Working Paper

Modeling crossroads in MATSim
The case of traffic-signaled intersections

Author(s):
Sallard, Aurore; Balać, Milos

Publication Date:
2020-08

Permanent Link:
https://doi.org/10.3929/ethz-b-000429954

Rights / License:
In Copyright - Non-Commercial Use Permitted
Modeling Crossroads in MATSim: the Case of Traffic-Signaled Intersections

Aurore Sallard
IVT
ETH Zurich
CH-8093 Zurich
aurore.sallard@ivt.baug.ethz.ch

Miloš Baláč
IVT
ETH Zurich
CH-8093 Zurich
milos.balac@ivt.baug.ethz.ch

Word Count: 7459 words + 0 table(s) \times 250 = 7459 \text{ words}

Submission Date: August 1, 2020
ABSTRACT

In transportation simulation, the prediction accuracy of travel times on road segments can have substantial impacts on the simulation outcomes. The travel times are impacted, among other things, by traffic signals. Modeling traffic signals is not straightforward in large scale simulations, especially when data on their characteristics is not available. Due to this, in most of the applications using the agent-based transport simulation software MATSim, traffic signals are not explicitly modeled. This paper addresses this issue and proposes a method based on Webster’s formula to compute delays experienced by motorists at traffic-signaled intersections without having access to the actual traffic signal data.

Within this study, results regarding the imputation of road flow capacities and estimating the impacts of congestion and crossroads on travel times will be presented and investigated. Evidence shows that Webster’s approach and road capacity modifications can substantially improve the quality of the travel time estimates for the regional model of Zurich, Switzerland.

Keywords: Traffic lights, crossroads, MATSim, Webster’s formula
INTRODUCTION

The goal of transportation simulation is to computationally reproduce travel conditions within a given area, in order to assess the effects of future policies, investments, and, more generally, changes in the infrastructure. For this purpose, it is of primary importance to calculate travel times as accurately as possible, because those strongly impact essential simulations’ results, such as transport mode choice or destination choice. Travel times obviously depend on the length of the chosen routes and of the speed limits enforced in the network. They are also influenced by other parameters, such as congestion level, road geometry or zebra crossings. Road intersections, as well, cause additional delays to motorists. This is why a special attention has to be paid to the modelling of crossroads before actually running the simulation.

This issue is only partially covered in the default settings of the agent-based transport simulation software MATSim (Horni et al. (1)). By default, no distinction is made in MATSim between the traffic-signaled intersections and the other ones. Additional modules presented in Grether (2) and Thunig et al. (3) offer however the possibility of adding traffic lights to the network and to implement different strategies for controlling the signal schedules. They rely on a detailed description of the intersections and on signal plans that had to be manually collected. Thus, those approaches could be counted as work within the field of microscopic scale simulation, whereas MATSim is usually used for mesoscopic scale applications.

This paper presents a contribution towards another possible way of modeling traffic-signaled intersections within the MATSim framework, which relies on the Webster formula (Webster (4)). This equation computes the optimal (in the sense of minimizing the average delay per motorist) traffic signal cycle length. The presented approach does not model each individual signal phase, but uses the average experienced delay, obtained through the Webster formula, to get more accurate estimates of the travel times. This study also provides a straightforward way of adding traffic lights’ localizations into an existing MATSim network and compares empirical travel times, computed according to this new approach, to estimates obtained from two web APIs.

After presenting with more details the best-known methods for traffic signal scheduling and the work already done within this field in MATSim, the present paper will guide the reader through the different implementation steps of this new approach and provide insights into validation results.

BACKGROUND

Webster’s approach for optimizing the cycle length of traffic signals

One of the first approaches aiming to optimize the traffic signals’ schedule was described by Webster. The chosen minimization criterion is the average vehicle delay, and, in this work, the author develops a formulation for the optimal cycle length derived from equations approximating the expected vehicle delay.

The computation of this average delay was thoroughly studied in Routhail et al. (5). A first exact equation was proposed in Beckman et al. (6); it is valid for fixed-time signals under the
assumption of a binomial arrival process. Knowing the signal cycle length $C$, the effective green-phase length on a given link $g$, the arrival rate $q$ on this link, the departure rate $S$ from the queue during the green phase and $Q_o$, the expected overflow queue remaining after previous cycles, the expected delay $d$ is given by

$$d = \frac{C g}{C \left(1 - \frac{q}{S}\right)} \left(\frac{Q_o}{q} + \frac{1 + C g}{2}\right).$$

Furthermore, efforts have been made to approximate the expected vehicle delay by combining theoretical analysis and numerical simulations in contexts where this exact equation cannot be used, which can be due to several reasons. First, it is difficult to access the exact arrival and departure rates. Secondly, computing the expected overflow queue $Q_o$ can also be a complex problem. Finally, the assumptions underlying the exact formulae do not necessarily cover a large range of real-world situations. The first approximate formula was developed in Webster (4). According to this work, the average delay per vehicle $d$, given in seconds, can be approximated by

$$d \simeq \frac{C}{2} \left(1 - \frac{g}{C}\right)^2 + \frac{x^2}{2q(1-x)} - 0.65 \left(\frac{C}{q^2}\right)^{\frac{1}{3}} x^{2.5} \frac{g}{C},$$

where

1. $C$ is the cycle length in seconds;
2. $g$ is the effective green time reserved to the link on which the vehicle approaches;
3. $x$ is the degree of saturation on this link, which means the ratio of the (hourly) flow on the link to the link capacity;
4. $q$ is the vehicle arrival rate in vehicles per second.

It has to be remarked that, if the first two terms of this expression have a theoretical justification, the last one was designed only so that the theoretical delays fit better with values obtained empirically.

Given this expression, Webster derives from it a formula for the optimal cycle length $C_{opt}$ in seconds:

$$C_{opt} := \frac{1.5 \times L + 5}{1 - Y},$$

where

1. $L$ is the total lost or unusable time during a signal cycle, in seconds. It corresponds to the sum of the time lost due to clearance times after each phase and of the time when all the traffic lights are red;
2. $Y$ is the sum of the saturation ratios $x$ from 2.

The effective green time, $G := C_{opt} - L$, can finally be split among the incoming links according to their contribution to the total flow.

Webster’s formula has been successfully used in transportation analysis for decades and remains one of the prevailing approaches to optimize traffic signal operations. It has been adapted in several recent works. For instance, in Cheng et al. (7), the same optimization criterion was considered, but another approximate equation for the expected delay was used. Furthermore, with the recent rise in interest for more environmentally sustainable transportation systems, this approach
was adapted in Calle-Laguna et al. (8) and Calle-Laguna et al. (9), considering new measures of traffic signals’ effectiveness, namely fuel consumption and emissions of pollutants.

Lämmer’s adaptive traffic signal control algorithm

Numerous other ways of scheduling traffic signal operations have been proposed. Some of them, such as Babicheva (10), rely on recent findings of queuing theory, others, such as Tonguz et al. (11), are based on cellular automata approaches. It would be impossible to review thoroughly all those methods; it was thus decided to present only one of them, underlying one of the MATSim implementations already mentioned (Thunig et al. (3)).

This approach was first presented in Lämmer and Helbing (12). It is based on a decentralized control algorithm, implementing both an optimizing strategy and a stabilizing strategy, using short-terms flow forecasts to adapt the green times of all incoming links. Communication between neighboring intersections is also possible and stability is guaranteed thanks to observance of minimal service intervals. Concretely, the optimizing strategy selects the link \( i \) that has the highest priority index \( \pi_i(t) \) at the current time \( t \), which is obtained from outflow rates, predictions of the number of approaching vehicles and the existing queue length. This index can be interpreted as a clearance efficiency rate – it represents basically the time necessary to clear the queue that forms during the intergreen time. A penalty is added to prevent the system from switching signals too frequently. The selected link is the one that will be served next, which means that it will be the next one to be assigned a green phase. The stabilizing strategy ensures afterwards that each link is served during a given minimal service time in order to prevent spillbacks.

An extension of this method – unfortunately not released as an open-source algorithm – could be tested with the traffic simulation VISSIM (13) and implemented in reality at two intersections in the city of Dresden (Germany) (14).

Traffic lights in the agent-based transport simulation framework MATSim

In MATSim, the network is represented by a set of nodes, representing road crossings, and by a set of links, that represent road sections. Detailed information is provided about the edges – one has access, among other data, to the link capacity, its type, the number of lanes it comprises and to the maximal speed allowed on it –, whereas only the coordinates of the vertices are known. In MATSim’s default implementation, the traffic is modeled mesoscopically: if the simulations detail the vehicles’ movements through the network at the link level – for instance, one knows the instant when a car enters and leaves a link –, the events happening within a link are not modeled. In particular, if a link ends at a node that should represent a traffic-signaled intersection, the turn pocket reserved to the left-turning cars is not clearly separated from the other ones. The traffic flow itself is modeled by first-in-first-out queues; those queues are not clearly split according to the agents’ intentions.

The fact that the traffic is modeled mesoscopically has impacts on the computation of the travel times. By default, the uncongested traversal time on a given link is given by

\[
t = \frac{L}{\min(v_f, v_{veh, max})},
\]
where $L$ is the length of the link, $v_f$ the speed limit on the link and $v_{veh,\text{max}}$ the maximum speed of the vehicle. If too many vehicles are present at the same time on a given link, a queue forms. Then, no vehicle can leave the link before all other vehicles, which are in front of it in the queue, leave themselves the link. On that way, congestion and delays caused by bottlenecks are modelled.

A simple approach, proposed in Hörl (15), was employed to model the delay caused by intersections: each time an agent, travelling by car, crosses a road which is more important than the one they are currently driving on, their travel time on this particular link is increased by a certain value called the crossing penalty. This penalty is manually calibrated, depending on the scenario, and usually ranges from 3 to 8 seconds. Moreover, it is applied both to traffic-signaled and non traffic-signaled intersections.

In Grether (2) and Grether and Thunig (16), the authors lay the foundation of an implementation of traffic lights in MATSim at a microscopic level. A subgraph representing the existing lanes is added on top of each link. This subgraph reflects the structure of a link ending at a traffic-signaled intersection: at the beginning, there is only one lane and, at its end, different lanes can be present that model the distinct turn pockets. With this representation, a vehicle has to be located in the appropriate turn pocket to access its next planned link. This allows to physically separate the different queues, as this is done in real life, without modifying the routing approach – the shortest path algorithm captures the effect of the lanes.

With this representation, traffic signals can be placed at the end of the lanes, allowing for an accurate representation of the way a traffic-signaled intersection is operated in reality.

This approach has been successfully tested in Grether (2) first with fixed-time signal plans and then with actuated signals. In Thunig et al. (3), another implementation, this time based on Lämmer’s approach, was proposed. Those three study cases are based on applications to the city of Cottbus, Germany. It can be remarked that all of them rely on fixed-time signal plans that were collected during a previous study.

One major drawback of this approach is that such data sets, gathering traffic signal schedules for a quite large area, are not always available. The present study addresses this issue by proposing a new method, based on Webster’s approach, which can be employed without having access to already existing traffic signal schedules. It also has the advantage that it remains in the field of mesoscopic simulations: no complex modification of the network structure (potentially at the expense of runtime efficiency) is required; nonetheless, the exact events happening at the intersection level will not be simulated. However, this is not an issue in the context of the present study: the main objective of this paper is to present a way of estimating the impacts of the delays caused by traffic-signaled intersections on the total travel times in MATSim simulations.

METHODOLOGY

The Zurich scenario

The study case which was chosen to implement and test Webster’s approach into MATSim is the city of Zurich, Switzerland. Zurich is the largest agglomeration of Switzerland, with a
population estimate of around 1.5 million inhabitants living in an area of 1,086 square kilometers.

The scenario was created as a section of the larger Switzerland scenario, which is presented, as well as the cutting process, in Hörl (15) and in Hörl et al. (17). It does not comprise all of Zurich’s agglomeration, but the city itself and the neighboring, smaller cities within a radius of 30 kilometers around the city center. Although this area is much smaller than the entire agglomeration, it gathers the vast majority of the population (around 1.3 million inhabitants).

The simulations were performed using a population sample of 10 percent – which means that the simulated population comprises of around 133,000 agents. They were run using the eqasim framework (18). Discrete choice models rely on travel times estimates and the implementation of Webster’s formula is likely to have significant influence on the travel times. As interference between the simulation itself (through the replanning strategy) and analysis of the impacts of the new method had to be avoided, it was decided not to use the MATSim discrete choice model extension, one of the most important innovations featured by the eqasim pipeline. Instead, at each iteration, during the replanning stage, 5 percent of the agents had the opportunity to choose a new route. This strategy is called ReRoute in MATSim.

### Extracting the traffic signals’ positions

The first step towards the implementation of Webster’s formula into MATSim consists in collecting the localizations of the traffic signals within the study area, and in adding them in some way into the existing network. Extracting the exact positions of the traffic signals can first be achieved with a single OSM request (19), for instance with the help of the QGis tool (20). Within this framework, for instance, all traffic lights within a given area can be extracted using the QuickOSM module – the “key” attribute was set to “highway” and the “value” attribute to “traffic_signals” for that purpose. The output of this step is a data set containing the coordinates of all traffic signals in the study area, as well as all their OSM attributes.

Such a database, however, is not sufficient for MATSim to understand which nodes have to be considered as traffic-signaled intersections, as traffic signals are themselves encoded as nodes in OSM. Hence, node identifiers of the extracted traffic signals cannot correspond to identifiers of nodes present in the MATSim network. In order to solve this issue, the file describing the MATSim network was used in a Python script to recreate the structure of the directed graph representing the road network thanks to the library NetworkX (21).

It was then possible to select, from this graph, all the vertices that represent potential traffic-signaled intersections. Not all nodes are indeed necessarily equivalent to crossroads in MATSim, as the network represents the roads’ geometry too. Particularly in Zurich, as the city is built in a hilly environment, it is quite rare that streets are perfectly straight, and each bend has to be modeled as one node to make sure that the network distances reflect the real ones. It was decided to define an intersection as a node where either strictly more than two links (the incoming links) end, or where exactly two links end and that has three or more neighbor nodes. This criterion was designed in this way because the graph edges are directed. If a node has only two neighbors and is such that exactly two links end in it, this means that the node is located within a road and
that it is thus not a real intersection. Otherwise, if three links end at a given node, it is guaranteed
that a conflict between at least two incoming flows will appear. The same situation is observed for
nodes with two incoming links but more than three neighbors: the incoming flows have to be split
and/or merged in or amongst all the possible outcoming links.

Once the candidate intersections have been designated, one \( k \)-dimensional tree is built.
It represents the locations of those intersections and is used to determine efficiently if, for each
intersection, one or more traffic signals are located nearby. Concretely, if a traffic signal was
found within a circle of radius 20 meters around the intersection, this intersection is alleged to be
traffic-signaled. Obviously, this method is not perfect and may erroneously classify intersections
as traffic-signaled, or miss intersections that are actually traffic-signaled, but are too large for the
node to be less than 20 meters away from the traffic lights. Manual validation – one can compare,
for instance on QGis, the locations of the traffic-signaled crossroads, obtained with this approach,
with the actual traffic lights – nevertheless confirmed the overall accuracy of this process. In total,
1326 traffic-signaled intersections were found in the study area. Unfortunately, no official figure is
available that would allow us to validate this result.

The output of this first step is an xml-file gathering all those intersections. The identifier
of the central vertex – corresponding to its identifier in the original MATSim network – is stored as
well as the identifiers of all links entering this intersection. As Webster’s approach is solely based
on the incoming flows, there is no need to provide more information about the links leaving the
intersection in the output file. Furthermore, within an intersection, incoming links corresponding
to one single road were grouped together. This was achieved by using a rather obvious criterion:
two links are alleged to belong to the same road if they have the same name (which can be accessed
in the OSM attributes present in the MATSim network) or if the angle between them is close to
180°. In particular, this will allow us to model with more accuracy the situation in which one
road ends in another one, as depicted in Figure 1. In this case, in reality, the distinction between
the primary and the secondary directions is reflected in the detailed traffic light schedules. In the
implemented MATSim approach, the links belonging to the same road will be first considered as
one single link, before being separated again for delay computation. More details will be presented
in the remaining paragraphs of this section.

Estimating road capacities
Traffic-signaled intersections have now been located and the necessary data about them
is stored in a way that can be used by MATSim. The formula that will be implemented depends to
a large extent on the road capacities. However, the road capacities were determined, for the Zurich
scenario, using the module pt2matsim. This tool defines road capacities solely according to their
road category in the OSM classification. For instance, by default, a motorway with three lanes has
a capacity of 6,000 vehicles per hour and a primary road has a capacity of 1,500 vehicles per hour.

Those capacities, however, do not reflect accurately what happens at a crossroad. Let us
imagine that a primary road, that was for instance designed to have a capacity of 1,500 vehicles
per hour, ends at a traffic-signaled intersection where the green signal is switched on for this link
on average 40 minutes per hour. Then, the capacity that has to be considered during a MATSim
simulation is equal to two thirds of its theoretical value, which is 1,000 vehicles per hour. On the one hand, this modifies the congestion level on the different network links, which ultimately influences the process determining the routes each agent will follow at each simulation stage. On the other hand, it means that Webster’s formula cannot be directly implemented within MATSim: experiments showed that those capacities are often too high, which leads to cycle lengths and delays too short to be consistent with reality.

A solution was to propose a heuristic updating the capacities before running the simulation. It is easy to notice that, in the context of a traffic-signaled intersection, not the link capacities are the limiting parameters, but the intersection capacity itself. It was thus decided to set the intersection capacity to 1,800 vehicles per hour, so as to account for the two-second rule. Some works, such as Riedel and Menendez (22), argue that this rule has to be adapted in an urban context such as Zurich’s and that the distance separating two vehicles is usually covered in less than two seconds, but, due to the unavoidable loss of time at the intersection, the value of 1,800 vehicles per hour was kept – this is also the value that was considered in the example detailed in Riedel and Menendez (22).

This heuristic consists in assuming the capacity of each link being equal to $1800 \times l$, where $l$ is the number of lanes of the link. In that way, Webster’s formula for the optimal cycle length becomes

$$C_{\text{opt}} := \frac{1.5 \times L + 5}{1 - \frac{1}{1800 \sum_{\text{link } i} f_i l_i}},$$

where $f_i$ is the flow on the link $i$ (the number of vehicles that left the link during the considered hour) and $l_i$ is the number of lanes of the link. This equation can be interpreted in the following way: first, the flow on the incoming link is downscaled according to the number of lanes since there is no conflict between vehicles entering the intersection from the same link. Secondly, these scaled flows are divided by the intersection capacity: those ratios represent the saturated flows, which are finally summed up together as in the original formula. The remaining of Webster’s approach can
then be used without any other change, as the next paragraph will explain. This approach ensures that motorists are not twice penalized by the delays they experience at crossroads – first through Webster’s delay equation, which takes overflow queues into account, and secondly through the additional travel time imposed by MATSim on congested links.

**Implementation of the Webster’s approach**

After the real intersection capacity was first roughly estimated, Webster’s formula can be implemented. Results from a baseline MATSim simulation, with 60 iterations, were used to obtain the theoretical flows on the links.

Those hourly flows are used to compute, hour by hour, the optimal cycle length on each intersection, the green time devoted to each link and, ultimately, the average delay experienced by motorists on the links. At the end of the first paragraph of this section, it was mentioned that links belonging to the same road were grouped together. Concretely, during the computation of the optimal cycle length, the ratios are computed group by group – and not link by link – before being summed and the effective green time was split among the groups and not among the links. Individual link flows are only used to compute the delays, both directly and indirectly, in the saturation flow.

The benefits of grouping the links if they belong to the same road are numerous. It allows first to model with more accuracy situations in which a small road ends in a more important one. Moreover, looking at the green times, it appears that the sum of the green time to cycle length ratios can be greater than one: this corresponds to what happens in reality, when green lights are switched on multiple times on given lanes, during one single cycle. This was impossible with the basic implementation of Webster’s approach.

**Enforcing the delays within the simulation**

Once the delays were obtained through Webster’s formula they can be enforced during the simulation. For a given link, the travel time is computed at each iteration: if an agent leaves this link by entering a link with a higher priority, the crossing penalty (defined in the introduction) is added to the time needed to go through the link. To enforce the new approach, this algorithm was completed with a simple test: if the link is part of an intersection, the corresponding delay is added to the default travel time instead of the crossing penalty. The hour of the day is known while computing those penalties; hence, the computation of delays based on hourly flows makes sense.

Moreover, at night, the traffic signals are often flashing yellow in Zurich. This was taken into account in the simulation: between 11 pm and 5 am, the modified algorithm was not applied and the default approach, with the crossing penalty only, was enforced.

**RESULTS AND DISCUSSION**

Several simulations were run using the methods and heuristics described in the previous sections of this paper to assess the impacts of the delays experienced by motorists due to traffic signals and to queues that form at crossroads. The first two correspond to the method implemented by default in the Zurich scenario, which penalizes motorists with additional delays each time they cross a road that have a higher priority than the one they are currently travelling on. In one of those
simulations, the crossing penalty was set to the value that is used by default in Zurich, which is 3 seconds, and, in the other one, a zero crossing penalty was considered. This helped to assess the impacts of a change of the crossing penalty on the travel times. In two other simulations, the capacity of all links in the network was changed to 1,800 vehicles/hour/lane and the same crossing penalties (respectively 0 sec. and 3 sec.) were chosen. Doing so, the possible interactions between crossing penalties and network capacities were investigated.

Results of the two last simulations were utilized to define hourly flows that served as inputs for two other simulations, where Webster’s formula was implemented. For these two simulations, road capacities of 1,800 vehicles/hour/lane were once again assumed, in order not to penalize twice the motorists as explained before.

**Capacities obtained with Webster’s approach**

As seen in the previous section, Webster’s approach was used to compute the average delay experienced by motorists on a link entering an intersection, depending on the hourly flow observed on that link. Webster’s formula can help re-evaluate the road capacities. Given an intersection, one knows the “green-time ratio” on each link, i.e. the ratio of the green time that was assigned to this link to the optimal cycle length. Multiplying this green-time ratio by the intersection capacity (which was estimated to 1,800 vehicles/hour in [subsection 4.3]) leads then to a new estimate of individual link capacities.

In the study case, as hourly flows were taken into account, the green-time ratios and, consequently, the capacities, depend on the hour of the day. In the remaining of this subsection, the results that will be presented use, for each link, the daily average of the capacities obtained from Webster’s approach, assuming in the first instance link capacities of 1,800 vehicles/hour/lane. In [Figure 2], those capacity estimates are compared to the default values that were used in the original MATSim network, according to the links’ OSM classification. Only road categories with more than 30 available observations were considered, and the estimations assumed one-lane roads (the total capacity estimate was divided by the number of lanes).

From the figure, one can see that, with Webster’s approach, the capacities are much more diverse than what is assumed in the original Zurich network. For instance, on secondary roads, a default capacity of 1,000 vehicles/hour is assumed. With Webster’s formula, those capacities range approximatively from 300 to 1800 vehicles/hour, which reflects more accurately the diversity of traffic situations that can be observed in reality. One can observe that, on the largest road categories, such as primary and motorway links, the capacity estimates obtained through Webster’s approach are significantly lower than the default values used by MATSim – for more than 50% of primary roads observations, the estimated capacity is smaller than 1150 vehicles/hour whereas a capacity of 1500 vehicles/hour is assumed in the MATSim simulations. This trend is less noticeable on smaller roads. The median road capacities obtained through Webster’s formula for secondary and residential roads are close to the default values used by MATSim. Nonetheless, the opposite trend can be observed on tertiary roads, whose capacities MATSim tends to underestimate (a default capacity of 600 vehicles/hour was assumed, almost 40% less than the observed empirical median which is close to 950 vehicles/hour). Finally, the figure suggests that, regarding the capacities,
FIGURE 2: Comparison between capacity estimates obtained through Webster’s approach and the default capacity values assumed in the original MATSim network. For each road category, the dark blue point indicates the default capacity obtained from OSM. The boxplots illustrate the capacity distributions estimated in this study, depending as well on the road category. The number written in the boxplot is the number of available observations. The default capacity assumed in MATSim for those links is 1,000 vehicles per hour, but the study indicates a capacity median slightly above 600 vehicles/hour.

From those observations, one can deduce that the default values used in MATSim simulations are not always the most appropriate ones. The employed approach to assign capacities to roads might thus be oversimplistic and should perhaps be adapted to perform simulations that are more consistent with reality. The OSM classification seems to play a major role here as it creates a distinction between secondary and tertiary roads that might not reflect the reality of traffic conditions in Zurich. The impacts of those changes in capacities on simulation outputs, such as flow distributions among the network, have however not been examined yet.

Consequences on the travel times

To compare the travel times in the different simulations, 5,000 origin-destination coordinate pairs were generated within the study area. A Java script was then used to compute the optimal route and the corresponding travel times according to simulation outputs, at five different hours of the day (6 a.m., 8 a.m., 12 noon, 6 p.m., 8 p.m.). These different time points correspond to different traffic situations (before the morning peak hours, the morning peak hours itself, day off-peak hours, evening peak hours and after the evening peak hours). Other travel time estimates,
for the same origin-destination pairs and for the same periods of the day, were also computed through two web APIs: Google Maps \(23\) and Bing Maps \(24\). At the end, for each comparison point (one of the 5,000 origin-destination pairs at one given time), eight travel time estimates were available: six from the various MATSim simulations, one from Google Maps and one from Bing Maps. Using two reference APIs helped to distinguish the differences existing between one online service and MATSim estimates from the bias that one of those two references can have compared to the other – from the results that will be presented, it will be obvious that Google Maps tends to underestimate travel times compared to Bing Maps.

Estimates obtained from Google Maps were considered as the reference and the result show the relative difference, in percentage points, between travel times obtained from the simulations or from Bing Maps and these reference values. Figure 3 illustrates the impact, in MATSim baseline simulations, of a change in the crossing penalty. Those relative differences were capped in absolute value at 100\% for the sake of visibility.

![FIGURE 3](image)

**FIGURE 3**: Relative difference between travel time estimates in the baseline simulations with crossing penalties equal to 0 (\(CP_0\)) and 3 seconds (\(CP_3\)) on the one hand, and estimates from Bing Maps on the other hand, and estimations given by Google Maps as a reference, in percentage points.

From the figure, one can see that the median relative difference increases from -7\% in the simulation with zero crossing penalty to 13.9\% in the one with a crossing penalty equal to 3 seconds. As the corresponding median from Bing Maps estimates amounts to 14.4\%, one understands why the value of 3 seconds was proposed as the default crossing penalty in Zurich: with it, the travel times estimates are as close as possible to the ones provided by this API. However, one can notice that the range between the first and ninth deciles remains unchanged, as well as the range between the first and the third quartiles, when the crossing penalty is modified. Those intervals are however large compared to the ones observed in the results from Bing Maps. With the default crossing penalty, it is neither unlikely to underestimate the travel times by 60\% nor to overestimate it by 100\% or more, whereas relative differences higher, in absolute value, than 30\%, are rare in the results from Bing Maps. To summarize, changing the crossing penalty seems to result only in a translation, upwards or downwards, of the boxplot depicting the relative difference of travel time estimates.
Figure 4 depicts the impacts of enforcing Webster’s approach, while keeping the crossing penalty unchanged. Interestingly, with Webster’s method, the travel times appear to be closer to the estimates provided by Google Maps (with a median relative difference of -2.2%) than to the ones obtained from Bing Maps, contrary to what is observed while only enforcing the crossing penalty. Moreover, one can see that both the interval between the first and ninth decile and the one between the first and third quartile shrink significantly when Webster’s approach is enforced. This implies that, on average, travel time estimates provided with this method are more reliable than the ones obtained with the crossing penalty alone. Figure 5 show the same results split according to the hour when the trip should take place. One can notice that the differences between Webster’s approach and the crossing penalty alone are much more significant at peak hours (8 a.m., 6 p.m. and, to a smaller extent, 8 p.m.) than off peak time (6 a.m. and 12 noon). This observation suggest that enforcing a crossing penalty besides penalizing separately motorists in congestion situations, as it is done in the baseline simulation, might not be the most relevant approach. With Webster’s method, as the computed delay simultaneously comprises the effects of the traffic lights and of congestion, the travel times estimates look more satisfactory, compared to the ones obtained from the two APIs.

This suggests that Webster’s approach is not the only reason why simulations implementing it provide better estimates than the baseline simulations, and that the road capacities themselves play a prominent role. This is what shows Figure 6, which compares the baseline simulation to the simulation where all link capacities were changed to 1,800 vehicles/hour/lane. The diagramm showing the results of this last simulation is indeed very similar to the one showing results of the simulation with Webster’s approach in Figure 4 – enforcing this approach for a network where no congestion can appear due to capacity limits leads to an increase of the median relative difference of around 3 percentage points.
Aurore Sallard, Miloš Balač

15

(a) At 6 AM

(b) At 8 AM

(c) At 12 PM

(d) At 6 PM

(e) At 8 PM

FIGURE 5: Relative difference between travel time estimates in the baseline simulations with crossing penalty equal to 3 seconds (CP3), in the simulation enforcing Webster’s approach with flows coming from the baseline simulation with the same crossing penalty (CP3_Webster) and in estimates from Bing Maps, and estimations given by Google Maps as a reference, in percentage points.

CONCLUSION

In this paper, a new approach to take into account traffic lights during MATSim simulations was proposed. This method uses Webster’s formula to compute, at the mesoscopic level, the delays experienced by motorists due to traffic signals themselves and queues forming because of them. A technique to extract the traffic lights’ localizations and to integrate them automatically in an already existing MATSim network was also developed and represents another significant contribution.

While analyzing results obtained through this new approach, the prominent role of road capacities within MATSim simulations was brought to light. It was shown that the capacities assumed in the default MATSim settings may not be the most appropriate ones to ensure reliable travel time estimates. Moreover, evidence was found that the approach used by default in order
FIGURE 6: Relative difference between travel time estimates in the baseline simulation with crossing penalty equal to 3 seconds (CP3), in the same simulation where the link capacities were set to 1,800 vehicles/hour/lane and in estimates from Bing Maps, and estimations given by Google Maps as a reference, in percentage points.

1 to take into account crossroads, which relies on the manually calibrated crossing penalty, might tend to overpenalize motorists, regarding travel times, in particular at peak hours. On the contrary, the presented method provides reliable travel time estimates, regardless of the hour of the day. Comparing those estimates to the ones provided by microscopic simulation tools, such as SUMO (25), would be worthwhile in order to provide more guarantees about the quality of those estimates.

Different parts in the article indicate potential future work. First, Webster’s method is implemented in a static way: final results from one previous simulation are needed to compute the delays that will be enforced, and those delays are not updated iteratively during the simulation to represent the fact that motorists adapt their behavior to them. A dynamic implementation would allow motorists and traffic lights to adapt simultaneously to each other. Secondly, because of this, it is not straightforward to adapt this method to assess the impacts of changes in infrastructure: if a modification is done to the network, a first simulation will have to be performed with the modified network before new delays can be computed and enforced. Besides that, Webster’s approach provides road capacity estimates, it could be interesting to integrate the presented approach in an iterative process, where flows, delays and road capacities could be updated simultaneously.

This work showed the drawbacks of the crossing penalty approach, which was employed in MATSim scenarios such as the Zurich scenario. However, the new approach is only suitable for traffic-signaled intersections. In the simulations that were performed for this study, the crossing penalty approach was thus still used for non traffic-signaled intersections. An additional improvement would be to propose another method to model in a more accurate way these non traffic-signaled crossroads. Doing so, a complete comparison between the crossing penalty approach and the new one could be performed. The hope is to work in this direction in order to be soon able to offer a new MATSim module that would model accurately all intersections and thus improve significantly the quality of travel time estimates, when access to detailed traffic signal schedules is not available.
AUTHOR CONTRIBUTION
The authors confirm contribution to the paper as follows: study conception and design: A. Sallard, M. Balač; data collection: A. Sallard; analysis and interpretation of results: A. Sallard, M. Balač; draft manuscript preparation: A. Sallard. All authors reviewed the results and approved the final version of the manuscript.

REFERENCES