



Intelligent Transport System auditing using road traffic micro-simulation

**M. BAZAN^a, P. CISKOWSKI^a, K. HALAWA^a, T. JANICZEK^b, P. KOZACZEWSKI^b,
Ł. MADEJ^a, A. RUSIECKI^a**

^a WROCLAW UNIVERSITY OF TECHNOLOGY, Faculty of Electronics, Department of Computer Engineering,
ul. Janiczewskiego 11/17, 50-372 Wrocław, Poland

^b WROCLAW UNIVERSITY OF TECHNOLOGY, Faculty of Electronics, Department of Control Systems and
Mechatronics, ul. Janiczewskiego 11/17, 50-372 Wrocław, Poland

EMAIL: marek.bazan@pwr.edu.pl

ABSTRACT

ITS systems have been deployed for last one and a half decades. Their aim is to increase a traffic flow rate in a road network of an urbanized area, to improve the comfort of driving as well as to decrease the pollution. A lot of commercial software are available to simulate road traffic in urbanized areas. Some of them are suitable to perform traffic simulations of intelligent transportation systems via traffic modeling. A process of traffic modeling exploiting numerical simulations for an urban area where an ITS is deployed requires provision of digital maps, traffic demand amongst city zones, traffic signalization micro-programs being executed in the environment that can imitate a dynamic behavior of traffic lights at intersections in the ITS. In this paper an execution environment, developed by the ArsNumerica Group, that launches a road traffic micro-simulation is used to audit a performance of intersections' signalization micro-programs on one of the main arteries in the city of Wrocław (Poland).

Keywords: traffic modeling, intelligent transportation system auditing, road traffic simulations

1. Introduction

Simulating an intelligent transportation system is a complicated task. It requires provision of

- digital maps,
- traffic demand amongst city zones,
- traffic signalization micro-programs being executed in the environment that can imitate a dynamic behavior of traffic lights at intersections in the intelligent transport system.

Digital maps may be provided by local geo-data providers (the reality in Poland means that it is e.g. [4]) or free global ones e.g. [5]. Usually traffic demand in the form of an origin destination matrix is calculated numerically from some kind of cordon measurements like e.g. [6] or can come from GPS data [7]. Full trip surveillance methods are expensive and are rarely used. For the simulation of an intelligent transportation system link count data is readily available. Such data was used to obtain

an input to a procedure described in [3] to calculate origination destination matrix.

The last crucial element is dynamic traffic lights control. In an intelligent transportation system traffic light phases are prolonged or shortened depending on the flow rate on particular lanes (c.f. [8] for Wrocław's instance of the ITS). That is why simple static settings of traffic light logic is not sufficient for the simulation.

In this paper we present an integrated execution environment to dynamically simulate traffic light phase adjustment on multiple intersections included into the intelligent transportation system. Such an environment enables us to calibrate the traffic model against any parameters of a single simulation. The idea of such an integrated environment was already used in the field of electro-magnetic designing (c.f. [12]). Due to this fact such an environment can be used to check whether a better performance can be achieved with different parameters. A numerical example concerns one of the main arteries in the city of Wrocław (Poland). In the example presented in this paper the sequence table is

parametrized and the simulations with flows from real life link counts from peak hours are tested. Such an assessment of an intelligent transportation system is a detailed extension of the assessment methodologies described in [9].

2. Dynamic simulation of the city

2.1. Static vs dynamic simulation

Dealing with a dynamic traffic simulation is far more complicated than with a static one. For a static simulation traffic light settings are set once for the whole simulation time. To define a static simulation it is sufficient to have

1. a road network definition,
2. defined traffic flows dependent on a traffic demand supplied,
3. static light signalization logic settings for a simulator such as SUMO [2].

For a dynamic simulation light signalization settings are adjusted according to changes in traffic flow. In the traffic simulation SUMO [2] such an adjustment can be performed using the TraCI extension [2]. To define the whole configuration one needs to get

1. a road network definition,
2. defined traffic flows dependent on a supplied traffic demand,
3. TraCI client configured so that it may connect to the SUMO simulator running the simulation defined by the latter two points,
4. an intersections' micro-programs executor that with the use of the TraCI client
 - may read current flow at the specified lanes of the roads,
 - may change current traffic light settings at the end time of the last sequence for the next cycle depending on a measured flow at specified detectors.

In [10] and [15] the implementation of an execution environment that fulfills the above conditions for a dynamic simulation was presented. Having defined a road network with validation detectors and traffic trips using origin destination matrix method [3] of a SUMO trip generator [2] for link count data – coming from an intelligent transportation system in a certain period of time. The last element needed are micro-programs to define a dynamic signalization control. For this purpose we extended a TCL language free interpreter PICOL [11] to be able to read validation detector counts and set traffic lights logic for the next cycle.

The original implementation of the above interpreter covered the following instructions

1. variables of simple types numeric, string and boolean,
2. substitution instruction,
3. loop instruction,
4. if instruction,
5. a calculator of expressions using variables and constants of the data types served,
6. procedure calls.

Our extension of the above interpreter covered

1. arrays,
2. access to predefined validation detectors,
3. access to predefined traffic light logic.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Fix	T
																5	30
																	5
																	15
																5	5
																	30
																	5
																	5
																	15
																	5

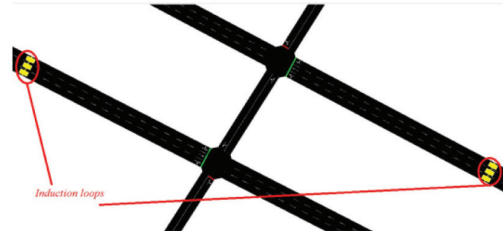


Fig. 1. An example of a sequence table. Intersection corresponding to a sequence table [own study]

Using the above extension allowed us to write and execute TCL microprograms for a sequence of intersections on one of the main arteries in the city of Wrocław – Legnicka Street. The microprograms correspond to those from the GERTRUDE system [8] in the intelligent transportation system deployed in the city. For each intersection its sequence table can be provided in an XML file. A sequence table is a table containing a description of the whole cycle of the specified intersection divided into sequences (phases) with their durations and possibility of the prolongation (see Figure 1). In Figure 2 we show an example of a microprogram to dynamically control an intersection.

```

set stc 0 set vehn 0 set lc 0
set pom 0 set ls 5000
set pg 60
set idS0 0
set idS1 1
set idS2 2
set lp1 50
set lp2 30
set wyd 5
set iddet 0
set prg1 grgr
set prg2 rgrg
while { <= $stc $ls }
{
  get $iddet pom
  set vehn [ + $pom $vehn ]
  while { <= $lc $lp1 }
  {
    setlights $idS0 $prg1 1
    setlights $idS1 $prg1 1
    setlights $idS2 $prg1 1
    get $iddet pom
    set vehn [ + $pom $vehn ]
    set lc [ + $lc 1 ]
  }
}
set lc 0
    
```

Fig. 2. An example of a microprogram for two intersections such as in Fig. 1 [own study]

3. Execution framework for calibration of parameters

On the top of the simulation environment there is a framework of the execution environment that is driven by a global optimization algorithm. A global optimization algorithm optimizes a quality of the simulation depending on the parameters. The structure of the implemented environment is shown in Figure 3.

3.1. Parameter space – constrained space

The parameters in the case of the auditing of the sequence of signalized intersections are their sequence tables. In the examples in the next section a time of the prolongation and the initial duration of the chosen sequence in the table are our optimized parameters. The parameters are changed in certain prescribed ranges. Of course the framework can serve also for calibration of parameters such as Krauss model parameters [13]. Into the parameter space we can also include offsets of the chosen phases so that the green wave optimization can be performed [14].

3.2. Objective functions

As an objective function we have chosen the average length of queues on the lights during a whole simulation. The total time of all trips can also be chosen. The objective functions used for calibration of the measurements are usually mean square error between measured data and the data obtained from the simulation. The nature of such functions is described in [13].

4. Global optimization algorithms

As a global optimization algorithm we used Matlab [18] implementation of the Genetic Algorithm based Augmented Lagrangian method for constrained optimization [16]. The augmented Lagrangian method is a connection of the penalty function method with the common Lagrange multipliers method. For the n -th intersection auditing the input vector has a form

$$\mathbf{x} = (r_{i_1}, \text{fix}_{i_1}, \dots, r_{i_1}, \text{fix}_{i_1}, r_{i_{1+1}}, \text{fix}_{i_{1+1}}, \dots, r_{i_2}, \text{fix}_{i_2}, \dots, r_{i_{(n-1)+1}}, \text{fix}_{i_{(n-1)+1}}, \dots, r_{i_{n+l}}, \text{fix}_{i_{n+l}})$$

where components $r_{i_k}, \text{fix}_{i_k}, \dots, r_{i_{k+1}-1}, \text{fix}_{i_{k+1}-1}$ of vector \mathbf{x} concern

the k -th intersection sequence table for $k = 1, \dots, n-1$. For n -th intersection corresponding components are $r_{i_{n+1}}, \text{fix}_{i_{n+1}}, \dots, r_{i_{n+l}},$

$\text{fix}_{i_{n+l}}$ where the sequence table is parametrized with $2l$ variables.

This notation corresponds to entries of the sequence table shown in e.g. 1 so the $r_{i_{k+j}}$ correspond to j -th sequence T column entry and

$\text{fix}_{i_{k+j}}$ correspond to j -th sequence Fix column entry.

Also Particle Swarm Optimization methods can be used such as [17] that were applied to traffic lights optimization in [15].

5. Auditing procedure

An overview of assessment methodologies of an intelligent transportation system can be found in [9]. The concept of the auditing procedure with the use of the framework described in the previous section relies on the as close as possible recovery of the traffic conditions in the analyzed network. The procedure consists of the following steps:

1. Prepare a road network model.
2. Prepare traffic demand in the form of a log of a link count data in a prescribed time window.
3. Calculate an origin destination matrix to establish routes.
4. Prepare sequence tables for all cross-sections in the analyzed network that is valid for the analyzed intelligent transportation system.
5. Prepare a table of offsets in case of an arteria simulation.
6. Prepare micro-programs in TCL for the ArsNumerica environment that correspond to those executed in the analyzed intelligent transportation system with the sequence tables defined in the previous step.
7. Run Global Optimization Algorithm with the ArsNumerica execution environment with the above settings of the network, sequence tables for the intersections included and microprograms. The aim is to minimize the objective function measuring a quality of the performance of traffic in the network for the specified settings of parameters. The parameters are
 - (a) sequence tables,
 - (b) offset table
8. if the traffic is recovered by the simulation with nominal settings (those corresponding to the real intelligent transportation system settings) then
 - (a) if the objective function can be radically decreased then something is wrong in the analyzed intelligent transportation system
 - (b) if the objective function cannot be improved the real setting seems to be OK.

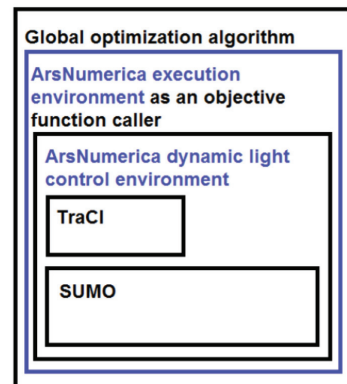


Fig. 3. Execution environment structure [own study]

6. Example of Legnicka optimization

With the use of the proposed procedure and the implemented environment we perform the assessment of three cross-sections on Legnicka Street – one of the main arteries in the City of Wrocław.

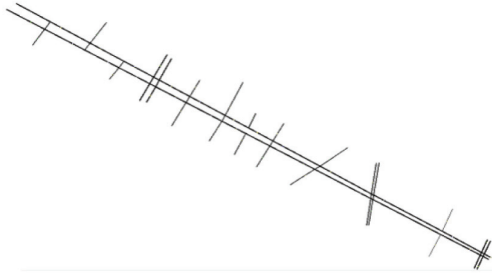


Fig. 4. Legnicka and Lotnicza Street in the city of Wrocław – the Open Street Map imported SUMO model [own study]

The steps from section 5 were performed as follows:

1. The network model was taken from open street maps [5]
2. Prepare traffic demand in the form of a log of a link count data in a prescribed time window.

The link count data were received from the Wrocław intelligent transportation system for the working days between 12-23 May 2014 in the morning peak hour i.e. 7:15-8:15. The data then is average link count from 10 working days.

3. Calculate origin destination matrix to establish routes. The SUMO tool DfRouter [2] was used. It has to be noted that this tool works only for relatively simple networks up to 20 cross-sections and without roundabouts. That means it cannot be used in the process of modeling of the city or larger regions. For this purpose the method described in [3] has to be used.
4. Prepare sequence tables for all cross-sections in the analyzed network that is valid for the analyzed intelligent transportation system.



Fig. 5. Sequence tables for (a) two first simple intersections (Bajana str. and Hutnicza str.), (b) a more complicated intersection (Na Ostatnim Groszu str.)

5. Prepare a table of offsets in case of an arteria simulation.
6. Prepare micro-programs in TCL for ArsNumerica environment that correspond to those executed in the analyzed intelligent transportation system with the sequence tables

defined in the previous step.

The example of the micro-program for our execution environment is shown in Figure 7. This program enables auditing three intersections on Lotnicza and Legnicka street i.e. Na Ostatnim Groszu, Bajana and Hutnicza.

7. Run Global Optimization Algorithm with the ArsNumerica execution environment with the above settings of the network, sequence tables for the intersections included and microprograms.

We use as parameters of the optimization only some entries of sequence tables of three cross-sections. The optimization concerned 6 variables in sequence tables shown in Figure 5. The variables enabled us to parametrize the duration of all main green phases and corresponding red phases and also prolongation for the intelligent transportation system. Offsets were taken from the intelligent transportation system such as shown in step 5. of this procedure. The objective function was the sum of all queue lengths during the time of the simulation at all intersections.

8. If the traffic is recovered by the simulation with nominal settings (those corresponding to the real intelligent transportation system settings) then

- (a) if the objective function can be radically decrease then something wrong is in the analyzed intelligent transportation system
- (b) if the objective function cannot be improved the real setting seems to be OK.

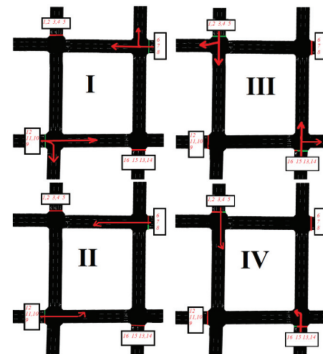


Fig. 6. Phases for the "Na Ostatnim Groszu str." intersection corresponding to the sequence table presented in Fig. 5b [own study]

As is clear from Figure 8 no substantial decrease of the objective function was achieved during optimization the Genetic algorithm using ITS flows. When the inflow is increased in the entering intersection and outflow increased at other three big intersections, then as it is seen at Figure 9 the objective function can be improved even by 25%.

Table 1. Offsets in the Intelligent Transport System in Wrocław for Lotnicza and Legnicka Street [own study]

Crosssection ID in the ITS	Crossed Street Name	Offset in [s]
1	Jana Pawła II	50
262	PdP	46
71	Rybacka	71

98	Pl. Strzegomski	27
100	Złotoryjska	38
69	Zachodnia	78
76	Niedźwiedzia	35
75	Małopanewska	22
73	Wejherowska	39
91	Milenijna (Na Ostatnim Groszu)	34
88	Horbaczewskiego	82
144	Bajana	35
74	Hutnicza	83
66	Metalowców wsch.	71
1066	Metalowców zach.	66
222	Pilczyce	41

7. Conclusion

In this paper we presented a concept of the methodology of numerical auditing of intelligent transportation system traffic light settings and their dynamic behavior. We presented a four level environment that enables us to simulate a dynamic behavior of the intelligent transportation system. This is done by the ability of the environment to execute to each intersection its micro-program that can dynamically – according to the sequence table – change duration of phases, keeping phases coordination

```

1 set tabtime18 455455
2 set tabsek18 rrrGGGGSSYYYYYYYGGGrrrrrSYYYYYYYO
3 set idS18 18
4 set tabtime17 455455
5 set tabsek17 rrrGGGGSSYYYYYYYGGGrrrrrSYYYYYYYO
6 set idS17 17
7 set idD1 validation_ST6
8 set idD2 validation_ST5
9 set idD3 validation_ST7
10 set idD4 validation_ST2
11 set idD5 validation_ST1
12 set idD6 validation_ST3
13 set IS 18
14 set stc 0
15 set fc 0
16 set c 17
17 set prog 1800
18 set TS 3400
19 set pom 0
20 set sekwencja 0
21 set vehn1 0
22 set vehn2 0
23 while ( <= $tco STS )
24 {
25     while ( <= $c SLS )
26     {
27         gettab tabsekfc pom $sekwencja
28         setlights $c $pom
29         set c [ + $c 1 ]
30     }
31     gettime tabtime17 time $sekwencja
32     set sekwencja [ + $sekwencja 1 ]
33     set c 17
34     if ( > $sekwencja 3 ) { set sekwencja 0 }
35     if ( > $tco 900 ) {
36         if ( > $vehn1 $prog ) {
37             set time [ + $time 5 ]
38             set vehn1 0
39         }
40         if ( > $vehn2 $prog ) {
41             set time [ + $time 5 ]
42             set vehn2 0
43         }
44     }
45     while ( <= $fc $time1 )
46     {
47         step
48         set fc [ + $fc 1 ]
49     }
50     set stc [ + $tco $tco ]
51     get $idD1 pom
52     set vehn1 [ + $pom $vehn1 ]
53     get $idD2 pom
54     set vehn1 [ + $pom $vehn1 ]
55     get $idD3 pom
56     set vehn1 [ + $pom $vehn1 ]
57     get $idD4 pom
58     set vehn2 [ + $pom $vehn2 ]
59     get $idD5 pom
60     set vehn2 [ + $pom $vehn2 ]
61     get $idD6 pom
62     set vehn2 [ + $pom $vehn2 ]
63 }

```

Fig. 7. Micro-program to audit three intersections along Legnicka Street in Wrocław [own study]

and drivers safety. The software that implements this environment was developed by the ArsNumerica Group. In the paper the auditing procedure was described and applied to audit ITS in the city of Wrocław. Using a genetic algorithm on the top of our environment we showed numerically that light settings on the main arteria of the city – namely Legnicka Street – are very well suited for flows currently measured. It means that the objective function in the optimization could not be substantially decreased for parametrized sequence tables for three intersections. The same was not true for flows increased by 15%. In this case queues could have been shortened by 25%.

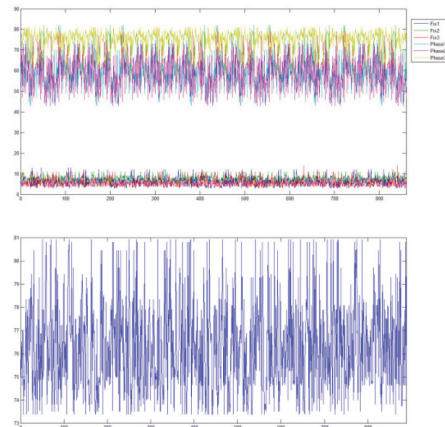


Fig. 8. (a) Parameters being optimized, (b) The objective function. Nominal settings of traffic flows [own study]

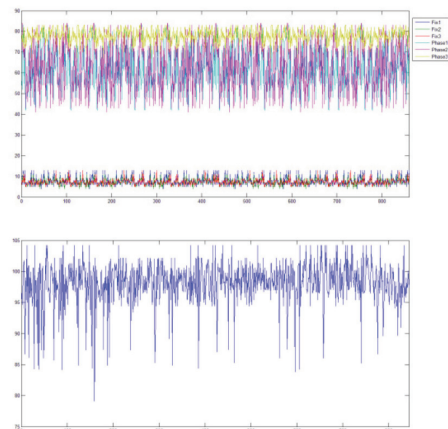


Fig. 9. (a) Parameters being optimized. (b) The objective function. Results for calculations performed for Legnicka Street with flows obtained from the Wrocław ITS in May 2014 increased inflows by 15% at the beginning of the arteria (South) and summed 15% of outflows at the main intersections [own study]

Bibliography

- [1] RATROUT N. T., RAHMAN S. M.: A Comparative Analysis of Currently used Microscopic AND Macroscopic Traffic Simulation Software, The Arabian Journal for Science and Engineering, Vol. 34, No. 1B, (2009)
- [2] KRAJZEWCZ D., et al.: Recent Development and Applications of SUMO – Simulation of Urban MObility, International Journal On Advances in Systems and Measurements, Vol. 5, No 3&4, Dec. 2012, 128-138.
- [3] SZYMAŃSKI A., et al.: Two methods of calculation of the origination destination matrix of an urban area, Raport W04/P-007/15, Wrocław University of Technology, (2015)
- [4] <http://geoportal.gov.pl/> [date of access: November 2014]
- [5] <http://www.openstreetmap.org/> [date of access: November 2014]

- [6] Biuro inżynierii transportu, Pentor Research International, Kompleksowe Badania Ruchu – Wrocław 2010, 2010 (in Polish).
- [7] J. HERRERA C.: Assessment of GPS-enabled smartphone data and its use in tra-c state estimation for highways, PhD Thesis, University of California, Berkeley, (2010)
- [8] FRANCERIERS E., LIVER K.: Centralized traffic management system as response to the effective realization of urban traffic fluency, Archives of Transport System Telematics, Vol. 4, Issue 4, (2011)
- [9] MBIYDZENYUY G.: Quantitative Assessment of Intelligent Transport Systems for Road Freight Transport, Blekinge Institute of Technology, BTH, Sweden, (2013)
- [10] MADEJ Ł.: Adaptive traffic control, Eng. Thesis, Department of Electronics, Wrocław University of Technology, (2014)
- [11] <http://oldblog.antirez.com/post/picol.html> [date of access: November 2014]
- [12] RUSSENSCHUCK S., et al.: Integrated design of superconducting magnets with the CERN Fields Computation Program ROXIE, CERN, LHC Report 446, (2000)
- [13] TREIBER M., KERSTING A.: Traffic Flow Dynamics, Data, Models and Simulation, ISBN: 978-3-642-32459-8, Springer, (2013)
- [14] NAGATANI T.: Vehicular traffic through a sequence of green-wave lights. Physica A (2007), nr 380, s.503-511
- [15] TYMIŃSKI P.: Optimization of the traffic flow along the sequence of intersections, M.Sc. Thesis, (2014)
- [16] SRIVASTAVA S., DEB K.: A Genetic Algorithm Based Augmented Lagrangian Method for Computationally Fast Constrained Optimization, Computational Optimization and Applications, 53.3: s. 869-902, (2012)
- [17] PEREZ R. E., BEHDINAN K.: Particle Swarm Optimization in Structural Design. Computers and Structures, nr 85, s. 1579-1588, (2007)
- [18] <http://www.mathworks.com/products/matlab/> [date of access: November 2014]