Integrating MATSim and ITSUMO for Daily Replanning Under Congestion

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

Kai Nagel
Inst. for Land and Sea Transport Systems
TU Berlin Sek SG12
Salzufer 17–19
10587 Berlin, Germany
nagel@vsp.tu-berlin.de

Franziska Klügl
Dep. of Artificial Intelligence
University of Würzburg
Am Hubland
97074 Würzburg, Germany
kluegl@informatik.uni-wuerzburg.de

ABSTRACT

One way to cope with the increasing demand in transportation networks 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 complex behavioral models of human travelers reacting to traffic patterns and control measures of these traffic patterns, focusing on distributed and decentralized methods. The traveler (and his/her reaction to traffic) is closely related to how commuters' trips are planned and executed along a day. The control measures are of course highly coupled with how travelers plan and divert a journey. These two problems have been addressed by the MATSim and ITSUMO 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 affecting both policy-makers and citizens/tax-payers. Especially in medium to big cities, the urban space has to be adapted to cope with the increasing needs of the commuters. In transportation engineering the expression of the transport needs is called *demand*. This demand (in terms of people, volume, etc.) is commonly used to quantify transport *supply*. This is the expression of the capacity of transportation infrastructures and modes. Supply is expressed in terms of infrastructures (capacity), services (frequency), and networks.

The increasing demand we observe nowadays has to be accommodated either with increasing supply (e.g. road capacity), or with a better use of the existing infrastructure. Since an expansion of the capacity is not always socially attainable or economically feasible, transportation and traffic engineering now seek to optimize the management of both the supply and the demand using concepts and techniques from intelligent transportation systems (ITS). These refer to the application of modern technologies to the operation and control of transportation systems [17].

From the side of supply, several measures have been adopted in the last years, such as congestion charging in urban areas (London), restriction of traffic in the historical center (Rome, Paris, Amsterdam), alternace of vehicles allowed to circulate in a given day (São Paulo, Mexico City). However, these measures impact the daily life of many commuters and tax-payers, as they pose restrictions to the freedom of movement. In order for these measures to be better accepted, traffic authorities can try to compensate the population integrating those solutions with more intelligent measures derived from ITS.

From the point of view of the demand, several attempts exist to distributed the demand within the available infrastructure. Besides, it is now recognized that the human actor has to be brought into the loop. With the amount of information that we have nowadays, it is almost impossible to disregard the influence of real-time information systems over the decision-making process of the individuals.

Hence, the general aim of our the project "Large Scale Agent-based Traffic Simulation for Predicting Traffic Conditions" is to address a complex problem like traffic from the point of view of information science. This project is the result of an accumulated experience with microscopic models modeling tools for traffic and transportation management. These range from traffic signal optimization [2], binary route choice, and effect of information on commuters [10], to microscopic modelling of physical movement [13] and modelling and simulation of full commuters' daily plans in MATSim.

An important milestone in our project is to propose a methodology to integrate complex 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 not done in a decentralized way. Rather network analysis is used. 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 [22]. This method predicts a long term *average* state of the network. By assuming steady state network supply and demand conditions from day-to-day, this equilibrium based method cannot, in most cases, cope with the dynamics of the modern transportation systems. Moreover, it is definitely not adequate for answering questions related to what happens in the network *within* a given day, as the variability in the demand and the capacity of the network tend to be high. Just think about changing weather conditions from day-to-day and within a single day! In summary, as equilibria based concepts overlook this variability, it seems obvious that it is not adequate to be used in microscopic modeling and simulation.

The reason why microscopic approaches are getting more and more popular within the community of transportation engineering is twofold. First, it is well recognized that individual decision making does affect the equilibrium and this cannot be disconsidered. The second factor is the improvements in hardware and in software paradigms (e.g. agent based simulation), which now allow the consideration of individual characteristics.

Based on this important assumption, the field of transportation engineering has seen a boom regarding methodologies for microscopic modeling as well as a trend towards development of real-time systems. As part of this effort, a new research area is studying how to integrate all pieces of work which have been produced in different fields such as traffic and transportation engineering itself, physics, psychology, computer science, geography, etc. Microscopic modeling can be accomplished in several ways. Agent-based modeling is one alternative.

This paper thus describes the integration of two tools for traffic simulation. One (MATSim) has its strenghts 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, focusses 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 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 get back to the issue of the paradigm change from macroscopic to microscopic, agent-based simulation. Section 3 describes MATSim; Section 4 describes ITSUMO. In Section 5 we give the details of the modules created to allow the feedback loop between planning in MATSim, control in ITSUMO, replanning in MATSim, etc. Section 6 returns to a case study that has appeared in a previous paper [3], this time using the integrated tool presented here, not the prototyping tool that was queue-based. Section 7 discusses ongoing work and some concluding remarks.

2. TOWARDS INTEGRATED AGENT-BASED TRAFFIC ASSIGNMENT, SIMULATION, AND CONTROL

2.1 Demand

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) OD matrices, and does, in consequence, not adapt to the toll. Adaptation would need to happen in the demand generation modules that generate the OD matrices, but that implies rather intricate coupling between demand generation and DTA.

Every model that uses single trips only will have problems predicting useful reactions of travellers that span the whole day. This is because trips in real life are embedded in a complete day plan. This means, for example, that travellers cannot escape a toll at their will, but have to trade off between different utilities (working eight hours, being at a shop when it has opened, ...) and disutilities (paying a toll, being late for work, ...).

2.2 Control

Given the current developments in communication and hardware, computer-based traffic control and management of the traffic system is now a reality. The main goals of the control are: to maximize the overall capacity of the network; to maximize the capacity of critical routes and intersections which represent the bottlenecks; to minimize the negative impacts of traffic on the environment and on energy consumption; to minimize travel times; and to increase traffic safety.

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.

Signalized intersections are controlled by signal-timing plans which are implemented at traffic signals. A signal-timing plan (signal plan for short) is a unique set of timing parameters comprising basically the cycle length (the length of time for the complete sequence of the phase changes), the split (the division of the cycle length C among the various movements or phases), pedestrian requirements for timing, and the phase-change interval.

There are several concepts of computer-based control of traffic lights. The most basic is the computer sends out commands that control the signals at crossings (isolated or in an arterial), receiving no feedback. Thus the traffic signal plans are not responsive to the actual traffic conditions. Signal plans are generated off-line based on historical or earlier traffic data, not in a real time fashion. The second concept for traffic lights control is similar to the former but uses detectors that feed information back to the central computer. However, this information is not used to influence the current plan selection, just for off-line creation of other plans. In full actuated operation, every lane of every approach must be monitored by a detector. Green time is allocated according to information from the detectors and programmed rules established in the controller. In this type of operation, the cycle length, sequence of phases, and green time split may very from cycle to cycle.

Another concept is operating coordinated systems (also called synchronized or progressive systems). We do not go into details here about coordinated systems. Readers can find more details in [2].

Optimization of the control of traffic lights has been done for several years using different techniques (see [20, 9, 12, 7] for some prominent ones). Regarding multiagent systems, there has been attempts using game theory ([2]), swarm intelligence ([15]), and reinforcement learning (e.g. [23, 14, 18]). Some of these were already implemented as traffic light agents in ITSUMO.

2.3 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 [21], a two-level, three-player game is discussed. the control of traffic lights and the route choices by drivers are considered. The control part involves two players, namely two road authorities, while the population of drivers is seen as the third player. These are modeled as a game and the main aim of this work is to analyze the outcome when players are able to observe the previous move. 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. The reason for this is the fact that 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 [11] describe a modeling approach which integrates microsimulation of individual trip-makers' decisions and individual vehicle movements across the network. Their focus is on the description of the methodology which incorporates both demand and supply dynamics, so that the applications are only briefly described and not many options for the operation and control of traffic lights are reported. One scenario described deals with a simple network with four possible routes and two control policies.

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 resulting problem is defined as a fixed point problem: A solution is found if the traffic control, via driver reaction and network loading, generates the same traffic pattern that was the basis for the traffic control. The approach, however, focuses on information as control input, not traffic signals.

Papageorgiou and co-workers look into the problem with a control-theoretic approach [16]. In that language, 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[1] 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:

- Each agent independently generates a so-called plan, which encodes its intentions during a certain time period, typically a day. Figure 1 depicts a plan for a synthetic person, including demographic data.
- 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.

---

[1] `www.matsim.org`

```
 <person id="393241" sex="f" age="27" license="yes" car_avail="always" employed="yes">
<travelcard type="regional-abo" />
<plan>
<act type="home" link="58" start_time="00:00" dur="07:00" end_time="07:00" />
<leg mode="car" dept_time="07:00" trav_time="00:25" arr_time="07:25">
<route>1932 1933 1934 1947</route>
</leg>
<act type="work" link="844" start_time="07:25" dur="09:00" end_time="16:25" />
<leg mode="car" dept_time="16:25" trav_time="00:14" arr_time="16:39">
<route>1934 1933</route>
</leg>
<act type="home" link="58" start_time="16:39" dur="07:21" end_time="24:00" />
</plan>
</person>
```

Figure 1: Plan of a synthetic person in MATSim

The current version of MATSim is a re-implementation in Java.

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:

- *Activity Times Generator:* This module 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.

- *Router:* 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 at will. 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.

In order to compare plans, it is necessary to assign a quantitative score to the performance of each plan. In principle, arbitrary scoring schemes can be used. In this work, travel time is used as the sole indicator of a plan's performance. The elements of our approach are as follows:

- The total score of a plan is computed as the sum of individual contributions, which consist of positive contributions for performing an activity, and negative contributions for travelling and for schedule delays.

- A logarithmic form is used for the positive utility earned by performing an activity.

- The (dis)utility of traveling is assumed as linear in the travel time.

- The (dis)utility of being late is assumed as linear in the delay.

It is important to note that the score thus takes into account the complete daily plan. More details can be found in [6].

4. ITSUMO

ITSUMO[2] – Intelligent Transportation System for Urban Mobility – [19] is a microscopic traffic simulator based on CA. The implementation uses agent technologies with a bottom-up philosophy in mind. One of the aims of this simulator was to integrate different functionalities, such as a simple traffic control (mostly based on control of traffic lights), traffic management, and real-time information providing via internet and/or mobile phone. Although ITSUMO has been used to investigate route choice scenarios [4], the focus has been primarily on control.

In order to achieve the necessary simplicity and performance, we use the Nagel–Schreckenberg CA model [13]. 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. Each vehicle travels with a speed based on the number of cells it currently may advance without hitting another vehicle. 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. Given that every vehicle has a nonnegative integer speed (v, limited to v_{max}), the following four rules are verified simultaneously for all vehicles in a CA way: movement (each vehicle advances v cells at each time step); acceleration (each vehicle's speed is increased by one unit, up to v_{max} or the *gap* – the number of empty cells in front of a vehicle); interaction (if the vehicle ahead is too close, v is decreased by one unit; and randomization (vehicles decelerate with probability p in order to simulate the nondeterministic dynamics of vehicle movement). Although these rules might seem too simplistic, investigations reported in the literature showed that the cellular automaton model is capable of reproducing macroscopic traffic flow features including realistic lane changing behavior.

The information regarding the topology of the traffic network is stored in a XML file (see Figure 2). This file also contains other information, as for instance, definitions of plans for drivers. However, as said, ITSUMO does not generate plans automatically. Thus in addition to the CA approach, one can also define other driver decision-making procedures via a special, optional, module. If the user does not define particular classes of drivers, then the simulation is performed using the standard driver model described above (CA rules).

A visualization module retrieves data originated from the microscopic simulation and exhibits a graphical representation of the traffic simulation. ITSUMO is thus composed by several modules: data module, simulation kernel, driver definition module, traffic lights agent module, and the visualization module.

```
<network_id> 1 </network_id>
<network_name> unnamed </network_name>
<settings>
<cell_size_in_meters> 5.0 </cell_size_in_meters>
<iteration_length_in_seconds> 1 <...>
<sensor_activation_frequency> 60 <...>
<cars_maximum_speed> 3 </cars_maximum_speed>
<trafficlight_agent_observation_frequency> 500 <...>
</settings>
<nodes>
<node>
<node_id> 2 </node_id>
<node_name> n0 </node_name>
<x_coord> 6250.0 </x_coord>
<y_coord> 6000.0 </y_coord>
</node>
...
<traffic_lights>
<traffic_light>
<traffic_light_id> 161 </traffic_light_id>
<located_at_node> 14 </located_at_node>
<signalplans>
<signalplan>
<signalplan_id> 1 </signalplan_id>
<phases>
<phase>
<phase_id> 1 </phase_id>
<iteration_start> 0 </iteration_start>
...
<section>
<section_id> 74 </section_id>
<section_name> (g2) &lt;-&gt; (n2) </section_name>
<is_preferencial> false </is_preferencial>
<delimiting_node> 25 </delimiting_node>
<delimiting_node> 4 </delimiting_node>
<lanesets>
<laneset>
<laneset_id> 75 </laneset_id>
<laneset_position> 1 </laneset_position>
<start_node> 25 </start_node>
<end_node> 4 </end_node>
<turning_probabilities>
<direction>
<destination_laneset> 78 </destination_laneset>
<probability> 100.0 </probability>
</direction>
</turning_probabilities>
<lanes>
<lane>
<lane_id> 76 </lane_id>
<lane_position> 1 </lane_position>
<maximum_speed> 10 </maximum_speed>
<deceleration_prob> 0.0 </deceleration_prob>
</lane>
</lanes>
</laneset>
</lanesets>
</section>
...
```

Figure 2: XML file (partially) describing a traffic network in ITSUMO

4.1 General Data Module

This 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 in-

---

[2] `http://www.inf.ufrgs.br/~mas/traffic/itsumo/index.html`

tersections. The main attributes are described below (see respective XML tags in Figure 2):

- General Settings: topology name, traffic system orientation (right-handed, left-handed), cell size, frequency of sensor measurements, deceleration probability, etc.;
- Network: network name and its settings;
- Node: cartesian coordinates of the intersection;
- Street, Section, Laneset, Lane: street name, section name, whether it is preferential, delimiting nodes, laneset length, maximum speed, and width for the lane; vehicle to a specific laneset upon passing through a node and the associated lane where these probabilities are valid
- 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;

4.2 Simulation Kernel

This module was developed using C++ over the microscopic model presented before. The simulation occurs in discrete steps and is implemented as a series of updates in the drivers' decisions of movement, followed by simultaneous updates in the vehicles' positions in the network. Each update in a node or traffic light may modify its current behavior. 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.

4.3 Driver Modelling Module

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 loosing significant simulation fidelity with reality. However, as said before, more sophisticated driver behaviors such as those based on route planning or en-route decision, are more difficult to implement in ITSUMO. Up to now ITSUMO provides a GUI for the user 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 probably very time consuming. Thus the integration with MATSim aims at addressing this shortcoming.

4.4 Traffic Light Agent Module

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.

Each agent is executed via one independent thread. The communication between agents and the simulator is established at the gate 30000, by blocking TCP, and accepts a maximum of 50 connections or any other configuration.

4.5 Visualization Module

This is the module that 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. It is useful to provide this kind of information via the internet, as it has been the case in [8]. 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.

The visualization of data at microscopic level is the kernel of the information system for the traffic engineer and/or the

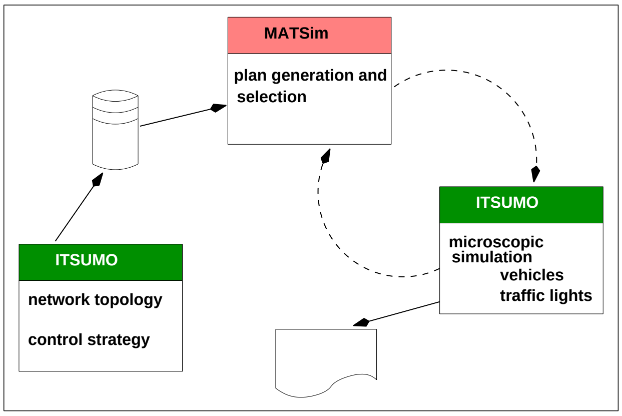

Figure 3: Scheme of the integration of MATSim and ITSUMO

```
<person id="100" sex="m" age="21" license="yes" car_avail="always" employed="yes">
 <plan score="-0.395" selected="yes">
  <act type="h" link="200" x="100.0" y="100.0" start_time="00:00:00" dur="00:00:00" end_time="00:00:00" />
  <leg num="1" mode="car" dep_time="00:00:00" trav_time="00:01:40" arr_time="00:01:40">
   <route dist="1500.0" trav_time="00:01:40">47 48 69 70 66 64</route>
  </leg>
  <act type="h" link="210" x="200.0" y="200.0" start_time="08:00:00" dur="00:00:00" end_time="00:00:00" />
 </plan>
</person>
```

Figure 4: An example of a plan resulting from the MATSim adaptation strategy.

urban planning experts. At this level, the whole system can be used to perform *what-if* simulations.

5. INTEGRATED TOOL

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

Figure 3 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 6 or any other kind of network) in a 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. Choosen 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 yield by the above choice to the ITSUMO format. An example of such plan is depicted in Figure 4 where the tag "plan score" appears;
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); an example of a log is showed in Figure 5;
6. MATSim reads the events log to calculate the plans' scores and stores them.

6. CASE STUDY

In order to illustrate the integration, we discuss next a case study. This deals with a 6x6 grid network which we have used previously [3]. At the end of this section we briefly discuss another smaller, two-route scenario.

The 6x6 grid used is a typical commuting scenario where drivers repeatedly select a route to go from an origin to a destination. Because it goes beyond simple binary choice scenarios, 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 6), all links are one-way and drivers can turn in each crossing. Although it is apparently simple, this kind of scenario is

```
T_GBL VEH_ID LEG_NR LINK_ID FROM_NODE_ID EVENT_FLAG DESCRIPTION

1 1 0 181 0 6 departure
1 1 0 181 0 4 wait2link
1 10 0 197 0 6 departure
1 10 0 197 0 4 wait2link

.
.
.

204 400 0 78 0 2 left link
204 400 0 81 0 5 entered link
204 50 0 110 0 2 left link
204 50 0 197 0 5 2 entered link
205 112 0 110 0 0 arrival
205 200 0 207 0 0 arrival
```

Figure 5: The result of each ITSUMO iteration: an events log

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%. Regarding capacity, all links can hold up to 15 vehicles, except those located in the so called "main street. These can hold up to 45 (one can think it has more lanes). This main street is formed by the links between nodes B3 to E3, E4, and E5.

Drivers in the integrated framework use the tools provided by MATSim. Henceforth we refer to adapting drivers when they use the MATSim strategy for replanning, and to fixed drivers when they only have one route so that they cannot change the route choice.

The adaptation strategy for MATSim 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. These signal plans are schematically shown in Table 1. As the table intends to be general we do not shown the lanes which form each phase, nor the begin of green and red time for each phase. The actions of the traffic light agents are: to run the default signal plan, or to select another plan in order to allow more green time to one phase. For the sake of the present paper this selection can be done 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 executing 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. Thus, the data depicted in Table 2 is averaged over 20 runs.

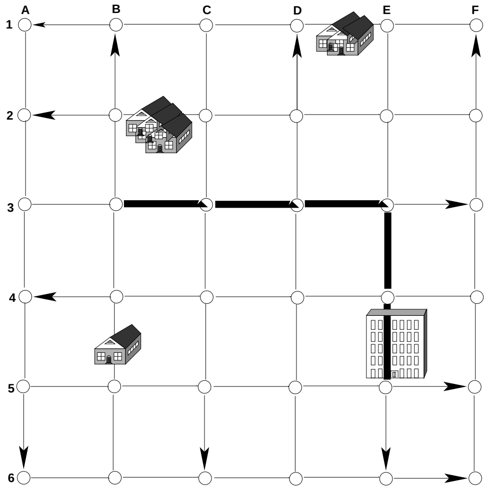

Figure 6: 6x6 grid showing the main destination (E4E5), the three main origins (B5B4, E1D1, C2B2), and the "main street" (darker line).

plan	one phase	other phase
1	20	20
2	30	10
3	10	30

Table 1: Template of signal plans for the traffic lights.

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

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

Table 2: Results for the grid network using MATSim and ITSUMO, showing the average trip duration of the last 5 iterations for different numbers of drivers.

Type of simulation		Nb. of drivers	
Drivers	TLs	400	700
Fixed	Fixed	112	346
Fixed	Greedy	94	256
Adapting	Fixed	149	442
Adapting	Greedy	143	261
Greedy	Fixed	100	411
Greedy	Greedy	106	380

Table 3: Results for the grid network using a queue based flow simulator, showing the average trip duration of the last 5 iterations for different numbers of drivers.

reach states that are difficult if not impossible to improve any further. Such a statement does not hold in general; we want to investigate this further with a two-route scenario.

We depict here the results of simulating the same scenario with a queue-based simulator. These results are reported in [3]. However, we do this just for sake of completeness as both implementations are hardly comparable. First, in the queue based simulation the actual movement of vehicles does not consume time, i.e. if the queue is empty, a vehicle is able to travel the link in one step. In the microscopic simulator, vehicles have a maximum speed and will advance v_{max} cells per time step at most. Therefore, travel times tend to be higher in the microscopic simulator, especially for the case with 400 drivers as this correspond to a not so high occupancy of the network (around 40%). In the case of 700 drivers, the occupancy is 72% and hence even in the queue based simulator it is rarely the case that a vehicle is in a nearly empty queue. Thus the travel times tend to be more similar.

Second, the adaptation mechanism of the traffic lights in the queue simulation were not exactly the same as implemented in ITSUMO. A greedy strategy in the former means that the traffic light will shift phases, while ITSUMO works with full signal plans. Thus this is another issue which renders the comparison difficult.

Third, it makes little sense to implement greedy drivers in MATSim which has a more sophisticated planning tool; we opt to use this planning instead of simplified strategies such as greedy selection of plans. In summary, the simulations performed in the queue-based simulator were necessarily simplified because both the control of traffic lights and the planning and selection mechanisms of drivers are very complex.

The second case study is a simple two route network. These two routes have a common junction and this has a traffic light. Although the route choice is binary here, thus simpler than the one discussed before for the case of the 6x6 grid, it permits the investigation of co-evolution i.e. drivers replan (using MATSim) and the traffic lights also use some kind of adaptation or learning (e.g. the Q-learning mechanism already provided by ITSUMO).

7. 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 scenario 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. In this scenario both the starting configuration (e.g. how drivers are divided in the two routes) and noise can have significant effects on the ultimate outcome of the system.

8. 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.

9. REFERENCES

[1] BALMER, M., NAGEL, K., AND RANEY, B. Large–scale multi-agent simulations for transportation applications. *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations 8*, 4 (2004), 205–221.

[2] BAZZAN, A. L. C. A distributed approach for coordination of traffic signal agents. *Autonomous Agents and Multiagent Systems 10*, 1 (March 2005), 131–164.

[3] BAZZAN, A. L. C., DE OLIVEIRA, D., KLÜGL, F., AND NAGEL, K. Adapt or not to adapt – consequences of adapting driver and traffic light agents. In *Adaptive Agents and Multi-Agent Systems III*, K. Tuyls, A. Nowe, Z. Guessoum, and D. Kudenko, Eds., vol. 4865 of *Lecture Notes in Artificial Intelligence.* Springer-Verlag, 2008, pp. 1–14.

[4] BAZZAN, A. L. C., AND KLÜGL, F. Case studies on the Braess paradox: simulating route recommendation and learning in abstract and microscopic models. *Transportation Research C 13*, 4 (August 2005), 299—319.

[5] BOTTOM, J., BEN-AKIVA, M., BIERLAIRE, M., AND CHABINI, I. Generation of consistent anticipatory route guidance. In *Proceedings of TRISTAN III* (San Juan, Puerto Rico, June 1998), vol. 2.

[6] CHARYPAR, D., AND NAGEL, K. Generating complete all-day activity plans with genetic algorithms. *Transportation 32*, 4 (2005), 369–397.

[7] DIAKAKI, C., PAPAGEORGIOU, M., AND ABOUDOLAS, K. A multivariable regulator approach to traffic-responsive network-wide signal control. *Control Engineering Practice 10*, 2 (February 2002), 183–195.

[8] Hafstein, S. F., Chrobok, R., Pottmeier, A., Schreckenberg, M., and Mazur, F. C. A high-resolution cellular automata traffic simulation model with application in a freeway traffic information system. *Comp-Aided Civil Eng 19*, 5 (2004), 338–338.

[9] Hunt, P. B., Robertson, D. I., Bretherton, R. D., and Winton, R. I. SCOOT - a traffic responsive method of coordinating signals. TRRL Lab. Report 1014, Transport and Road Research Laboratory, Berkshire, 1981.

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

[11] Liu, R., Van Vliet, D., and Watling, D. Microsimulation models incorporating both demand and supply dynamics. *Transportation Research Part A: Policy and Practice 40*, 2 (February 2006), 125–150.

[12] Lowrie, P. The Sydney coordinate adaptive traffic system - principles, methodology, algorithms. In *Proceedings of the International Conference on Road Traffic Signalling* (Sydney, Australia, 1982).

[13] Nagel, K., and Schreckenberg, M. A cellular automaton model for freeway traffic. *Journal de Physique I 2* (1992), 2221.

[14] Nunes, L., and Oliveira, E. C. Learning from multiple sources. In *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multi Agent Systems, AAMAS* (New York, USA, July 2004), N. Jennings, C. Sierra, L. Sonenberg, and M. Tambe, Eds., vol. 3, New York, IEEE Computer Society, pp. 1106–1113.

[15] Oliveira, D., and Bazzan, A. L. C. Emergence of traffic lights synchronization. In *Proceedings of the 20th European Conference on Modelling and Simulation (ECMS 2006, ABS Track)* (May 2006), R. Z. Wolfgang Borutzky, Alessandra Orsoni, Ed., ECMS, pp. 572–577.

[16] Papageorgiou, M. Traffic control. In *Handbook of Transportation Science*, R. W. Hall, Ed. Kluwer Academic Pub, 2003, ch. 8, pp. 243–277.

[17] Roess, R. P., Prassas, E. S., and McShane, W. R. *Traffic Engineering.* Prentice Hall, 2004.

[18] Silva, B. C. d., Basso, E. W., Bazzan, A. L. C., and Engel, P. M. Dealing with non-stationary environments using context detection. In *Proceedings of the 23rd International Conference on Machine Learning ICML* (June 2006), W. W. Cohen and A. Moore, Eds., New York, ACM Press, pp. 217–224.

[19] Silva, B. C. d., Junges, R., Oliveira, D., and Bazzan, A. L. C. ITSUMO: an intelligent transportation system for urban mobility. In *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006) - Demonstration Track* (May 2006), P. Stone and G. Weiss, Eds., ACM Press, pp. 1471–1472.

[20] TRANSYT-7F. *TRANSYT-7F User's Manual.* Transportation Research Center, University of Florida, 1988.

[21] van Zuylen, H. J., and Taale, H. Urban networks with ring roads: a two-level, three player game. In *Proc. of the 83rd Annual Meeting of the Transportation Research Board* (January 2004), TRB.

[22] Wardrop, J. G. Some theoretical aspects of road traffic research. In *Proceedings of the Institute of Civil Engineers* (1952), vol. 2, pp. 325–378.

[23] Wiering, M. Multi-agent reinforcement learning for traffic light control. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000)* (2000), pp. 1151–1158.