EPiC Series in Computing

Volume 62, 2019, Pages 107–123

SUMO User Conference 2019

SUMO Based Platform for Cooperative Intelligent Automotive Agents [*]

Levente Alekszejenkó[1] and Tadeusz Dobrowiecki[2]

[1] Budapest University of Technology and Economics, Budapest, Hungary
ale.levente@gmail.com
[2] Budapest University of Technology and Economics, Budapest, Hungary
dobrowiecki@mit.bme.hu

Abstract

Starting from the problems of nowadays' urban traffic (congestions, imperfect timing of traffic lights, high impact of lane changes) we investigate the feasibility of a cooperative intelligent agent based solution as an overall control scheme governing the car flow in congested urban intersections.

The proposed complex solution features both the intelligent traffic control and the car platooning. In order to test and verify the merits of the proposed solution in urban intersection of a widely variable topology, but also to support our future research aims, a simulation platform, extending the basic functionalities of SUMO with the options of intelligent communication and cooperative co-acting, was designed and developed.

1 Introduction

As a part of our research in cooperative schemes to mitigate the congestion problems in the urban traffic, we have developed a simulation platform making it possible to investigate the vehicle behavior as a cooperative multi agent system. The platform was based on the Simulation of Urban MObility [10] platform and extended its basic functionalities with both smart traffic controlling systems and a basic platooning concept.

The aim was to implement a communication based, fully cooperative, multiagent system of self-driving cars and intelligent traffic controlling devices with an eye on only a modest modification to the already existing traffic infrastructure [2]. The proposed system is expected to produce a better flow of traffic and reduced waiting time at traffic intersections.

To this purpose different scheduling algorithms were investigated to control traffic flow in the intersections, analogous to the Round Robin and the Shortest Job First (implemented as Minimal Destination Distance First) methods known in operating system scheduling. By forming platoons of cars and by introducing a new lane-change model (extending the SL2015 model [6] of SUMO), we attempted to reduce the impact of lane changes on the traffic flow.

[*] 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), as well as by the BME- Artificial Intelligence FIKP grant of EMMI (BME FIKP-MI/SC).

M. Weber, L. Bieker-Walz, R. Hilbrich and M. Behrisch (eds.), SUMO2019 (EPiC Series in Computing, vol. 62), pp. 107–123

According to our measurements, the proposed scheme is able to reduce the average waiting time even to only 20% of that experienced with the traditional intersection control systems, increasing meanwhile the traffic flow and the throughput of the network, both in (rectangular grid) US and (highly irregular) European road-network topology.

After a literature review in Section 2, the architecture of the developed platform is shortly introduced in Section 3. A new simulation abstraction device is described in Section 4, the mechanism of the platooning and a new lane change model is presented in Section 5. Section 6 deals with the static nodes of the traffic control implementing the scheduling algorithms. Finally Section 7 presents the concrete results of simulations on different intersection and road topology. Section 8 concludes the discussion.

2 Related Works

The principal difference between the solutions available in the literature and the one proposed in the present paper is that the usual platooning, as found in the literature, is a long term coalition. It has as its aim the coordinated and close driving to improve the capacity of the road network, reduce air drag, improve fuel consumption, etc. [12] [14] [15].

In our approach platooning is a short term organization meant to facilitate the decision making (light switching) of the urban intersection controllers. It is based on the goal of a "common direction of intersection crossing" and is valid so long as the cars pass the intersection.

Among the numerous extensions to SUMO in the topic of platooning, [7] mainly focuses on keeping gap between the members of a platoon. It uses Traffic Control Interface (TraCI) to influence the movements of the simulated vehicles. As an essential part of platooning, a simple gap keeper algorithm was also implemented in our solution, but instead of TraCI, our extension is based on C++.

Plexe [16] is one of the most commonly used platooning extension to SUMO. This extension features cooperative adaptive cruise control (CACC), as well as simulation of communication and networking. In our research, we assumed that a faultless, well scalable communication was available for every smart car, so we did not want to simulate the communication network itself. In slowly moving, urban traffic CACC would not be clearly beneficial either. Permit [12] extends Plexe even further with complex platoon movements. In urban traffic, unfortunately, there is not enough space and time to make such maneuvers. For this reason, we need a simple platoon definition, where platoons are easy to create, and have only a limited set of known maneuvers. Our (further) research aims require also such knowledge about SUMO state variables which cannot be transmitted via TraCI. In our development work, we used native C++ code to extend the capabilities of the original system.

Besides platooning, intelligent traffic control (ITC) is also a hot topic of the current research. In intelligently controlled intersections significant amount of fuel can be saved, therefore air-pollution can be reduced, as it is mentioned in [13]. [17] proves that in ITCs almost every well-known artificial intelligence technology can be profitably used, such as artificial neural networks, reinforcement learning, fuzzy-logic, or (multi) agent based and game theoretic approaches. In our research, we focused solely on multi agent based solutions without any game theoretic constructions.

[9] proposes a system which is capable to improve the traffic flow by using deep reinforcement learning of the simulated vehicles. In our belief, proving the correctness of the results of a machine learning system is far too difficult,[1] particularly in a multiagent system. Instead, we

[1] In case of a real-world implementation, a formal (safety) proof of the system cannot be avoided.

propose a centralized and easy to verify solution for improving the traffic flow. This solution is the operating system scheduler based intersection control, since those algorithms are already formally verified. This idea is basically similar to the process synchronization based approach, which is described in [8].

Our proposed ITC system can be categorized, accordingly to [4], as a trajectory modeling based (which means that the surface of the intersections are divided by the trajectories of the smart cars, which can pass through it simultaneously), centralized intersection controller with virtual traffic lights.

Norm-based traffic [3] is an interesting multi-agent system solution. It requires, however, also some central controlling agents with norm data bases and well-distributed information among the agents. We aimed at creating a simpler system, with a significantly smaller state-space.

A similar approach could be the reservation-based traffic control [5]. By good implementation, this method can operate with relatively few messages between the central, management unit and smart cars, with averagely 6 messages per vehicles. In our preliminary experiments, *cancellations* and *change-requests* were so common, that we tried to find another solution. This solution and the simulation platform, used to verify its merits, is presented in the following sections.

3 Architectural Overview

We decided to design the simulation platform by modifying the SUMO's base code as little as possible. This decision implied that new layers of software architecture had to be added.

SUMO's base code is the lowest layer (SUMO) in our software model (see Figure 2). The functions of this layer are used by the middle layer, called Simulation Abstraction Layer (SAL). On the top of the architecture sits Intelligent Logic Layer (ILL). The new, modified lane change model has some components that logically belongs to ILL but the major part of this model also belongs to SUMO itself, so our modified SL2015 model, called SmartSL2015 is realized as vertical layer of the architecture.

Basically SUMO layer contains only the source code which can be fetched from GitHub of the original SUMO repository. A new device, called Simulation Abstraction Device (SAL) is introduced in the Simulation Abstraction Layer. Such devices connect the vehicles (Smart Cars) with their agent representations resident in the upper layer of ILL. Through these devices, objects in the logical layer (like Judges or Groups, see Figure 1) are notified about Smart Cars' movements and requests, and inversely, these intelligent components can control the Smart Cars. Since SUMO's lane change models are truly sophisticated, we also defined a separate object to control SmartSL2015 model.

In the ILL layer Marker System, Groups and Judges are created. These are basic classes for the further development or for defining more specific components. Markers are intelligent sensors and traffic signs. Entry markers indicate the entrance to the intersection and exit markers show the exit. Markers also trigger the cooperative protocols. Groups are the representation of platoons in the software architecture. Judges are the intelligent traffic lights. They work with Conflict classes, i.e. those groups of smart cars which can pass through the intersection simultaneously. Two concrete types of judges are implemented, differing in the scheduling policy used.

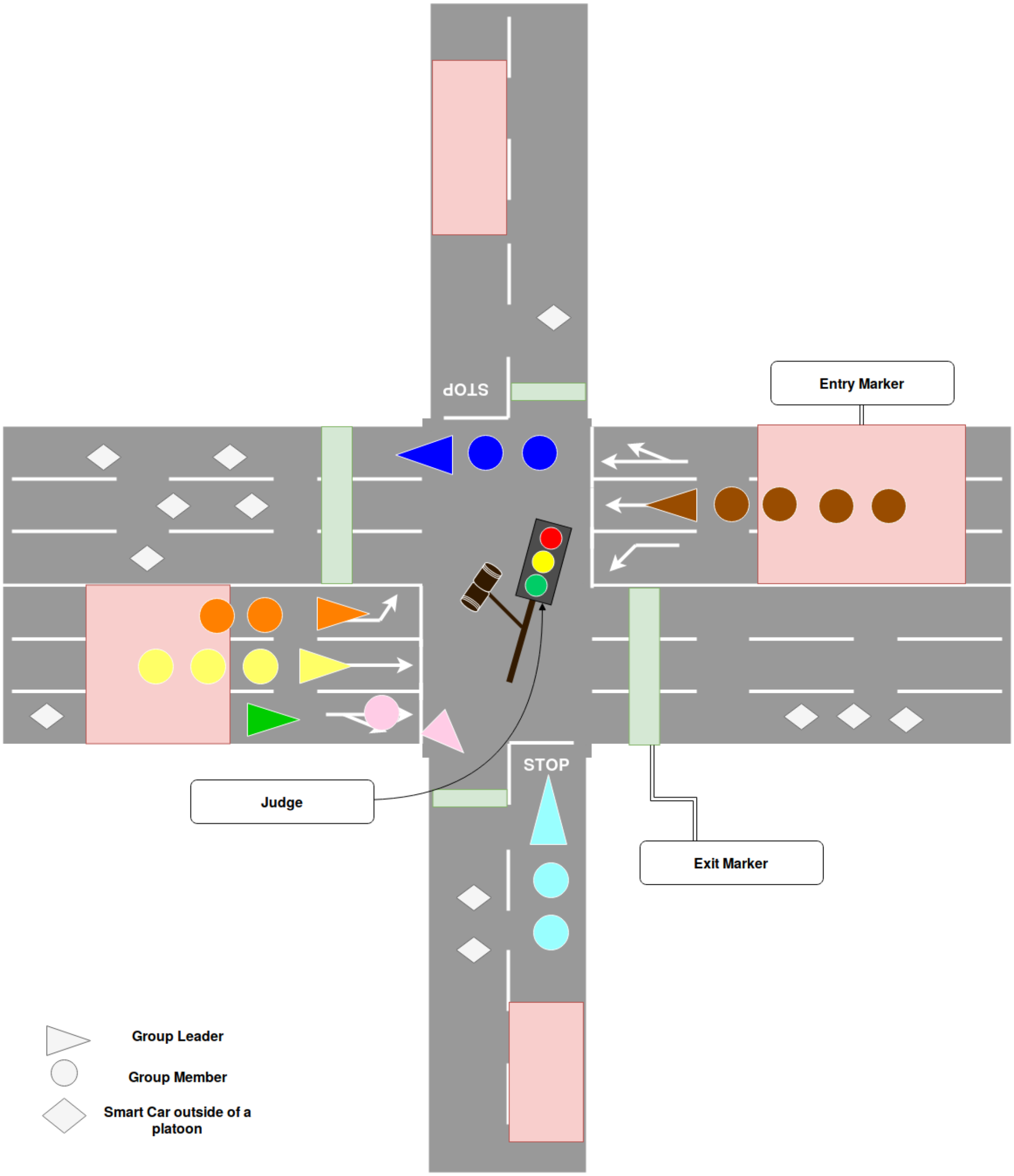

Figure 1: Components of the urban traffic intersection controlled by the proposed architecture

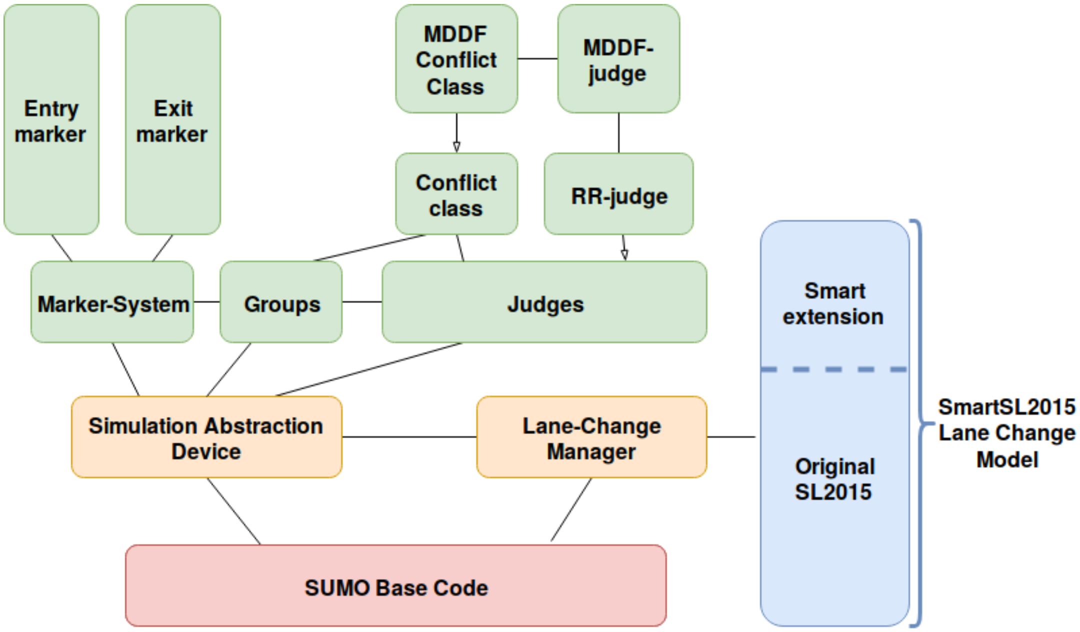

Figure 2: Extension to SUMO. SUMO layer is salmon-colored, meanwhile SAL layer is apricot-colored and ILL layer is emerald-colored.

4 Simulation Abstraction Device

Using SUMO's device system, a new device, called MSDevice_SAL, was designed. This device is responsible for the execution of the protocols dictated by the upper, ILL layer, using the basic functions of SUMO and some method from TraCI library.

In the `notifyMove` method of this device, communication with judges and markers is embedded. Here also the speed control of smart cars is managed, taking into account the prescribed follow-up distance of group members and the judges' decisions.

Because of the way SUMO does operate, the `notifyEnter` method only changes flags of some events. These events (protocol triggers) are later processed by `notifyMove` method.

Since the introduced device turns to be the only connection between the SUMO basic code and the intelligent layers, some getter and setter functions are placed into this object. These are in connection with the grouping state (member, leader, outside), speed, position of a smart car, etc.

5 Implemeting Platooning in SUMO

Traditional platooning of the vehicles can be beneficial by reducing the impact of lane changes on the traffic flow. It can also reduce the space between smart cars and at higher velocities it can help to save fuel, therefore reducing pollution and costs.

Marker System

The proposed platoons are formed only near the traffic junctions. At certain positions of the track, Entry Markers are placed in order to indicate an upcoming intersection. Sensing these markers, smart cars attempt to join, or to create platoons. The exits of intersections are indicated by Exit Markers. At these points platoons dissolve.

In the simulator itself, the markers are represented as IDs of edges of the road network and also as independent classes containing specific data about the intersection. Marker system is a singleton class which has the collection of the markers.

Groups and Their Members

In our terminology platoons are called groups, reflecting that in our system platoons are groups of smart cars with exactly the same trajectory along the track. This means that platoons consist of smart cars which arrive at an intersection from the same direction, in the same lane, and also leave towards the same direction and in the same lane.

When a smart car passes an entry marker, it is obligated to join a platoon. If the car in front is moving along the exactly same trajectory, the incoming smart car joins it. Otherwise the newly coming smart car creates a new group.

In a platoon smart cars can have to following roles:

- Group leaders are moving in the front of the platoons. Group leaders make the decision about changing lanes.
- Platoon members follow the group leader. If the leader changes lanes, group members act accordingly. Members keep a predefined distance (approx. 10 m) from the car in front of them.

Platoon Lane Changes

In order to make it possible to change lane by a whole group, we have modified SUMO's SL2015 lane change model to the SmartSL2015 Lane Change Model.

In the new model the way the groups operate implicates that a lane change model with platooning capabilities has to provide three operating states:

1. Normal working when a smart car is not a part of a group.
2. Working as a group leader needs to calculate with the platoon.
3. Group members have to follow their leaders.

The first and the third operating modes can be provided as the normal operation of the SL2015 model. For the first case, the default operation is suitable, and the third case can be achieved if the decision of SL2015 model is based only on TraCI calls. On the other hand, the second mode requires new computations. Observing the operation of SL2015, it becomes clear that these computations are achievable if the length of a platoon is used instead of that of a single vehicle's. Of course, new communication is also needed to make SL2015 work as desired. For that reason, we use the original messages defined by SL2015. In SmartSL2015 we only change the vehicle which submit those messages accordingly. In order to correctly do so, Lane Change Manager software component was introduced. This component is also responsible for changing the operating mode described above.

6 Intelligent Traffic Control

Concept of Judges

Traditional traffic control systems are based on timing. This idea is relatively simple and was easy to implement. However, it is not scalable at all to the traffic flow. For this reason, in some intersections, induction-loop detectors are placed to measure the incoming traffic and to switch light-phases accordingly. Unfortunately these detectors are not completely reliable. Yet another option at the intersections is to change the program of the traffic lights regularly, for example six times a day.

Utilizing the communication ability of the smart cars of the future, a scalable and robust traffic control system can be created. By an analogy, schedulers known in operating systems can be applicable here, since the basic problem is the same: competing entities in need of a unique undivisible resource. As a part of our research, we implemented such control system based on operating system schedulers. We call these controller devices judges. One judge is placed in every traffic junction.

Distances

The protocols of intersection control are triggered by the approaching cars. We use the following specific distances (see Figure 3):

1. *Reporting distance:* at this point, smart cars report their approach to the Judges.

2. *Stopping distance:* if the judge decision is to stop the vehicle, then this vehicle has to stop at this point.

3. *Point of No Return:* if a vehicle passes this point, it explicitly enters the intersection. Due to safety reasons, judges do not change the "green phase" when a smart car is in the intersection.

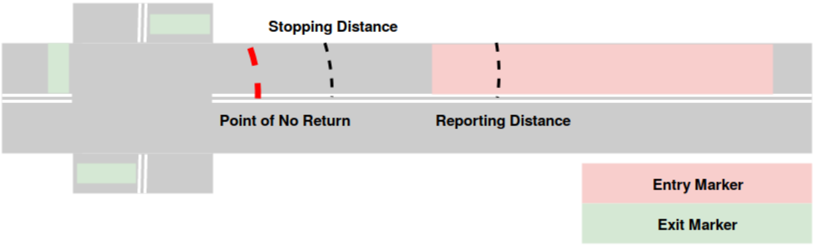

Figure 3: Distances defined on road network

Conflict Classes

To make the scheduling control working, schedulable objects have to be defined. Regarding the granularity, a single vehicle can be schedulable, however it would be very expensive computationally. For that reason larger, so called conflict classes were introduced.

A set of smart cars (a platoon or a group of platoons) which can pass through an intersection simultaneously is called a conflict class (see Figure 4). Every judge knows which directions are passable in the intersection at the same moment. This information is stored in a file and our simulation platform implementation uses it as one of its input. We call a conflict class active, if currently it is able to pass through the intersection. The judges decide which conflict class a smart car belongs to. The judges also decide which conflict class is currently active.

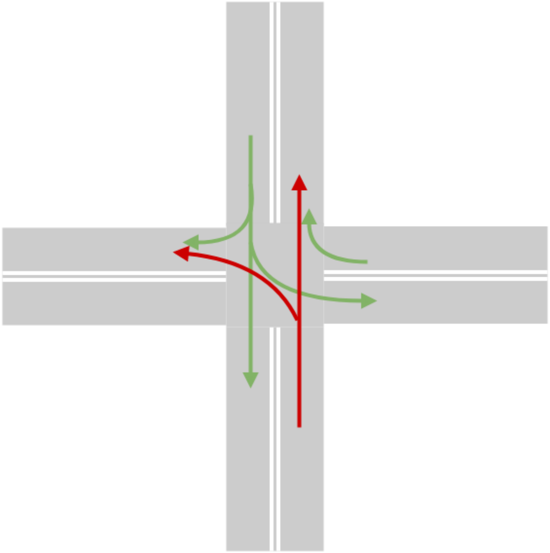

Figure 4: Two possible conflict classes. Vehicles with green directions can form one conflict class. Vehicles with red directions can form another one.

Implementing Judges

In the current implementation an abstract class of judges was created which can be used as the basis class for the future development. A super-class for conflict classes was also created. Considering the concrete schedulers, we decided to implement the simplest Round Robin (RR) scheduler and the theoretically optimal Shortest Job First (SJF) scheduler as intelligent traffic light controllers. A preemptive Round Robin scheduler can be easily adapted to such task. However, SJF needs a good estimate of the remaining time. This can be guessed as the remaining distance from the destination of a smart car, so Minimal Destination Distance First (MDDF) algorithm was used.

Round Robin Judge

This type of scheduler was developed on the base of a simple, preemptive Round Robin scheduler. This scheduler switches phases if the active conflict class contains no cars or the maximum amount of time is exceeded in this period (preemption).

The mentioned maximum time is identical to the one used by the traditional traffic light systems. Because of this, when all directions have sufficient amount of vehicles, then this scheduling provides exactly the same operation as the traditional traffic controlling system. In every other case (eg. there are no cars approaching from one direction) some time can be saved by skipping this direction's time of green phase.

MDDF Judge

The idea of MDDF had been published in [1] and in [2] it was shown that it is an unfair scheduler. Our implementation (solving the fairness problem) operates with two priority levels. At the low priority level the MDDF scheduler is active, meanwhile at the high priority level the Round Robin implementation is in operation, see Figure 5. If a conflict class is not active in the last 90 seconds, the controller automatically switches to a higher priority level. RR scheduling ensures that from this point the conflict class can pass through the intersection in a finite time.

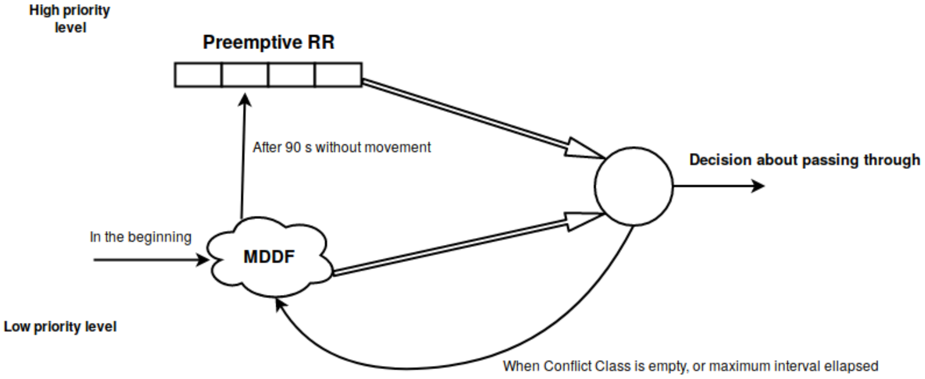
High priority level
Preemptive RR
After 90 s without movement
Decision about passing through
In the beginning
MDDF
Low priority level
When Conflict Class is empty, or maximum interval ellapsed

Figure 5: An overview of the implemented MDDF scheduler. Single arrows depict Judge actions regarding the MDDF-conflict classes. Double arrows represent candidates for the next active conflict class proposed by both the Preemptive RR and the MDDF scheduling logic. Decision maker (shown as a circle) always selects the Preemptive RR's candidate if there is any. Otherwise, MDDF's candidate is selected as the next active conflict class.

7 Simulations and Results

The developed simulation platform was used extensively to compare various traffic situations. In every simulation the same vehicle-type definition, depicting approximately the well-known and typical car, Volkswagen Golf mark IV, was used.

Measured Traffic Parameters

In the simulations the following parameters were measured:

- *Arrived vehicles* = Started vehicles = Running vehicles.
- *Average Time:* a parameter provided by SUMO, referring to the time between the insertion and arrival of a vehicle in simulation.

- *Average waiting time:* a parameter provided by SUMO, referring to the time which a vehicle spent stopped at intersections.
- *Lost time:* a parameter provided by SUMO, referring to the time loss which is caused that a vehicle is not the only one on the route.
- *Sensitivity* = Arrived vehicles/Started vehicles. As it can be seen, it is a value which helps to compare the different methods implemented.

Setup of the Tests

Since the road network of older cities (eg. European cities) basically differs from that of the newer cities (eg. American cities), two different road network topologies were created to test and verify the merits of the proposed system. The results of the intelligent system (with RR or MDDF judges) were compared with the working of a traditional light switching system.

Test Track

The first road network is called Test Track. It is a fictional, rectangular grid network (see Figure 6), simulating an US-like road reality. Some of the roads are one way only, and some of the roads are supporting higher traffic flows as well. For this track only one kind of traffic flow input was defined.

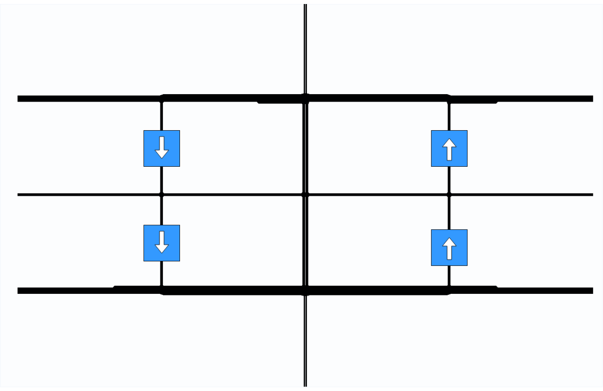

Figure 6: Test Track

BAH Intersection

BAH intersection is one of the major intersections of the inner Budapest. It is an intersection of six streets with relatively heavy traffic (connecting to the highway and the Danube bridges). Some other roads are also added to this network in order to simulate the escape routes (see Figure 7).

Four different traffic input flows were defined for this road network. Two of them simulate the regular traffic flow. One is for the night traffic (with low demand) and one is for the morning (with high demand) traffic. Another two test cases simulate irregular traffic flows,

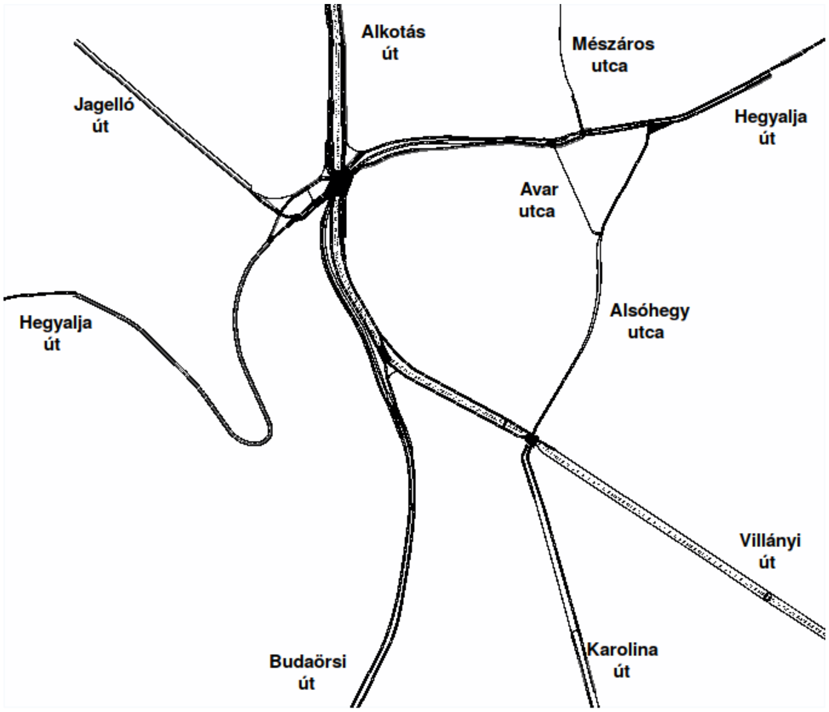

Figure 7: BAH intersection as simulated in SUMO [11]

with the main roads closed (e.g. due to an accident)[2]. In these cases cars are trying to find alternative routes to get through.

Results of the Measurements

On the Test Track, the proposed intelligent system (with both RR or MDDF judges) outperforms the traditional system in every measured value. The amount of the arriving vehicles could be increased up to approx. 160% (for more details, see Figure 8).

Our first configuration[3] of judges for BAH intersection was not satisfactory [2]. Our second attempt was more promising. Average waiting time (see Figure 9) was reduced in every test case and surprisingly the Round Robin scheduler provided better results than the MDDF one did. In the case of the regular, high traffic flow MDDF showed a bit increased average time (see Figure 10) compared to the traditional system. We believe that was caused by the priority switching of the scheduler. However MDDF provided a little bit lower lost time in irregular cases even better than the RR scheduler did (see Figure 11).

The sensitivity of the intelligent system is almost identical as the traditional system, in regular cases. In the most irregular (Irreg2) case however, the intelligent system can provide better results with more than 10% better sensitivity compared to the traditional traffic control, see Figure 12.

[2] One side of Budaörsi street is closed. For 'Irreg1' case it is northbound, and for 'Irreg2' case it is southbound.

[3] The placements of the markers, specifications of the corresponding distances were not correct.

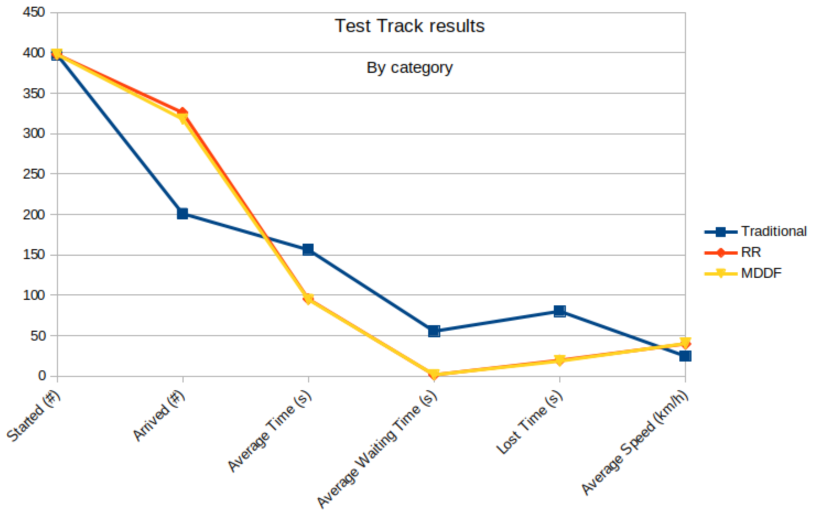

Figure 8: Measurement results on Test Track

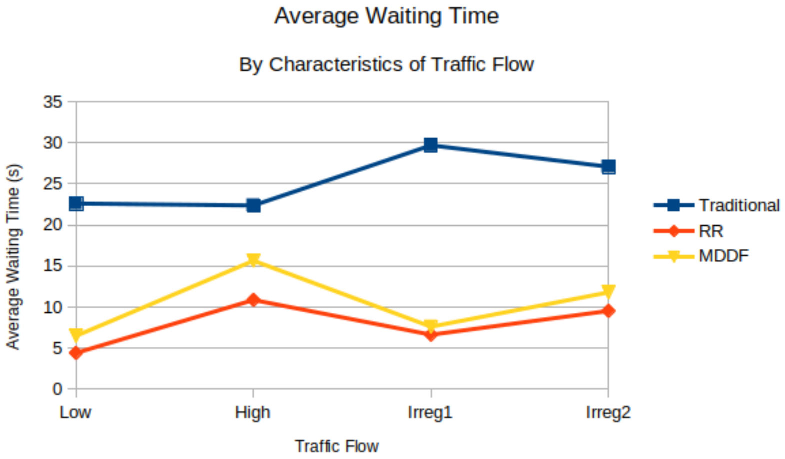

Figure 9: Average waiting time measurement results in BAH intersection

Discussion

Importance of Grouping

The impact of maximum size of platoons on the traffic flow was also investigated. Since only the leader of each platoon has to communicate with judges, theoretically grouping can help to reduce the number of messages in the communication channel. It is roughly equivalent to the speed of the simulation, which is called *Duration Factor* in SUMO. The value of the Duration Factor means how many times faster the simulation time is than the real time.

By using the regular, high traffic flow situation (see Section 7) some additional measurements

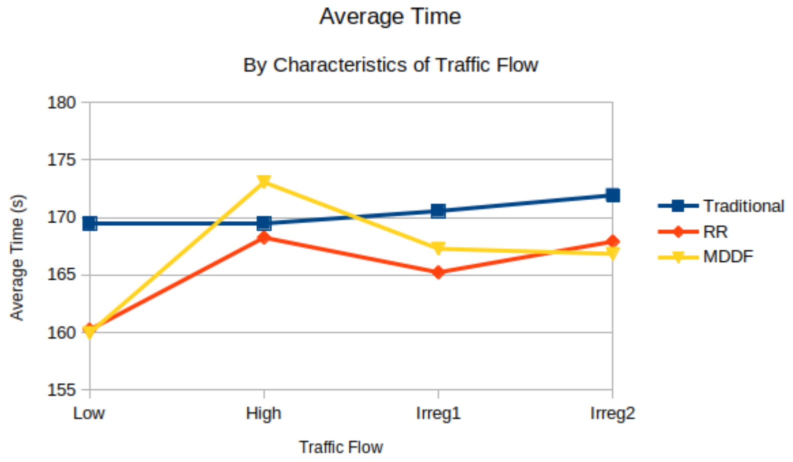

Figure 10: Average time measurement results in BAH intersection

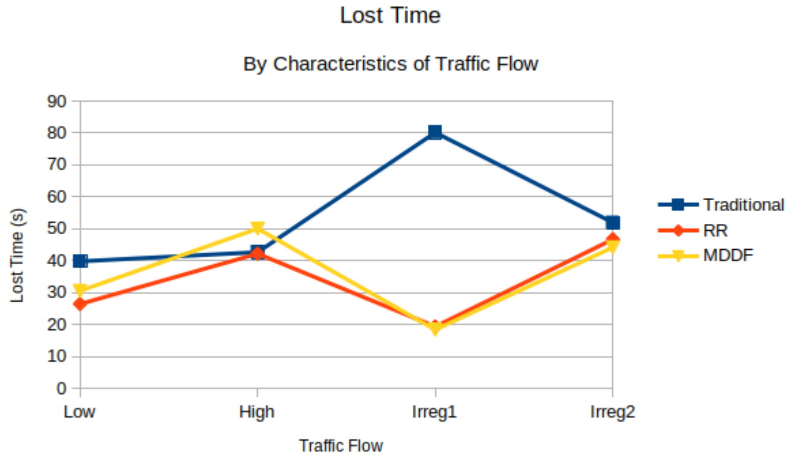

Figure 11: Lost time measurement results in BAH intersection

were made to check how the maximum size of a group[4] influences *Sensitivity*, Average Speed and *Duration Factor*.

As shown in Figure 13, both Sensitivity and Average Speed reach maximum at particular group sizes. In our scenario the places of these maxima are different, so it could be a traffic engineering decision which one is more favorable, and to adjust the maximum group size accordingly. In addition, the results also show that bigger group sizes cause less load on the communication channel and also can be simulated faster.

It is also worth mentioning that Sensitivity is also slightly improved by introducing platooning (setting the maximum group size higher than 1).

[4] Naturally, if the maximum size of a group is 1 then practically no platooning is performed at all.

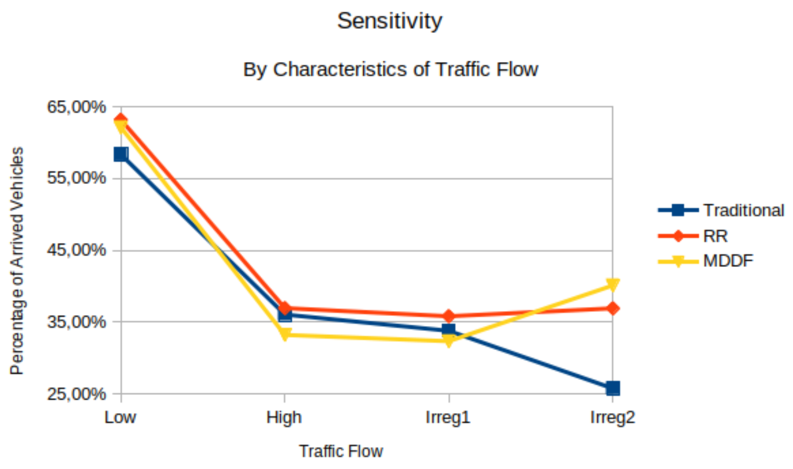

Figure 12: Sensitivity of the methods measured in BAH intersection

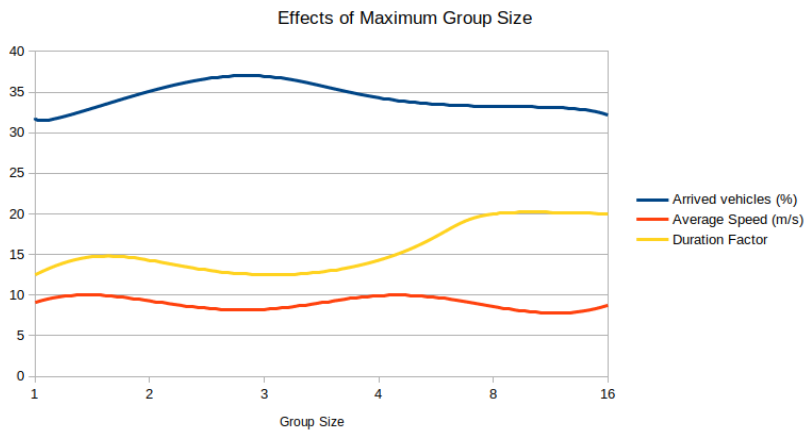

Figure 13: Impact of Different Sizes of Groups on the traffic flow and the simulation kernel

For these two reasons the defined ad-hoc group-forming can be beneficial for improving the traffic flow.

Statistical Significance of the Results

The statistical significance of the obtained results also need to be investigated. To clarify whether the improvements are caused by random events or they are systematic, some further experiments were performed.

In the regular, high traffic flow case (see Section 7) uncertainity was introduced by `--random --random-depart-offset 60` command line options of SUMO. With these randomized input, thirty simulations were performed. In ten cases, the traditional traffic control system, and in twenty other cases (ten cases for RR and ten for MDDF judges) our intelligent solution were tested.

Since the Average Speed only differs by slightly less than 2 $\frac{km}{h}$ (37 $\frac{km}{h}$ vs. 39 $\frac{km}{h}$ in favour of the intelligent solutions)[5], *Sensitivity* is better measure of the improvement of the traffic flow. Our results can be seen in Figure 14.

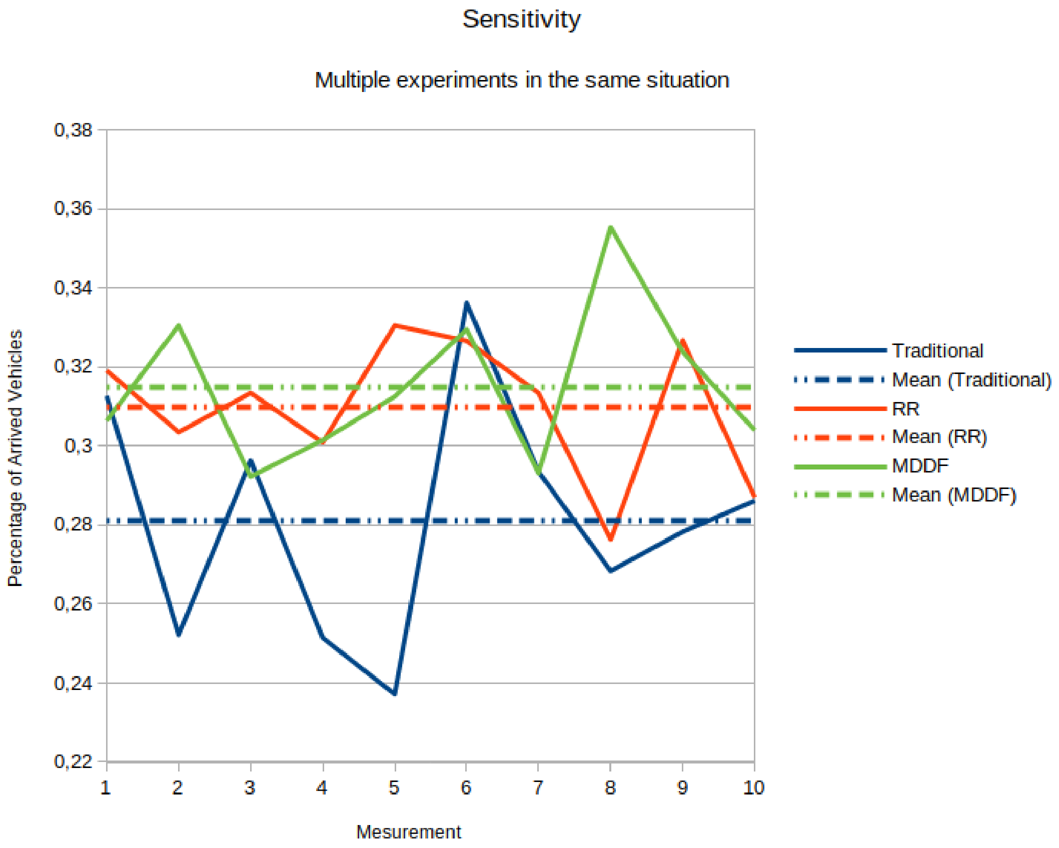

Figure 14: Comparison of the traffic controlling methods, in a randomized, regular, high traffic scenario.

These measurements clearly show that the improvement, compared to the traditional system, is systematic, and has a statistical significance. Moreover, according to these results, MDDF judges statistically slightly outperform RR judges in these situations.

8 Conclusions

By using SUMO based intelligent agent simulation platform, we managed to prove that coupling the platooning concept and the intelligent intersection control can improve traffic flow. Such solution can be implemented without the need of any significant change to the already existing

[5] with a variance of roughly 10 km/h

infrastructure. Since the proposed system is based on simple communication, the required modifications are relatively inexpensive. The European Union has a strategy which states that the vehicles of the future would be equipped with communication devices. Traffic lights also require regular maintenance, so these modules also can be replaced by new intelligent ones.

The development work proved that SUMO is a good basis platform, easily extendable to more sophisticated protocols, permitting experimenting with various adds-on modeling car-level and traffic-level intelligence. The developed simulation platform is a good basis for the future research. It also provides base classes for the development of further conflict class and judge implementations. The knowledge available to the platoon leaders and members can be expanded too. In the future, we plan to investigate the possible benefits of communication between judges. The impact of non-cooperative (human) agents on the system also can be analyzed.

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)*, pages 271–276, Las Vegas, NV, USA, 2013.

[2] L. Alekszejenkó and T. Dobrowiecki. Intelligent vehicles in urban traffic – communication based cooperation. *The IEEE 17th World Symp. on Applied Machine Intell. and Inform. (SAMI 2019), Herl'any, Slovakia*, Jan 2019.

[3] J. Baumfalk, M. Dastani, B. Poot, and B. Testerink. A SUMO extension for norm-based traffic control systems. In M. Behrisch and M. Weber, editors, *Simulating Urban Traffic Scenarios*, pages 55–82. Springer International Publishing, Cham, 2019.

[4] L. Chen and C. Englund. Cooperative intersection management: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 17(2):570–586, February 2016.

[5] K. Dresner and P. Stone. Multiagent traffic management: an improved intersection control mechanism. In *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems - AAMAS '05*, page 471, The Netherlands, 2005.

[6] J. Erdmann. SUMO's lane-changing model. In M. Behrisch and M. Weber, editors, *Modeling Mobility with Open Data*, pages 105–123. Springer International Publishing, Cham, 2015.

[7] P. Fernandes and U. Nunes. Platooning of autonomous vehicles with intervehicle communications in SUMO traffic simulator. In *13th International IEEE Conference on Intelligent Transportation Systems*, pages 1313–1318, 2010.

[8] K. h. N. Bui, O. j. Lee, J. J. Jung, and D. Camacho. Dynamic traffic light control system based on process synchronization among connected vehicles. In *Ambient Intelligence- Software and Applications – 7th International Symposium on Ambient Intelligence (ISAmI 2016)*, pages 77–85. Springer International Publishing, Cham, 2016.

[9] N. Kheterpal, K. Parvate, C. Wu, A. Kreidieh, E. Vinitsky, and A. Bayen. Flow: Deep reinforcement learning for control in SUMO. *SUMO 2018 – Simulating Autonomous and Intermodal Transport Systems, Berlin*, pages 14–16, May 2018.

[10] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner. Microscopic traffic simulation using SUMO. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2575–2582, Nov 2018.

[11] Google Maps:. https://www.google.com/maps/place/Budapest,+BAH+csomopont/@47.4836497,19.0242423,2828m.

[12] J. Mena-Oreja and J. Gozalvez. Permit - a SUMO simulator for platooning maneuvers in mixed traffic scenarios. In *2018 21st International Conference on Intelligent Transportation Systems*

(ITSC), pages 3445–3450, Maui, HI, USA, 2018.

[13] J. Rios-Torres and A. A. Malikopoulos. A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps. *IEEE Transactions on Intelligent Transportation Systems*, 18(5):1066–1077, May 2017.

[14] S. Santini, A. Salvi, A. S. Valente, A. Pescapè, M. Segata, and R. Lo Cigno. Platooning maneuvers in vehicular networks: a distributed and consensus-based approach. *IEEE Transactions on Intelligent Vehicles*, 4(1):59–72, March 2019.

[15] M. Segata. Platooning in SUMO: An open source implementation. *SUMO User Conference 2017, Berlin, Germany*, pages 51–62, May 2017.

[16] M. Segata, S. Joerer, B. Bloessl, C. Sommer, F. Dressler, and R. L. Cigno. Plexe: A platooning extension for Veins. *2014 IEEE Vehicular Networking Conference (VNC)*, pages 53–60, 2014.

[17] Dongbin Zhao, Yujie Dai, and Zhen Zhang. Computational intelligence in urban traffic signal control: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(4):485–494, July 2012.