

Simulating Urban Traffic as a Multilayered Multiagent System

Levente Alekszejenkó

Department of Measurement and Information Systems
Budapest University of Technology and Economics
Budapest, Hungary
ale.levente@gmail.com

Tadeusz Dobrowiecki

Department of Measurement and Information Systems
Budapest University of Technology and Economics
Budapest, Hungary
dobrowiecki@mit.bme.hu

Abstract—The vehicles of the future will be capable of communicating with each other and with the road infrastructure as well. Based on this ability, complex multiagent systems can be designed, including smart cars and intelligent traffic control systems (referred to as judges).

Such system was implemented by extending an open-source traffic simulation tool, called Simulation of Urban MObility platform (SUMO). The implemented system can be used to experiment with and to verify various algorithmic approaches aimed to increase the intelligence of autonomous drivers and urban traffic controllers.

Our study investigated the adaptation of the operating system task schedulers and the Explicit Congestion Notification algorithm of computer networks. It resulted in a layered cooperative multi agent system composed from platooning car drivers in the lower layer and the cooperating intersection judges in the upper layer.

Results indicate that the implemented system can organize the traffic better in extraordinary cases (e.g. an accident, road works on some major streets, etc.). The regulatory capability of the proposed system depends greatly on the topology of the road connections. This aspect (especially the problem of congestions) is currently under investigation.

Index Terms—intelligent traffic control, smart vehicles, multiagent system, platooning, explicit congestion notification, scheduling algorithms

I. INTRODUCTION

One of the major problems of our cities is the regular congestion of road networks. As the number of vehicles is rapidly increasing, improving the flow of the traffic and reducing traveling time becomes an even more challenging task. The connected vehicles of the future and the wide variety of IoT devices implemented in the road infrastructure may create new ways to optimize the traffic.

For example, smart cars can form groups, so called *platoons*, near intersections. The cars which form a platoon can change lanes or can pass through intersections together, therefore causing less impact on the traffic.

Another possibility is to create intelligent traffic controllers, so called *judges*. Let us suppose that the number of incoming vehicles from each direction is known. In this case, as the

demand is known for the near future, theoretically well established scheduling algorithms can be applied. Moreover, these judges can be made cooperative as well in order to make a globally optimal solution.

Some ideas have already been implemented as a multilayered multi-agent system. In our implemented system there are two types of agents, i.e. smart cars and judges. They can communicate with each other in order to perform some intelligent actions. For example, smart cars can form platoons, or ask the judges whether they can pass through an intersection. Judges can also send messages to each other, cooperatively evaluating the state of the roadnetwork, to attempt to avoid the congestion. The performance of our system was validated by simulations. The used simulation platform was created by extending the Simulation of Urban MObility (SUMO) [5] microscopic traffic simulation program.

II. LITERATURE REVIEW

Creating platoons of smart cars, besides reducing the computational demands on intelligent traffic controllers, results in a more efficient lane-changing strategy. Let us suppose that the lane-changing of a platoon can be modeled as a single lane-change of a truck. In [9] the authors have shown that a double semi-trailer truck is equivalent to 3 personal cars. It takes, however, more space on the road than those 3 cars. Consequently, platooning also seem to be an effective way to reduce the impact of lane-changes.

Consider now the perspective of a judge, i.e. an intersection controller. The task of the judge is analogous to that of the scheduler of an operating system. Both are responsible for deciding which competing entity (task or vehicle) can use a unique resource (the processor or the part of the intersection). A scheduling solution to control intersection lamps was suggested by [1], where so called Minimal Destination Distance First (MDDF) method was used, based on the well known Shortest Job First scheduler (SJF)¹ of operating systems. Unfortunately, that proposed algorithm is not fair and was verified only in a highly regular intersection environment.

Coordination of the traffic signals is an old idea, and for example can be achieved by green-waves. There are some

¹To be precise it is based on the Shortest Remaining Time First, the preemptive version of SJF.

The research has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.2-16-2017-00013, Thematic Fundamental Research Collaborations Grounding Innovation in Informatics and Infocommunications).

traditional algorithms, like TRANSYT and SCOOT [7], which try to shorten the queue lengths behind the traffic lights. In the last decades, new methods were published, for example the one based on a reservation system [2]. In this algorithm, smart cars have to book time and space slots when they are permitted to pass through the intersection. An intersection manager stores these bookings and checks whether an incoming booking is feasible. This system has a major disadvantage, namely when there is a vast amount of vehicles with a vast amount of bookings, the feasibility check would be a really processing-intensive task. An agent based solution was proposed in [10]. In this approach every neighboring intersection was connected, therefore it is theoretically possible to create unstable states. The green time of a traffic signal is only modified a little bit, making its neighbors also to modify a little bit more, and so on. As a result of this butterfly-effect, the whole system might become unstable, causing unpredictable traffic flows, therefore increasing the risk of an accident.

III. INTERACTION BETWEEN SMART CARS – PLATOONING

A. Formation of a Platoon

Smart cars are basically competing agents (for the green light slot), but they are willing to form a coalition – a *platoon* – in order to go through an intersection as efficiently as possible, if their interests (path) coincide.

When a smart car approaches an intersection², it has to join a platoon. The cars of a platoon have the exactly same trajectory: they arrive at the intersection from the same direction, in the same lane and leave via the same exit lane. For the time the platoon exists, its cars are joined virtually into a chain maintaining about 5 m of distance between each other.³

If the platoon in front of a smart car is not suitable, the smart car has to create a new platoon.

In the front of the platoon is the *platoon leader*, all the other vehicles in the platoon are the *platoon members*. Platoon leaders are responsible for their platoons, and the platoon members have to follow the platoon leader.

After crossing the junction, the platoon leader exits its platoon and passes over its prerogatives to the next-in-line in the platoon. This smart car will be the new platoon leader. It is an easy and effective way to avoid the problems which can be caused by the preemptive scheduler of the judges.

B. Lane-Changing of Platoons

Reducing the impact of changing lanes before intersections can provide a significant improvement in the traffic flow. In the SUMO platform sophisticated lane change models are already implemented. In our research we modified the SL2015 model [3] to calculate also with the platooning concept. So while a smart car belongs to a platoon, it has to behave differently, depending on whether it is the leader or a simple member.

²Some markers are placed as new traffic signs which instruct the smart cars to join or leave a platoon.

³Platooning in this case is an adhoc formation and slightly differs from platoons created on highways. The aim of our platoons is to pass through intersections more effectively than individual vehicles are able to do so.

In a platoon only the platoon leader can make a lane-change decision. All the other members have to follow the car in front of them.

If the platoon leader finds out that a lane change is needed⁴, it makes contact with the platoon leader (if there is any) in the target lane. The two platoon leaders make an agreement whose platoon will be ahead of the other.⁵ Platoon in the target lane will slow down or even stop if necessary to make sure that the maneuver will be successful. Another possibility is that the mover leader has to wait until the asked leader and its platoon leaves the target lane. The platoon manages the lane change car-by-car, sending lane-change command down the platoon chain.

IV. INTERSECTION CONTROLLING ALGORITHMS OF UNCONNECTED JUDGES

In our first approach, unconnected (i.e. non-communicating) judges were implemented. From the operating system field we borrow two simple scheduling algorithms. One is the Round Robin (RR) algorithm, which is fair (free of starvation) and the other is the Shortest Job First (SJF), which yields an optimal response time, but is unfair. These two simple schedulers (and their preemptive versions respectively) provide the basics of all kinds of much more complex scheduling algorithms. Due to this fact, we decided to try out these two methods, as conflict class⁶ selector algorithms of an intelligent judge agent.

1) *RR*: A simple round robin scheduler can be implemented as a traffic controlling method without any significant modification to the original algorithm. We prescribe time slices to each conflict class. This will be the maximal amount of time in which a conflict class can be active. After this time slice is elapsed, we simply select the next conflict class from the list.

2) *MDDF*: [1] *Minimal Destination Distance First* traffic controlling system is based on the optimal scheduler, called Shortest Job First (specifically its preemptive version, the so-called Shortest Remaining Time first). The problem with this solution is that it is not fair.

Let us suppose that a lonely car is waiting in an intersection to pass. This car is at the beginning of its route to a very distant destination, but vehicles with significantly shorter routes are continuously arriving. The car with the long route to its destination can wait forever without getting through this intersection.

To make the algorithm fair, we redefined our scheduler as a two-level scheduler. On the higher priority level a simple Round Robin scheduler is running, and on the lower priority level a scheduling algorithm similar to the implementation of [1] is used. At first, every conflict class is scheduled by the lower priority level. If a conflict class was not active in the last 90 s, it would change its priority to the high level. This

⁴In this state, smart cars' lane change model calculates with length of platoons instead of single vehicles.

⁵[3] has already worked out the protocol and algorithm of this agreement. We modified the existing solution to have the contract made only between platoon leaders instead of single vehicles.

⁶A conflict class is a group of cars, which are permitted to pass through an intersection simultaneously.

way it is guaranteed that every vehicle will be scheduled in a limited amount of time. The Round Robin also prevents the occurrence of starvation.

V. INTERSECTION CONTROLLING ALGORITHMS OF CONNECTED JUDGES

A. The ECN-based method

It is a simple idea to connect the judges (i.e. to permit them to communicate) with each other in order to improve the capabilities of the system. This improvement is a signal coordination which aims to prevent the formation of congestions. Such algorithms are already in use in the domain of computer networking. However, the character of the traffic is different and the majority of them cannot be applied in the road traffic environment. An algorithm which could be applied, or at least be experimented with, is the so called Explicit Congestion Notification (ECN) method [4], which we implemented in our simulations.

The basic idea behind ECN is that the receiver node (an intersection manager or a network router) can inform the sender if the queue length (of vehicles or datagrams) at the receiver side reaches a certain level (let us call this signal the *ECN-signal*). This means that a congestion is about to form. To avoid the congestion, the sender must decrease its output in this case. A new kind of judge, the so-called ECN-judge, was implemented which is based on this discussed method.

The ECN algorithm has a great advantage that it does not require the definition of arterial directions⁷. Defining arterials would demolish the merits of the intelligent system in extraordinary situations, when the proposed system can clearly outperform the traditional system, for details, see Section VI.

B. Challenges in the Implementation

The state-space of the ECN judge can be enormous since it depends both on the number of incoming vehicles and on the number of the neighboring intersections. Therefore, storing a signal plan for all of the states is quite memory-consuming. Instead of doing this, a dynamical signal plan generation method was implemented (for an overview, see Figure 1).

The calculation of simple signal phase can be formalized as an *integer programming problem (IP)*. Our goal is to maximize the number of directions which receive green light at the same time, subject to the actual state of the network. This state consists of dynamic parts, like the incoming ECN-signals or the decision of a scheduling algorithm (eg. a Round Robin) as well as static parts, which describes which directions cannot pass through an intersection simultaneously.

If the IP is solved, we only know the signal phase for a given moment. In order to generate a signal plan (which describes how long a direction should get a green or a red light), it is necessary to recalculate this IP problem from time to time. In our implementation the recalculation time is a linear function of the number of incoming vehicles, but cannot exceed 45 seconds.

⁷Arterial direction is the main route which for example receives a green-wave.

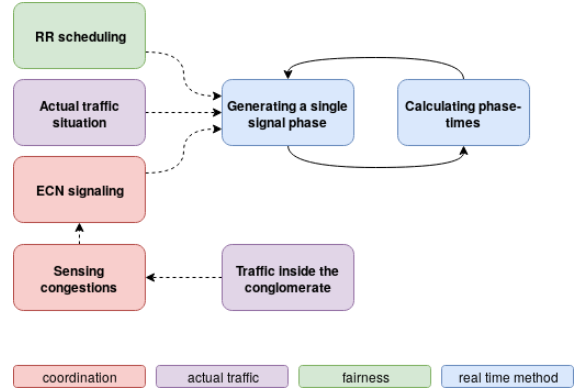


Fig. 1. Overview of the ECN-judge

There is another topic which should be discussed, namely the identification of the forming congestions. It is a quite difficult task [6], [8], and our research did not focus on solving this problem, thus based on preliminary simulations, we simply calculated the traffic density, which can provide the highest traffic flow. We say that there is a congestion forming when the 90% of this level is reached, so the ECN-signal is sent at this event.

VI. SIMULATIONS

Our solutions were tested by an extension to the Simulation of Urban MObility program (see Figure 2 for details). The simulated network was the BAH intersection⁸ of Budapest and its close neighborhood.

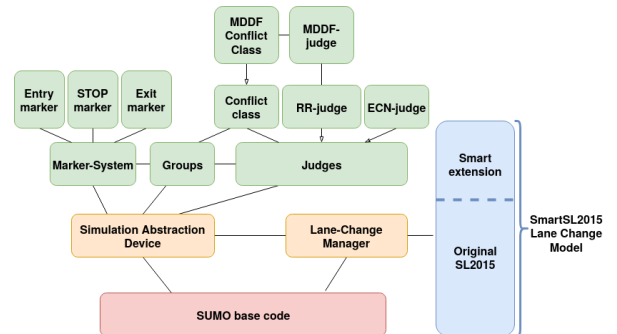


Fig. 2. The developed extension of the SUMO. The components of the multilayered, multiagent system are shown in green. Some modules are necessary to create an abstraction layer between the original source code of the SUMO and the intelligent system's layer. This abstraction layer is presented in orange and blue in this figure.

Basically two types of traffic demand were modeled: some cases of regular traffic (eg. night traffic, morning traffic, noon traffic) and irregular traffic (Budaörsi út is closed⁹) were fed into the simulator.

⁸Where streets of Hegyalja út, Jagelló út, Villányi út, Budaörsi út and Alkotás utca intersect.

⁹*Irregular1* case: Obstacle is northbound of “Budaörsi út”, can be bypassed via Karolina and Villányi streets.

Irregular2 case: Obstacle is southbound of “Budaörsi út”, bypass route is via “Hegyalja út”.

A. Simulating The Unconnected Judges

As a first attempt, we tested the behavior of the platooning system and the simple, unconnected judges. These measurements basically show that such kind of systems may be able to reduce waiting and traveling times through this intersection.

The results show (see Table I and Table II) that such a system is able to decrease the waiting (for example at red lights) and average traveling times in irregular situations. On the other hand, the improvement of the traffic flow¹⁰ is not so obvious in regular cases, see Figure 3.

TABLE I
SIMULATION RESULTS OF "IRREGULAR1" CASE

Test case	Arrived (%)	Waiting Time (s)	Average Traveling Time (s)
Traditional	33.81	29.68	170.55
RR	29.19	12.117	174.87
MDDF	22.77	12.41	154.02

TABLE II
SIMULATION RESULTS OF "IRREGULAR2" CASE

Test case	Arrived (%)	Waiting Time (s)	Average Traveling Time (s)
Traditional	38.48	36.44	199.38
RR	32.71	11.43	170.07
MDDF	34.39	10.74	176.72

B. Simulating The Connected Judges

In order to improve the traffic flow, some judges¹¹ were re-programmed to ECN-judges. Theoretically this system would have greater chance to find a globally optimal solution, than the unconnected judges, which are only capable of finding a locally optimal scheduling.

The trial of the system gave surprising results. Instead of improving the flow of the traffic in the BAH-intersection, this method rather reduced this value. As it can be seen in Figure 3, the new judges limit the density of the traffic to around 65-70 vehicles/km, almost regardless of the height of the traffic demand. (With a combined system, which contains both connected ECN-type and unconnected Round Robin-type judges, this limit is slightly higher.) Partly by this density limitation, partly by some yet unknown effects, the traffic flow is strongly reduced by the ECN-judge system.

VII. CONCLUSION AND FURTHER RESEARCH AIMS

As the traditional system is likely to be numerically optimized, it is a challenging task to achieve the same or even better results with a new intelligent solution in regular cases. On the other hand, in extraordinary situations, an intelligent,

¹⁰The traffic flow is a commonly calculated value. It is the product of the traffic density ($\frac{\text{vehicles}}{\text{km}}$) and the mean velocity of the vehicles ($\frac{\text{km}}{\text{h}}$). These values can be measured by different types of detectors, cameras, etc.

¹¹Namely the judge supervising the intersection of Villányi and Budaörsi streets, the one supervising the Budaörsi, Hegyalja and Alkotás street intersection and the one placed at the Jagelló and Hegyalja crossing.

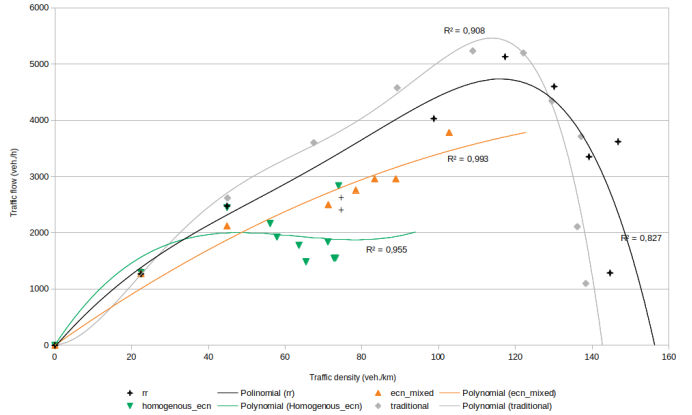


Fig. 3. Traffic flow in the traditional and in the unconnected intelligent system, consisting of RR-type judges.

multi-agent based solution can be much more flexible. This flexibility provides better traveling times by reducing the unnecessary waiting times.

The ECN-judges have no benefits in the BAH intersection scenario, if our goal is to improve the flow of the traffic. Supposing that there are situations where the traffic density (and therefore the flow) limitation is a desired effect, our proposed system might be beneficial as well. Such situations can be the limitation of the traffic going through residential areas or nature reserves.

Further research is needed to verify that the ECN-judge system is able to cause such effect in these kinds of networks.

REFERENCES

- [1] F. Ahmad, S. A. Mahmud, G. M. Khan, and F. Z. Yousaf, "Shortest remaining processing time based schedulers for reduction of traffic congestion," in 2013 International Conference on Connected Vehicles and Expo (ICCVE), Las Vegas, NV, USA, 2013, pp. 271–276.
- [2] K. Dresner and P. Stone, "A Multiagent Approach to Autonomous Intersection Management," *JAIR*, vol. 31, pp. 591–656, Mar. 2008.
- [3] J. Erdmann, "SUMO's Lane-Changing Model", in *Modeling Mobility with Open Data*, M. Behrisch and M. Weber, Edit. Cham: Springer International Publishing, 2015, pp. 105–123.
- [4] S. Floyd, K. K. Ramakrishnan, and D. L. Black, "The Addition of Explicit Congestion Notification (ECN) to IP." [Online]. Accessible: <https://tools.ietf.org/html/rfc3168#section-1>. [Accessed: 2019. okt. 25.].
- [5] P. A. Lopez et al., "Microscopic Traffic Simulation using SUMO," in 2018 21st International Conference on Intelligent Transportation Systems (ITSC), 2018, pp. 2575–2582.
- [6] K. Nagel, P. Wagner, and R. Woessler, "Still Flowing: Approaches to Traffic Flow and Traffic Jam Modeling," *Operations Research*, vol. 51, no. 5, pp. 681–710, Oct. 2003.
- [7] D. I. Robertson, "Research on the TRANSYT and SCOOT Methods of Signal Coordination," *ITE Journal*, Jan 1986 pp. 37–40.
- [8] M. Treiber and A. Kesting, *Traffic Flow Dynamics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.
- [9] N. Webster and L. Elefteriadou, "A simulation study of truck passenger car equivalents (PCE) on basic freeway sections," *Transportation Research Part B: Methodological*, vol. 33, no. 5, pp. 323–336, 1999.
- [10] J. Withanawasam and A. Karunananda, "Multi-agent based road traffic control optimization," in 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, 2017, pp. 977–981.