This article can be cited as R. P. Alvarez Gil, Z. C. Johanyák and T. Kovács, Surrogate Model based Optimization of Traffic Lights Cycles and Green Period Ratios using Microscopic Simulation and Fuzzy Rule Interpolation, International Journal of Artificial Intelligence, vol. 16, no. 1, pp. 20-40, 2018.

Surrogate Model based Optimization of Traffic Lights Cycles and Green Period Ratios using Microscopic Simulation and Fuzzy Rule Interpolation

Rafael Pedro Alvarez Gil[1], Zsolt Csaba Johanyák[2] and **Tamás Kovács[3]**

John von Neumann University, GAMF Faculty of Engineering and Computer Science,
Department of Information Technology, Izsáki út 10., H-6000 Kecskemét, Hungary

[1] Email: alvarez.rafael@gamf.uni-neumann.hu

[2] Email: johanyak.csaba@gamf.uni-neumann.hu

[3] Email: kovacs.tamas@gamf.uni-neumann.hu

ABSTRACT

In this paper, first the travel and delay times in a road ending in a traffic light are determined under different traffic flows and traffic light cycles using a microscopic traffic simulator. The obtained results are analyzed and compared with the results of other models reported in the literature. In addition, the optimal traffic light cycle times under different loads are determined for one road and a method is shown to obtain the optimal green period ratios of a traffic light as well as the cycle time in an intersection. Finally, the generation of a fuzzy model is presented and a methodology is suggested, where the fuzzy system is applied as a surrogate model for the determination of the optimal green period ratios and traffic light cycle times.

Keywords: fuzzy rule interpolation, traffic lights cycles optimization, microscopic traffic simulator, IDM, surrogate model.

Mathematics Subject Classification: 03B52, 93C42, 90B99

Computing Classification System: Applied computing---Operations research--- Transportation

1. INTRODUCTION

The optimization of traffic lights cycles has been intensively investigated in the last decade. The calculation or measurement of delays caused by a road ending in a traffic node containing traffic lights is one of the key issues in this field. Different models and methods have been used to solve this task so far, and the results may be significantly different from each other when the node is overloaded or close to the overloaded state.

In this problem it is important to distinguish between the optimization of particular traffic lights of an intersection and the optimization of a network of traffic lights containing traffic nodes of a larger area.

The research being reported in this paper had two goals. The first one was to apply the Intelligent Driver Model (IDM) (Treiber, Hennecke, Helbing, 2000) based microscopic traffic simulator (Kovács et al., 2016) developed at our department, that we have called IT MICROSIM, in order to determine the traffic delay in a simple test traffic node equipped with traffic lights, and to compare the results to those of other models. The optimal cycle time for traffic signals can be determined later based on delay values. The second goal was to create a fuzzy model that describes the connection between the total delay at an intersection and its influential factors as well as to describe a methodology that uses a fuzzy model as a surrogate model for the determination of the optimal green period ratio and traffic light cycle time in case of given traffic flow values.

The rest of this paper is organized as follows. Section 2 recalls some significant results obtained in field of fuzzy logic based traffic light related solutions. Section 3 presents the significant parameters of IT MICROSIM. Section 4 discusses the simulation results related to a road ending in a traffic light. Section 5 investigates the issues of the optimal cycle time. Section 6 extends the investigation to a whole intersection. Section 7 presents the fuzzy model generation and the optimization of the green time ratios and traffic light cycle of the intersection. The conclusions are drawn in section 8.

2. RELATED WORKS

Fuzzy controllers have been proposed in several works for the optimization of a traffic light at an intersection.

As early as 1977 Pappis and Mamdani (Pappis, Mamdani, 1977) presented a fuzzy logic controller for a traffic light at a single intersection of two one-way streets considering average delay of vehicles as performance criterion. They compared the fuzzy logic controller to the conventional effective vehicle-actuated controller showing that the use of a fuzzy logic controller results in a better performance.

More recently Karakuzu and Demirci (Karakuzu, Demirci, 2010) designed a fuzzy controller for a traffic light at a four-directional intersection with two bidirectional arteries, and developed a simulator hardware to test the controller. They designed a fuzzy controller with two inputs, i.e. the lengths of the queues of waiting vehicles in each artery, and two outputs, i.e. the lengths of the green and red lights of the main artery.

Postorionoa and Versacia (Postorinoa, Versacia, 2014) proposed a fuzzy demand-responsive control approach based on real-time measures and traffic lights adaptation to speed up traffic flows and reduce delay at signalized intersections.

Baydokht, Noori, Siamak, and Azhang zad (Baydokht, Noori, Azhang zad, 2014) presented a mathematical model to describe the control of the traffic lights at a four-way intersection with eight traffic lights. Due to the limitations of giving precise and constant values to parameters of the mathematical model a fuzzy model was used. The fuzzy model more closely resembled real life situations and therefore it proved to be practical.

Fuzzy controllers have been used to control freeways entrances (Murat, Cakici, Yaslan, 2012) as well. In this case the objective was to avoid the congestion of the highways. Delay control was also introduced, discouraging drivers from entering the freeway at peak times, and encouraging them to use them during off-peak hours.

Other computational intelligence based methods have been also proposed for the traffic lights problem like case-based reasoning systems (Castan et al., 2014), genetic algorithms (Teo, Kow, Chin, 2010) as well as non-computational intelligence based techniques and methods (Khanjary, Navidi, 2012).

Traffic lights optimization in traffic networks has been investigated by several researchers as well. For example, Rouphail et al. (Rouphail, Park, Sacks, 2000) suggested genetic algorithm using the microscopic traffic simulator CORISM for the coordination of nine intersections of Chicago. García-Nieto et al. (García-Nieto, Alba, Carolina Olivera, 2012) presented a Swarm Intelligence approach complemented with a microscopic traffic simulator to find optimal traffic lights timing programs for areas with different characteristics of Malaga and Seville. Metaheuristics were also applied to find efficient traffic light programs aiming the reduction of fuel consumption and harmful emissions in the cities of Malaga and Seville (García-Nieto, Ferrer, Alba, 2014).

3. PARAMETERS OF THE IDM MODEL IN IT MICROSIM

In the microscopic traffic simulator IT MICROSIM (Kovács et al., 2016) used in course of this research the speed and instantaneous acceleration of each vehicle is determined by the so-called “car-following model”, where the instant attributes of each vehicle depend on the kinematics of the vehicle moving in its front.

The two most popular models in this field are the Intelligent Driver Model (IDM) (Treiber, Hennecke, Helbing, 2000), (Treiber, Helbing, 2001) and the Wiedemann Model (Wiedemann, 1974). In both models the behavior of the vehicle is defined by a parameter set. Here the most important environmental inputs are the kinematics variables, i.e. the relative position and the speed of the vehicle traveling in front of the current vehicle. These models have also been improved in the last few years (Wiedemann, 1991), (Derbel et al., 2012), (Derbel et al., 2013), and comparative tests of the two models have also been carried out (Apeltauer et al., 2013).

The dynamic model of the developed simulation software has been designed to be freely interchangeable and upgradeable. The current measurements were carried out in the main IDM model, in which the instantaneous acceleration of a vehicle is given by the equations (1) and (2)

$$\frac{dv}{dt} = a\left[1-\left(\frac{v}{v_0}\right)^{\delta}-\left(\frac{s^*(v,\Delta v)}{s}\right)^2\right], \tag{1}$$

$$s^*(v,\Delta v) = s_0 + \max\left[0,\left(vT+\frac{v\Delta v}{2\sqrt{ab}}\right)\right], \tag{2}$$

where v is the speed of the vehicle, Δv is the difference between the speed of the vehicle in its front and the speed of the current vehicle, s is the distance between the two vehicles and s^* is the ideal following distance.

The optimal values of the model parameters were determined in course of the investigation reported in (Kovács et al., 2016). They are:

- the maximum acceleration (a = 1.6 m/s^2),
- the minimum acceleration, i.e. the maximum braking deceleration (b = 2 m/s^2),
- the maximum speed (v_0 = 55 km/h),
- the follow-up time (T = 0.86 s),
- the distance between non-moving vehicles (s_0 = 2 m),
- the acceleration exponent (δ = 4).

These parameter values determine how realistic the movement of vehicles will be in different simulated situations, i.e. how the simulation results are in correspondence with measurements taken in real traffic flow.

An important component of the dynamic module is the controller for lane changes of vehicles. This is currently partly based on the model proposed by Kesting (Kesting, Treiber, Helbing, 2007). If the driver believes that an adjacent lane is more advantageous for faster progress the lane change will start using Kesting's model. However, the lane change can occur independently from the model as well. When the path-tracking unit indicates the vehicle to follow a chosen direction it will have to change the lane.

4. SIMULATION BASED INVESTIGATION OF AVERAGE DELAY TIME VALUES ON A ROAD

In our previous work (Johanyák, Alvarez, 2017) we investigated average delay times on a road that ends at an intersection with traffic lights (Figure 1). Experiments were carried out using IT MICROSIM with a wide range of parameter values. In this section we recall the main points of the investigation extending it with further discussion of the results and comparing them to results obtained by other models available in the literature.

,

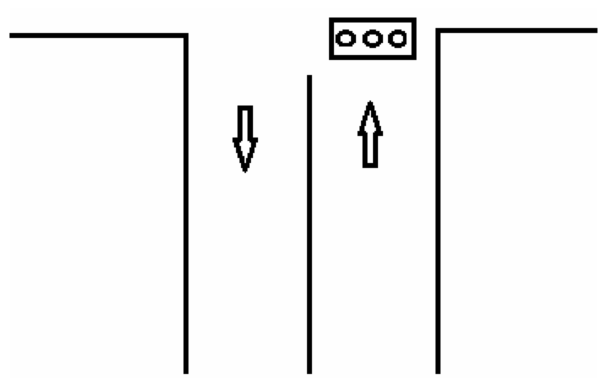

Figure 1. Road ending in node with a traffic light (Johanyák, Alvarez, 2017).

The simulation runs were carried out considering a 500 m long road ending in a node with a traffic light and randomly arriving vehicles. The traffic light cycle was composed from two periods with the same durations, the first one being the green period, and the second one being the red period. The green period was formed from three sections (1) green light time, (2) yellow light (3 seconds), and (3) red light (2 seconds). The red period contained only red light, simulating the time when the cross road gets the green period. Although in this section only the results related to the 50% green period ratio of the total cycle time are presented, experiments were carried out for 25%, 33%, 67% and 75% green period ratios as well.

Although the simulator software makes possible the measurement of travel time, in practice delay time is used more commonly for examination issues related to the traffic lights. It is calculated from the travel time by subtracting the free-flow travel time, i.e. the travel time without traffic light. For example, if vehicles travel at 55 km/h, the theoretical free-flow time is 32.73 s. Besides, similarly to the real-world situation, in the micro-simulator the free-flow travel time also depends on the traffic flow value, as it is shown in Table 1.

Table 1: Free-flow travel times (Johanyák, Alvarez, 2017).

Flow [vehicle/h]	Free-flow travel times [s]
200	34.58
400	35.59
600	36.19
800	36.86
850	37.03
900	37.17
950	37.30
1000	37.41
1050	37.50
1100	37.53
1200	37.58

Travel times were obtained in the micro-simulator under condition of stationary traffic inflow. Delay times are shown in Table 2 and in Figure 2. Similar tables also were obtained for the cases when the duration of the green period was 25%, 33%, 67% and 75% of the total duration of the traffic light cycle.

Investigating the results one can recognize that the delay time values obtained using the micro-simulator meet the Beckmann model (Beckmann, McGuire, Winsten, 1956) until the traffic light cycle capacity. They can be described by functions that are convex, continuous, non-negative and non-decreasing and which depend on the current traffic. For traffic flow values greater than the traffic light cycle capacity, the delay increases until it reaches its maximum value when the road is filled with waiting vehicles.

The capacity of a traffic light cycle is near the value of the abscissa of the inflection point of the graph corresponding to the cycle.

One can see in Figure 2 that the higher the traffic light cycle time is, the larger the capacity of the traffic light cycle becomes.

Table 2: Average delay times for the 50% green time period ratio [s] (Johanyák, Alvarez, 2017).

	Traffic light cycle time [s]				
Flow [v/h]	50	76	100	150	200
200	**10.63**	14.90	17.23	24.42	31.28
400	**12.81**	17.03	19.60	27.12	35.65
600	**16.22**	19.62	22.34	31.53	39.41
800	44.45	**26.45**	27.62	35.58	45.52
850	185.68	31.67	**30.27**	38.11	46.94
900	218.98	45.83	**35.21**	42.09	49.38
950	217.97	162.33	56.06	**47.76**	52.22
1000	221.38	191.40	155.29	62.90	**61.67**
1050	220.07	191.98	176.02	144.25	**95.30**
1100	221.18	192.36	177.33	162.73	**153.34**
1200	222.56	189.96	174.87	162.02	**152.21**

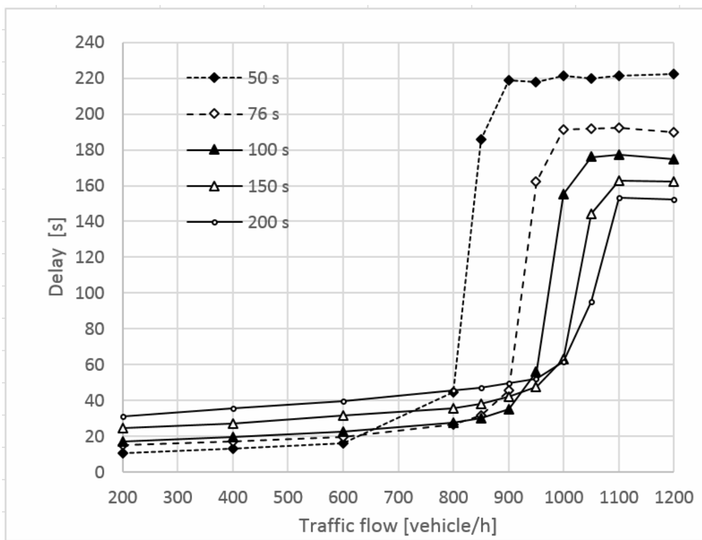

Figure 2. Average delay times (Johanyák, Alvarez, 2017).

A previous work in this field written by Dion, Rakha and Kang (Dion, Rakha, Kang, 2004) analyzed and compared several models used to determine the delay time. In this paper, our results have been compared with the following three methods.

a. *Webster's stochastic model* (Webster, 1958)

The model calculates the delay time with a three-factor formula. The first member is the delay caused by uniform arrivals while the second one considers additional delay values due to the random arrival of vehicles. The third factor is a correction factor.

b. *Queueing model with supersaturation* (Dion, Rakha, Kang, 2004)

The delay is calculated considering uniform arrivals. The model allows that the number of vehicles arriving in a cycle to be more than the number of vehicles that can go through the traffic light during the cycle.

c. *Microscopic traffic simulation system INTEGRATION* (Karakuzu, Demirci, 2010)

The INTEGRATION microscopic traffic simulator includes a dynamic traffic simulator and a traffic assignment model. The simulator was used in (Dion, Rakha, Kang, 2004) to obtain delay estimates, which were obtained by calculating the difference between the simulated travel times and the ideal travel times at free speed.

The comparison of the delay times calculated by the different models is shown in Figure 3, where a traffic light cycle of 76 seconds was used in correspondence with the traffic light cycle applied in (Dion, Rakha, Kang, 2004). The vertical dashed line gives the value of the traffic flow corresponding to the traffic light cycle capacity.

The comparison of the delay times calculated by the different models is shown in Figure 3, where a traffic light cycle of 76 seconds was used in correspondence with the traffic light cycle applied in (Dion, Rakha, Kang, 2004). The vertical dashed line gives the value of the traffic flow corresponding to the traffic light cycle capacity.

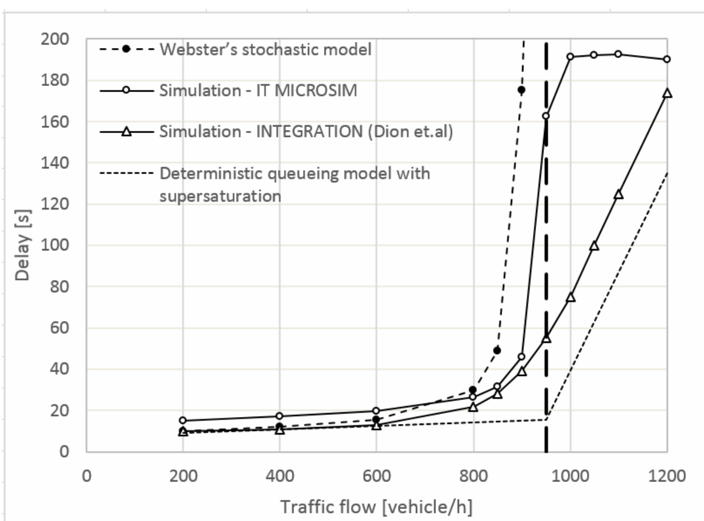

Figure 3. Comparison of models for delay times estimations.

Figure 3 shows that in Webster's model the delay time goes to infinity when traffic flow is approaching the capacity of the traffic light cycle (vertical dashed line). Meanwhile, the queueing model with supersaturation and the INTEGRATION system can handle the case when traffic flow exceeds the capacity of the traffic light calculating the delay time in this case as well. The behavior of the delay times in IT MICROSIM was described above, reaching a maximum value of 190 s in a 500 m long road. This value is the time required to travel the filled road with the specific traffic light cycle capacity.

In the left vicinity of the capacity the results obtained by IT MICROSIM are closer to the results calculated by the stochastic model than to the values determined by the other two models. If the length of the road is not limited and the traffic flow exceeds the traffic light cycle capacity the number of waiting vehicles will grow indefinitely, and thus the delay will increase as well.

5. OPTIMAL CYCLE TIME OF THE TRAFFIC LIGHT FOR ONE ROAD

Still focusing on the traffic on only one road (e.g. the main road) one could be interested in determining the optimal value of the traffic light cycle in order to ensure the minimal average delay times for the vehicles on this road. In the case of the previously presented simulations the optimal delay times for the different traffic flow values are marked in bold in Table 2. One can see that the 50 seconds traffic light cycle time is the best for low traffic flow values, but as the traffic flow value increases, longer and longer traffic light cycle times become the best options.

This phenomenon is also illustrated in Figure 4, where the graphs for five traffic flow values (600, 800, 850, 950 and 1000 vehicle/h) are shown. For each of them the best traffic light cycle is different: 60, 76, 100, 150 and 200 s, respectively. In case of the other green period ratios (25%, 33%, 67%, 75%) the modification of the average delay time in function of the cycle time and traffic flow is similar with different actual values.

IT MICROSIM ensures a proper architecture to determine the optimal cycle time values for all green period ratios and traffic flow values by using an arbitrary search technique or design of experiments (DOE) methodology in case of a static traffic flow. However, its application in a real-time system is not feasible owing to the high time demand of the simulations.

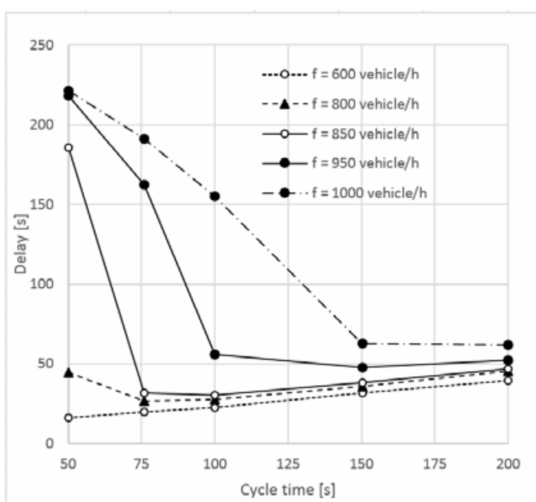

Figure 4. Average delay time in function of the traffic light cycle time.

6. OPTIMAL GREEN PERIOD RATIO AND TRAFFIC LIGHT CYCLE OF AN INTERSECTION

In order to determine the optimal green period ratio and traffic light cycle of an intersection we consider an intersection with two two-way roads. Both roads have two lanes with opposite travel directions. The intersection has a traffic light system with a cycle similar to the one described in Section 4, and the two parts of the cycle may have different durations, i.e. different green period ratios. In the first part of the cycle one of the roads gets the green period in both directions, in the second part the other road comes. The traffic flow of a road is the maximum of the traffic flows in both directions. Figure 5 illustrates the intersection.

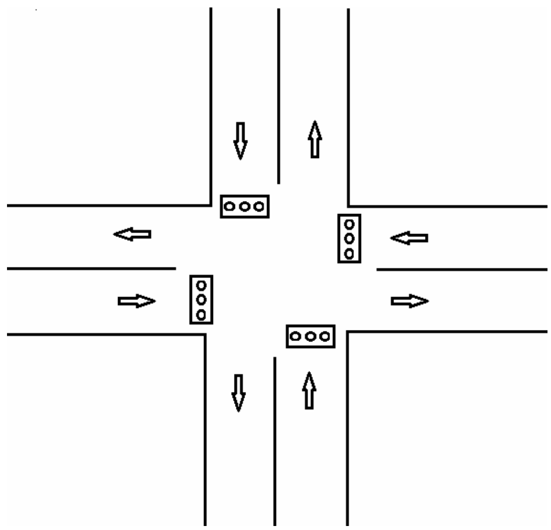

Figure 5. The test intersection.

Knowing traffic flows in both roads a separate table can be built for each possible traffic light cycle time containing the average delay times for each road and for each considered green time ratio based on the simulations carried out previously. For example, Table 3 was created for the 50 seconds traffic light cycle time supposing that the traffic flow is 200 vehicle/h in the W – E (West – East) road and 400 vehicle/h in the S – N (South – North) road. The table also contains the absolute values of the differences between the delay times of the two roads.

The quality of the service provided by the traffic light system can be measured by the total delay time (T_{td}), which is

$$T_{td} = TF^{WE} \cdot T_d^{WE} + TF^{SN} \cdot T_d^{SN}, \quad (3)$$

where TF^{XX} is the current traffic flow on the road *XX*, and T_d^{XX} is the average delay time on the same road. T_{td} indicates the sum of all delay times related to the examined intersection for a period of one hour.

The smallest T_{td} value in Table 3 is 6 697 seconds, which was measured when the green time ratio was 33% (W – E) - 67% (S – N). Similar tables to Table 3 were created for the same traffic flows with 100, 150 and 200 seconds long traffic light cycles, but in none of them a total delay inferior to the one obtained for a cycle of 50 seconds was obtained. The 50 seconds traffic light cycle with a 33% (W – E) - 67% (S – N) green time ratio is the optimal traffic light setting to reduce the total delay time among the performed simulations.

Table 3: Delays for different green time ratios (Traffic flows: 200 vehicle/h in W – E, 400 vehicle/h in S – N; Traffic light cycle time: 50 s).

Green time ratios [s]		Delay time [s]			Total delay [s]
W-E	S-N	W-E	S-N	Diff.	
75%	25%	4.15	734.34	730.19	294 566.00
67%	33%	6.71	39.53	32.82	17 151.00
50%	50%	10.63	12.81	2.18	7 250.00
33%	67%	19.00	7.25	11.75	6 697.00
25%	75%	30.20	4.37	25.83	7 788.00

Although the above presented solution ensures the minimal total delay in some cases a “more fair” ratio could be demanded by requesting similar or the closest possible delay times for both roads. This requirement can be fulfilled by the 50% - 50% green time ratio in case of Table 3. Here by choosing the second best T_{td} value one can ensure a significant reduction of the difference between the average delay times of the two roads.

For larger values of traffic flows we got different optimal cycle time values from tables similar to the Table 3 obtained from the simulations. For example, for 600 vehicles/h on a road and 800 vehicles/h on the other, the optimal cycle time is 100 seconds long with 50% - 50% green time ratio.

This method can provide an interesting alternative to other established methods (e.g. (Péter, 2012)). The above presented simulation results can be used in the green time ratio optimization of different intersection types as well.

7. FUZZY MODEL BASED DETERMINATION OF THE OPTIMAL GREEN PERIOD RATIO AND TRAFFIC LIGHT CYCLE

The previously described approach gives a simple solution for the determination of the optimal green time ratio and traffic light cycle time in case of the presented intersection type. However, it also has two shortcomings, namely it can choose only from a predetermined set of options and its application is cumbersome.

To overcome the above mentioned problems a computational intelligence based approach is considered in this section. Computational intelligence based methods have become widely applied solutions in case of several problem types like fuzzy modeling (Precup et al., 2015), fuzzy control (Blažič et al., 2010), (Precup et al., 2013a), (Precup et al., 2013b), cognitive maps (Vaščák, 2012), natural language processing (Kovács, Szabó, 2017), failure analysis (Pokorádi, 2015), mechanical properties prediction (Karkalos et al., 2017), text classification (Sharif et al., 2017), etc. Based on the wide applicability of the approach a fuzzy model was created that describes the relation between the total delay time and its four influential factors.

$$T_{td} = f\left(TF^{WE}, TF^{SN}, GPR^{WE}, TLC\right), \tag{4}$$

where GPR^{WE} is the green time ratio for the West-East direction, and *TLC* is the traffic light cycle time. The green time ratio of the South-North direction being a dependent variable it was omitted from the inputs of the system. To keep low the complexity of the system a sparse rule base was created and the fuzzy rule interpolation based inference technique presented in Section 7.1 was applied. The parameters of the rule base were optimized by the Particle Swarm Optimization algorithm, which is described in Section 7.2. The details of the fuzzy model are presented in Section 7.3. Section 7.4 describes how the *TLC* and *GPR* values can be found for the smallest possible T_{td} using the fuzzy system.

7.1 Fuzzy Inference Based on Rule Interpolation

Although the classical fuzzy inference techniques like Zadeh's (Zadeh, 1965), Mamdani's (Mamdani, Assilian, 1975), or Takagi-Sugeno's (Takagi, Sugeno, 1985) are widely applied in practice they cannot properly handle cases when the available rule base does not ensure a full coverage of the input space, i.e. there is at least one point of the input space which is not overlapped (covered) by any rule antecedents. Rule bases having this characteristic are called sparse ones. Figure 6 illustrates a sparse rule base in case of a two-dimensional input space where all the membership functions are trapezoidal ones and thus the rule antecedents can be represented as truncated pyramids. Here for example in case of the input values (0.1, 0.1) there is no rule whose antecedent part would overlap the input point at least partially, i.e. there is no rule which could be applied.

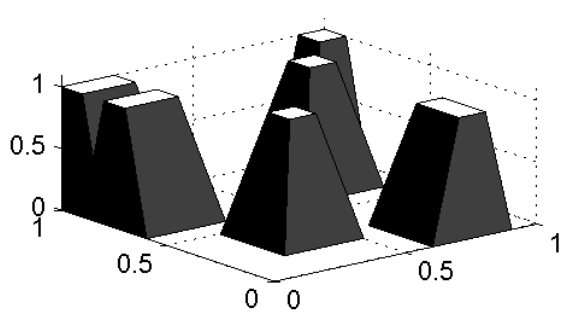

Figure 6. Antecedent space of a sparse rule base.

Sparse rule bases can arise either owing to the lack of information or due to an intentional approach. The first situation can occur both in case of human created and in case of automatically generated rule bases. The second situation could appear when the full coverage of the input space could be reached only with a huge number of rules, which would slow down the system considerably, and therefore the creator of the rule base opted for a sparse solution.

The discovery of the existence and sometimes necessity of the sparse cases led to the development of interpolation based fuzzy inference methods. The key idea here is that the output of the fuzzy inference (conclusion) is determined as a result of interpolative calculations that take into consideration the similarity between the current input (observation) and the antecedent parts of the known rules. Although the related research began in the early 1990s fuzzy rule interpolation (FRI) is still an intensively investigated field.

FRI methods form two main groups, i.e. the one-step and the two-step methods. The one-step methods determine the conclusion directly from the observation, and thus the creation of an auxiliary rule is not necessary. Typical members of this group are e.g. the linear interpolation (Kóczy, Hirota, 1993), and the vague environment based reasoning FIVE (Kovács, 2006).

In contrast, two-step FRI methods apply the concept of Generalized Methodology of fuzzy rule interpolation (GM) (Baranyi et al., 2004). In the first step they interpolate a new rule in the position of the observation and next they calculate the conclusion using a special single rule reasoning technique. Typical members of this group are the technique family suggested in (Baranyi et al., 2004), LESFRI published by (Johanyák, Kovács, 2006), and the transformation based technique (Chen, Ko, 2008).

The FRI method chosen for the current research was the Fuzzy Rule Interpolation based on Method of Least Squares and vector calculations (VLESFRI) (Johanyák, 2011) is a vector calculation and α-cut based FRI method, which follows the two-step approach. Its algorithm is presented below.

1. *Create a list of all breakpoint levels considering all antecedent and consequent partitions as well as the observation sets.*
2. *Calculate the α-cut for each level in case of each fuzzy set.*
3. *In each antecedent partition*
    a. *Calculate the Euclidean distances between the observation and the fuzzy sets of the partition based on their reference points.*
    b. *Interpolate a new fuzzy set in the position of the observation α-cut wise based on the weighted average of the α-cuts of the fuzzy sets.*
4. *For each rule*
    a. *Calculate the Euclidean distance between the rule antecedent and the observation considering the reference points.*
5. *Calculate the position of the rule consequent (interpolated rule) set using extended Shepard interpolation (Shepard, 1968).*
6. *Interpolate a new fuzzy set in that position based on the weighted average of the α-cut endpoints of the fuzzy sets in the consequent partition.*
7. *Modify the shape of this set with the average of the deviations between the observation sets and the antecedent sets of the new rule.*
8. *Defuzzify the resulting set.*

Please refer to (Johanyák, 2011) for a detailed description of VLESFRI.

7.2. Particle Swarm Optimization

Particle Swarm Optimization is a population based heuristic optimization method. Its original version was proposed by Kennedy and Eberhart (Kennedy Eberhart, 1995). In PSO the population is called swarm while the individuals of the population representing candidate solutions are the particles.

A particle is represented by two vectors. One contains the actual values of the parameters whose optimization is the aim of the method. This vector is also mentioned in the literature as position vector. The other one is the so called velocity vector of the particle, which is used for the determination of the next position vector of the particle.

PSO is an iterative process. Its key idea is that the particles move in the search space in order to find the optimal solution. It means that they have a new position in each iteration cycle, which is mainly determined by their previous position, the best position of the same particle (its previous position vector that ensured the best fitness value so far), and the position of the so far overall best particle. The iteration continues until the stopping criterion is met. The algorithm is described by the following steps.

1. *Generate an initial swarm with N particles (position and velocity vectors).*
2. *Evaluate the performance (fitness) of each particle using* (6).
3. *Store for each particle its best position so far.*
4. *Store the position vector of the best particle.*
5. *Update the velocity vector of each particle.*
6. *Update the position vector of each particle by adding the velocity vector to the position vector.*
7. *GO TO step 2 if stopping criterion is not met.*

The initial velocity vectors of the particles (d_{ij}, $j=\overline{1,N}$) are defined as random values from the [-0.5,0.5] interval. The new values of the velocity vectors in step 5 are calculated by

$$d_{ij} = c_3 \cdot d_{i-1,j} + c_1 \cdot r_1 \cdot \left(PB_j - P_{i-1,j}\right) + c_2 \cdot r_2 \cdot \left(OB - P_{i-1,j}\right), \tag{5}$$

where i is the iteration number; c_1, c_2 and c_3 are constant parameters of the method; r_1, $r_2 \in [0,1]$ are random values; PB_j is the best position vector of the particle j so far; OB is the position vector of the overall best particle; $P_{i-1,j}$ is the position vector of the j^{th} particle in the previous iteration; and $d_{i-1,j}$ is the velocity vector of the j^{th} particle in the previous iteration.

The basic stopping criterion is the number of allowed iterations (n_i).

7.3. Fuzzy Model Generation

As presented before, several experiments were carried out to get enough descriptive information about the variation of the total delay time (T_{td}) in function of its four influential factors. Based on the simulation runs 1500 data tuples were identified, which set was divided into three groups using random selection. First, 150 data points were selected for validation purposes, followed by the selection of 150 data points for test purposes, and finally, the remaining 1200 data rows were kept for training of the fuzzy system.

The fuzzy model was generated in an iterative process starting with an initial rule base containing only two rules corresponding to the minimum and maximum experienced outputs. In each iteration cycle the actual fuzzy system was evaluated against both the training and the validation data sets, but only data rows belonging to the training set were used for the creation of the next version of the fuzzy system and for the creation of a new rule in the position where the difference between the actual output and the desired output was maximal. We used root mean squared error expressed in percentage of the output range (RMSEP) as performance indicator.

$$PI = \frac{\sqrt{\frac{\sum_{i=1}^{n_d}(y_i - \hat{y}_i)^2}{n_d}}}{R_o} \cdot 100\,, \tag{6}$$

where n_d is the number of the used data points, R_o is the range (width) of the output, y_i is the i^{th} desired output value and $\hat{y}_i$ is the i^{th} calculated output value. The applied algorithm is presented below.

1. *Create a rule describing the lowest output value of the system.*
2. *Create a rule describing the highest output value of the system.*
3. *Create a new rule in that position where the difference between the actual output of the system and the desired output is maximal.*
4. *In each partition*
    a. *IF the new fuzzy set created with the new rule is identical or very close and similar to a previously created fuzzy set THEN*
        i. *merge the two sets.*
5. *Optimize the position of the fuzzy sets of the new rule.*
6. *GO TO step 3 if performance criteria are not met.*

We decided to use at the beginning triangle shaped membership functions with support width defined as 15 % of the range of the actual partition. Later, some of the sets became trapezoidal shaped as a result of the merging process in step 4.a.i. The closeness and similarity measurement was done α-cut wise, which in case of piece-wise linear membership functions can be reduced to the examination of the breakpoint levels. It was expressed as the average difference measured at the endpoints of the α – cut endpoints

$$\overline{d_{ij}} = \frac{\sum_{\alpha}\left(\left|\inf\left\{\left[A_i\right]_{\alpha}\right\} - \inf\left\{\left[A_j\right]_{\alpha}\right\}\right| + \left|\sup\left\{\left[A_i\right]_{\alpha}\right\} - \sup\left\{\left[A_j\right]_{\alpha}\right\}\right|\right)}{2n_{\alpha}}, \quad (7)$$

where A_i and Aj are the two fuzzy sets that are examined, α is the actual level, n_α is the number of α levels. The key idea of merging is that if $\bar{d}$ is below a threshold expressed in proportion of the range of the actual partition the two sets are merged. Merging means that the given two sets are placed by a new set that wraps the two original fuzzy sets. The new set is also defined α-cut wise

$$\inf\left\{\left[A_n\right]_{\alpha}\right\} = \min\left(\inf\left\{\left[A_i\right]_{\alpha}\right\}, \inf\left\{\left[A_j\right]_{\alpha}\right\}\right), \quad (8)$$

$$\sup\left\{\left[A_n\right]_{\alpha}\right\} = \max\left(\sup\left\{\left[A_i\right]_{\alpha}\right\}, \sup\left\{\left[A_j\right]_{\alpha}\right\}\right), \quad (9)$$

where A_n is the new fuzzy set.

In case of step 5 PSO was used as optimization technique applying the following parameters. The number of the particles in the swarm was N=30, self-confidence coefficient was c_1=0.2, social coefficient c_2=0.2, c_3=0.6, number of allowed generations was n_g=5, and the type of the newly created values was real number.

As presented in (4) the fuzzy system has four input variables. Their partitions are presented in Figures 8-10. The first two input partitions representing the linguistic variables GPR^{WE} and TLC are sparse with 5-5 fuzzy sets in each one. The remaining two input partitions (TF^{WE} and TF^{SN}) are also sparse and have 8-8 fuzzy sets.

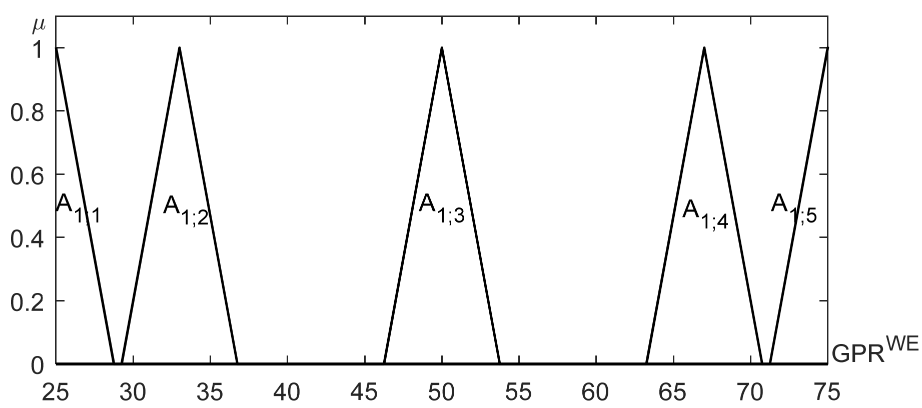

Figure 7. GPR^{WE}.

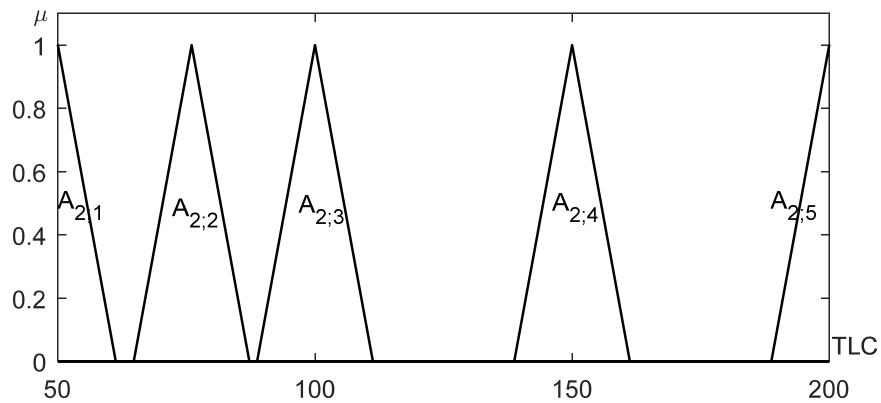

Figure 8. TLC.

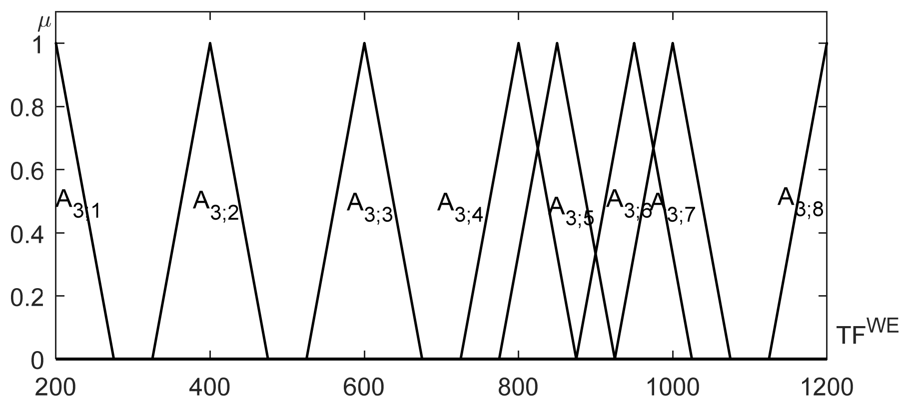

Figure 9. TF^{WE}.

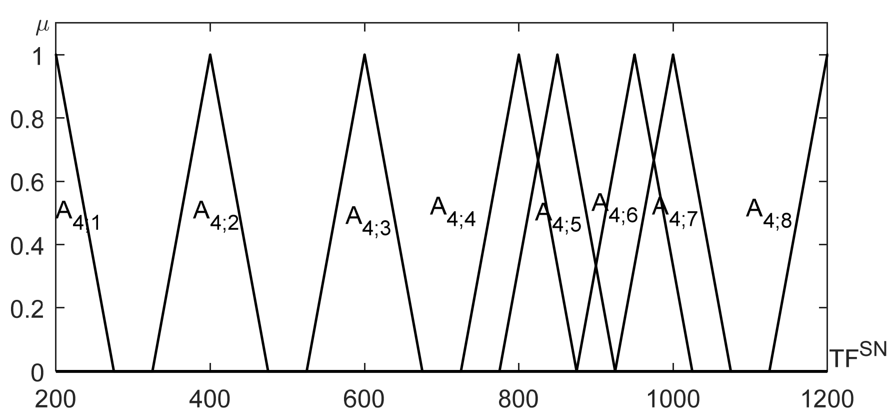

Figure 10. TF^{SN}.

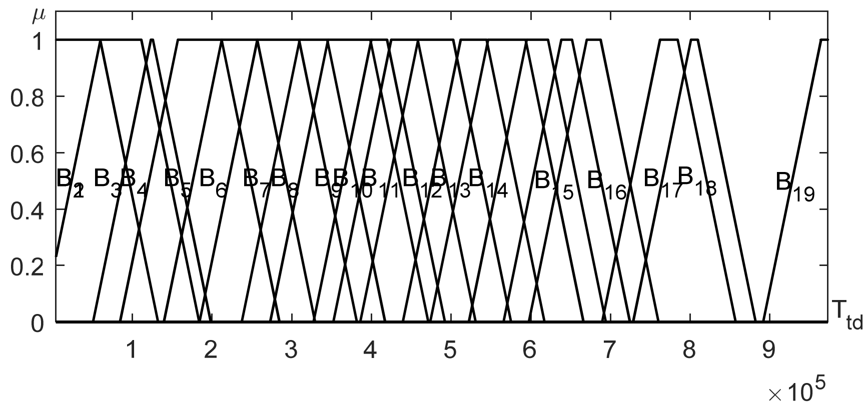

Figure 11. T_{td}.

The number of output fuzzy sets resulted in a bigger number (19) even after several merging operations. Owing to the sparse character of the input partitions the generated rule base was also sparse. However, even the coverage of these few fuzzy sets would require 5*5*8*8=1600 rules. Applying the above presented approach, the rule base generation task could be solved with a reduced number of 1284 rules.

The stopping criterion of the algorithm was defined by two components related to the performance of the system calculated against the training and the validation data set, respectively. In case of the training data set we defined an arbitrary threshold of PI_{tr}=4 % expressing that we want a *PI* value smaller than 4%. In case of the validation data set the threshold was PI_{va}=5 % expressing that in case of this dataset we do not want greater *PI* values than 5 %.

The performance of the final system became PI_{tr}=3.58 %, PI_{va}=4.33 %. The system performed well in case of the test data set as well providing PI_{te}=3.79 %

7.4 Surrogate based TLC and GPR optimization

Surrogate based optimization (Han, Zhang, 2012) is a special optimization approach developed for the case when the calculation of the objective function using the real phenomena or process is too expensive from computational point of view (e.g. long simulation runs), it requires expensive experiments or special environmental factors that cannot be always easily ensured. In such cases the original system/process/model is replaced by a surrogate model that can provide less expensive and/or faster information for the objective function.

In our case the fuzzy model presented in the previous section serves as surrogate model in course of the optimization of the traffic light cycle time and the green period ratio in order to obtain the smallest possible total delay in the intersection.

Having the model that describes the total delay time in function of the traffic flow values, the green time ratio and the traffic light cycle time as well as knowing the actual traffic flow values (TF^{WE} and TF^{SN}) the problem is reduced to the optimization

$$\underset{GPR^{WE},TLC}{\arg\min}\left(f\left(TF^{WE}, TF^{SN}, GPR^{WE}, TLC \right), TF^{WE} = const, TF^{SN} = const \right), \qquad (10)$$

where the minima of function *f* can be found by the help of an arbitrary optimization technique. In our case the application of the particle swarm optimization technique would be a straightforward option.

8. CONCLUSIONS

The optimal traffic light cycle time depending on traffic flow on a road with a traffic light at the end of the road was determined using the IDM based simulator our department has developed. The obtained results were compared with the results of other previously reported models. For small traffic flow values the results of the different models including IDM simulation are close to each other. In the region close to the road capacity determined by the traffic light capacity, but under the capacity, the results of the IDM model are closer to the results of the stochastic model, than to the other two compared models results.

It was shown that the delay times obtained in the micro-simulator for different traffic flows and different traffic light cycle's types in a road with a traffic light at the end of the road may be used to determine the optimal green light ratios intersections as well.

A fuzzy model was also created that can predict the total delay time in an intersection. Using this model, one can replace the simulations - which are time consuming steps – in course of the total delay calculations making possible the creation of a real-time solution that determines the optimal green time ratio and traffic light cycle in function of the actually observed traffic load values on the two intersecting roads. The actual values of the two mentioned parameters can be found by applying an arbitrary optimization method.

The proposed system is a proof-of-concept solution. Further research will consider other FRI and optimization methods (e.g. Vincze, 2017) as well aiming the reduction of the time necessary for the calculation as well as the improvement of the accuracy of the model. Furthermore the results obtained for the green time ratios in intersections should be compared with the results reported in the corresponding literature, and the application of the IDM based microscopic traffic simulator developed in our department to reduce the delay times at red lights of a traffic network has to be examined.

ACKNOWLEDGEMENT

This research is supported by EFOP-3.6.1-16-2016-00006 "The development and enhancement of the research potential at John von Neumann University" project. The Project is supported by the Hungarian Government and co-financed by the European Social Fund.

9. REFERENCES

Apeltauer, T., Macur, J., Holcner, P., Radimsky, M., 2013, Validation of Microscopic Traffic Models Based on GPS Precise Measurement of Vehicle Dynamics, *PROMET - Traffic&Transportation*, **25**, 2, 157-167.

Baranyi, P., Kóczy, L.T., Gedeon, T.D., 2004, A generalized concept for fuzzy rule interpolation, *IEEE Transactions on Fuzzy Systems*, **12**, No. 6, 820-837.

Baydokht, R.N., Noori, S., Azhang zad, A., 2014, Presenting a fuzzy model to control and schedule traffic lights, *Journal of Intelligent & Fuzzy Systems*, **26**, 2, 1007-1016.

Beckmann, M., McGuire, C., Winsten, C., 1956, *Studies in the Economics of Transportation*, Yale University Press, CT.

Blažič, S., Matko, D., Škrjanc, I., 2010, Adaptive law with a new leakage term, *IET Control Theory & Applications*, **4**, 9, 1533-1542.

Castan R., J.A., Ibarra M., S., Laria M., J., Castan R.E., 2014, An implementation of case-based reasoning to control traffic light signals, *Proceedings of the World Congress on Engineering 2014 (WCE 2014)*, London, UK, vol. I.

Chen, S.M., Ko, Y.K., 2008, Fuzzy interpolative reasoning for sparse fuzzy rule-based systems based on alpha-cuts and transformations techniques, *IEEE Transactions on Fuzzy Systems*, **16**, 6, 1626-1648.

Derbel, O., Péter, T., Zebiri, H., Mourllion, B., Basse, M., 2012, Modified intelligent driver model, *Periodica Polytechnica / Transportation Engineering*, **40**, 2, 53-60.

Derbel, O., Péter, T., Zebiri, H., Mourllion, B., Basse, M., 2013, Modified intelligent driver model for driver safety and traffic stability improvement, *7th IFAC Symposium on Advances in Automotive Control*, Tokyo, 734-739.

Dion, F., Rakha, H., Kang, Y., 2004, Comparison of delay estimates at under-saturated and over-saturated pre-timed signalized intersections, *Transportation Research Part B*, **38**, 99-122.

García-Nieto, J., Alba, E., Carolina Olivera, A., 2012, Swarm intelligence for traffic light scheduling: Application to real urban areas, *Engineering Applications of Artificial Intelligence*, **25**, 2, 274-283.

García-Nieto, J., Ferrer, J., Alba,E., 2014, Optimising traffic lights with metaheuristics: Reduction of car emissions and consumption, *International Joint Conference on Neural Networks*, Beijing, China, 48-54.

Han, Z.H., Zhang, K.S., 2012, Surrogate-based optimization, in: Real-World Applications of Genetic Algorithms, O. Roeva (Ed.), InTech, Available from: http://www.intechopen.com/books/real-world-applications-of-genetic-algorithms/surrogate-based-optimization.

Johanyák, Z.C., Kovács, S., 2006, Fuzzy rule interpolation based on polar cuts, in: Computational Intelligence, Theory and Applications, Springer-Verlag, Berlin, Heidelberg, 499-511.

Johanyák, Z.C., Alvarez Gil, R.P., 2017, Fuzzy model for the average delay time on a road ending with a traffic light, in: Lecture Notes in Electrical Engineering, K.J. Kim et al. (Eds.), IT Convergence and Security 2017, Springer Nature, vol. 449.

Johanyák, Z.C., 2011, performance improvement of the fuzzy rule interpolation method LESFRI, *Proceeding of 12th IEEE International Symposium on Computational Intelligence and Informatics*, Budapest, Hungary, 271-276.

Karakuzu, C., Demirci, O., 2010, Fuzzy logic based smart traffic light simulator design and hardware implementation, *Applied Soft Computing*, **10**, 66-73.

Kennedy, J., Eberhart, R., 1995, Particle swarm optimization, *Proceedings of IEEE International Conference on Neural Networks*, Perth, Australia, 1942-1948.

Karkalos, N.E., Kundrák, J., Markopoulos, A.P., 2017, Assessment of the performance of neural networks models for the prediction of surface roughness after grinding of steels, *International Journal of Artificial Intelligence*, **15**, 1, 55-75.

Kesting, A., Treiber, M., Helbing, D., 2007, General lane-changing model MOBIL for car-following models, *Transportation Research Record: Journal of the Transportation Research Board*, **199**9, 86-94.

Khanjary, M., Navidi, H., 2012, Optimizing traffic light of an intersection by using game theory, *Proceedings of 3rd World Conference on Information Technology*, AWERProcedia Information Technology & Computer Science (2013), **3**, 1163-1168.

Kóczy, L.T., Hirota, K., 1993, Approximate reasoning by linear rule interpolation and general approximation, *International Journal of Approximative Reasoning*, **9**, 197–225.

Kovács, L., Szabó G., 2017, Generalization of string transformation rules using optimized concept lattice construction method, *Procedia Engineering*, **181**, 604-611.

Kovács, S., 2006, Extending the fuzzy rule interpolation "FIVE" by fuzzy observation, in: *Advances in Soft Computing, Computational Intelligence, Theory and Applications*, B. Reusch (Ed.), Springer-Verlag, Berlin, Heidelberg, New York, 485-497.

Kovács, T., Bolla, K., Alvarez Gil, R., Csizmás, E., Fábián, Cs., Kovács, L., Medgyes, K., Osztényi, J., Végh, A., 2016, Parameters of the intelligent driver model in signalized intersections, *Technical Gazette*, **23**, No. 5, 1469-1474

Mamdani, E.H., Assilian, S, 1975, An experiment in linguistic synthesis with a fuzzy logic controller, *International Journal of Man Machine Studies*, **7**, 1-13.

Murat, Y.S., Cakici, Z., Yaslan, G., 2012, Use of fuzzy logic traffic signal control approach as dual lane ramp metering model for freeways, *Proceedings of 17th Online World Conference on Soft Computing in Industrial Applications*, 339-349.

Pappis, C., Mamdani, E., 1977, A fuzzy logic controller for a traffic junction, *IEEE Transactions on Systems, Man, and Cybernetics*, **SMC-7**, 10, 707-717.

Péter, T., 2012, Modelling of nonlinear road traffic networks for junction control, *International Journal of Applied Mathematics and Computer Science*, **22**, 3, 723-732.

Pokorádi L., 2015, Fuzzy failure modes and effects analysis with weigthed summarized center of gravity defuzzification, in: *Proceedings of 14th International Conference on Mathematics and its Applications*, I. Golet, L. Cadariu (Eds.), Politehnica University of Timisoara, Editura Politehnica, Timisoara, 93-98.

Postorinoa, M.N., Versacia, M, 2014, Upgrading urban traffic flow by a demand-responsive fuzzy-based traffic lights model, *International Journal of Modelling and Simulation*, **34**, 2, 102-109.

Precup, R.-E., Sabau, M.-C., Petriu, E.M. 2015, Nature-inspired optimal tuning of input membership functions of Takagi-Sugeno-Kang fuzzy models for anti-lock braking systems, *Applied Soft Computing*, **27**, 575-589.

Precup, R.-E., Radac, M.-B., Tomescu, M.L., Petriu, E.M., Preitl, S., 2013a, Stable and convergent iterative feedback tuning of fuzzy controllers for discrete-time SISO systems, *Expert Systems with Applications*, **40**, 1, 188-199.

Precup, R.-E., Tomescu, M.L., Preitl, S., Petriu, E.M., Fodor, J., Pozna, C., 2013b, Stability analysis and design of a class of MIMO fuzzy control systems, *Journal of Intelligent and Fuzzy Systems*, **25**, 1, 145-155.

Rouphail, N.M., Park, B.B., Sacks, J., 2000, Direct signal timing optimization: Strategy development and results, *XI Pan American Conference in Traffic and Transportation Engineering*, 195-206.

Sharif, W., Samsudin, N.A., Deris, M.M., Aamir, M., 2017, Improved relative discriminative criterion feature ranking technique for text classification. *International Journal of Artificial Intelligence*, **15**, 2, 61-78.

Shepard, D., 1968, A two dimensional interpolation function for irregularly spaced data, *Proceedings of 23rd ACM International Conference*, 517–524.

Takagi, T., Sugeno, M., 1985, Fuzzy identification of systems and its applications to modeling and control, *IEEE Transactions on System, Man, and Cybernetics*, **15**, 116-132.

Teo, K.T.K., Kow, W.Y., Chin, Y.K., 2010, Optimization of traffic flow within an urban traffic light intersection with genetic algorithm, *Proceedings of Second International Conference on Computational Intelligence, Modelling and Simulation*, 172-177.

Treiber, M., Hennecke, A., Helbing, D., 2000, Congested traffic states in empirical observations and microscopic simulations, *Physical Review E*, **62**, 1805-1824.

Treiber, M., Helbing, D., 2001, Microsimulations of freeway traffic including control measures, *Automatisierungstechnik*, **49**, 478-484.

Vaščák, J., 2012, Adaptation of fuzzy cognitive maps by migration algorithms, *Kybernetes*, **41**, 3/4, 429-443.

Vincze, D., 2017, Fuzzy rule interpolation and reinforcement learning, in: *IEEE 15th International Symposium on Applied Machine Intelligence and Informatics*, A. Szakál (Ed.), Herlany, Slovakia, 173-178.

Webster, F.V., 1958, Traffic signal settings, *Road Research Technical Paper No. 39*, Road Research Laboratory, Her Majesty Stationary Office, London, UK.

Wiedemann, R., 1974, *Simulation des Straßenverkehrsflusses*, *Schriftenreihe des Institutes für Verkehrswesen der Universität Karlsruhe*, Heft 8, Karlsruhe, Germany.

Wiedemann, R., 1991, Modelling of RTI-elements on multi-lane roads, *Advanced Telematics in Road Transport*, Proceedings of the DRIVE Conference, II, Elsevier, New York, 1007-1019.

Zadeh, L.A., 1965, Fuzzy sets, *Information and Control*, **8**, 338-353.