Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 130 (2018) 894–899

Procedia Computer Science

www.elsevier.com/locate/procedia

9th International Conference on Ambient Systems, Networks and Technologies, ANT-2018 and the 8th International Conference on Sustainable Energy Information Technology, SEIT 2018, 8-11 May, 2018, Porto, Portugal

Implementing an adaptive traffic signal control algorithm in an agent-based transport simulation

Nico Kühnel[a,*], Theresa Thunig[b], Kai Nagel[b]

[a] Technical University of Munich, Department of Civil, Geo and Environmental Engineering, Arcisstr. 21, 80333 Munich, Germany
[b] Technische Universität Berlin, Transport Systems Planning and Transport Telematics, Salzufer 17-19, 10587 Berlin, Germany

Abstract

This paper describes the implementation of the fully traffic adaptive signal control algorithm by Lämmer[2] in the agent-based transport simulation MATSim. The implementation is tested at an illustrative, single intersection scenario and compared to the results of Lämmers MATLAB simulation. Plausibility of the self-controlled signals and overall results can be confirmed. Small deviations can be explained by differences in flow simulation and resolution of simulation time steps. In the simulation of the illustrative intersection, the adaptive control is proved to be stable and, overall, superior to a fixed-time control. Constant vehicle arrivals are simulated to show the performance of the control and its underlying sub-strategies. The expected behavior of the algorithm and its implementation are validated by analyzing queue lengths over time. The adaptive control significantly outperforms the fixed-time control for stochastic demand, where its ability to dynamically react to changes in flow becomes important.

Keywords: adaptive traffic signal control; self-controlled traffic signals; agent-based transport simulation; MATSim

1. Introduction

In contrast to fixed-time signals that cyclically repeat a given signal plan, adaptive traffic signals dynamically decide for signal states based on real-time information on traffic conditions. They can, therefore, react to changes in demand and reduce emissions and waiting times more efficiently. A variety of adaptive traffic signal control algorithms have been developed. An overview is e.g. given by Friedrich[1]. This paper addresses the adaptive control algorithm developed by Lämmer[2,4] and describes the implementation of his self-controlled signals in the agent-based transport simulation MATSim. Unlike other microscopic simulations like VISSIM, where this adaptive control has already been implemented[3], MATSim offers the possibility to analyze long-term reactions of travellers to their environment including traffic signals. As a first step, the adaptive signal control is implemented in MATSim to reproduce and verify the results of Lämmer[2]. It is tested with a simple case study – a single intersection with constant and stochastic arrivals. Lämmer[2] applied his algorithm to a similar case study in a MATLAB simulation, such that results are comparable. The

* Corresponding author. Tel.: +49-89-289-28598
E-mail address: nico.kuehnel@tum.de

1877-0509 © 2018 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of the Conference Program Chairs.
10.1016/j.procs.2018.04.086

paper is organized as follows. Lämmers self-controlled signals and the transport simulation MATSim are described in section 2 and 3. The illustrative scenario and the evaluation of the adaptive signal control is presented in section 4. Section 5 concludes and brings up further research.

2. Self-controlled signals

The control algorithm presented and implemented in this paper was developed by Lämmer[2,4] in 2007. Since this initial approach, the algorithm has been further developed and is already tested in the field[5]. Lämmers self-controlled signals locally optimize signal settings based on sensor-data, whereas a decentralized supervision ensures stability throughout the network. Similar to other approaches, the main goal is to minimize waiting times of all vehicles.

2.1. Assumptions

Lämmer assumes a queue-representation of traffic flow. If a link i is served, vehicles can leave the link with a constant outflow rate q_i^{max}. These outflow rates as well as the average arrival flow rates $\bar{q}_i$ are assumed to be known. Additionally, queues are assumed to be non-spatially, i.e. vehicles do not spill back to upstream lanes or links. Demand is supposed to be manageable on average with the desired cycle time T to ensure stability. Another assumption of Lämmers signal control is, that all links are in conflict to each other, i.e. each link has to be served in a separate signal stage. Intergreen times τ^0 (i.e. all-red times between green phases) only depend on the next served signal with no regard to the current active signal.

2.2. Sensors

Two sensors are used to predict the number of waiting vehicles per link and time. One is positioned at the end of the link to detect waiting and outflowing vehicles; the second one is located further upstream to detect approaching vehicles. Assuming free flow conditions at link i, one can estimate the length of the queue $n_i(t)$ at time t and predict the expected queue length $\hat{n}_i(t, \tau)$ at a time $\tau > t$. While the estimation of queue lengths allows uncertainty, the mere presence of a queue is definite. Given $\hat{n}_i(t, \tau)$ and the maximum outflow rate q_i^{max} for link i, one can derive the expected required green time $\hat{g}_i(t, \tau)$ for clearing the queue at time t using $\hat{g}_i(t, \tau) = \frac{\hat{n}_i(t,\tau)}{q_i^{max}}$.

2.3. Optimizing Regime

The core of the traffic-adaptive behavior is depicted by the optimizing regime which aims to minimize waiting times. In each time step, it cyclically estimates priority indices π_i for each signal i (which belongs to link i) and selects the signal $\sigma(t)$ with the highest priority index to be served next.

The priority index of a signal i is calculated using the equation

$$\pi_i(t) = \begin{cases} \max\limits_{\tau_i(t) \le \tau \le \tau_i^0} \frac{\hat{n}_i(t,\tau)}{\tau + \hat{g}_i(t,\tau)} \,, & \text{if } i = \sigma(t) \\ \frac{\hat{n}_i(t,\tau_i^0)}{\tau_{\sigma(t)}^{pen}(t) + \tau_i^0 + \hat{g}_i(t,\tau_i^0)} \,, & \text{if } i \neq \sigma(t) \,. \end{cases} \tag{1}$$

Two cases are distinguished depending on whether the signal i is already selected or not. In either case, the equation basically divides the number of vehicles by the time needed to clear the queue including the (remaining) intergreen time. The priority index could, therefore, be interpreted as a clearance efficiency rate. If the signal is selected (i.e. $i = \sigma(t)$), the priority indices of multiple time horizons are compared depending on the remaining intergreen time $\tau_i(t)$ that needs to pass until the signal can be switched to green. Once the signal shows green, $\tau_i(t)$ equals zero which results in $\pi_i(t) = \frac{\hat{n}_i(t,0)}{0+\hat{g}_i(t,0)} = q_i^{max}$ if $\hat{n}_i(t, 0) > 0$. This describes the fact that vehicles leave a link at its maximum outflow rate, once the signal turns green. This is also the highest possible priority index a link can get.

If a signal is not active, the priority index is reduced by a canceling penalty $\tau_{\sigma(t)}^{pen}(t)$. This prevents the optimizing regime from frequently switching signals which would cause a high sum of intergreen, i.e. lost time. The penalty can be interpreted as the average additional waiting time for vehicles at the previously served link that would occur upon cancelation.

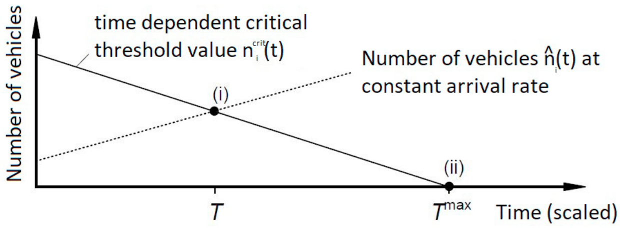

Fig. 1. Critical threshold value n^{crit} over time.[2]

2.4. *Stabilizing Regime*

Under the assumptions formulated in section 2.1, the optimizing regime minimizes waiting times. However, due to the fact that the priority index is closely related to the maximum outflow rate q_i^{max} of a link, which is also the highest possible index value, stability can break down under high traffic loads. A minor road with low maximum outflow rate is not able to exceed the priority index of a major road under overloaded conditions.

The stabilizing regime solves this problem, by ensuring a maximum waiting time at a signal. Therefore, a stabilizing queue Ω is used. A signal i is added to Ω, if the number of waiting vehicles $\hat{n}_i(t)$ at time t is greater than a threshold value n_i^{crit}. This threshold value reduces over time when vehicles are waiting and the signal is not served (see figure 1). After a maximum cycle time $T^{\max}$, a single vehicle can trigger stabilization. If vehicles constantly arrive with the average expected flow rate $\bar{q}^{exp}$, the signal is stabilized once during the desired cycle time T.

Signals in the stabilization queue Ω are served first-in-first-out: In each time step t, the first element of the queue is stabilized, i.e. selected and served for a guaranteed green time g_i^s. Once the guaranteed green time has passed or the waiting queue at the signal has cleared, the signal is switched to red and removed from the queue. If the stabilization queue is empty, no signal is selected. The guaranteed green is a fraction of T and depends on the average occupancy rate $\bar{\lambda}_i$ of link i. The remaining part of T is distributed among the approaches, according to their outflow rate q_i^{max}.

2.5. *Combined Algorithm*

For Lämmers self-controlled signals, the two strategies are combined to achieve both – optimality and stability. Therefore, the optimizing regime is used whenever the stabilization queue Ω is empty. Once a signal is inserted into Ω, the stabilizing regime takes over. The combined logic can be expressed as

$$\sigma(t) = \begin{cases} \text{head } \Omega, & \text{if } \Omega \neq \emptyset \text{ (stabilizing Regime)}, \\ \arg\ \max_i \pi_i(t), & \text{else (optimizing Regime)}. \end{cases} \tag{2}$$

3. MATSim

Lämmers signal control is implemented in the agent-based transport simulation MATSim[6]. In MATSim, the population is modeled as agents with daily plans. Plans consist of activities with start and end times, and trips between activities that specify which mode of transport to use and which route to take. These plans are executed by the mobility simulation (see *mobsim* in figure 2). Traffic is modeled by spatial queues representing streets. Vehicles at the head of a queue can leave the link, when their earliest exist time (depending on the links free flow travel time) has past, link flow capacity is fulfilled, and enough space is left on the next link (i.e. spill back effects are modelled). With this, plans can get delayed which is evaluated by the *scoring* module. Some agents are allowed to adjust their plan, e.g. their route, departure time, or mode of transport (*replanning*). The co-evolutionary process of execution, evaluation and replanning is repeated iteratively to achieve a state where no agent has an incentive to unilaterally chance his plan – a (stochastic) user equilibrium.

The simplistic representation of traffic flows allows MATSim to simulate large-scale scenarios in reasonable time. MATSim is written in Java, developed open-source[7], and offers various extension points to plug in custom functionality – the so called *contribs*. For this work, the *signals* contrib is used, to simulate traffic lights. Traffic signals

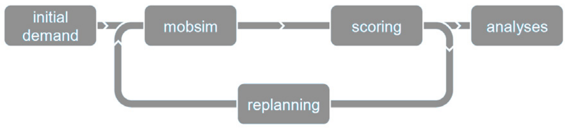

Fig. 2. Iterative process of the transport simulation MATSim

in MATSim constitute an additional requirement for leaving a link queue: The signal has to show green. For each signalized intersection a signal system is installed. It consists of multiple signals that control one or more links or lanes. Fixed-time signals in MATSim can easily be modeled by following a given signal control. To model traffic-adaptive signals, a signal controller has to be implemented that is called every second of the simulation to decide about all signal states of a signal system. Therefore, link and lane detectors are modeled that listen to the queue-simulation and can be queried by the signal controller to obtain current and predicted queue lengths. There have been previous implementations of traffic-adaptive signals in MATSim. Grether[8] implemented the SYLVIA control, which uses predefined signal programs that can be adjusted to the current traffic situation by extending green times. Applied to a real-world scenario with high, unexpected traffic demand, the traffic-actuated control showed significant improvements in reducing waiting times when compared to the default plan-based signal control[9].

4. Evaluation of the implemented signal control

The implemented signal controller is evaluated by comparing its behavior with the results of Lämmers MATLAB simulations[2]. Therefore, Lämmers test scenario is rebuilt in MATSim. Queue lengths for different occupancy rates are analyzed and compared to the results of Lämmers thesis. Besides testing the combined algorithm, both individual strategies are simulated on their own. A fixed-time control is used as an additional comparison. Similar to the self-controlled signal, it only serves one signal at a time.

The test case constitutes a single two-road crossing. The major road consists of two lanes with a total flow capacity of 3600 veh/h. With one lane per direction, the flow capacity of the minor road is 1800 veh/h. Each approach of the roads is controlled by a signal. The fixed-time cycle length and the desired cycle time T for the self-controlled signals are both set to 120 s. The maximum cycle time T^{max} for the stabilizing regime is set to 180 s.

Intergreen times τ^0 are set to 5 s for every signal, and green times for the fixed-time control are distributed relatively to the respective link's occupancy. In contrast to Lämmers original setup with time steps of 0.5 s, the MATSim scenario uses a time step size of 1 s.

While the demand on the minor road is constant with $q^{exp}_{minor} = 180$ veh/h per direction, the demand on both directions of the major road rises from $[0 \dots 1440]$ veh/h. With this, the occupancy rates of the minor road are fixed as $\bar{\lambda}_{minor} = 0.1$; the occupancy rates of the major road vary from $\bar{\lambda}_{major} = [0 \dots 0.4]$. This implies a total intersection occupancy of $\Lambda = [0.2 \dots 1.0]$.

4.1. Constant Arrival Times

At first, the scenario is run with constant arrival times. Figure 3 shows the results of the MATSim simulation compared with Lämmers results. In each diagram, the average sum of queue lengths for fixed-time control and combined, optimizing and stabilizing self-controlled signals are compared. As expected, the average sum of queue lengths increases with overall occupancy. All strategies become unstable at occupancy rates of 0.833, as this violates the stability condition mentioned in section 2.1 (see Lämmer[2] for respective equations).

In both simulation tools, the stabilizing strategy performs worse than the other strategies for most occupancy rates. But it stays stable for occupancy rates lower than 0.833. The fixed time control performs better than the stabilizing strategy and is mostly worse than the combined strategy. It also stays stable until the occupancy rate of 0.833, where an average of 30 vehicles are queued up at each time step. The optimizing strategy results in smaller queues than the fixed-time and stabilizing strategy for lower occupancy rates but becomes unstable at medium rates around $\Lambda = 0.6$. For most parts, the combined algorithm leads to the best performance in terms of queue lengths. For low occupancies, it is congruent with the optimizing strategy. Instead of becoming unstable, the stabilizing part in the combined

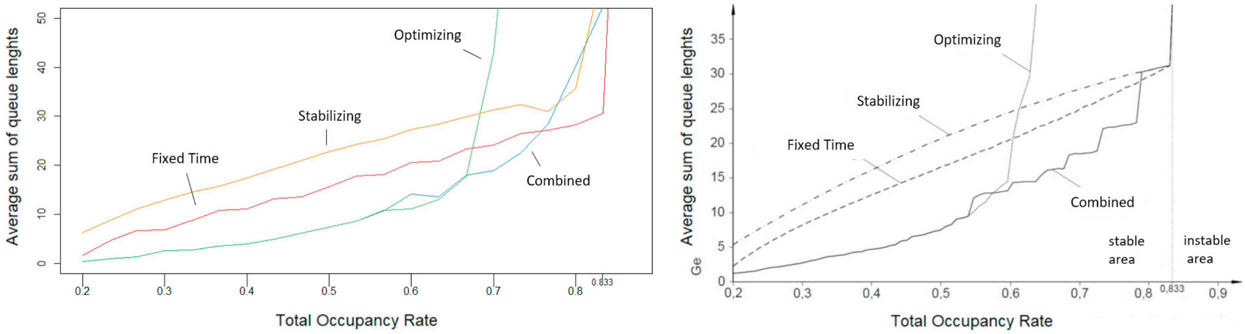

Fig. 3. Simulation results for constant arrival times: MATSim (left), MATLAB simulation by Lämmer[2] (right)

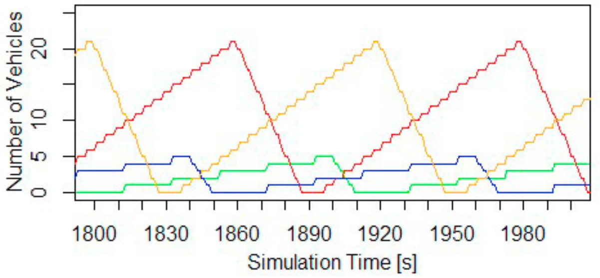

Fig. 4. MATSim simulation result for the single intersection scenario with constant arrival times: Course of queue lengths in time (red/orange: major roads; blue/green: minor roads)

algorithm leads to a stable course of queue lengths until the maximum occupancy rate of 0.833. In both simulations, combined and optimizing strategy diverge at occupancies between 0.5 and 0.6.

As the simulation results are deterministic, differences between the MATLAB simulation and MATSim cannot be explained by stochastic variations. One explanation could be the different resolution of simulation time steps. As some demand values lead to non-discrete time intervals between agent departures, e.g. 3.33 s, constant arrival times periodically change from four to three second time intervals in MATSim, as only discrete time steps are possible. This potentially affects the adaptive behavior especially in high demand situations, where forecasting and calculation on time is crucial. Also, all calculations and predictions are only executed in one second time steps, which also leads to a coarser resolution.

To analyze the temporal behavior of the signal algorithm, queue lengths of a simulation with fixed demand have been analyzed over time. Figure 4 shows vehicle queues for each link from second 1800 to 2000 of the simulation for a total occupancy rate of 0.7 (flow of 900 veh/h on major roads), at which the stabilizing regime is already intervening. One can clearly identify a cyclic profile in the service of signals. As the critical threshold value is defined to trigger a service once during T (assuming an average flow rate, see figure 1), the cyclic profile is regularly repeated in constant arrival situations. Looking e.g. at the blue line, it can be seen that the link is served around second 1840, having five vehicles in the queue. About 120 seconds later, at second 1960, the system is in the same state. This confirms the theory of the algorithm.

4.2. Stochastic Arrival Times

Secondly, the scenario is run with stochastic arrival times. The demand is inserted in exponential distributed time gaps. The time gap between vehicle platoons is the platoon size divided by flow, with the platoon size also being exponential distributed with an expected value of 5. During simulation, it is ensured that demand per link does not exceed q_i^{max}. For this setup, only fixed-time control and the combined self-controlled signals are compared.

Figure 5 shows the averaged result of 30 simulation runs for each occupancy. Results vary, as the simulation of stochastic arrivals is no longer deterministic. The fixed-time control shows significant higher variances in queue

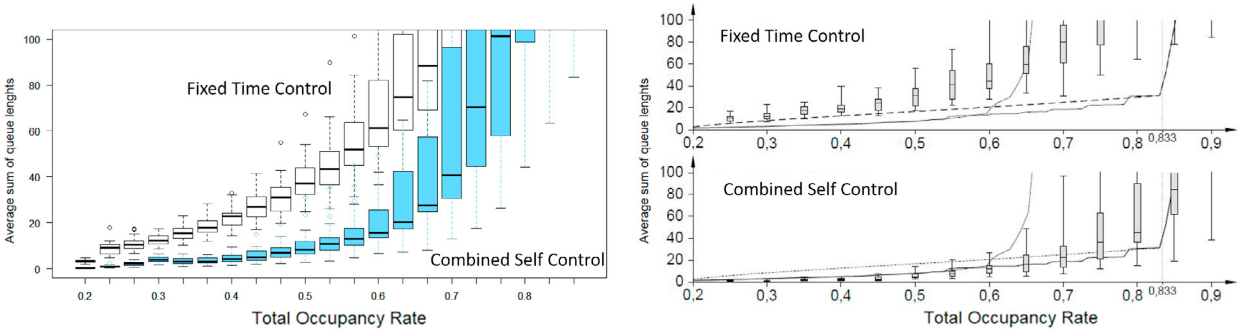

Fig. 5. Simulation results for stochastic arrival times: MATSim (left), MATLAB simulation by Lämmer[2] (right)

lengths than the self-controlled signals. This is especially true for small occupancies where the adaptive control can react more flexible. Also, overall queue lengths are considerably smaller for the adaptive control. The results show that the positive impact of self-controlled signals is higher in stochastic situations, that are also more likely in reality.

5. Conclusion

The presented work shows that agent-based transport simulations are a suitable tool to model and evaluate adaptive signals. The simulation results confirm the positive effects of Lämmers self-controlled signals. Drawback of this first implementation (and, also, of the algorithm described by Lmmer[2]) is the lack of transferability to more realistic scenarios, especially because of the assumption that all approaches have to be served in separate signal stages. Also, minimum green times and overload situations that break the assumptions given in section 2.1 are not captured by the current algorithm. An extension of the algorithm that is applicable to real-world scenarios, is already under development in MATSim. This will allow to analyze long-term effects of de-centralized, delay-minimizing traffic-adaptive signals on user-behavior, e.g. travel time in the user equilibrium.

Acknowledgements

The authors wish to thank Stefan Lämmer for encouraging research on this topic and for all discussions about his self-controlled signals. The authors also thank the Deutsche Forschungsgemeinschaft (DFG) for funding the project *Optimization and network wide analysis of traffic signal control*.

References

1. B. Friedrich, Adaptive signal control – an overview, in: Proc. of the 9th Meeting of the Euro Working Group Transportation, Bari, Italy, 2002.
2. S. Lämmer, Reglerentwurf zur dezentralen Online-Steuerung von Lichtsignalanlagen in Straßennetzwerken, Ph.D. thesis, Dresden Institute of Technology (2007).
3. S. Lämmer, D. Helbing, Self-stabilizing decentralized signal control of realistic, saturated network traffic, Working Paper 10-09-019, Santa Fe Institute, Santa Fe (2010).
4. S. Lämmer, D. Helbing, Self-control of traffic lights and vehicle flows in urban road networks, Journal of Statistical Mechanics: Theory and Experiment 2008 (04) (2008) 04–019. doi:10.1088/1742-5468/2008/04/P04019.
5. S. Lämmer, Die Selbst-Steuerung im Praxistest, Straßenverkehrstechnik 3 (2016) 143–151.
6. A. Horni, K. Nagel, K. W. Axhausen (Eds.), The Multi-Agent Transport Simulation MATSim, Ubiquity, London, 2016. doi:10.5334/baw. URL `http://matsim.org/the-book`
7. MATSim webpage, MultiAgent Transport Simulation (accessed 2016). URL `http://www.matsim.org`
8. D. S. Grether, Extension of a multi-agent transport simulation for traffic signal control and air transport systems, Ph.D. thesis, TU Berlin, Berlin (2014). arXiv:urn:nbn:de:kobv:83-opus4-53257.
9. D. Grether, J. Bischoff, K. Nagel, Traffic-actuated signal control: Simulation of the user benefits in a big event real-world scenario, in: 2nd International Conference on Models and Technologies for ITS, Leuven, Belgium, 2011, also VSP WP 11-12, see `http://www.vsp.tu-berlin.de/publications`.