it can easily be extended to capture congestion shock waves using for instance Newell's simplified model for kinematic waves (Newell, 1993; Zhou et al., 2014).

### 2.1.2. A strategy for en-route decision making

Initially each agent plans a nominal shortest path source-destination, assuming he can travel at free-flow speed, i.e.

$$l_v = 0 \quad \forall l \in L. \tag{1}$$

This shortest-path computation implies that the agents have a complete knowledge of the network topology and have a fully rational behaviour, which are strong assumptions[1] (Downs and Stea, 1977), let us observe however that such knowledge is limited to the nominal state of the network, that is agents do not have access to real time information on the dynamics state of the network and even if this information was available, it is not trivial to determine how agents cope with the latter. Agents choices being independent from each other, it can result in some links being overcrowded and therefore presenting a severe speed reduction. Hence these parts of the path will no longer be the optimal ones.

As proposed by Bonsall (1992) we thus introduced a possibility for each agent to apply a strategy allowing it to re-route his path or stay on it (without modifying its source, destination and departure time) given perceived local network conditions. This strategy is realised using a neural network: a machine learning method used to solve a wide variety of tasks that are difficult to solve using ordinary rule-based programming. We refer the reader to Kriesel (2007) for a comprehensive description of these methods.

As this work is exploratory, we decided to use a simple and straightforward neural network implementation, where the strategy is realised with two inputs and one output only. Its simple design is presented in Fig. 1. The input nodes respectively read.

- the normalised[2] time spent from the source up to the current position, $x_1$;
- and the saturation[3] of the next link on the path, $x_2$.

The binary output node $y_{out}$ gives 1 if the agent strategy is to change his path, or 0 otherwise. For the sake of simplicity, there is no hidden layer between the input and output nodes, thus the output is given by

$$y_{out} = \Theta(\cos(\alpha)x_1 + \sin(\alpha)x_2 - \theta) \tag{2}$$

where $\Theta$ is the Heaviside step function, $\cos(\alpha)$ and $\sin(\alpha)$ are synapses weights and $\theta$ the threshold of the output node. Let us observe that smoother decision functions, such as a logistic one, could be used. If the agent chooses the re-routing, then he computes a new shortest-path avoiding the congested link between his current location and his destination. It is important to note that each agent is associated with its own neural network, i.e. different agents can have different values of $\alpha$ and $\theta$, introducing heterogeneous mobility behaviours amongst the agents.

Despite its simplicity, the neural network can be easily improved to take into account many other inputs and/or agent's characteristics as explained below and it has a straightforward interpretation; from the local knowledge of the actual route usage, each agent takes his decision based on the importance he attributes to the (normalised) number of cars in front of him - $x_{1,min}$ in the right panel Fig. 1 - and to the (normalised) time his is willing to loose to perform his trip - $x_{2,min}$ in the right panel Fig. 1. As in real life, every agent has his own personal preferences that can be faithfully represented by an agent based model.

This strategy seems reasonably behaviourally consistent. Previous research works (Wachs, 1967; Ueberschaer, 1971; Bonsall and May, 1986) highlighted the fact that if a traveller already spent a larger amount of time en-route than it should have taken, and if he perceives congestion on the next road he intents to take, then the agent may reconsider a re-routing to avoid it. Hence, the re-routing decision depends on the experience of the current journey and expectation of congestion ahead.

More sophisticated neural network could be considered by adding hidden layers or inputs (Bonsall, 1992), such as the theoretical time saving the agent would experiment if he re-routes his initial path, the number of re-routing he already achieved, the remaining distance to his destination, the next road type and/or its length, the congestion level of the links to reach the destination (not only the next one on the path), a memory of previous choices, global information provided by a navigation system, *etc.*

Nevertheless we focused on a simple strategy in the present work in order to reach a trade-off between simplicity and efficiency of the strategy. Furthermore even if the agent had access to global information, it is likely that most of the re-routing will occur near the congested link. It can also be argued that the memory of previous choices is already embedded in the values of the parameters of the strategy.

Each time an agent reaches the end of a link, it computes its strategy and decides what to do: stays on the path or reroutes.

---

[1] Note that this hypothesis is also assumed when dealing with the user equilibrium models.
[2] Normalised means divided by the nominal time one should have spent, i.e. in free flow conditions.
[3] The ratio of the combined length of the cars in the link divided by its length.
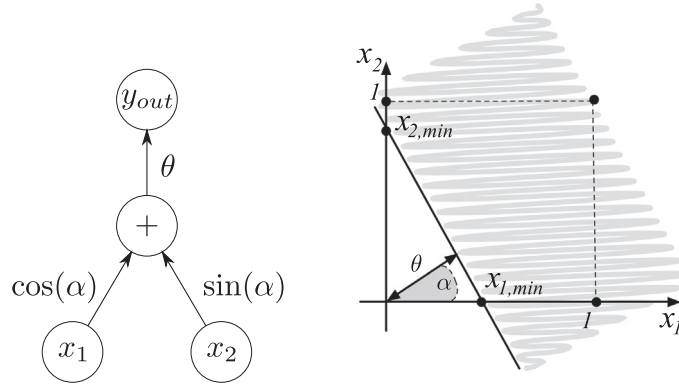
**Fig. 1.** The neural network for a strategic agents. Left panel: schematic representation of the neural network, the input layer consists of nodes $x_1$ and $x_2$ which are respectively weighted by the parameters $\cos(\alpha)$ and $\sin(\alpha)$. If the weighted combination of the inputs exceeds a threshold $\theta$ then output node is activated and $y_{out} = 1$; otherwise $y_{out} = 0$. Right panel: the strategy, depending of the parameters $\alpha$ and $\theta$. Each agent, measuring $x_1$ and $x_2$, decides to re-route or not his path. The dashed area corresponds to values of saturation and time already spent for which the agent will re-route his trip.

As already stated such decision depends strongly on the network topology and on the whole set of agents' behaviour, the aim of the next section is to provide the necessary details of the strategy optimisation.

### 2.2. Strategy learning with genetic algorithm

In our context we aim to minimise the time needed to perform a given source-destination trip. Stated differently, the goal is to allow the agent to choose the best links, i.e. the less congested ones on his path taking into account the dynamically varying traffic conditions. The neural network has two strategy parameters to be determined, the weight $\alpha$ and the threshold $\theta$. Note that standard continuous optimisation methods are not applicable in our context of fitness maximisation. Indeed an agent's fitness depends on number of agents he encounters on each link he passes by. Hence it is a discrete optimisation problem whose mathematical formulation depends on every agent's path and becomes intractable, being the objective function widely oscillating (see Fig. 2). Consequently we choose to solve this optimisation problem using a heuristic method, our choice was oriented towards Genetic Algorithm, because of their simplicity and robustness, but other choices could be possible.

A genetic algorithm is a heuristic search that mimics the process of natural selection in order to find an optimal solution to a given problem. This methodology belongs to the class of evolutionary algorithms, which generate solutions using techniques inspired by natural evolution such as mutation, selection, and crossover. We refer the reader to Eiben and Smith (2003) for a detailed overview of this methodology.

The parameters $\alpha$ and $\theta$ have been cast into a chromosome $\chi(\alpha, \theta) \in [0, \pi] \times [-1, 1]$ to which we associate a fitness value $\in [0, 1]$ that reflects the optimality of the solution it encodes: the higher the fitness, the lower the travel time. We decided to use a standard genetic algorithm. Thus, we first generate an initial population of chromosomes $G_0 = (\chi_1^0, \ldots, \chi_n^0)$ where $\chi_i^0$'s components are randomly drawn from a joint Uniform distribution $U([0, \pi] \times [-1, 1])$ and their fitness are evaluated. The following steps are then executed to iteratively generate $h$ populations $G_1, \ldots, G_h$. At step $k \geqslant 1$.

1. A virtual population $G\prime_{k-1}$ formed by $m$ elements $\in G_{k-1}$ is generated by repeating $m/2$ times:
   - randomly draw a couple of chromosomes (the parents), with replacement. The weights are set accordingly to the chromosomes fitness values: the higher the fitness, the larger the probability to be selected[4];
   - the parents are crossed[5] with probability $q_X$ and the offspring added to $G\prime_{k-1}$;
   - if the crossover is not performed, the parents are directly added to $G\prime_{k-1}$;
2. A mutation operator[6] acts on all element of $G\prime_{k-1}$ with probability $q_M$;
3. Finally the next population $G_k$ is constructed by selecting the $n$ best fitted elements from the larger population $G_{k-1} \cup G_{k-1}\prime$.

The last point concerns the computation of the fitness. Every agent is randomly assigned one origin-destination pair. One particular agent $a_s$ is then provided with a strategy while all the remaining ones will not change their path, that is they will always and unconditionally follow the shortest path computed using the nominal network conditions. We then compute the

---

[4] This process is also known in the literature as the roulette wheel selection.

[5] Crossing chromosomes means exchanging some of their parameters.

[6] As the chromosomes parameters are real numbers, a *continuous* mutation is adopted, i.e. we add to the parameter to be mutated a random number drawn from an Uniform distribution $U[-\delta, \delta]$ for a small positive $\delta$.
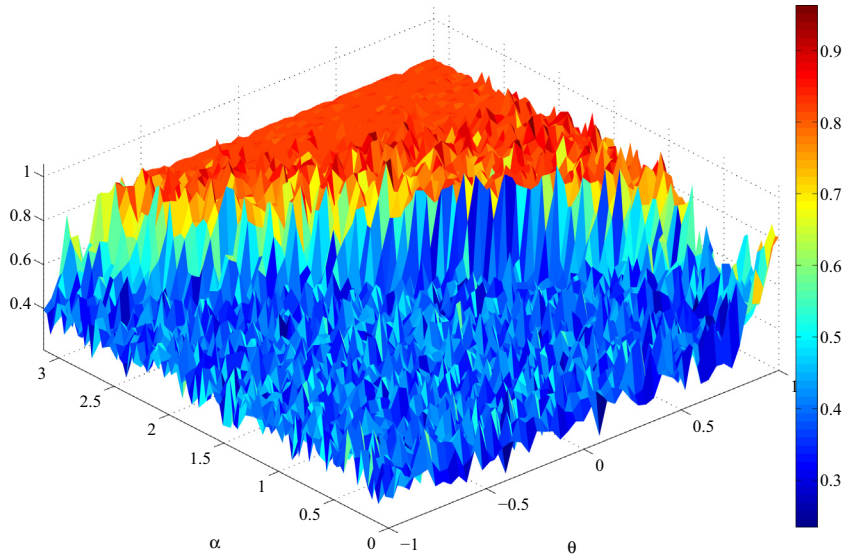
**Fig. 2.** Fitness value as a function of $\theta \in [-1, 1]$ and $\alpha \in [0, \pi]$.

nominal time $t^*$ needed by $a_s$ to perform $S$ times his source-destination trip and we divide it by the actual time $t$ needed. Assuming that $(l_1, \ldots, l_d)$ is the sequence of links covered by $a_s$ to reach his destination, then the ratio is given by

$$f(a_s) = \sum_{i=1}^{d} \frac{t_{l_i}^*}{t_{l_i}} \qquad (3)$$

The fitness associated to this agent-chromosome is the average of the value obtained using the previous formula, by repeating $N_{rep}$ times the above scheme, paying attention that the origin-destination nodes will be different in each replica. In this way we are able to reduce the impact of extreme events in the computation of the chromosome fitness.

In Fig. 2 we show a possible fitness function for the *2 cities* scenario detailed in Section 2.2.1, computed along the previously described algorithm and taking a fine mesh of values for the parameters $\alpha$ and $\theta$. It can be observed that the objective function presents a large number of local maxima, which further justifies the use of a genetic algorithm to explore the parameters space.

### 2.2.1. Application to the 2 cities scenario

The goal of the Section is to introduce the reader to our model more than provide a precise application. We decided to apply the previous learning process with the parameters given in Table 1 to the artificial scenario *2 cities* whose details can be found in Table 2. Fig. 3 illustrates the associated road network, which was artificially constructed for the agents learning purposes and consists of 2 urban centres linked by 3 roads playing the role as possible bottlenecks. The origin and destination of each agents are randomly chosen among the network nodes with uniform probability.

The parameters of the genetic algorithm have been empirically determined to obtain a trade-off between the solution quality and the computation times, that is essentially sensitive to the parameters $N_{rep}$ and $h$, because the total number of iterations is given by $N_{rep} \times h$. Increasing the (virtual) population size only marginally affected the solution quality. Note that the higher the mutation and crossovers probabilities, the more the feasible solution space is explored.

Fig. 4 reports the convergence of the genetic algorithm given by the fitness evolution (maximum and average values) as a function of the generation number. One can easily observes that the algorithm converges quickly, i.e. a limited number of generation steps produces a set of strategies with high fitness, close to the optimal one. Based on this remark, we thus decided to limit the number of generations to 10 since genetic algorithms are typically very computationally intensive. The current execution time is approximately 1h45 with a Matlab implementation, running on an Intel(R) Core(TM) i7-7700HQ CPU @ 2.80 GHz and 16 Gb of RAM.

### 2.3. Sensitivity of the learning process

As only local information is used by the strategic agent, the shape of the network used to train the strategies is not important. On the other hand the set of optimal values for the parameters defining efficient strategies can be sensitive to the level of congestion in the training network.

**Table 1**
Genetic algorithm parameters.

| Parameter | Value | Description |
|-----------|-------|-------------|
| $S$ | 5 | Stopping criterion |
| $n$ | 16 | Population size |
| $m$ | 8 | Virtual population size |
| $q_X$ | 0.1 | Crossover probability |
| $q_M$ | 0.1 | Mutation probability |
| $h$ | 10 | Number of generations |
| $N_{rep}$ | 2 | Number of origin-destinations pairs |

**Table 2**
Learning scenario characteristics.

| Parameter | Value |
|-----------|-------|
| Nodes | 59 |
| Links (50 km/h) | 182 |
| Total length (km) | 196 |
| Mean capacity per km (agents/hour) | 1.9 |
| Number of agents | 100 |
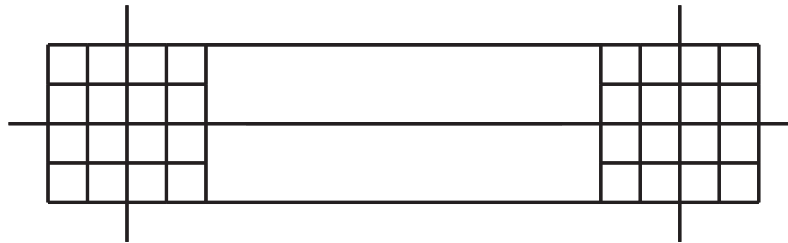


**Fig. 3.** 2 cities network.
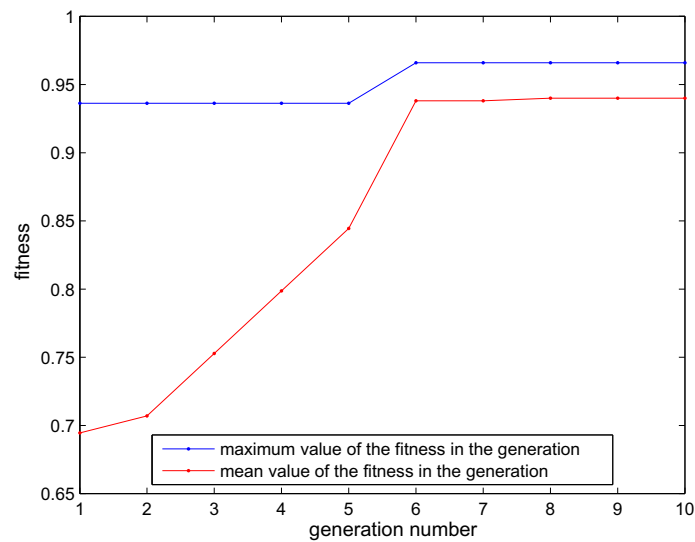


**Fig. 4.** Fitness evolution over the generations.

In order to investigate this sensitivity, we modify the *2 cities* scenario by scaling the original capacities with the following ratios:

$$200\%, \quad 150\%, \quad 125\%, \quad 100\%, \quad 75\% \quad \text{and} \quad 50\%.$$

The other parameters of the learning scenario remain unchanged.

To each agent but one, we randomly assign one origin-destination pair and it will perform its journey using the shortest path, computed using the nominal network conditions, never deviating from it whatever it is the congestion level of the used streets. To the remaining particular agent we assign a strategy defined through the parameters $\theta$ and $\alpha$.

We then use a process similar to the one used to obtain Fig. 2. For each ratio, 400 combinations of values for $\theta \in [-1, 1]$ and $\alpha \in [0, \pi]$ are tested on 10 different origin-destination pairs. The fitness values resulting from these trips are then averaged. It should be noted that the same pairs of origin-destination are used for testing every ratio.

The results of these experiments are illustrated in Fig. 5. It can be seen that the set of optimal values remain in the same area as congestion occurs in the network (i.e. when the ratio is lower than 150%). When there is no or little congestion (scaling of 200% and 150%), every strategy seems to be equivalent as their fitness values are not significantly different. This indicate that the values defining optimal strategies are robust against the level of congestion in the training network as soon as it reaches a minimum amount.

## 3. Results

The goal of this Section is to present some preliminary results of our re-routing model. In particular we are interested in analysing the efficiency and robustness of the strategies obtained in Section 2.2.1, under new environments and the impact of the proportion of strategic agents present in the population.

We initially conducted experiments over the *2 cities* scenario and two additional scenarios detailed in Table 3 with different proportions of strategic agents to test their adaptability to new conditions. The scenarios attributes were chosen to explore various network conditions in terms of mean capacity per km in order to assess the strategy adaptability with respect to new environments. Besides the *2 cities* network used at the learning stage, the additional scenarios also involved are:

- an artificial network consisting of 3 urban centres surrounded by main roads and joined by highways, represented in left panel of Fig. 6;
- the Chicago road network, available at https://github.com/bstabler/TransportationNetworks, and represented in right panel of Fig. 6. Note that the original capacities of the Chicago network have been downscaled to obtain congestion with less agents in order to keep reasonable computation times.

The goodness of the strategy will be evaluated using 2 indicators: the ratio of used links in the network and the fitness of the agents, i.e. the ratio defined by Eq. (3). The former indicator is network related: the larger number of used streets indicates a better exploitation of the network by the strategic agents because the traffic load is more balanced over its links, thus less traffic jam are likely to occurs. The latter is dedicated to the agents satisfaction: a high value indicates that the agent average speed is close to the one he would experience on an empty, or very diluted, network.

These initial experiments are then followed by three cases applications: the traffic simulation for the cities of Sioux-Falls, Namur and Randwick. The generated traffic flows will be compared against the ones produced by an existing traffic micro-simulator and real traffic data.

### 3.1. Impact of the strategic agents proportion

Let us first examine how the proportion of strategic agents in the simulation influences the average fitness of every agents. The following proportions

$$0\%, \quad 10\%, \quad 25\%, \quad 50\%, \quad 75\% \quad \text{and} \quad 100\%$$

are retained in our experiments. The efficiency of the learning process is also investigated by comparing trained agents against agents following a nominal shortest path trip, that would be used as a baseline model, and agents with random strategies[7]

The evolution of the fitness is reported in Fig. 7. One can observe that the proportion of strategic agents (both optimised and random) does have an impact on the average fitness experienced by the agents:

- For every tested scenario there is an increase of the average fitness value for proportions of optimised agents up to 50% of the total number of agents, indicating that the strategy is efficient. Moreover a proportion up to 75% and 100% of such agents has more significant and positive impact in the scenario involving the most congested road networks, i.e. with lower mean capacities (*2 cities* and *Chicago*). For less congested network (*3 cities*), in case the number of optimised agents is high, the overall performance may not increase and even drop. Indeed when they re-route themselves, they may encounter links being more congested than the initial ones, resulting in a lower fitness as they always suffer a speed reduction;

---

[7] Meaning characterised by weights and threshold randomly drawn and not optimised.
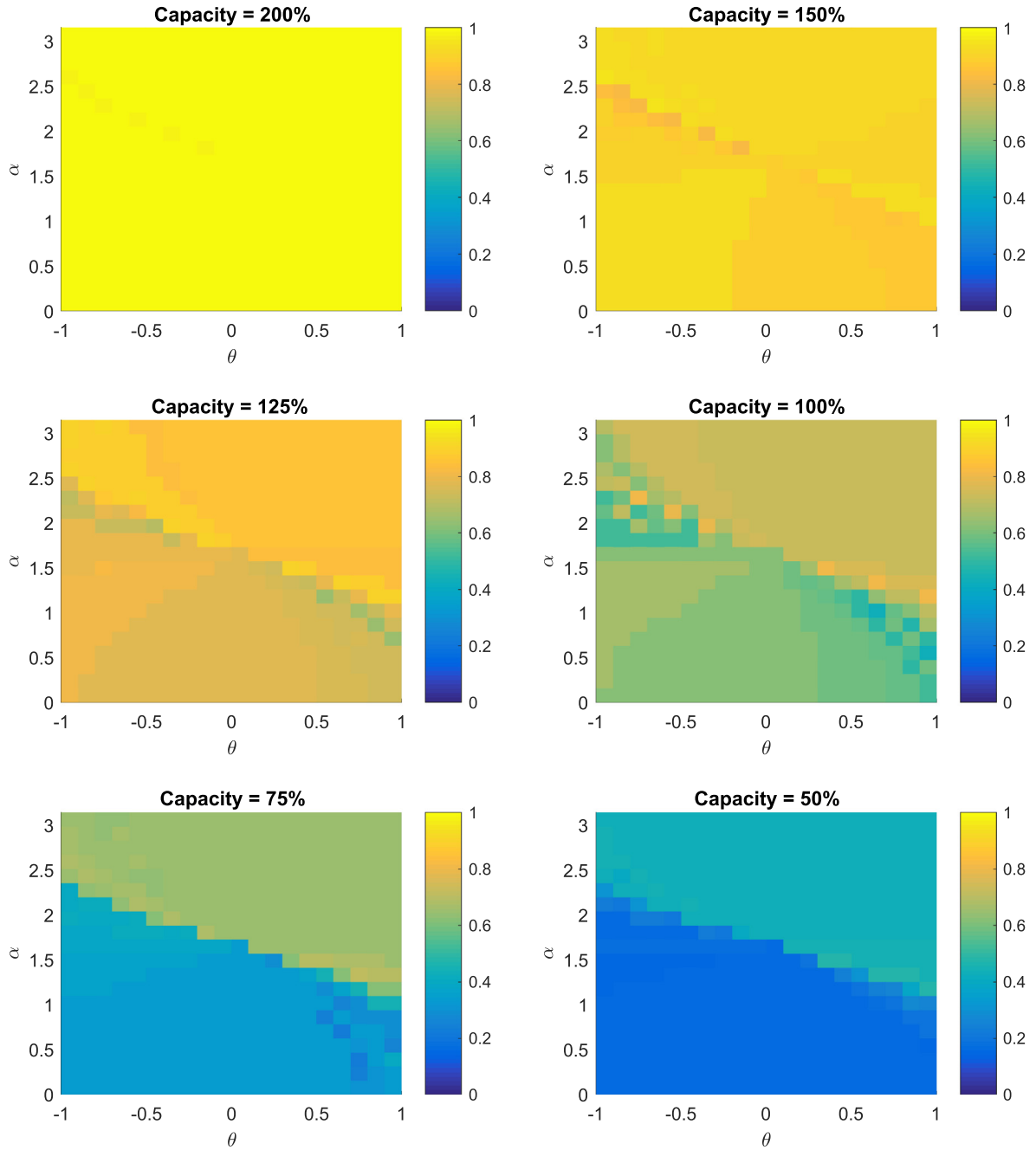
**Fig. 5.** Fitness values for one strategic agent as a function of $\theta$ and $\alpha$ for different scaling of link capacities in the scenario *2 cities*. The fitness values have been averaged on 10 runs for each scaling. Each run specifies a different origin-destination pair for the strategic agent.

- Agents performing random re-routing experience a decrease of the overall performance as their proportion increases with the exception of the *3 cities* scenario. This behaviour is certainly imputable to the uncongested nature of the network.

Moreover the trained agents perform better than the random agents in term of average fitness for a majority of the conducted experiments, demonstrating that the learning process is necessary and produces efficient strategies.

Similar behavioural patterns can be observed for the ratio of used streets with respect to the proportion of strategics agents depicted in Fig. 8. As previously observed, a proportion of 50% optimised agents seems to be a good trade-off. Again agents with random strategies performed worse than agents with optimised strategies in the experiments involving

**Table 3**
Testing scenario characteristics.

| Parameter | 3 cities | Chicago |
|---|---|---|
| Nodes | 99 | 933 |
| Links (50 km/h) | 264 | 354 |
| Links (90 km/h) | 54 | 2532 |
| Links (120 km/h) | 6 | 64 |
| Total length (km) | 504 | 13,190 |
| Mean capacity per km (agents/hour) | 11.5 | 5.7 |
| Number of agents | 750 | 4000 |
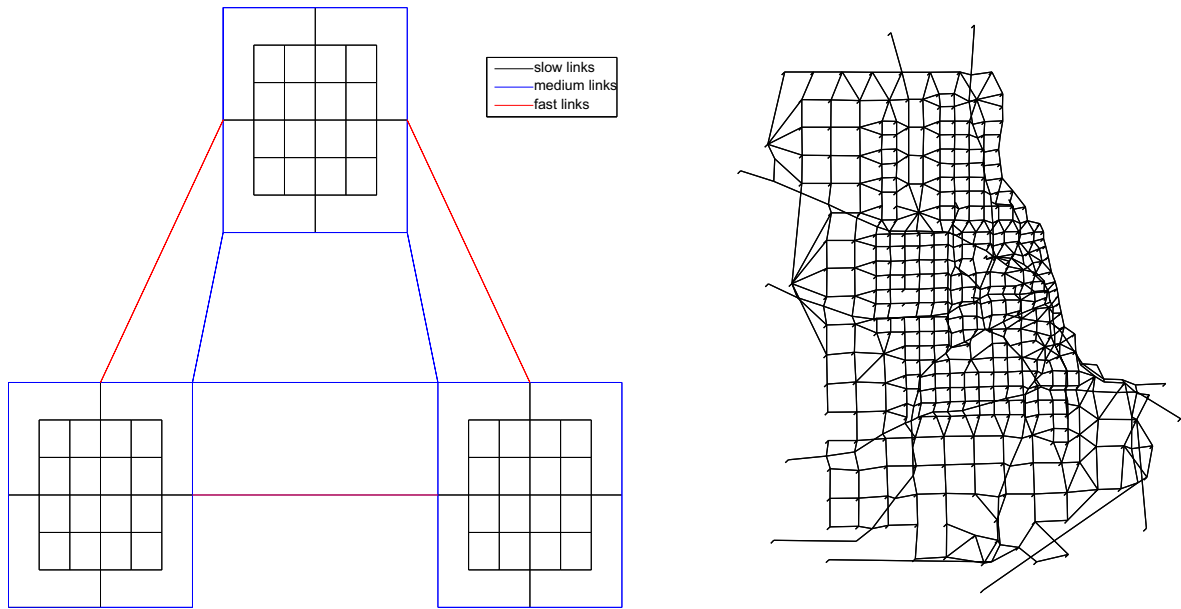


**Fig. 6.** Left panel: 3 cities network. Right panel: Chicago network.
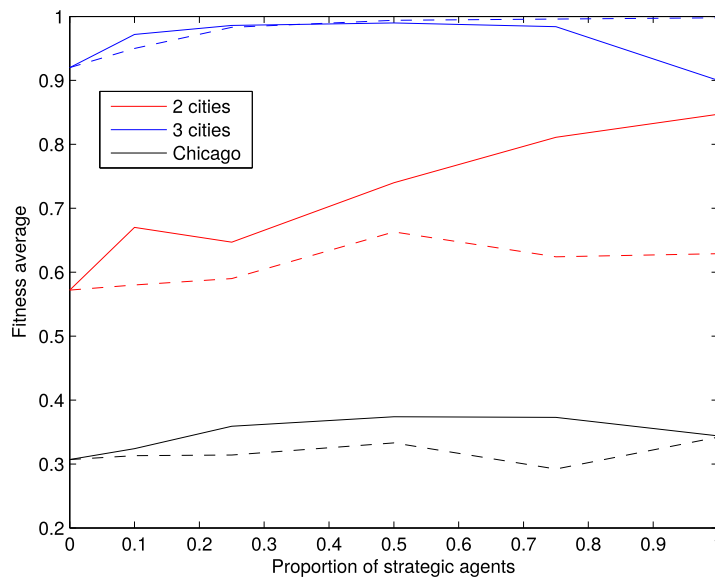


**Fig. 7.** Evolution of the agent average fitness with respect to the proportion of strategic agents in various scenarios. The average fitness is computed at the end of each simulation. The solid lines represent the agents provided with a strategy optimised by a genetic algorithm while the dashed lines correspond to agents with random strategies.
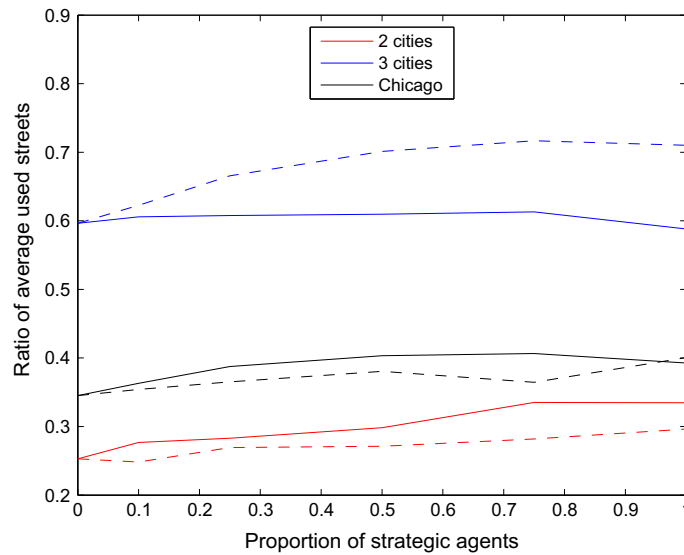
**Fig. 8.** Evolution of the average ratio of used streets over the total number of streets with respect to the proportion of strategic agents. The average ratio is computed across the whole simulation. The solid lines represent the agents provided with a strategy optimised by a genetic algorithm while the dashed lines correspond to agents with random strategies.

congested networks. On the other hand, the random agents seems to perform more re-routing and using more road than necessary in the uncongested scenario.

These observations show that the provided strategy, optimised with a preliminary learning process, is effective compared to random behaviours and even having no strategy. Moreover these findings hold even if the learning phase has been performed on a network different from the one used in the simulation phase.

### 3.2. Performance profiles

In order to have a deeper insight of the agents behaviours over the simulation, we propose a performance profile based on 4 indicators:

- the ratio of used links with respect to the total number of streets;
- the ratio of jammed links[8];
- the probability density function, estimated by a Normal kernel function (Bowman and Azzalini, 1997), of the average fitness across the simulated agents at the simulation final time step;
- the average fitness evolution, which is computed across the simulated agent at each time step.

Let us observe that we stop each simulation whenever half of the agents have performed 5 round-trips between their respective origin-destination, in this way we reduce the impact of extreme cases where a single agent could have covered his path very fast by chance. On the other hand this halting procedure avoids too long simulations where extreme events of very long travel times can arise. Such performance profiles are shown in Figs. 9–11 for some scenarios simulated with a proportion of 25% of optimised or random agents.

From these profiles, and more specifically from the fitness value distribution and evolution over time, it appears clearly that trained agents perform better that the non-strategic.

Regarding the agents behaviour with respect to the level of traffic density, the observations made previously are also confirmed:

- in presence of high congestion, trained strategic agents have at better fitness than the one associated with the other agents;
- similarly in uncongested conditions, strategic agents are able take advantage of the network and their fitness is significantly superior than the fitness of the non-strategic agents;
- generally the random agents have a fitness similar to the one of the non-strategic agents except in uncongested network where they have slightly better fitness.

---

[8] A link is said to be jammed if its speed computed by the BPR equation is reduced at least by half with respect to its free flow value.
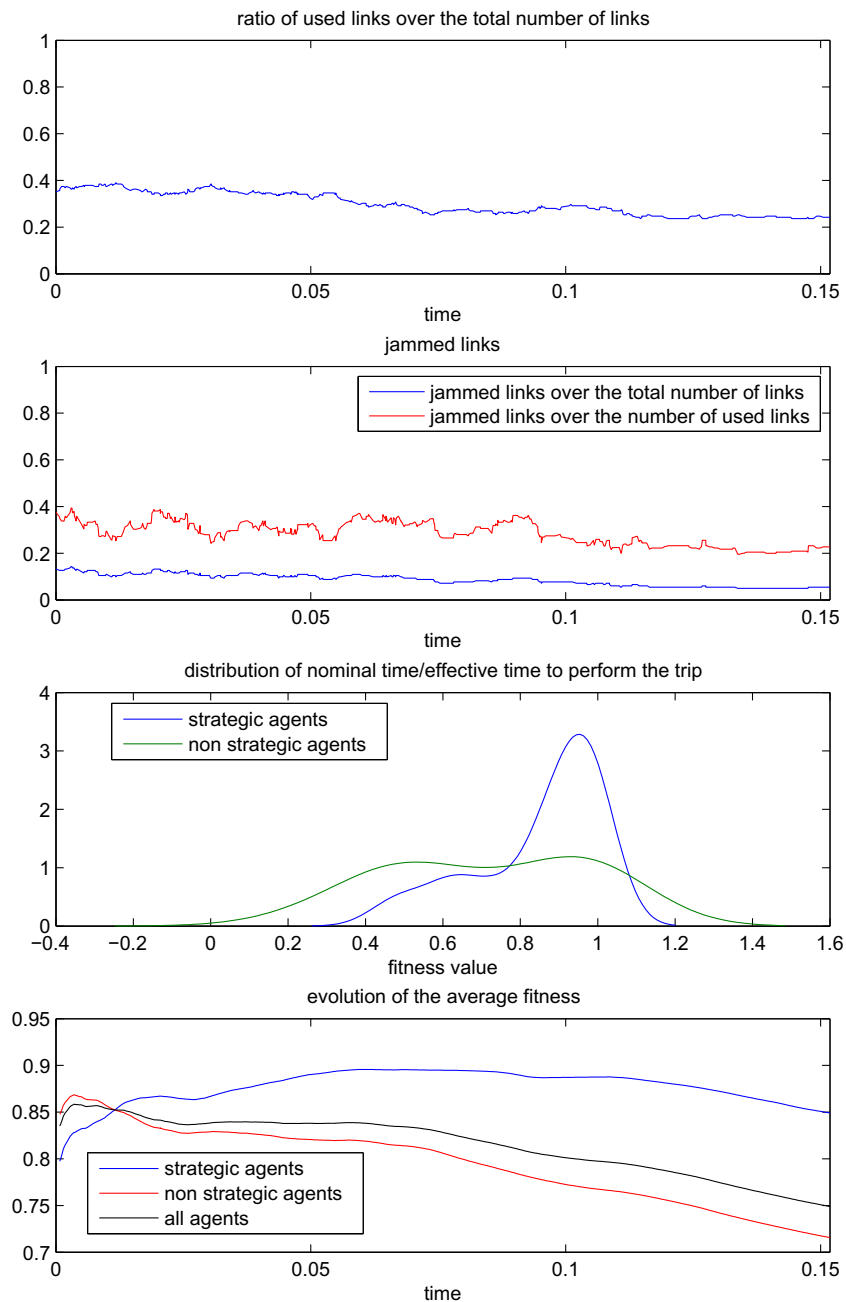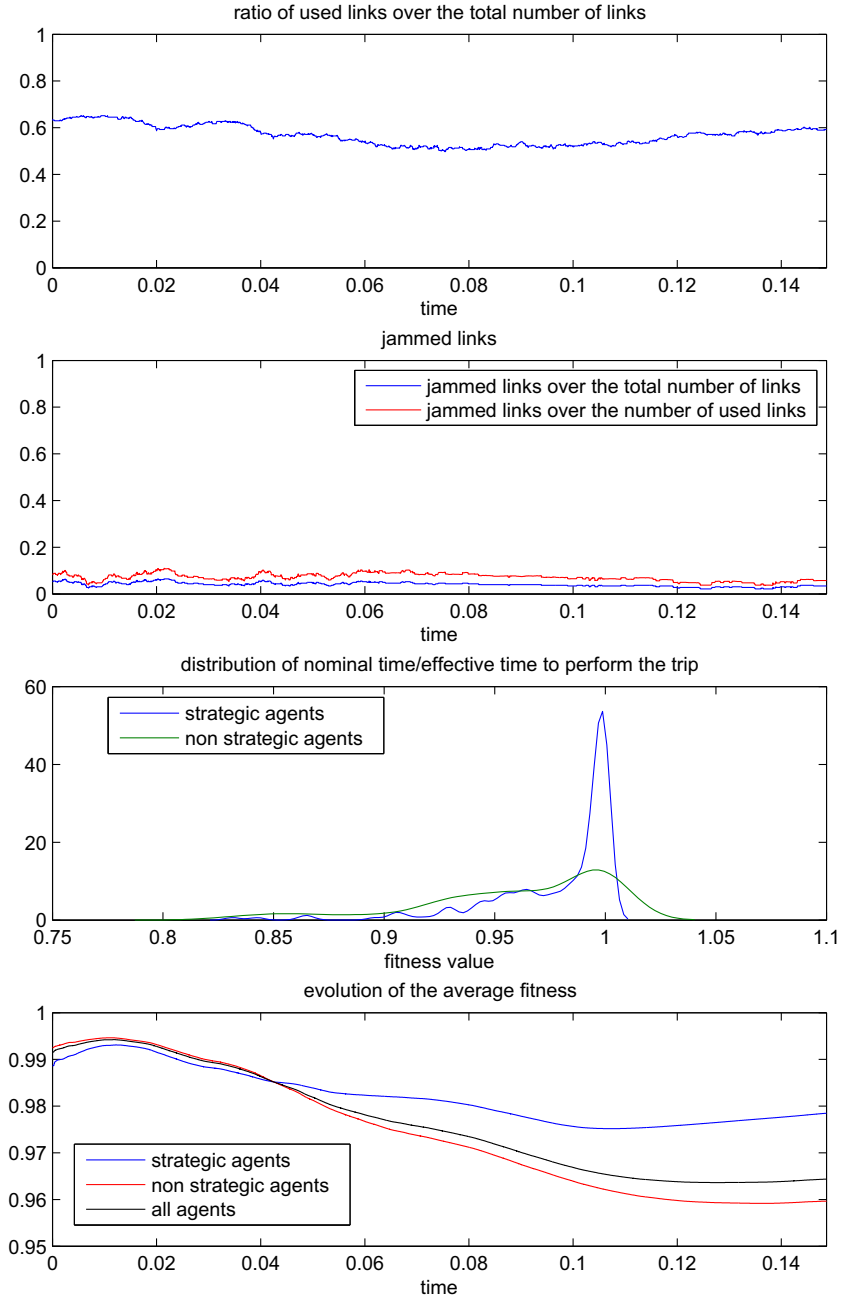
**Fig. 9.** Performance profile for scenario 2 cities - strategic agents.

Note that the ratios of used links and jammed links seems to rapidly converge to an equilibrium in every simulation. The initially observed decrease corresponds to the time needed for every agent to enter the simulation, namely a sort of transient time in the simulation. As the number of agents on the network remains constant, this observation comes as no surprise.

### 3.3. Agents' robustness to network modifications

We now examine the robustness of the agents performances against network modification. This test has been performed by sequentially taking out links from the *3 cities* network. First, we start to remove the fast links between the three cities,

**Fig. 10.** Performance profile for scenario 3 cities - strategic agents.

taking care to still have a connected network, then we removed randomly the links but still to keep the network connected.[9] An example of the final network we got is given in Fig. 12.

The robustness tests have been perfumed using a fixed proportion of 25% of strategic agents in the total population, the results are illustrated in Fig. 13. One can observe that the proportion of removed links does not affect too much the proportion of used streets. On the other hand, as the network is being degraded, the median value of the fitness for the strategic agents remains higher than the one associated with the non-strategic agents, indicating that the strategic agents can adapt their routes to cope with the modifications.

___

[9] This is important, in order to keep always at least one feasible path between any origin-destination pair.
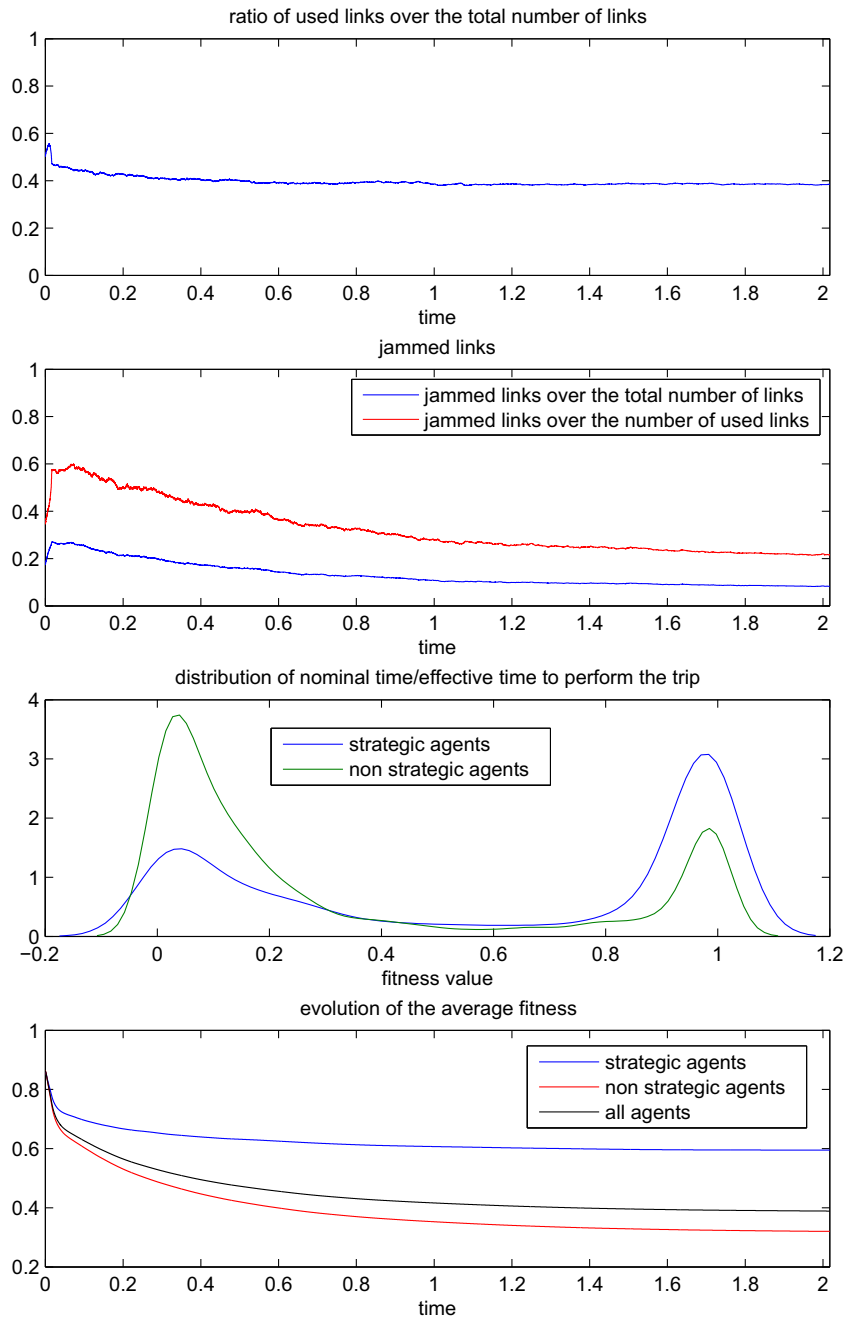
**Fig. 11.** Performance profile for scenario Chicago - strategic agents.

## 3.4. Comparison with MATSim

MATSim is an agent-based traffic micro-simulator which has been used in several applications all around the world (see http://www.matsim.org/scenarios for a listing of different scenarios). The simulator initially assigns to every agent a *plan* for each of his trip, i.e. a pre-defined path, a time to leave and a desired arrival time. Then MATSim tries to reach an user equilibrium by repeating $K$ times the following iterative process:

1. each agent carries out his plans (simulation step);
2. after the simulation step, MATSim assesses the performance of every agent;
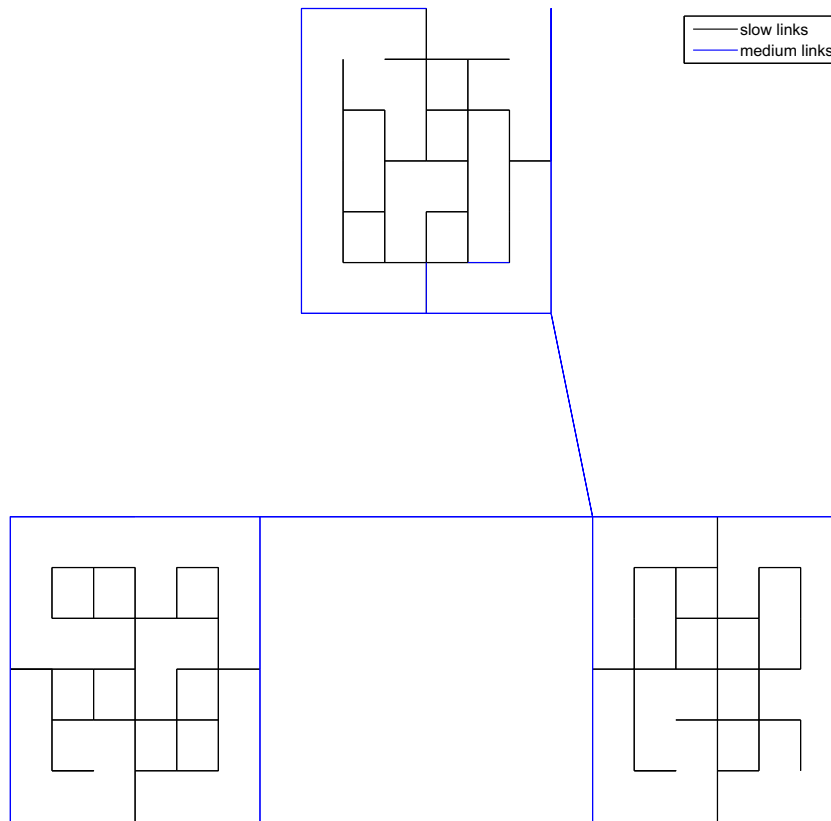3. MATSim then modify the plans of the agents with the worst performance.

**Fig. 12.** Road network of *3 cities* after the removal of 20% of its links.
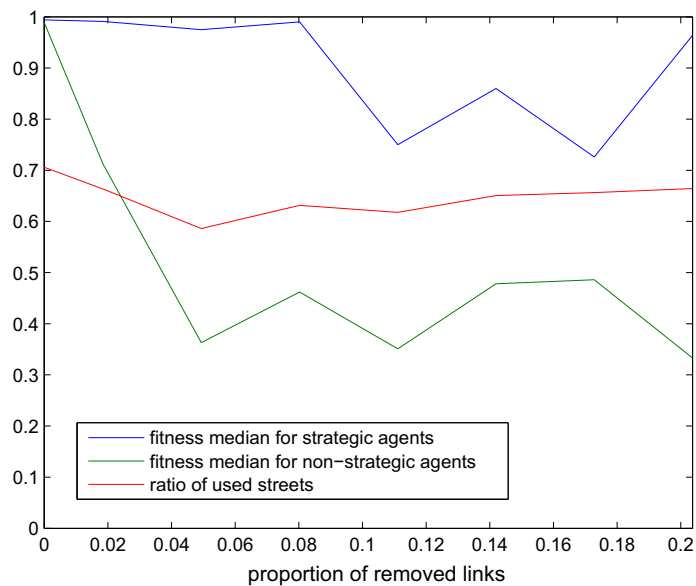


**Fig. 13.** Evolution of fitness median for strategic and non strategic agents and the ratio of used streets as a function of the proportion of removed links from the original *3 cities* network. A proportion of 25% strategic agents was used in the conducted experiments.

In this Section we compares the results of our strategic agents-based simulator (using 25% of strategic agents) against the ones generated by MATSim for Namur, a city in the Walloon Region of Belgium (after 20 iterations, only allowing the modification of the pre-defined paths for 10%). The simulation involved 100,000 agents performing a total of 436,612 trips over a

typical weekday in a road network made of 23,000 nodes and 36,700 links. The travel demand generation is fully detailed in Barthélemy (2014) and Barthelemy and Toint (2015).

During the simulation, the strategic agents chose to change their initial route 17,600 times. The predicted traffic at 8:30 am by both the proposed approach and MATSim are illustrated in Figs. 14 and 15 wherein it can be observed similarities in terms of link congestion between the two methods. A comparison of the simulated daily counts on each link by the two approach is shown in Fig. 16. One can easily observe the strong positive correlation (0.951) between the counts, indicating that our methodology and MATSim produce similar traffic patterns. A simple linear regression predicting the MATSim counts from the ones generated by the proposed method further confirmed this observation with an associated adjusted $R^2$ of 0.905.

## 3.5. Comparison with an user-equilibrium approach

In order to assess the validity of the proposed approach, we can compare our results with the ones generated by the Origin-based assignment (OBA) algorithm developed by (Bar-Gera, 2002). This algorithm determines the classical deterministic user equilibrium as defined by Wardrop (1952). The retained network for these experiments is the well-known *Sioux-Falls* network[10] represented in Fig. 17. Although this network is not considered to be a realistic one, it has been already used in many publications for testing and benchmarking new methodologies.

Fig. 18 shows the impact of the proportion of strategic agents on various indicators. It can be observed that:

- as the proportion of strategic agents increases, the average fitness, the ratio of used streets and the agents average speed slightly improve;
- the number of time steps required for every agents to reach his destination also decreases as the proportion of strategic agent increase up to 75%.

The comparison is performed by computing the average and maximum absolute deviations between the distribution of the traffic flows amongst the links produced by our approach and the ones determined by the OBA algorithm. These deviations, respectively denoted by $D_a$ and $D_m$, are formally defined by

$$D_a = \sum_{l \in L} \frac{|v_l - v_l^*|}{l_{tot}}, \tag{4}$$

and

$$D_m = \max_{l \in L} |v_l - v_l^*|, \tag{5}$$

where $L$ is the set containing $l_{tot}$ links, and $v_l$ and $v_l^*$ correspond respectively to the normalised traffic flow going through a link $l \in L$ generated by the proposed approach and the OBA algorithm. In this context, normalised means divided by the sum of the flows over every link $k \in L$.

Fig. 19 illustrates the evolution of theses deviations as a function of the proportion of strategic agents. For a proportion of 75%, $D_a$ is less than 0.5% while $D_m$ is about 1.6%, showing that the solution of the proposed approach is close to the theoretical one. Hence the solution proposed by the genetic algorithm is very close to the optimal one.

MATSim can also be used to perform the dynamic traffic allocation for this scenario[11] (Chakirov and Fourie, 2014). The daily traffic counts are then compared with the ones generated by the proposed approach where 75% of the agents have a strategy. As previously, it can be observed that the counts are strongly correlated (Pearson correlation coefficient of 0.852). This further indicates that our methodology and MATSim produces similar traffic assignment.(see Fig. 20)

Another interesting indicator of the agents' behaviour is given by the number of re-routing performed during the course of the simulation. This is illustrated in Fig. 21 where it can be seen, as expected, that the number and triggering of re-routing decisions (for a total of 88,080) is strongly correlated with the number of agents on the network.

## 3.6. Transmob: simulating the transport demand of Randwick

The strategic agents have been used in Transmob, an agent-based model which simulates the dynamics between demographic evolution, transport demands, housing needs and the eventual change in the average satisfaction of the residents of Randwick, a suburb of Sydney (Australia). The different modules of Transmob, including the traffic demand generation and assignment as well as their validations are detailed in (Huynh et al., 2015). The simulation involves 100,000 agents (of which 25% are strategic) performing 360,000 trips during a typical weekday in a network made of 7,000 nodes and 17,000 links. During the course of the simulation, the strategic agents have performed 34,750 re-routing.

---

[10] Available at http://www.bgu.ac.il/~bargera/tntp/.
[11] The data is available at http://www.matsim.org/scenario/sioux-falls-sd-usa.

**Fig. 14.** Saturation of the Namur road network at 8:30 am predicted by the proposed approach. Red links present a saturation of at least 75%. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)
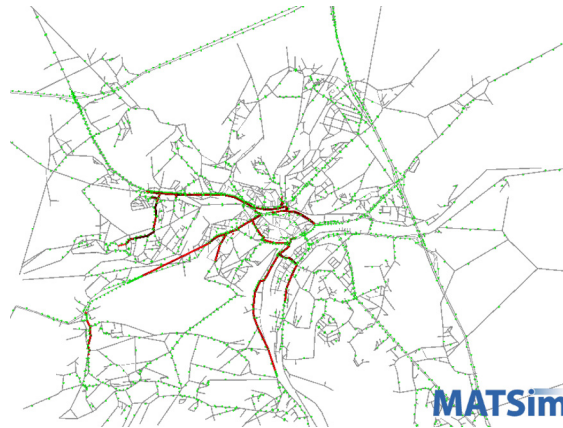


**Fig. 15.** Traffic density road network at 8:30 am predicted by MATSim. Red agents are stuck in a traffic jam. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The simulated traffic density on road links around the University of New South Wales at 8:00am are compared against the corresponding congestion profiles from Google Traffic, as shown in Fig. 22. The proposed approach is able to reproduce relatively accurately the observed congestion patterns from Google Traffic. Fig. 23 also shows the simulated congestion level of every road link on the road network in the study area at 8.00 am, which highlights Anzac Parade (the main road running across the area) as the most congested, agreeing with the observed traffic. However such agreements do not occur on all parts of the network. This can be due to missing data from Google Traffic and/or the assumptions of Transmob.

### 3.7. Computation time

In this Section we examine the computation time required by the current implementation, called BATSim,[12] of the proposed approach and its ability to take advantage of the multiple computing cores present in modern computers. The implementation is written in C++14 and relies on Repast HPC 2.2 (Collier and North, 2012).

---

[12] Available at https://github.com/smart-facility/BATSim.
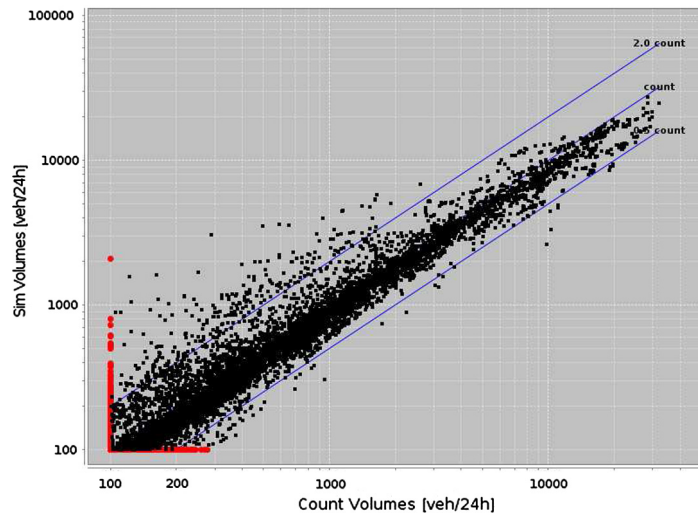
**Fig. 16.** Strategic agents traffic counts (*x* axis) against MATSim traffic counts (*y* axis). Each dot represents a link and the daily traffic volume predicted by the proposed model and MATSim. Pearson correlation coefficient $\rho = 0.951$, adjusted $R^2 = 0.905$.
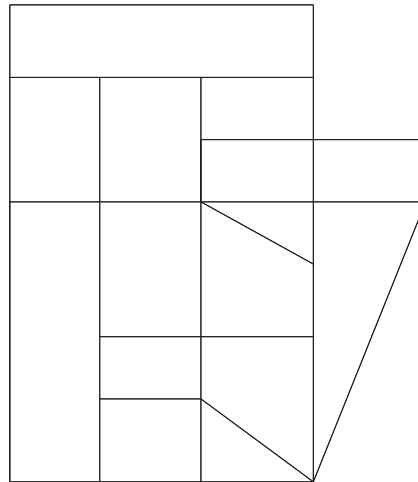


**Fig. 17.** *Sioux-Falls* road network: 282 nodes, 334 links (60 km/h) and 360,600 trips performed by 65,408 individuals.

Fig. 24 illustrates the time required by both our approach and MATSim to simulate the *Sioux-Falls* scenario (detailed in Section 3.5) with respect to the number *c* of computational cores used. The results have been averaged on 10 runs and were obtained on a workstation equipped with a 4-core Intel(R) Core(TM) i5-4570 CPU @ 3.20 GHz and 16 Gb of RAM.

On this small-scale scenario it can be seen that as *n* increases the time required by the strategic agents decreases significantly. This trend indicates that the current implementation is able to take advantage of multiple processing cores. This is not the case for MATSim as the time slightly increases with the number of cores. This last observation is due to the small size of the scenario: the communication overhead between MATSim's processes overcomes the gain of using parallel computing. Nevertheless when only one core is used to run the simulation MATSim is faster, but this is no longer the case for $c \geqslant 2$.

Next we consider the larger problem of simulating the traffic in Namur detailed in Section 3.4 to further investigate the speed up that can be obtained using multiple computing cores. The simulations have been run on a computer equipped with an Intel(R) Core(TM) i7-7700HQ CPU @ 2.80 GHz and 16 Gb of RAM, with $c = 1, \ldots, 8$. As previously, the time required by each simulation has been averaged on 10 runs. Fig. 25 illustrates the speed up obtained by increasing *c*. It can be seen that a speed up of nearly 3 (meaning that the simulation is nearly 3 times faster) when using 8 cores. For that scenario, increasing the number of cores above that threshold would not improve the simulation time as the speed up seems to reach a plateau.

It should be noted that the best computing times (using 8 cores) for running MATSim and the proposed approach in this case are respectively 21.4 min and 25.7 min. MATSim is slightly faster, but the small difference tends to indicate that the proposed approach is close in terms of computational costs.
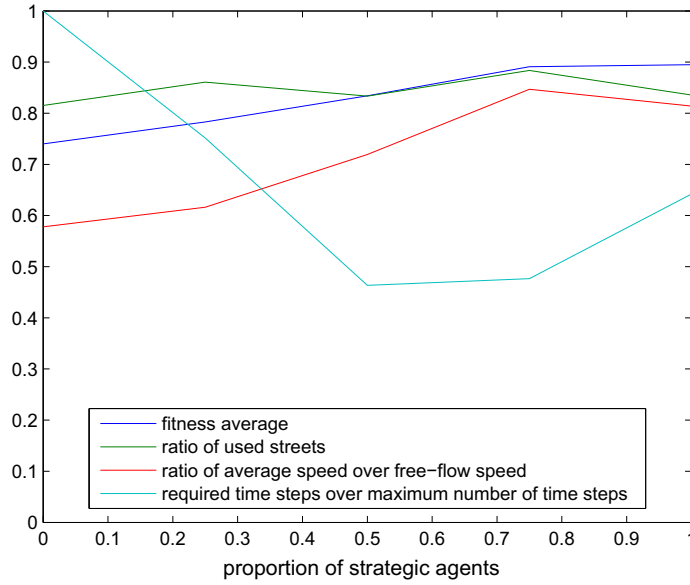
**Fig. 18.** Evolution of some performance indicators with respect to the proportion of strategic agents in the *Sioux-Falls* network.
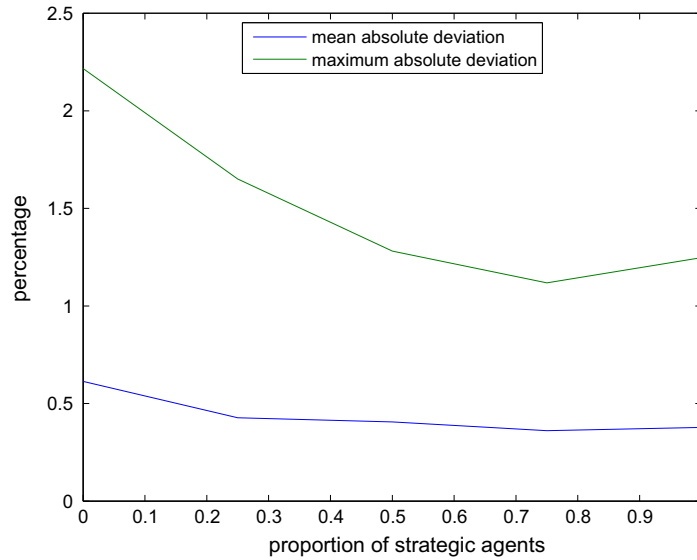


**Fig. 19.** Average and maximum absolute deviations between theoretical flows distribution computed by the Origin-based assignment algorithm and the proposed approach as a function of the proportion of strategic agents.

We also compare the number of shortest path computations performed by MATSim and the new methodology, denoted by $SP_{mat}$ and $SP_{sa}$ respectively and given by:

$$SP_{mat} = T + (pT)K \quad \text{and} \quad SP_{sa} = T + r, \tag{6}$$

where $T$ is the initial number of trips, $p$ is the proportion of agents modifying their route at each iteration $k = 1, \ldots, K$ performed by MATSim and $r$ corresponds to the number of re-routing done by the strategic agents during the simulation. Those indicators can be seen as a rough approximation of the complexity of the methods. The values of $SP_{mat}$ and $SP_{sa}$ for *Sioux-Falls* and *Namur* are presented in Table 4. In both cases the strategic agents-based approach perform less shortest path computation.

Compared to MATSim, the new approach has then similar computation time and performs less shortest path computation. It has also been shown that the current code is also able to uses efficiently multiple multi-cores computers. Hence
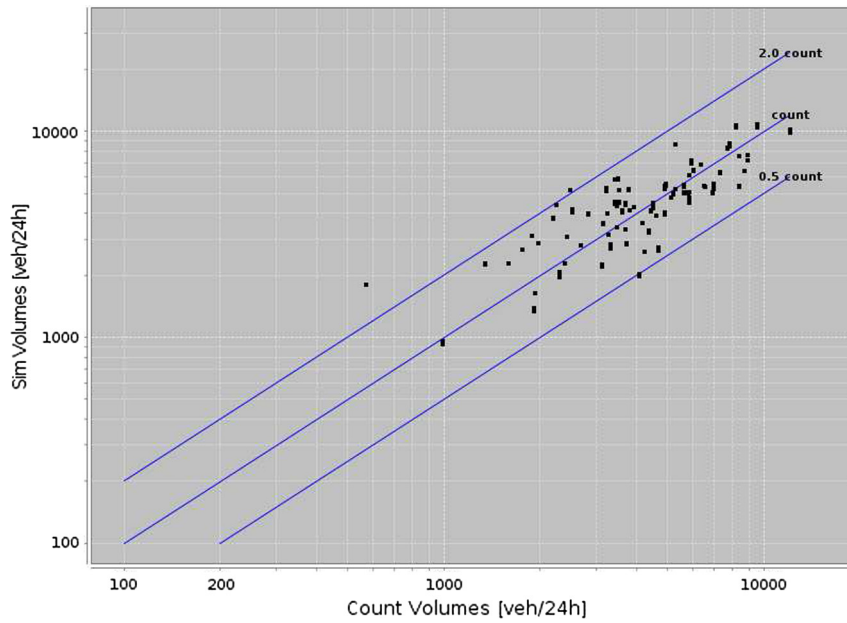
**Fig. 20.** Strategic agents traffic counts (*x* axis) against MATSim traffic counts (*y* axis). Each dot represents a link and the daily traffic volume predicted by the proposed model and MATSim. Pearson correlation coefficient $\rho = 0.852$, adjusted $R^2 = 0.724$.
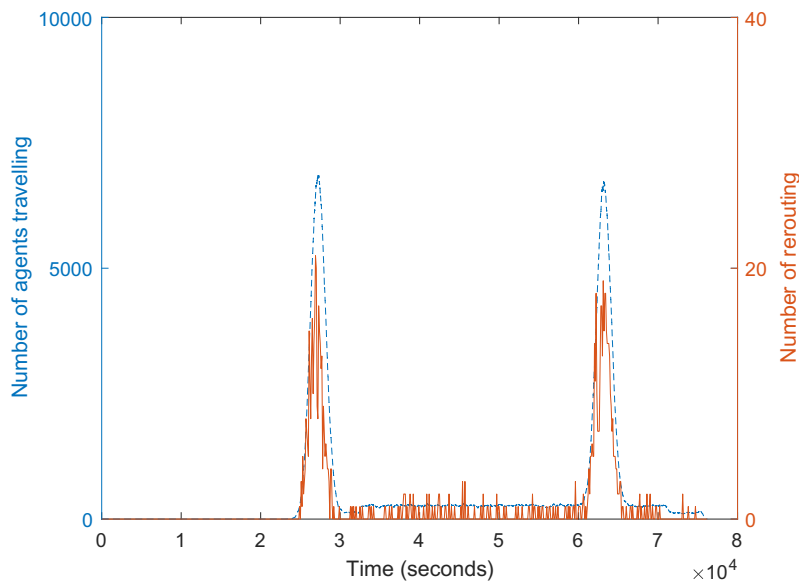


**Fig. 21.** Number of agents travelling and re-routing during the simulation. The *x* axis represents the number of simulated seconds since midnight. The total number of re-routing is 88,080.

the results presented in this Section are encouraging. Indeed this work is exploratory and as such there is a lot of room to improve the current implementation.

## 4. Conclusions and discussions

In this paper, we presented the premises of an alternative to existing simulation-based dynamic traffic assignments models. Our starting point is to provide travelling agents with a strategy in such a way they evaluate in real time the possibility to re-route their path at any intersection they encounter to consider instantaneous changes in the road network condition.
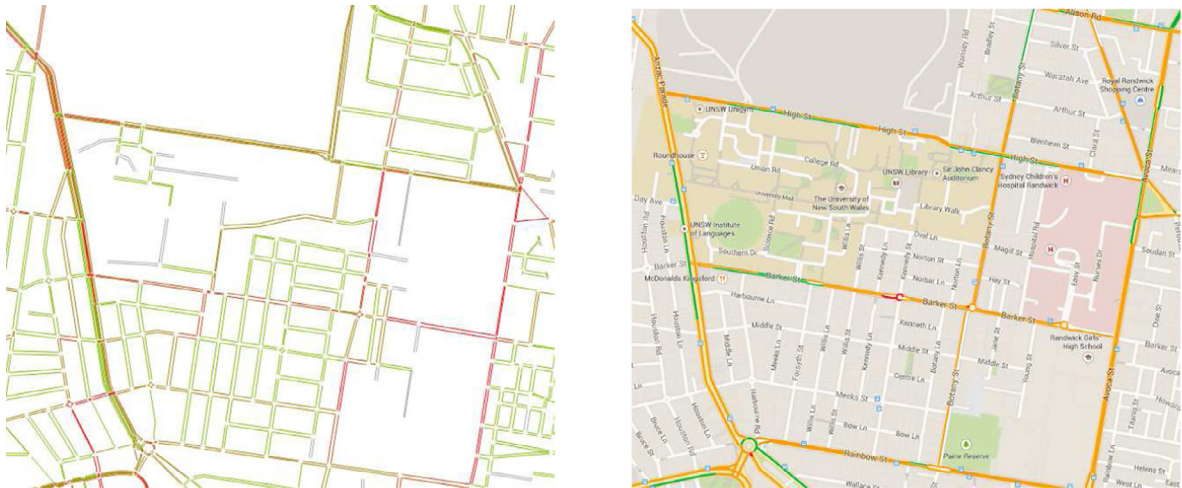
**Fig. 22.** Traffic density at 8:00 am. Left panel: Strategic agents. Right panel: Google Traffic (accessed on 12 March 2015).
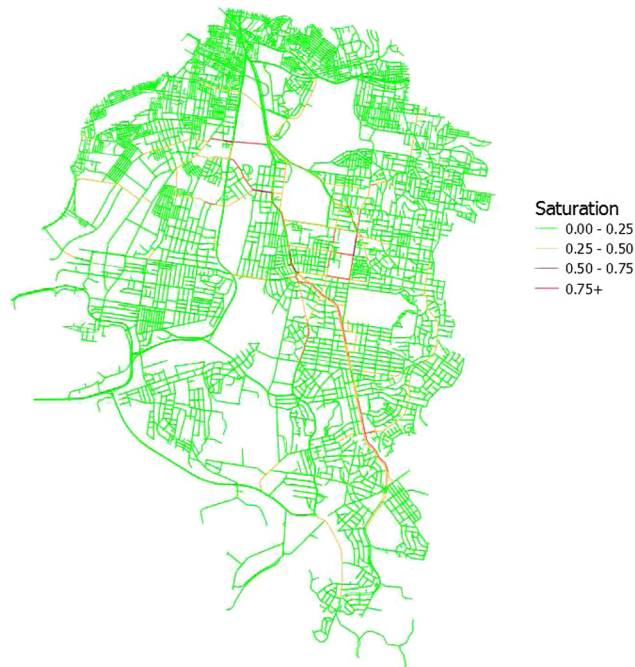


**Fig. 23.** Saturation of the Randwick road network at 8:00 am predicted by the proposed approach.

The proposed strategy for dynamic re-routing is coded using a neural network where its inputs rely on the current trip duration and the perceived traffic conditions on the successive link on the agent's path. Instead of fine tuning the neural network parameters, we decided to use a training phase on a simple and small congested network; such training phase has been performed using a standard genetic algorithm allowing us to derive a set of optimal neural network parameters. The trained agents can then use their own strategy to face changing conditions. Moreover, as the agents use only local information, the overall network topology does not really matter, thus the strategy is be able to cope with (larger) networks different from the one used for training.

In the conducted experiments, despite the simplicity of their neural networks, the strategic agents have shown an efficient behaviour in terms of road network use. The robustness and adaptability of the trained strategies to new environments, i.e. different networks and congestion levels, have also been demonstrated. Comparisons with MATSim, OBA and Google Traffic tend also to indicate that the generated traffic patterns are not unrealistic. This is an encouraging outcome for a work-in-progress traffic micro-simulator.
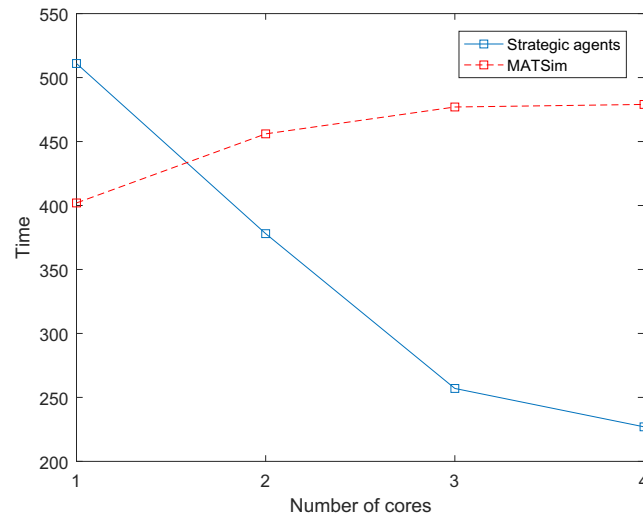
**Fig. 24.** Computation time for the *Sioux-Falls* scenario against the number of cores used by MATSim and the proposed approach.
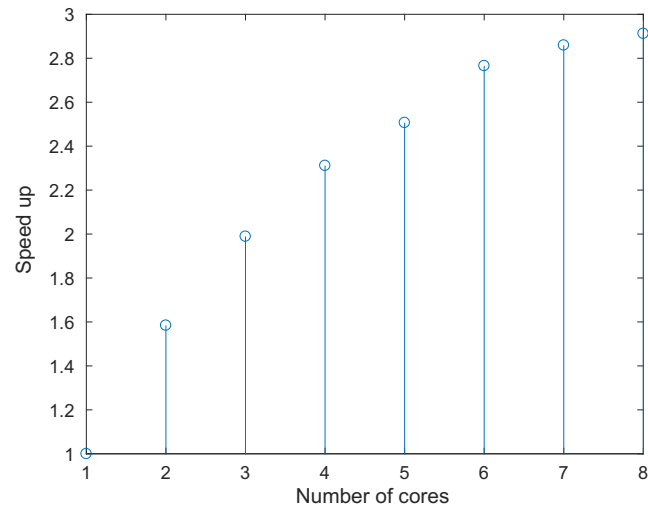


**Fig. 25.** Speed up obtained when increasing the number of computing cores for simulating the *Namur* scenario.

**Table 4**
Number of shortest path computations per method.

| Scenario | Initial trips | Strategic agents | MATSim |
|---|---|---|---|
| Sioux-Falls | 130,816 | 218,896 | 379,366 |
| Namur | 436,612 | 453,612 | 1,266,175 |

A key feature of this approach is that it does not require several computationally expensive iterations to take account of network modifications or changing traffic conditions. Indeed, the agent strategies need to be trained only once during a preliminary learning stage. As a result, this model seems well-suited to large-scale applications. With its current implementation the framework is already able to take advantage of multi-core processors resulting in computational costs similar to existing simulation-based approaches in the conducted experiments.

This work is a first step of a larger framework that will be deeply analysed with successive studies. Many improvements, such as the ones mentioned below, must be considered but are beyond the scope of this work.

Firstly, a current limitation of the approach lies in the strong (and unrealistic) assumption that the agent has a perfect knowledge of the road network when it performs a re-routing.

The strategy can be also improved, for instance by adding memory of the previous choices or accounting for additional, global, information about the network state. The former must be completed, as already stated, with a behavioural model of the agent in order to understand of the received information is processed. Instead of considering only a preliminary off-line training, a reinforcement learning during the simulation could also be considered.

Additionally, the queue model currently used to simulate the traffic dynamic can be easily extended with the Newell's model to capture congestion shock waves. Further validation work and comparison with other micro-simulation frameworks should also be conducted.

The strategic agents could also be implemented within a traffic micro-simulation based on the Beliefs, Desires, Intentions (BDI) framework (Rossetti et al., 2002; Kim et al., 2014). The agents will use their strategy for the re-routing choices based on their beliefs and goals. This framework has been shown to produce realistic traffic patterns and the strategic agents could lower its computational cost.

The approach can be considered for simulating the route choice made by the users of means of transportation that do not typically rely on information provided by navigation systems and radios. This can include people riding a bicycle or bicycle and even pedestrian.

The computational efficiency and scalability of the micro-simulator using both high performance computing facilities (thanks to Repast HPC) and GPU computing (following the work initiated by Strippgen and Nagel (2009) and Sano et al. (2016)) will aso be investigated further. Using those parallel computing techniques will also allow the approach to simulate large scenarios, such as assigning the traffic demand of Belgium detailed in Barthelemy and Toint (2015).

Finally, the proposed behavioural framework is a step towards better tools and models to investigate and understand drivers' behaviours when they are provided with traffic information. This knowledge can improve the design of existing and new traffic information delivery systems to optimise the use of the road network. For instance a connected navigation system that provides rerouting instructions to only a fraction of the drivers (for which the system knows that they are likely to follow the instructions) as we have seen in the numerical experiments that having too many strategic agents is not optimal.

## Acknowledgements

## References

Asimov, I., 1988. Prelude to Foundation. The Foundation Series. Random House Publishing Group.
Auld, J., Hope, M., Ley, H., Sokolov, V., Xu, B., Zhang, K., 2016. Polaris: Agent-based modeling framework development and implementation for integrated travel demand and network and operations simulations. Transp. Res. Part C: Emerg. Technol. 64, 101–116.
Bar-Gera, H., 2002. Origin-based algorithm for the traffic assignment problem. Transp. Sci. 36 (4), 398–417.
Barceló, J., Casas, J., 2005. Dynamic network simulation with aimsun. In: Simulation Approaches in Transportation Analysis. Springer, pp. 57–98.
Barthélemy, J., Carletti, T., 2017. An adaptive agent-based approach to traffic simulation. Transp. Res. Procedia 25, 1238–1248.
Barthelemy, J., Toint, P., 2015. A stochastic and flexible activity based model for large population. Application to Belgium. J. Artif. Soc. Soc. Simul. 18 (3), 15 <http://jasss.soc.surrey.ac.uk/18/3/15.html> .
Barthélemy, J., 2014. A parallelized micro-simulation platform for population and mobility behaviour-application to belgium (Ph.D. thesis). University of Namur.
Bazghandi, A., 2012. Techniques, advantages and problems of agent based modeling for traffic simulation. Int. J. Comput. Sci. 9 (3).
Ben-Akiva, M., Bierlaire, M., Koutsopoulos, H., Mishalani, R., 1998. Dynamit: a simulation-based system for traffic prediction. In: DACCORS Short Term Forecasting Workshop, The Netherlands. Citeseer.
Ben-Akiva, M.E., Gao, S., Wei, Z., Weng, Y., 2012. A dynamic traffic assignment model for highly congested urban networks. Transp. Res. Part C: Emerg. Technol. 24, 62–82.
Benekohal, R.F., Treiterer, J., 1988. Carsim: Car-following model for simulation of traffic in normal and stop-and-go conditions. Transp. Res. Rec., 1194
Bonsall, P., 1992. The influence of route guidance advice on route choice in urban networks. Transportation 19 (1), 1–23.
Bonsall, P., May, A.D., 1986. Route choice in congested urban networks. In: Research for Tomorrow's Transport Requirements. Proceedings of the Fourth World Conference on Transport Research.
Bowman, A.W., Azzalini, A., 1997. Applied Smoothing Techniques for Data Analysis: The Kernel Approach with S-Plus Illustrations: The Kernel Approach with S-Plus Illustrations. Oxford University Press.
Chakirov, A., Fourie, P., 2014. Enriched sioux falls scenario with dynamic and disaggregate demand. In: Technical Report. Future Cities Laboratory, Singapore-ETH Centre (SEC).
Charypar, D., Axhausen, K., Nagel, K., 2015. Event-driven queue-based traffic flow microsimulation. Transp. Res. Rec.: J. Transp. Res. Board.
Chiu, Y., Bottom, J., Mahut, M., Paz, A., Balakrishna, R., Waller, T., Hicks, J., 2011a. Dynamic traffic assignment: a primer. Transp. Res. E-Circ. E-C153.
Chiu, Y., Nava, E., Zheng, H., Bustillos, B., 2011b. Dynust user?s manual.
Collier, N.T., North, M., 2012. Parallel agent-based simulation with repast for high performance computing. Simulation. http://dx.doi.org/10.1177/0037549712462620.
Cornélis, E., Toint, P.L., 1998. Pacsim: a new dynamic behavioural model for multimodal traffic assignment. In: Labbé, M., Laporte, G., Tanczos, K., Toint, P.L. (Eds.), Operations Research and Decision Aid Methodologies in Traffic and Transportation Management.. Springer Verlag, Heidelberg, Berlin, New York, pp. 28–45.
Daganzo, C.F., Sheffi, Y., 1977. On stochastic models of traffic assignment. Transp. Sci. 11 (3), 253–274.
Dehoux, P., Toint, P., 1991. Some comments on dynamic modelling in the presence of advanced driver information systems. In: Argyrakos, G., Carrara, M., Cartsen, O., Davies, P., Mohlenbrink, W., Papageorgiou, M., Rothengatter, T., Toint, P.L. (Eds.), Advanced Telematics in Road Transport, Commission of the European Communities - DG XIII, pp. 964–981.

Downs, R.M., Stea, D., 1977. Maps in minds. Harper and Row, New York.

Eiben, A., Smith, J.E., 2003. Introduction to Evolutionary Computing. Springer.

Friesz, T.L., Bernstein, D., Smith, T.E., Tobin, R.L., Wie, B.W., 1993. A variational inequality formulation of the dynamic network user equilibrium problem. Oper. Res. 41 (1), 179–191.

Gao, S., 2012. Modeling strategic route choice and real-time information impacts in stochastic and time-dependent networks. IEEE Trans. Intell. Transp. Syst. 13 (3), 1298–1311.

Gao, S., Frejinger, E., Ben-Akiva, M., 2010. Adaptive route choices in risky traffic networks: a prospect theory approach. Transp. Res. Part C: Emerg. Technol. 18 (5), 727–740.

Helbing, D., 2016. <http://futurict.inn.ac/overview/>.

Huynh, N., Perez, P., Berryman, M., Barthélemy, J., 2015. Simulating transport and land use interdependencies for strategic urban planning–an agent based modelling approach. Systems 3 (4).

Kim, S., Son, Y.-J., Tian, Y., Chiu, Y.-C., 2014. Drivers' en-route divergence behavior modeling using extended belief-desire-intention (e-bdi) framework. In: Proceedings of the Winter Simulation Conference 2014. IEEE, pp. 344–355.

Kriesel, D., 2007. A Brief Introduction to Neural Networks. on-line, available at <http://www.dkriesel.com>.

Lam, W.H.K., Huang, H.J., 1992. A combined trip distribution and assignment model for multiple user classes. Transp. Res. B 26 (4), 275–287.

Ma, J., Smith, B.L., Zhou, X., 2016. Personalized real-time traffic information provision: agent-based optimization model and solution framework. Transp. Res. Part C: Emerg. Technol. 64, 164–182.

Mahmassani, H., Hu, T., Jayakrishnan, R., 1992. Dynamic traffic assignment and simulation for advanced network informatics (dynasmart). In: Proceedings of the 2nd international CAPRI seminar on Urban Traffic Networks, Capri, Italy.

Mahmassani, H.S., Chang, G.-L., 1987. On boundedly rational user equilibrium in transportation systems. Transp. Sci. 21 (2), 89–99.

Meister, K., Balmer, M., Ciari, F., Horni, A., Rieser, M., Waraich, R.A., Axhausen, K.W., 2010. Large-scale agent-based travel demand optimization applied to switzerland, including mode choice. In: Paper Presented at the 12th World Conference on Transportation Research.

Merchant, D.K., Nemhauser, G.L., 1978a. A model and an algorithm for the dynamic traffic assignment problems. Transp. Sci. 12, 183–199.

Merchant, D.K., Nemhauser, G.L., 1978b. Optimality conditions for a dynamic traffic assignment model. Transp. Sci. 12 (3), 200–207.

Nagel, K., Flötteröd, G., 2009. Agent-based traffic assignment: going from trips to behavioral travelers. In: 12th International Conference on Travel Behaviour Research (IATBR). Jaipur.

Newell, G.F., 1993. A simplified theory of kinematic waves in highway traffic, part i: General theory. Transp. Res. Part B: Methodol. 27 (4), 281–287.

Nguyen, S., Pallotino, S., 1988. Equilibrium traffic assignment for large scale transit network. Oper. Res. Int. Journal 37 (2), 176–186.

Nguyen, S., Pallotino, S., Gendreau, M., 1998. Implicit enumeration of hyperpaths in a logit model for transit networks. Transp. Sci. 32 (1), 54–64.

Pan, J., Khan, M.A., Popa, I.S., Zeitouni, K., Borcea, C., 2012. Proactive vehicle re-routing strategies for congestion avoidance. In: 2012 IEEE 8th International Conference on Distributed Computing in Sensor Systems (DCOSS). IEEE, pp. 265–272.

Peeta, S., Mahmassani, H.S., 1995. Multiple user classes real-time traffic assignment for on-line operations: a rolling horizon solution framework. Transp. Res. C 3 (2), 83–98.

Peeta, S., Ziliaskopoulos, A.K., 2001. Foundations of dynamic traffic assignment: the past, the present and the future. Netw. Spat. Econ. 1 (3–4), 233–265.

Pel, A.J., Bliemer, M.C.J., Hoogendoorn, S.P., 2012. A review on travel behaviour modelling in dynamic traffic simulation models for evacuations. Transportation 39 (1), 97–123.

Rathi, A.K., Nemeth, Z.A., 1986. Freesim: A microscopic simulation model of freeway lane closures (abridgment). Transp. Res. Rec., 1091

Rossetti, R.J.F., Bordini, R.H., Bazzan, A.L.C., Bampi, S., Liu, R., Vliet, D.V., 2002. Using bdi agents to improve driver modelling in a commuter scenario. Transp. Res. Part C: Emerg. Technol. 10 (5), 373–398.

Salvini, P., Miller, E.J., 2005. Ilute: An operational prototype of a comprehensive microsimulation model of urban systems. Netw. Spat. Econ. 5, 217–234. http://dx.doi.org/10.1007/s11067-005-2630-5.

Sano, Y., Kadono, Y., Fukuta, N., 2016. A performance optimization support framework for gpu-based traffic simulations with negotiating agents. In: Recent Advances in Agent-based Complex Automated Negotiation. Springer, pp. 141–156.

Spiess, H., Florian, M., 1989. Optimal strategies: a new assignment model for large scale transit network. Transp. Res. B 23 (2), 83–102.

Strippgen, D., Nagel, K., 2009. Multi-agent traffic simulation with cuda. In: International Conference on High Performance Computing & Simulation, 2009. HPCS'09. IEEE, pp. 106–114.

Ueberschaer, M.H., 1971. Choice of routes on urban networks for the journey to work. Highway Res. Rec. 369.

Wachs, M., 1967. Relationships between drivers'attitudes toward alternate routes and driver and route characteristics. Highway Res. Rec. 197, 70–87.

Waddell, P., 2002. Urbansim: modeling urban development for land use, transportation and environmental planning. J. Am. Plann. Assoc. 3 (3), 297–314.

Wardrop, J., 1952. Some theoretical aspects of road traffic research. Proc. Inst. Civil Eng. 1, 325–378, part II.

Wright, M., 2017. Google traffic map shows bumper-to-bumper congestion shadaowing track of the eclipse as solar tourists head home in epic jams that could last all week Daily Mail. <http://www.dailymail.co.uk>.

Xiong, C., Chen, X., He, X., Lin, X., Zhang, L., 2016. Agent-based en-route diversion: dynamic behavioral responses and network performance represented by macroscopic fundamental diagrams. Transp. Res. Part C: Emerg. Technol. 64, 148–163.

Xiong, C., Zhou, X., Zhang, L., 2017. Agbm-dtalite: an integrated modelling system of agent-based travel behaviour and transportation network dynamics. Travel Behav. Soc.

Zhou, X., Taylor, J., Pratico, F., 2014. Dtalite: A queue-based mesoscopic traffic simulator for fast model evaluation and calibration. Cogent Eng. 1 (1), 961345.