

# Integrating MATSim and ITSUMO for Daily Replanning Under Congestion

Ana L. C. Bazzan<sup>1</sup>, Kai Nagel<sup>2</sup> and Franziska Klügl<sup>3</sup>

<sup>1</sup> Instituto de Informática, UFRGS  
Caixa Postal 15064, 91.501-970 Porto Alegre, RS, Brazil  
*bazzan@inf.ufrgs.br*

<sup>2</sup> Inst. for Land and Sea Transport Systems, TU Berlin  
Salzufer 17–19, 10587 Berlin, Germany  
*nagel@vsp.tu-berlin.de*

<sup>3</sup> Örebro University, Örebro, Sweden  
*franziska.klugl@oru.se*

**Abstract.** One way to cope with the increasing demand in transportation is to integrate standard solutions with more intelligent measures. This problem has been approached from different sides such as the study of the assignment of the demand in the network, and the investigation of the effects of control measures. However, given that most of these approaches are complex and deal with different levels of abstraction of the original problem, there has been few attempts to address it from a more general perspective. Thus our aim is to propose a methodology to integrate behavioral models of human travelers reacting to traffic patterns and control measures of these traffic patterns, focusing on distributed and decentralized methods. Issues related to control measures are of course highly coupled with how travelers plan and divert a journey. These two problems have been addressed by the ITSUMO and MATSim tools respectively. In this specific paper we discuss the integration of both tools and illustrate its use with a case study.

## 1 Introduction

Urban mobility is one of the key topics in modern society. Our long term agenda is to propose a methodology to integrate behavioral models of human travelers reacting to traffic patterns and control measures of these traffic patterns, focusing on distributed and decentralized methods. Classically, this is done via network analysis. To this aim, it is assumed that individual road users seek to optimize their individual costs regarding the trips they make by selecting the “best” route. This is the basis of the well known traffic network analysis based on Wardrop’s equilibrium principle. There are many variants of the Wardrop equilibrium, such as the dynamic user equilibrium, or the so-called stochastic user equilibrium (which is, in effect, a deterministic distribution of traffic streams across alternatives). It is even possible to apply the dynamic user equilibrium in a truly stochastic situation, where the traffic situation changes from day to day.

In that situation, however, the definition of the game is such that the players can at best play strategies that optimize average reward. Although it is clearly possible to simulate situations where either drivers or traffic lights or both are within-day adaptive to variable traffic, few if any investigations exist that attempt to clarify the overall system effects of such adaptiveness. In summary, as equilibrium-based concepts generally overlook the within-day variability regarding demand and capacity, it seems obvious that they are not adequate to be used in microscopic, decentralized approaches.

This paper thus describes the integration of two tools for traffic simulation. The first one (MATSim) has its strengths on the planning side, without considering fine control measures such as the existence of traffic lights in the network. The second (ITSUMO), on the other hand, focuses on short time control by means of agents in charge of optimization of signal plans, but provides just basic tools for the definition of routes and plans for drivers. In fact, because ITSUMO is based on cellular automata (CA) model, drivers are no more than particles that have no a priori routes. Rather they are re-routed at each crossing (junctions) according to macroscopic rules. With the integration of both tools, we aim at having the best of both, namely a planning capability for the drivers and the movement of the vehicles in the CA based simulator, which considers different control measures by the traffic light agents.

In order to describe the integration of those tools, this paper is organized as follows. In the next section we discuss background and related works. Section 3 describes MATSim; Section 4 describes ITSUMO. In Section 5 we discuss the feedback loop between planning in MATSim, control in ITSUMO, replanning in MATSim, and so on, as well as a case study. Section 6 presents concluding remarks.

## 2 Background

### 2.1 Demand and Control

On the side of demand forecasting, the arguably most used computational method is the so-called 4-step-process consisting of: trip generation, destination choice, mode choice, and route assignment. The 4-step-process has several drawbacks. For a discussion of these issues see [1]. Traditional transportation planning tools work macroscopically, distributing static traffic flows onto a network. While this is a well-established technology, it is not able to fully model all aspects that are of interest when modelling tolls. In particular, they usually lack any meaning of time-of-day. Dynamic traffic assignment (DTA) explicitly models the temporal development of the traffic. Demand, however, is typically given as fixed-period (e.g. hourly).

Given the current developments in communication and hardware, computer-based traffic control and management of the traffic system is now a reality. From the side of control, a popular method is to use traffic lights. Several signal plans are normally required for an intersection to deal with changes in traffic volume.

Thus, there must be a mechanism to select one of these plans. Optimization of the control of traffic lights has been done for several years using different techniques. Regarding multiagent systems, there has been attempts using game theory, swarm intelligence, and reinforcement learning. Some of these were already implemented as traffic light agents in ITSUMO. Due to lack of space we refer the reader to [2, 4] for more details in control and for a collection on papers dealing with aspects of demand and control respectively.

## 2.2 Previous Attempts to Address the Integration of Demand and Supply

Regarding integration of traffic assignment and control, there are a number of works which represent different views of this issue. In [11], a two-level, three-player game is discussed. The control part involves two players, namely two road authorities, while the population of drivers is seen as the third player. Complete information is assumed, which means that all players (including the population of drivers) have to be aware of the movements of others. Moreover, it is questionable whether the same mechanism can be used in more complex scenarios, as claimed because when the network is composed of a high number of links, the number of routes increases and so the complexity of the route choice, given that now it is not trivial to compute the network and user equilibria.

Liu and colleagues [7] describe a modeling approach which integrates microsimulation of individual trip-makers' decisions and individual vehicle movements across the network. However, their focus is on the description of the methodology only.

Ben-Akiva and co-workers have investigated in some detail the issue of so-called self-consistent anticipatory route guidance [5]. In this, a loop "traffic control – driver reaction – network loading" is defined. The loop is closed by the traffic control being reactive to the result of the network loading. The approach, however, focuses on information as control input, not traffic signals.

Papageorgiou has looked into the problem with a control-theoretic approach [9]. So posed, human behavior and network loading are combined into the dynamical update of the system, and the goal is to search for a control input that optimizes some aspect of the output from the system. However, human behavior is by necessity of the mathematical formulation very much reduced, and no results about the emergent properties from system-wide signal control seem to be known.

## 3 MATSim

MATSim<sup>4</sup> is constructed around the notion of agent-based simulation. Each traveler of the real system is modeled as an individual agent in our simulation. The overall approach consists of three important pieces:

---

<sup>4</sup> [www.matsim.org](http://www.matsim.org)

- Each agent independently generates a so-called plan, which encodes its intentions during a certain time period, typically a day.
- All agents’ plans are simultaneously executed in the simulation of the physical system. This is also called the traffic flow simulation or mobility simulation.
- There is a mechanism that allows agents to learn. In our implementation, the system iterates between plans generation and traffic flow simulation. The system remembers several plans per agent, and scores the performance of each plan. Agents normally chose the plan with the highest score, sometimes re-evaluate plans with bad scores, and sometimes obtain new plans by modifying copies of existing plans.

A plan contains the itinerary of activities the agent wants to perform during the day, plus the intervening trip legs the agent must take to travel between activities. An agent’s plan details the order, type, location, duration and other time constraints of each activity, and the mode, route and expected departure and travel times of each leg. This paper concentrates on “home” and “work” as the only activities, and “car” as the only mode. A plan can be modified by various modules. This paper makes use of two modules only: the *Activity Times Generator* and the *Router*. The former is called to change the timing of an agent’s plan. At this point, a very simple approach is used which just applies a random “mutation” to the duration attributes of the agent’s activities. Although this approach is not very sophisticated, it is sufficient in order to obtain useful results. This is consistent with our overall assumption that, to a certain extent, simple modules can be used in conjunction with a large number of learning iterations. The router is implemented as a time-dependent Dijkstra algorithm. It calculates link travel times from the output of the traffic flow simulation. The link travel times are encoded in 15 minute time bins, so they can be used as the weights of the links in the network graph.

The traffic flow simulation executes all agents’ plans simultaneously on the network, and provides output describing what happened to each individual agent during the execution of its plan. The traffic flow simulation is implemented as a queue simulation, which means that each street (link) is represented as a FIFO queue. The outcome of the traffic flow simulation (e.g. congestion) depends on the planning decisions made by the decision-making modules. However, those modules can base their decisions on the output of the traffic flow simulation, using this as feedback from the multi-agent simulation structure.

This sets up an iteration cycle which runs the traffic flow simulation with specific plans for the agents, then uses the planning modules to update the plans; these changed plans are again fed into the traffic flow simulation, etc, until consistency between modules is reached. However, currently MATSim does not consider fine grained control as for instance, the presence of traffic lights.

The feedback cycle is controlled by the agent database, which also keeps track of multiple plans generated by each agent, allowing agents to reuse those plans. The repetition of the iteration cycle coupled with the agent database enables the agents to learn how to improve their plans over many iterations. This cycle

continues until the system has reached a relaxed state. At this point, there is no quantitative measure of when the system is “relaxed”; we just allow the cycle to continue until the outcome seems stable. More details can be found in [6].

## 4 ITSUMO

ITSUMO<sup>5</sup> – Intelligent Transportation System for Urban Mobility – [10] is a microscopic traffic simulator based on CA. The implementation uses agent technologies with a bottom-up philosophy in mind. Although ITSUMO has also been used to investigate route choice scenarios, the focus has been primarily on control. In order to achieve the necessary simplicity and performance, we use the Nagel–Schreckenberg CA model [8]. In short, each road is divided in cells with a fixed length. This allows the representation of a road as an array where vehicles occupy discrete positions. The vehicle behavior is expressed by rules that represent a special form of car–following behavior. This simple, yet valid microscopic traffic model, can be implemented in such an efficient way that is good enough for real-time simulation and control of traffic.

The information regarding the topology of the traffic network is stored in an XML file. ITSUMO is composed by several modules: data module, simulation kernel, driver definition, traffic lights agent, and the visualization module. The data module creates, updates, and stores (XML file) both the static and the dynamic objects to be used in the simulation, as for instance the cartesian coordinates of the intersections. The main attributes are: General Settings (topology name, traffic system orientation, cell size, frequency of sensor measurements, deceleration probability, etc.); Network (name and settings); Node (cartesian coordinates of the intersection); Street; Section; Laneset; Lane; Turning Probabilities (set of allowed vehicle movements and their probabilities); Signal Plan (set of lane-to-laneset allowed movements in a specific order, cycle time and offset); Sink and Source (nodes for removal and insertion of vehicle according to given values).

The simulation kernel was developed using C++ and implements the microscopic model presented before. The simulation output can be formatted according to the user needs. The most usual formats are the “cell map” and the “laneset occupation map”. The former indicates which portions of the lane are occupied by which vehicle, providing the most detailed output possible. On the other hand, the “laneset occupation map” is a high-level output which specifies the rate of occupation (density) for each laneset in the network.

Modeling drivers’ behavior can be approached in different ways, depending on the purpose of the simulation. In most cases, the objective is to simulate the collective or macroscopic behavior. However, this behavior emerges out of individual ones. Simple algorithms, like the CA model, can be used to describe vehicles movement without losing significant simulation fidelity with reality. However, as said before, more sophisticated driver behaviors such as those based

---

<sup>5</sup> <http://www.inf.ufrgs.br/~mas/traffic/itsumo/index.html>

on route planning or en-route decision are more difficult to implement in ITSUMO. Up to now ITSUMO provides a GUI to define a route for a handful of drivers, the so-called “floating cars”. This process of route definition is manual and can of course be done for more drivers but it is time consuming. Thus the integration with MATSim aims at addressing this shortcoming.

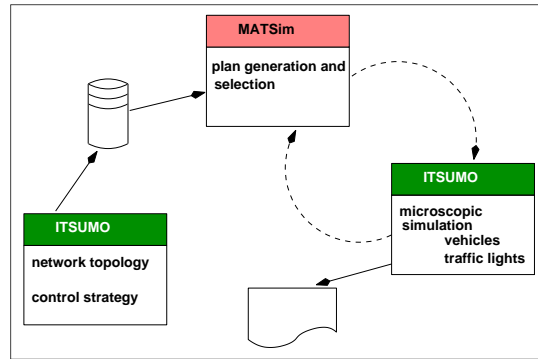
In ITSUMO the control of traffic lights is implemented and executed via traffic light agents. These can control one or more crossing or junction. Basic classes for creation of traffic light agents have been implemented in order to facilitate the development of traffic controllers. These agents are organized in a data structure that is kept separated from the simulator. Therefore, the user does not need to manipulate the simulator code. This makes the independence of this module possible. Moreover if any user wishes to develop its own code in order to implement a different control method, this can be easily done.

Between the traffic light agents and the simulator, a communication is established using sockets, where it is possible to receive the necessary information (from the simulation kernel) for the implementation of the control. Such information can be e.g. number of stopped vehicles, density, speed etc. of the lanes under control by the given agent. The agent can then send a control action back to the simulation kernel (normally this control action takes the form of a signal plan). This will cause the kernel to run the action (signal plan) selected by the traffic light agent.

The visualization module allows the graphical visualization – either in a macroscopic or microscopic level – of the simulation results. At a macroscopic level, the visualization considers only data which reflect the overall behavior of the network, providing an useful tool to capture the big picture of what is happening in a specific scenario. The microscopic level provides an interface through which one can see individual vehicles movement. In order to obtain a more realistic and detailed visualization, these modules are developed using OpenGL, enabling features such as walk-through navigation and detail-focused interfaces.

## 5 Integrated tool and Case Study

The aim of the integrated tool is to have a feedback loop between planning in MATSim and execution of plans in ITSUMO. As already said, MATSim’s plans can be executed in MATSim itself (simulation of traffic flow module) but this does not consider sophisticated control measures. On the other hand, not all ITSUMO drivers have plans because there is no mechanism for automatic plan generation. Plans, if used, must be provided manually by the user, in a time consuming way. Thus, once MATSim generates the plans, ITSUMO reads them, makes drivers execute their routes, while some control is also carried out. For example, one may have traffic lights all running a greedy strategy that gives priority (e.g. more green time) to the more congested approaching lanes. Once the trips are over, ITSUMO provides information for MATSim to replan (if



**Fig. 1.** Scheme of the integration of MATSim and ITSUMO

necessary) drivers' trips. Plans are resent to ITSUMO, a new execution of them is performed, information is given back to MATSim etc.

Figure 1 shows a scheme of the integrated tool based on the interaction between MATSim and ITSUMO. The approach works by having ITSUMO generating the representation of the network (e.g. a grid like in Figure 2 or any other kind of network) in an XML file. In ITSUMO it is also possible to design traffic light agents that implement any particular kind of control algorithm that is based on signal plans. These are optional and, if not provided, the control will take place by running the default signal plan. Given the XML file which is read by MATSim, routes are created for each driver (route library), and the algorithms for route choice is run. Chosen routes are passed back to ITSUMO for execution.

The following cycle is repeated for as long as previously defined.

1. MATSim makes a choice of route for each driver based on given strategies (see Section 3);
2. MATSim exports the plans yielded by the above choice (in the ITSUMO format).
3. MATSim calls for an execution of ITSUMO and waits;
4. ITSUMO simulates the plans execution using traffic light control;
5. ITSUMO generates a log file in MATSim's format with the events of the simulation (events represent actions such as departure, left link, entered link and arrival time; the log of an event also includes time and place in which it occurred);
6. MATSim reads the events log to calculate the plans' scores and stores them.

In order to illustrate the integration, we discuss a case study. This deals with a 6x6 grid network which we have used previously [3]. However the latter has not included a microscopic model of flow, which is now possible due to the integration of ITSUMO to MATSim. The 6x6 grid used is a typical commuting scenario where drivers repeatedly select a route from an origin to a destination. Because

it is not a simple binary choice, it deals with route choice in a network with a variety of possible routes. Thus, it captures desirable properties of real-world scenarios. In the case of the grid with 36 nodes (depicted in Figure 2), all links are one-way and drivers can turn in each crossing. Although it is apparently simple, this kind of scenario is realistic and, from the point of view of route choice and equilibrium computation, it is also a very complex one as the number of possible routes between two locations is high. In contrast to simple two-route scenarios, it is possible to set arbitrary origins (O) and destinations (D) in this grid. For every driver agent, its origin and destination are randomly selected according to probabilities given for the links. To render the scenario more realistic, neither the distribution of O-D combinations, nor the capacity of links is homogeneous. On average, 60% of the road users have the same destination, namely the link labelled as E4E5 which can be thought as something like a main business area. Other links have, each, 1.7% probability of being a destination. Origins are nearly equally distributed in the grid, with three exceptions (three “main residential areas”): links B5B4, E1D1, and C2B2 have, approximately, probabilities 3, 4, and 5% of being an origin respectively. The remaining links have each a probability of 1.5%.

Drivers in the integrated framework use tools provided by MATSim. Henceforth we refer to adapting drivers when they use the MATSim strategy for re-planning, and to fixed drivers when they only have one route so that they cannot change their route choice. The adaptation strategy for drivers is as follows: *ReRoute* is performed with 10% of drivers replanning their trips. This means that, after each ITSUMO execution, and according to the scores provided (a function of the travel time), 10% of the drivers are allowed to replan their trips.

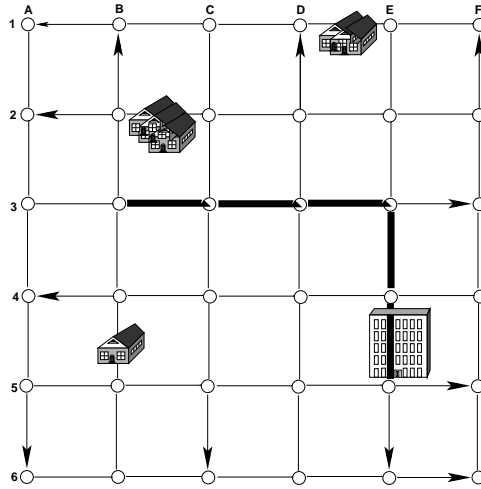
Regarding the control, in ITSUMO, the control is performed by the traffic light agents located in each node. Each of them has similar signal plans, with a cycle length of 40 time steps. One signal plan gives equal green time to both directions (default). Two others allocate 30 time steps to one direction and 10 to the other. Thus there are three signal plans for each intersection. The actions of the traffic light agents are: to run the default signal, or to select another plan in order to allow more green time to one phase. For the sake of the present paper traffic light agents can control intersections in the following ways:

- fixed: no selection of signal plan; always run the default one which splits the green time equally between the approaching lanes
- greedy: choose the plan that allows more green time for the approaching lanes with higher current occupancy

The metric used to evaluate the different combination of strategies is as in [3]: we execute a batch of  $T$  trips, measure drivers’ travel time, and depict the mean travel time over the last 5 trips. In the experiments we have used 400 and 700 driver. We execute each simulation 20 times with  $T = 15$ . Thus, the data depicted in the table appearing in Fig. 3 is averaged over 20 runs.

One observes the following trends: When drivers are fixed, then making the traffic signals adaptive improves the performance of the system considerably.





**Fig. 2.** 6x6 grid showing the main destination (E4E5), the three main origins (B5B4, E1D1, C2B2), and the “main street” (darker line).

Type of simulation		Nb. of drivers	
Drivers	TLs	400	700
Fixed	Fixed	212	330
Fixed	Greedy	185	293
Adapting	Fixed	176	255
Adapting	Greedy	170	253

**Fig. 3.** Results for the grid network using MATSim and IT-SUMO, showing the average trip duration of the last 5 iterations for different numbers of drivers.

This tendency holds rather independently from the level of congestion (number of drivers). This result can be intuitively understood by considering that fixed drivers means that drivers insist on a certain route, no matter what the congestion level. Clearly, traffic lights that reduce congestion levels are beneficial. When drivers are day-to-day adaptive, the additional gains of also making the traffic signals adaptive are not significant. Also, the performance of the system with adaptive drivers but fixed traffic signals is similar to the performance of the system with fixed drivers but adaptive signals. This indicates that in the particular situation of this study, either the adapting drivers or the adapting traffic lights can reach states that are difficult if not impossible to improve any further.

## 6 Conclusion and Outline

We have discussed the integration of MATSim and ITSUMO which now allows us to simulate traffic flow in a microscopic way while also considering a microscopic simulation of individual drivers. In order to illustrate this integration we have used a scenario already employed by us before. Previously, the simulation of this scenario has not included a microscopic model of flow, which is now possible due to the integration of ITSUMO to MATSim. This is a significant improvement given that it uses microscopic models and replanning. The scenario used is complex to evaluate given that the number of possible routes is very high. This allows us to only make general conclusions about the overall travel time in the network when we have both drivers and traffic lights adapting. Thus, the next

step of this work is to investigate this problem of co-adaptation in a simpler, two route scenario and later extend it to more complex cases.

## Acknowledgments

The authors would like to thank CAPES (Brazil) and DAAD (Germany) for their support to the joint, bilateral project “Large Scale Agent-based Traffic Simulation for Predicting Traffic Conditions”. Ana Bazzan is partially supported by CNPq and Alexander von Humboldt Stiftung.

## References

1. M. Balmer, K. Nagel, and B. Raney. Large-scale multi-agent simulations for transportation applications. *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, 8(4):205–221, 2004.
2. A. L. C. Bazzan. Opportunities for multiagent systems and multiagent reinforcement learning in traffic control. *Autonomous Agents and Multiagent Systems*, 18(3):342–375, June 2009.
3. A. L. C. Bazzan, D. de Oliveira, F. Klügl, and K. Nagel. Adapt or not to adapt – consequences of adapting driver and traffic light agents. In K. Tuyls, A. Nowe, Z. Guessoum, and D. Kudenko, editors, *Adaptive Agents and Multi-Agent Systems III*, volume 4865 of *Lecture Notes in Artificial Intelligence*, pages 1–14. Springer-Verlag, 2008.
4. A. L. C. Bazzan and F. Klügl, editors. *Multi-Agent Systems for Traffic and Transportation*. IGI Global, Hershey, PA, April 2009.
5. J. Bottom, M. Ben-Akiva, M. Bierlaire, and I. Chabini. Generation of consistent anticipatory route guidance. In *Proceedings of TRISTAN III*, volume 2, San Juan, Puerto Rico, June 1998.
6. D. Charypar and K. Nagel. Generating complete all-day activity plans with genetic algorithms. *Transportation*, 32(4):369–397, 2005.
7. R. Liu, D. Van Vliet, and D. Watling. Microsimulation models incorporating both demand and supply dynamics. *Transportation Research Part A: Policy and Practice*, 40(2):125–150, February 2006.
8. K. Nagel and M. Schreckenberg. A cellular automaton model for freeway traffic. *Journal de Physique I*, 2:2221, 1992.
9. M. Papageorgiou. Traffic control. In R. W. Hall, editor, *Handbook of Transportation Science*, chapter 8, pages 243–277. Kluwer Academic Pub, 2003.
10. B. C. d. Silva, R. Junges, D. Oliveira, and A. L. C. Bazzan. ITSUMO: an intelligent transportation system for urban mobility. In H. Nakashima, M. P. Wellman, G. Weiss, and P. Stone, editors, *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, pages 1471–1472. ACM Press, May 2006.
11. H. J. van Zuylen and H. Taale. Urban networks with ring roads: a two-level, three player game. In *Proc. of the 83rd Annual Meeting of the Transportation Research Board*. TRB, January 2004.