EWGT2013 – 16th Meeting of the EURO Working Group on Transportation

Capability-Enhanced AIMSUN with Real-Time Signal Timing Control

Cristina Vilarinho\textsuperscript{a}, *, Guilherme Soares\textsuperscript{b}, José Macedo\textsuperscript{b}, José Pedro Tavares\textsuperscript{a}, Rosaldo J. F. Rossetti\textsuperscript{b}

\textsuperscript{a}Centro de Investigação do Território, Transportes e Ambiente, Departamento de Engenharia Civil
\textsuperscript{b}Laboratório de Inteligência Artificial e Ciência de Computadores, Departamento de Engenharia Informática
Faculdade de Engenharia da Universidade do Porto, Rua Roberto Frias, 4200-465 Porto, Portugal

Abstract

In this paper, we enhanced the capability of Aimsun simulation including signal timing optimization of traffic control with real-time information on the network dynamics. The problem is formulated so as to find the duration of maximum green time for each stage in response to recurrent traffic flow fluctuation at an intersection. The approach used a simple version of the Webster method for determining the cycle length and green time split. The resulting algorithm was coded in Java and used TraSMAPI to dynamically link it to Aimsun’s API, which allows the user to change the cycle length and green time duration of each traffic light’s stage. TraSMAPI is a Traffic Simulation Manager API, designed to provide real-time interaction with traffic simulators. So far, this tool has been only tested with Sumo and Itsumo microscopic traffic simulators. The paper presents an example of the communication protocol using the API module linked to TraSMAPI, and contributes to the implementation of a novel real-time traffic control in Aimsun.

© 2013 The Authors. Published by Elsevier Ltd.
Selection and/or peer-review under responsibility of Scientific Committee.

Keywords: Aimsun; traffic simulation; traffic light control; Webster method;

1. Introduction

In the last decades automobile fleets in urban areas have been increasing all over the world bringing serious congestion problems to large cities, affecting them both socially and economically. In view of the fact that travel demand increases at a greater rate than adding to the road capacity, this problem will continue increasing unless

* Corresponding author. Tel.: +351-225-082-161; fax: +351-225-081-902.
E-mail address: cvilarinho@fe.up.pt.
efficient traffic management strategies are implemented. In this way, traffic signal timing is one of the most cost-effective methods for improving mobility within the urban transportation system (Park and Schneeberger, 2003).

Traffic simulation models are a powerful tool for testing and analyzing a wide variety of dynamic transportation problems that are difficult to perform in real-world environment. This technique simulates the real conditions of a network and carries out analysis and forecasts on it by replacing the physical experiments with representations in a computer program. This ability to draw conclusions and to test new techniques without having to disturb the real system and to undertake new data collection is the great advantage of simulation models, which makes their use so important.

Over than eighty traffic simulation software packages (Boxill and YuAn, 2000) have been developed by public institutions, research organizations and consultants. Usually these models have specific functions such as signal control at isolated intersections, signal control at coordinated intersections, freeway simulation, etc. According to their scope, they can be classified as either microscopic, mesoscopic or macroscopic simulation model. 77% of traffic simulators were developed for the microscopic level, which means that each vehicle in the network is detailed described (e.g. maximum speeds, acceleration rates and deceleration rates) and continuously modeled during the simulation period while it travels throughout the traffic network, according to vehicle behavior models such as car following, lane changing, etc.

However, simulation models have limitations in their use because each one is designed to perform a specific traffic analysis. There is no single software package suitable for all traffic situations (Alexiadis, Jeanotte and Chandra, 2004). Besides it would be an exercise of dubious utility to try to integrate all functionalities in a single traffic simulator.

In some simulation packages such as Aimsun, Vissim and Paramics, there is an Application Programming Interface (API) module available that provides users with the ability to implement an interface connection to user-defined and third-party applications. This simple and yet powerful mechanism allows data exchange between running applications and eventually elements of Aimsun’s simulation models to be influenced by external procedures. There are some studies on traffic control strategies that describe the use of an API in Paramics (for examples, see Liu, Chu, and Recker, 2000; Park, Yun and Choi, 2004; Srinivasan, Min Chee Choy and Cheu, 2006; Yang et al., 2012) but such methods are less explored for Aimsun (Fang and Elefteriadou, 2008; Pham, 2009; Fang et al., 2013).

In this work a different API approach is used. TraSMAPI is a Traffic Simulation Manager Application Programming Interface, designed to provide real-time interaction with traffic simulators by collecting relevant metrics and statistics, and offering an integrated framework to develop multi-agent systems (MAS) (Timóteo, Araújo, Rossetti and Oliveira, 2010). So far, this tool has been only tested with Sumo and Itsumo microscopic traffic simulators (Timóteo, Araújo, Rossetti and Oliveira 2012).

In this paper we extended the capability of Aimsun simulations improving signal timing of traffic control with real-time information on the network dynamics. The problem is formulated to find the maximum duration of green time for each stage in response to recurrent traffic flow fluctuations at an intersection. The algorithm used a simple version of the Webster method (Webster, 1958) for determining the cycle length and green time split. Software packages such as Syncro, Passer II and Passer V uses as well Webster method (Chaudhary, Kovvali and Alam, 2002).

The algorithm was coded in Java and used TraSMAPI framework to dynamically link it to Aimsun’s API (C/C++), which allows the user to change the cycle length and green time duration of each traffic light’s stage. The paper contributes to the implementation of a novel real-time traffic control in Aimsun and presents an example of the devised communication protocol using the API module and TraSMAPI.

This paper is organized as follows. In section 2, the proposed signal timing control is described. Section 3 consists of a brief description of the software used, whereas the communication protocol is presented in Section 4. Section 5 presents an application example to support our approach. In section 6, we draw conclusions and discuss on future work directions.
2. Description of Signal Timing Control

Signal timing control of traffic lights is one of the less expensive and most effective means of reducing vehicles traffic congestion in metropolitan road networks (Spall and Chin, 1997). We propose a signal timing control based on the Webster method for an actuated-operation signal. For each five-minute simulated interval, it is decided the new cycle length and the maximum green time period for all stages using the traffic flows of the turning movements of the last five minutes in the node (Figure 1). Up-to-date signal timing brings benefits in air quality, traffic safety and reduces drivers’ frustration (Ratting, Chapline and Williams, 2002).

Fig. 1. Flowchart of the signal timing control proposed
In the first part of the algorithm, the information about the traffic light diagram (stages, lane group movements and intergreen) is read and stored in our signal timing solution state (Java classes).

In each stage, the lane group movements are analyzed in order to find the critical flow ratio (Equation 1), denoted by the movement with higher ratio (y-value). The green split allocation is based on this ratio by stage. In case the stage is either an interphase or an intergreen, no critical flow is determined and a total of five seconds (yellow and all red time) should be added to the lost time variable.

\[ Y_i = \max \left( \frac{q_j}{s_j} \right) \]  

where \( Y_i \) is the critical flow ratio of stage \( i \), \( q_j \) is the traffic flow of lane group movement \( j \) in vehicles per hour, and \( s_j \) is the saturation flow of lane group movement \( j \) in vehicles per hour.

The traffic flow used in Equation 1 is the turning flow simulated in the model and sent to the algorithm. The saturation flow is encoded in the algorithm according to the movement type (straight, right turn and left turn).

Webster’s formulation is used to determine the optimal cycle time for the minimum delay (Equation 2). Webster (Webster, 1958) used field observations and computer simulation to develop a cycle-optimization equation intended to minimize delays when arrivals are random.

\[ C = \frac{1.5 \times L + 5}{1 - \sum_{i=1}^{n} Y_i} \]  

where \( L \) is the total lost time per cycle in seconds, and \( \sum_{i=1}^{n} Y_i \) refers to the intersection critical flow ratio, i.e., the sum of flow ratios of all stages \( i \).

The cycle length value should be in the range between 30 seconds and 150 seconds (Costa, Seco and Vasconcelos, 2010). In the case cycle be out of these limits, cycle length takes the value closest acceptable.

Several strategies for allocating effective green time are available (Koonce et al, 2008). We selected one of the most popular strategies, which consist in distributing the available green time in proportion to the critical flow factors on each stage (Equation 3). The total green time is equal to the cycle length without the lost time per cycle.

\[ g_i = \frac{Y_i}{Y} \times (C - L) \]  

where \( g_i \) is the effective green period of stage \( i \) in seconds.

The green time period determined in Equation 3 is the maximum green time period, which should be higher than or equal to the minimum green period. The minimum green period defined is 8 seconds (Costa, Seco and Vasconcelos, 2010). In case of a value lower than the limit, the green time of the stage assumes the minimum value and it is counted as a lost time of the cycle. The cycle length is defined again, as well as the green time period allocation until every established condition is satisfied. If the maximum green time period is equal to the minimum, then it means that the stage will behave as pre-timed. The extension defined is 3 seconds.

3. Aimsun Overview

Aimsun means Advanced Interactive Microscopic Simulator for Urban and Non-Urban Networks and has been developed by Universidad Politecnica Catalunya and Transportation Simulation System of Barcelona, Spain (Barceló, 2002). Aimsun is a microscopic traffic simulation model that can simulate the individual behavior of a vehicle on the network over time and in accordance to the various theories of vehicle behavior. This model was developed to be a tool to support traffic engineers so that they can analyze and design traffic models. Aimsun has proven to be very useful in testing new traffic control systems and management policies based on traditional technologies, as well as in the implementation of Intelligent Transport Systems (ITSs) (TSS, 2011a). The SMARTTEST project (Barceló et al, 2002) classified Aimsun as high-applicability and suitable for different networks (urban, highways).
Aimsun simulations can be classified as hybrid processes, combining an event approach with activity scanning. At each simulation step, the simulation cycle updates the events scheduling list (traffic light changing does not depend on the conclusion of other activities). After this updating process, a set of loops starts to update the states of the entities (links and nodes) and vehicles. The last tasks include instantiating new vehicles, collecting statistics and updating the simulation clock (TSS, 2011a).

Three different types of traffic light control can be defined in Aimsun, namely pre-timed, actuated and external. There are some customization options such as: coordination (i.e., the possibility of synchronizing a system of traffic lights controllers), implementation of actuated parameters (e.g., minimum green, rest in red, allowance gap, passage gap, recall), and implementation of multi-ringer or pre-emption controllers, depending on the traffic control type selected.

As for signal timing, the actuated traffic control allows adjusting traffic signal timing during the simulation period for a more efficient control. Adjustments result as a reaction to real-time traffic demand measured by vehicle detectors, installed in a shallow slot sawed in the pavement. However the signal timing change is limited to a minimum and to a maximum green previously defined by the user. In this work, we used the actuated traffic control to which a new cycle length and a maximum green period are calculated in each time interval based on the turning flow of a node. This adaptive approach is less selfish than the actuated one because the new times have in attention the node as a whole.

Traffic lights control in a node is organized in signal groups that consist of a set of turning movements that are controlled by the same light indications of traffic lights. Then, a sequence of phases is defined as is its duration of green time for the entire node. Each phase has a set of signal groups associated with it. Another type of phase is defined for the intergreen period, the yellow time and the following all-red interval, defined as an interphase.

For traffic light control modeling, Aimsun micro-simulator uses fictitious stopped vehicles, which are instantiated and placed at the stop line when the light turns red, and are eliminated when it turns green. In this way, the car-following model can be used to model braking to stop in front of a red light (TSS, 2011a).

Aimsun was chosen because it is a sturdier traffic simulator providing the possibility of customization trough the API module, which ultimately leverages traffic light control with a lot of potential to be enhanced.

4. Communication Protocol

To include the algorithm described in the Section 2, it was necessary to develop a communication protocol to link it to the traffic simulator, described in Section 3. The proposed signal timing control is carried out by a multi-agent framework leveraging MAS-based simulation over multiple microscopic simulators, coined TraSMAPI (Timóteo, Araújo, Rossetti and Oliveira, 2010). This tool was developed in an abstraction level in order to be independent from the simulator used, which allows modelers to test the same approach in different traffic simulation software without changing their solution code. For each coupled traffic simulator, TraSMAPI implements a dedicated communication module, which interconnects with the simulator’s API.

The Aimsun’s API module allows the interface of almost any external application that may need access to some internal data of Aimsun during simulation run time since it has direct access to the simulation functions. The interaction between Aimsun simulator and its API module is performed by a set of functions provided by an interface of Aimsun. This set of functions allows the Aimsun’s API module to obtain information from the simulation model and modify the simulation state (TSS, 2011b).

Moreover, the external application must connect and establish a communication process with the API to take advantage of these functionalities. Therefore, to test the algorithm described in the section 2 it was necessary to develop a communication protocol to link the implemented Java solution with the Aimsun’s environment.

In this work we used the C/C++ programming language to link the dedicated TraSMAPI’s communication module (in Java) to the Aimsun’s API, allowing access to traffic light control and turning flows statistics.
In short the overall information flux between the several components is presented in Figure 2.

![Communication protocol scheme](image.png)

A node controlled by traffic lights is modeled in Aimsun simulator. The TraSMAPI framework allows building an abstraction of this traffic light, and controlling the simulation lifecycle.

Every simulation step our algorithm runs and in 300 time steps – equivalent to 300 seconds - new green time durations are determined. Otherwise, simulation control is handled back to the simulator with the previous calculated green time durations. To determine a new maximum green time duration, it is necessary to feed the algorithm with simulation data. The algorithm requests information invoking the appropriate function through the API, which searches for the invoked function and asks to the algorithm for inputs. The input data is passed through the API that asks the simulator for the function result and returns it back to the algorithm.

5. Application

5.1. Case study and scenario description

We tested the communication protocol and evaluated the proposed signal timing algorithm through a case study based on a small network of two nodes controlled by traffic lights within the city of Porto, in Portugal (Figure 3). The two adjacent nodes are defined by four streets each. All movements of road traffic and pedestrians are controlled by traffic lights, though in this study only the road traffic is considered. We used a sub-network of a calibrated and validated microscopic traffic simulation model of the Asprela Campus, at the University of Porto (Vilarinho and Tavares, 2012).

![Case study](image.png)

The actual control plan of the traffic light for the peak morning hour is shown in Figure 4. This plan is used as baseline for scenario comparison and for the first ten minutes of scenario simulation in order to have some input before our algorithm starts to calculate new times.
The demand is codified in matrices of origin and destination of 15-minutes each. The total time of simulation is two hours (7:45 am to 9:45 am) that follows a 15-minute warm-up period. Three scenarios of demand were defined (low, medium and high), with the morning peak-hour traffic corresponding to the third scenario of high demand.

The evaluation of the signal timing control uses measures of effectiveness (MOEs), which are indicators that quantify in which extent a particular scenario meets the study objectives. In traffic control, MOEs normally used include delay, travel time, stops, etc. (Akçelik, 1981; Koonce et al, 2008). According to the study purposed and the available simulation model outputs, we selected the total travel time of the system, the average speed of the system, the total delay time of each node and the mean queue of each node.

5.2. Results and discussion

After the successful implementation of the algorithm through the API in Aimsun, we conducted comparison tests between the network featuring traffic lights controlled by the implemented algorithm (API) and the same network featuring a full-actuated traffic control for the three demand scenarios. We run each scenario ten times and calculated the average result. The simulation results show that the system controlled by our algorithm has superior performance in total travel time and speed for low and medium demand (Table 1).

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Total travel Time (h)</th>
<th>Speed (km/h)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Actuated</td>
<td>API</td>
</tr>
<tr>
<td>1</td>
<td>1,7</td>
<td>1,6</td>
</tr>
<tr>
<td>2</td>
<td>32,0</td>
<td>31,4</td>
</tr>
<tr>
<td>3</td>
<td>1452,7</td>
<td>1785,3</td>
</tr>
</tbody>
</table>

Table 1. Simulation Performance Comparisons - system

The results for node detail show that nodes follow the system trend with the exception of total delay of node 2. For the medium demand, total travel time of node 2 is very similar for both traffic control strategies. For the high demand the traffic control implemented by the algorithm has lower total travel time.
Table 2. Simulation Performance Comparisons - nodes

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Node</th>
<th>Total delay (veh.h/h)</th>
<th>Queue (veh)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Actuated</td>
<td>API</td>
</tr>
<tr>
<td>1</td>
<td>0.19</td>
<td>0.16</td>
<td>-16.3%</td>
</tr>
<tr>
<td>2</td>
<td>3.93</td>
<td>3.59</td>
<td>-8.6%</td>
</tr>
<tr>
<td>3</td>
<td>545.17</td>
<td>739.80</td>
<td>35.7%</td>
</tr>
</tbody>
</table>

The depicted plots in Figures 5, 6 and 7 show the MOEs of each node, in each time period for the different traffic demands. The plot analyses confirm the above remarks. The suspicious of nodes congestion is proved in Figure 7, since from 7:55 am of node 1 and 8:20 am of node 2 the MOE values are exponentially increasing. However, after 9:10 am, the traffic flow reduced; for node 2 the total delay becomes higher for the actuated strategy and for node 1 queue length becomes also higher for the actuated strategy too.

Figure 5. Indicators results of each node (node 1 right, node 2 right) - scenario 1

Figure 6. Indicators results of each node (node 1 right, node 2 right) - scenario 2
As a consequence of the result analysis, it is possible to conclude that the proposed signal timing reduces the system total travel time, the system speed, the nodes total delay time and the nodes queue length when the network/intersection is not saturated. The reduction in algorithm performance is explained by the approach used to calculate the maximum green time and the cycle length. As the simulation flow of the last period increased, the cycle lengths also increased until they reached the maximum cycle length (150 sec) so as to improve capacity in the node. The increases in cycle length influenced the capacity of the intersection positively as the proportion of lost time between stages decreased. The use of our algorithm increased the actual traffic flow in 6%. However long cycles increased the delay time and queue length as vehicles had long waiting time periods during the red signal period. Once we used turning flows as parameter, in green time allocation it was not possible to take into account the traffic pressure in each section after it reached the maximum cycle length.

6. Conclusion and Future works

A novel signal timing control algorithm was developed and implemented in the microscopic traffic simulator Aimsun, using TraSMAPI and the Aimsun API module for communication with the simulator. The simulated network encompassed two adjacent intersections operated by a full actuated traffic control. The algorithm employed allowed us to find new maximum green time durations for each stage in order to handle recurrent demand fluctuations. The comparison study in the simulated system permitted us to conclude that the proposed control method had a better performance when the intersection is not saturated. This paper introduces a simple adaptive signal control model that can be easily applied to an existing actuated signal control system to improve the performance, especially during off-peak periods.

The paper also contributes to the implementation of a novel real-time traffic control in Aimsun and presented an example of the communication protocol between the Aimsun’s API module and TraSMAPI.

In the future, the signal timing control algorithm will be revised in order to become more general in terms of application, suitable for different intersections geometries and management, as well as for networks with several intersections controlled by traffic lights. The algorithm will also be enhanced in order to consider other variables of signal timing such as vehicle delays and queue lengths on the links of the intersection, as well as other network variables. Finally, the algorithm will be tested in other simulators taking advantage of the simulator-independent nature of TraSMAPI, which allows a solution once designed to be tested in different platforms with no need for recoding.
Acknowledgements

This project has been partially supported by FCT (Fundação para a Ciência e a Tecnologia), the Portuguese Agency for R&D, under grant SFRH/BD/51977/2012.

References


