View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Institute of Transport Research:Publications

Simulation of modern Traffic Lights Control Systems using the open source Traffic Simulation SUMO

Daniel Krajzewicz, Elmar Brockfeld, Jürgen Mikat, Julia Ringel, Christian Rössel, Wolfram Tuchscheerer,
Peter Wagner, Richard Wösler
German Aerospace Centre, Institute for Transportation Research
Rutherfordstr. 2
12489 Berlin
Germany
E-mail: Daniel.Krajzewicz@dlr.de, Elmar.Brockfeld@dlr.de, Jurgen.Mikat@dlr.de, Julia.Ringel@dlr.de,
christian.roessel@gmx.de, Wolfram.Tuchscheerer@dlr.de, Peter.Wagner@dlr.de, Richard.Woesler@dlr.de

KEYWORDS

Microscopic traffic simulation, open source, traffic lights, traffic research

ABSTRACT

Within the project "OIS" (optical information systems) new traffic control mechanisms had to be invented and tested. One of the most important topics was to optimize the flow over a junction using information from the OIS sensors which can not be measured using normal sensors such as induct loops. For this purpose, an "agentbased" traffic lights logic algorithm was used, which uses the length of a jam in front of a traffic light as input. As we had no possibility to test the traffic lights control within the reality, the improvement of the flow throughput of such junctions was shown using the open source traffic Simulation "SUMO" (Simulation of Urban MObility) [1, 2].

This publication describes the algorithm itself and how it was embedded within the simulation. Furthermore, the simulation results are given.

INTRODUCTION

Simulations are often used to test new systems before implementing them into the real world. This also counts for traffic simulations. Within a project which has been done in a co-operation with universities and a number of companies from Berlin and Brandenburg, new optical sensors have been developed (see [3, 4]). Using digital cameras, we are able to observe traffic and gain areal information, including the length of jams on a street or trajectories of vehicles. Besides developing these systems themselves, another goal of the project was to invent mechanisms which use such information for traffic optimization.

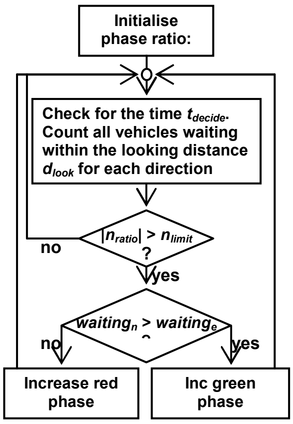
To show the capability to improve traffic flow, we decided to implement our agentbased algorithm (see [5] for a description) for traffic lights controls within the microscopic road traffic simulation "SUMO" ([1, 2]). SUMO is an open source traffic simulation developed at our institute. As the OIS-system has been tested within a certain area around our institute's place, we have used a digital network representing it as our simulation scenario.

The OIS sensors themselves will be not discussed, herein, but only their simulation and the simulation of the traffic lights control algorithm. At first, this algorithm will be presented, followed by a description of how the simulation was prepared. We will then give the simulation results, followed by some conclusions and a discussion of open questions.

THE TRAFFIC LIGHTS CONTROL ALGORITHM

The agentbased traffic lights control algorithm was firstly presented in [5]. The main idea is that each traffic light is trying to solve the jams in his front by itself. To achieve this, he looks into the incoming lanes and measures the jam lengths on these lanes. If at one of these lanes the jam gets longer, this lane gets green for a longer time. Beside these assumptions, several parameters prevent the system from oscillating and from adapting too fast or too strong. This is done by increasing a green phase's duration only if a jam is longer than a threshold. Furthermore, the jam has to occur for a certain amount of time. There are further boundaries for the duration of a phase – beside the standard value given at the begin, a phase must not be longer or shorter than predefined thresholds. The whole algorithm is shown within picture 1.

Beside the advantage to be very simple, the agentbased traffic lights logic can be set on top of existing traffic lights and tries to adapt them to the current traffic amount.


Initialise phase ratio:
Check for the time t_{decide}. Count all vehicles waiting within the looking distance d_{look} for each direction
$|n_{ratio}| > n_{limit}$?
no
yes
$waiting_n > waiting_e$
no
yes
Increase red phase
Inc green phase

***tr* , *tg* : red, green phase proportion**
rph* = *tr* / *tg
***tcycle* = *tr* + *tg* : cycle time**
***dlook* : looking distance**
***tdecide* : decision time interval**
nratio* = (*waiting n* - *waiting e*) / *waiting n
***n* : northbound**
***e* : eastbound**
***nlimit* : decision threshold**

Picture 1: The traffic lights control algorithm

SIMULATION PREPARATION

Setting Up The Scenario

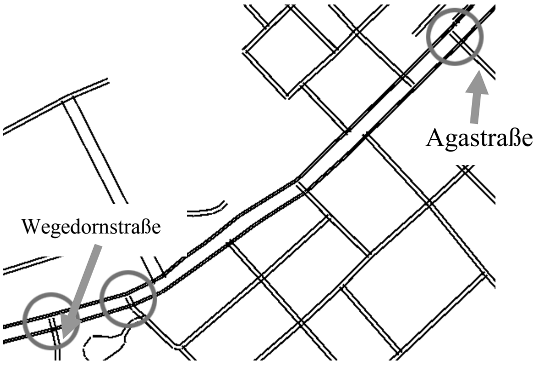
The description we used was a database containing the road network of Berlin, converted from a digital map supplied by NavTech. To extract the area we wanted to use, we had to extract all nodes (junctions) and edges (streets) within a certain rectangle. This was done straightforward using simple SQL-queries.

For the final presentation, two situations had to be shown simultaneously, one showing the scenario running with normal traffic lights as they are implemented in reality, and one using the OIS-detectors. To achieve this, we have duplicated the lists of edges and nodes, shifted them by several hundred meters – the width of the scenario plus an offset – and included this second list within the first one. These lists were then converted into XML files which the SUMO network converter is capable to read. Doing this, we gained two networks combined into a single which could be used for simulating the scenario.

Furthermore, traffic lights definitions computed by the “NETCONVERTER” had been replaced by definitions that exactly match the real world traffic light logics. This has been done for both parts of the network – the one using normal traffic lights logics and the one using agentbased traffic lights logics.

The routes were retrieved from measures of the real world scenario area. Picture 2 shows the junctions the traffic was counted at. From these measures, junction turning percentages for each of the participating, incoming edges were computed. These were then fed, together with the traffic amount to one of SUMO’s routing modules for generating single vehicle trips.

Of course, the routes had to be duplicated for the second network, too, in order to gain exactly the same flows for both parts of the network.



Picture 2: Positions of the OIS-controlled junctions within the simulation (marked by circles)

Implementing The OIS Sensors

The OIS sensors were simulated using areal detectors (see [2] for a further description) which look at all lanes in front of the junctions which are meant to be equipped with OIS sensors. Beside the jam length in meters, other values are collected, such as the jam length in vehicles, the number of vehicles, the occupancy degree, the vehicles’ mean speed, the halting durations, and several more.

Implementing The Traffic Lights Control

The traffic light control was implemented by extending SUMO’s representation of normal traffic lights logics (done using derivation in C++). The application’s interface was changed in a way which allows the traffic light to extend its phases. Besides implementing the traffic lights control themselves this way, some further changes had to be done. Among them was the addition of the ability to change the traffic lights parameter on application start and to read additional information about the phases’ minimum and maximum lengths from the network description.

Below, the algorithm that is executed every time the light switches is given:

```
- find the maximum queue length for the lanes that have green light during the current phase
- save this queue length into a list A which holds such values for the last l_h cycles (remove past values if needed)
- if the time after the last decision > t_decide:
    - compute the mean queue length for all phases by averaging the values within list A with the length of list A
    - compute the phase with the largest queue in front
    - compute the phase with the smallest queue in front
    - quit computation if either the phase with the largest queue in front may not be lengthen or the phase with the shortest queue in front may not be shorten
    - compute n_ratio:
        - n_ratio = (max queue length – min queue length) / max queue length
    - if n_ratio > n_limit:
        - increment phase length of the phase with the largest queue length by one
        - decrement phase length of the phase with the shortest queue length by one
```

With:

l_h:	learn horizon
t_decide:	decision time interval (see Pic. 1)
n_ratio:	normalised queue length delta between the longest and the shortest queue
n_limit:	decision threshold (see Pic. 1)

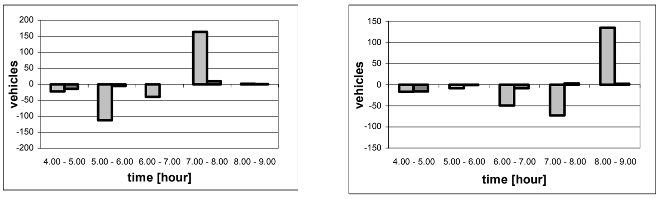
SIMULATION RESULTS

The comparisons were created by writing detector measures into files and evaluating them after the simulation has ended. For this purpose, areal detectors, the same as used by the agentbased traffic lights logics, were laid in front of the junction. Two measures were used to compare the traffic lights controls: a) the jam in front of them and b) the throughput of the junctions. The second was computed by summing up the number of vehicles that leave the junction and subtracting the number of vehicles that approach the junction.

While using the original traffic amount, almost no difference between the OIS controlled and normal traffic lights controls could be observed. For this reason a further scenario has been implemented where the vehicle flow on one of the incoming edges was incremented for the time between 5am and 5.30am. After doing this, the configurations showed large differences, which prove the improvement of the junctions' phases when areal sensors combined with the agentbased traffic lights control are used.

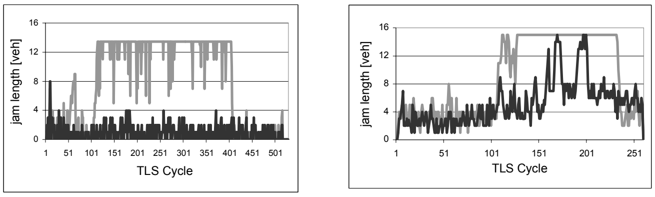
Below, comparisons for two of the three regarded junctions are given, using the junctions' throughput (picture 3) and the jam lengths in front of the junctions (picture 4). We will now discuss it, briefly.

As the amount of approaching vehicles increases, a normal junction (shown in light grey within pictures 4 and 5) is not "prepared" to solve all incoming vehicles. Most of them get stucked in front of the junction. Due to this, there are more approaching than leaving vehicles and the "throughput" gets negative. After the additional flow has been inserted into the network, the number of vehicles that try to pass the junction decreases and the vehicles that were waiting in front of it are leaving the junction consecutively. This is the reason for the high positive peak in the normal junctions' throughput. As one can see, there are neither positive nor negative peaks within the agentbased junctions (shown in dark grey), what shows that such junctions are capable to solve the additional demand.



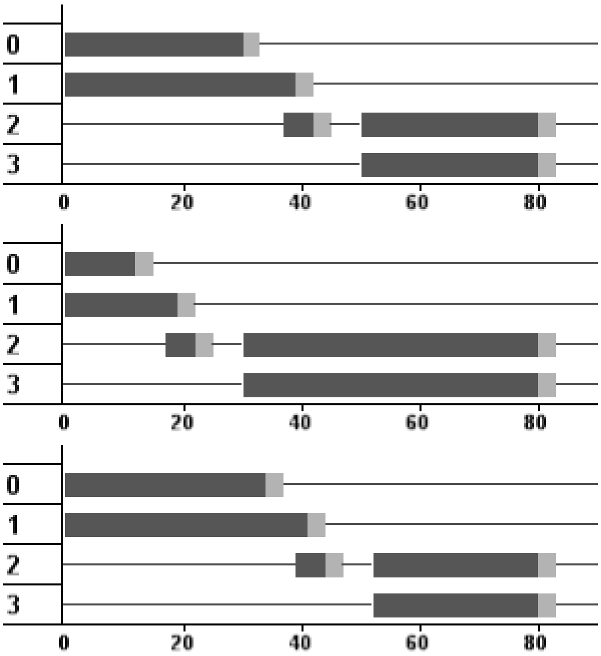
Picture 3: Comparison of the throughput of fixed and agentbased traffic lights logics; light: throughput of fixed tls, dark: throughput of agentbased tls; left: junction Agastraße, right: junction Wegedornstraße

This ability is even more visible when the jams in front of the traffic lights are examined (picture 4). Here, one can see that jams are much longer in front of normal traffic lights when compared with agentbased traffic lights. The constant maximum in front of normal traffic lights is due to a certain length of the detectors that were used to compute the jam lengths.



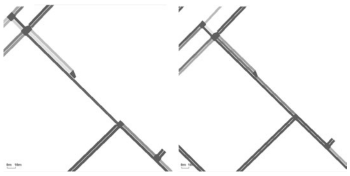
Picture 4: Comparison of the jam lengths in front of fixed and agentbased traffic lights logics for a certain lane; light: throughput of fixed tls, dark: throughput of agentbased tls; left: junction Agastraße, right: junction Wegedornstraße

Picture 5 shows the difference in the traffic lights logic located at the bottom of the rightmost junction from picture 2. One can see here how agentbased traffic light logics adapt to an increased flow and how they readapt to a normal situation as soon as the queues in front of it get similar long.



Picture 5: Comparison of the traffic lights logic before (top) and after the optimization (middle); the display at the bottom shows the phases after the increased flow has passed

Besides collecting the values as shown above, a visualisation component was developed which allows viewing the aggregated values over all lanes. Picture 6 shows the situation at about 6am comparing a normal adaptive traffic lights logic and the agentbased logic equipped with OIS sensors. One can see that the jam, displayed by the luminance is much longer when the normal traffic lights logic is used.



Picture 6: Comparison of the density on the streets when using normal (left) and agentbased traffic lights logics (right); the lighter the street, the higher the density

This increased throughput is what is wanted, but it also contains traps. The simulation shows that the succeeding traffic lights may not be able to solve the greater incoming amount of vehicles. If this happens, the system's jam length may even increase, although locally the throughput of a junction is improved.

CONCLUSIONS

Although the agentbased traffic lights logics show clearly their benefits at certain circumstances, no improvement could be achieved as long as the flows were small and balanced between the different directions of a traffic light.

This means that the best place for such controls in real life would be junctions where the flow is changing tremendously from time to time, for example in places near to venues.

A second result is that we have to investigate the interrelationship between consecutive traffic lights to avoid generation of larger jams in front of succeeding traffic lights. Without such coordination mechanisms, agentbased traffic lights are most effective when used solely, in a larger distance to other traffic lights.

At last, we want to point out that this research would cost a much greater effort when a commercial simulation application had to be used. Especially the implementation of new on-road systems would not be possible or at least very time consuming as long as the simulation's source code is not open. That's why we want to encourage you to take a look at SUMO's project pages located at http://sumo.sourceforge.net/ and try out the software.

Unfortunately, we are not able to make the scenario downloadable, because it is based on commercial NavTech data.

FUTURE WORK

We have seen that some further research has to be done on coordinating the lights. A mechanism for this is not yet designed and should be done as next.

There are also further systems to be evaluated. Currently, we develop a system which reidentifies vehicles that pass a set of camera-equipped junctions (see [6] for a description). Among other things, travel times are computed from these reidentified cars. These travel times are valuable data, e.g. for further optimization of traffic light control. Such traffic improvement methods, could be simulated, too, and may also be one of the further steps.

REFERENCES

[1] D. Krajzewicz, G. Hertkorn, C. Rössel, P. Wagner. 2002. "SUMO (Simulation of Urban MObility); An open-Source Traffic Simulation" Proceedings of the 4th Middle East Symposium on Simulation and Modelling (MESM2002), Edited by: A.~Al-Akaidi, pp. 183 - 187, SCS European Publishing House, ISBN 90-77039-09-0

[2] D. Krajzewicz, G. Hertkorn, C. Rössel, P. Wagner. 2002-2005. SUMO Homepage. http://sumo.sourceforge.net

[3] http://www.dlr.de/vf/forschung/projekte/ois

[4] http://www.fav.de

[5] J. Mikat, E. Brockfeld, P. Wagner. 2003. "Agent Based Traffic Signals on a basic grid". Proceedings of the 4th Workshop on Agent-Based Simulation. Edited by: Jean-Pierre Müller, Martina-Maria Seidel. SCS European Publishing House.

[6] R. Woesler. 2004. Real-time recognition and reidentification of vehicles from video data with high reidentification rate, Proceedings of the 8th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2004), Orlando, Florida, USA. Edited by: Nagib Callaos, William Lesso, Ashraf Ahmad, Vol. VI, pp. 347-352, International Institute of Informatics and Systemics, ISBN 980-6560-13-2

BIOGRAPHY

Born in Bydgoszcz, Poland, 1972, **Daniel Krajzewicz** has finished his study of computer science at the Technical University in Berlin by the middle of the year 2000 with artificial intelligence and computer graphics as main topics. After work on text classification he changed to the Institute for Transportation Research of the German Aerospace Centre in 2001 where he now works on a cognitive driver model and the open-source urban traffic simulation "SUMO".