Queue discharge-based emergency vehicle traffic signal preemption [⋆]

Vít Obrusník [*] **Ivo Herman** [**] **Zdeněk Hurák** [***]

[*] *Faculty of Electrical Engineering, Czech Technical University in Prague (e-mail: obrusvit@fel.cvut.cz).*
[**] *Herman Elektronika, Brno (e-mail: ivo.herman@herman.cz).*
[***] *Faculty of Electrical Engineering, Czech Technical University in Prague (e-mail: hurak@fel.cvut.cz)*

Abstract: In this paper, we document a novel method for emergency vehicle preemption at an intersection controlled by traffic lights. The method relies on wireless vehicle-to-infrastructure (V2I) communication between the emergency vehicle and the traffic lights controller, availability of an accurate estimate of the number of vehicles in the queue, and a mathematical model of dynamics of discharging of the queue. Unlike some occasionally deployed methods that trigger the preemption the instant that the emergency vehicle appears at a prespecified distance from the intersection, the proposed method adapts the activation moment to the actual traffic conditions so that the preemption is as short as possible, thus reducing the impact on the other traffic. The method has been finetuned using numerical simulations in SUMO simulator and experimentally verified in real urban traffic.

Keywords: emergency vehicle preemption, queue modelling, traffic control, intelligent transportation systems.

1. INTRODUCTION

1.1 Motivation, goals

In this paper, we document a method for granting a right-of-way to an emergency vehicle (EV) such as an ambulance, a fire rescue truck and a police car approaching an intersection controlled by traffic lights.

The primary motivations for developing such systems based on *preemption of traffic signals* are the reduction of the travel times of EVs and mitigating the risk of accidents involving EVs passing through the intersection during the red light phase. More elaborate arguments in favor of emergency preemption techniques (as well as related definitions and explanations) are given in the report by U.S. Department of Transportation (2006).

The challenge in developing such preemption schemes is that any interference into the finetuned traffic light signal plan can have an undesirable impact on the smoothness of the other traffic. The more so that actions taken at one intersection influence the situation at neighbouring intersections. One approach to minimize the impact of the emergency vehicle preemption on the other traffic is to keep the preemption period as short as possible. The method proposed in this paper aims at achieving it.

To explain the essence of the proposed method, we first explain the key principle of a family of emergency vehicle preemption methods that are currently deployed in urban traffic control. An area is defined around a given road intersection (typically in the form of a polygon drawn by a transportation engineer into a map, stretching typically a few hundred meters), which, if entered by an EV, triggers a traffic signal preemption in favour of the approaching emergency vehicle. The entry of the EV into the activation polygon around the intersection is detected either by some road-side sensors (infrared) or through onboard GNSS/GPS position sensors and wireless (radio) communication with the traffic lights controller.

In our method, the key enabling technology is the wireless *vehicle-to-everything (V2X)* communication, and, in particular, its subset called *vehicle-to-infrastructure (V2I)* communication. Through its onboard V2I unit, the emergency vehicle approaching the intersection communicates continuously its position and velocity to the road-side V2I unit located at the intersection. This unit fuses the incoming data with an estimate of the number of vehicles in the queue, a mathematical model of dynamics of the queue discharging, and the information about the current phase of the traffic signal cycle, and generates an activation signal for the traffic light controller to trigger a preemption.

A simplified description of our algorithm is that the time needed for all vehicles in the queue to reach the saturation speed is computed first, taking the dynamics of the queue into consideration, and then the time to start the preemption is determined so that the EV joins the queue while the tail vehicle is already travelling at the saturation speed.

Compared to the commonly deployed *distance-based* preemption triggering mentioned above, the proposed method

[⋆] The research was funded by Technology Agency of the Czech Republic within the program Epsilon, the project TH03010155.

can be regarded as *estimated time of arrival*-based. Yet another interpretation is that the preemption polygon *adapts* to the velocity of the approaching EV, the number of vehicles in the queue, and the (possibly time-varying) parameters of the intersection.

1.2 State of the art

A survey paper by Humagain et al. (2019) gives an overview of recent techniques to reduce travel times of EVs, namely, they categorize routing, preemption and approaches combining both of these. Based on their categorization, our work would fall into a preemption technique using Vehicular Adhoc Network (VANET) concept, initiated by queue length and utilizing method of requesting green light at an appropriate time.

Our work is close to the work of Noori et al. (2016), where authors also use the number of vehicles in a queue in front of an intersection to vary the time to request preemption based on the computed time needed to discharge the queue. The work of Bieker-Walz and Behrisch (2019) uses a similar approach and shows simulations on a series of consecutive intersections. Wang et al. (2013) estimate the queue length and describe simplified queue tail speed model to get the time for discharging the queue. Our paper differs in a level of details we use to derive necessary parameters. The model used in this paper is more detailed and can be adjusted to different types of intersections.

We already mentioned that the algorithm relies on the availability of an estimate of the number of vehicles in the queue. There are a few methods published in the literature that are relevant for real-time use, for example Tiaprasert et al. (2015), Liu et al. (2009), and Li et al. (2013). In this paper, we do not elaborate on this topic and assume that the (estimate of the) number of vehicles in the queue is available to the controller.

1.3 Outline of the paper

This document is structured as follows: in Section 2 we describe a chosen mathematical model of dynamics of queue discharging that is needed for determining the control parameters. The algorithm is then explained in Section 3. Some comments on the simulation are provided in Section 4. An experiment in real traffic is documented and discussed in Section 5. Conclusions as well as plan for future work are given in the final Section 6.

2. QUEUE DISCHARGE MODEL

The queue model is adopted from Akçelik Rahmi (2002), which we review in this section. We also briefly describe how to obtain the parameters.

Considering a queue of vehicles as an entity, its behaviour can be described by the following set of functions of the time since start of a green phase t:

$$v_\mathrm{s}(t) = v_\mathrm{n}\left(1 - e^{-m_v(t-t_\mathrm{r})}\right), \tag{1}$$

$$q_\mathrm{s}(t) = q_\mathrm{n}\left(1 - e^{-m_q(t-t_\mathrm{r})}\right), \tag{2}$$

$$h_\mathrm{s}(t) = \frac{h_\mathrm{n}}{1 - e^{-m_q(t-t_\mathrm{r})}}, \tag{3}$$

where

$$\begin{aligned}
t &= \text{time since start of green phase } [s],\\
t_\mathrm{r} &= \text{response time of first vehicle } [s],\\
v_\mathrm{s}(t) &= \text{discharge speed at time } t\ [km/h],\\
q_\mathrm{s}(t) &= \text{queue discharge volume at time } t\ [veh/h],\\
h_\mathrm{s}(t) &= \text{queue discharge headway at time } t\ [s],\\
v_\mathrm{n} &= \text{maximum queue discharge speed } [km/h],\\
q_\mathrm{n} &= \text{maximum queue discharge volume } [veh/h],\\
h_\mathrm{n} &= \text{minimal queue discharge headway } [s],\\
m_v &= \text{a parameter in the speed model } [-],\\
m_q &= \text{a parameter in the discharge volume model } [-].
\end{aligned}$$

Parameters v_n, m_v and t_r can be identified directly by fitting the function (1) to the velocity measurements at a stop line of an intersection. Such simulated measurements and a fitted function (1) can be seen in Fig. 1 (top right). We simulated a queue of vehicles discharging at traffic light on a straight road. The vehicles are governed by Intelligent Driver Model car-following model (Treiber and Kesting, 2013). The simulation was done in SUMO framework (Lopez et al., 2018).

Maximum queue discharge volume (or saturation flow rate) q_n was set according to an empirically observed formula 16.4 from Akcelik et al. (1999) relating the maximum discharge speed to the saturation flow rate:

$$q_\mathrm{n} = 1012 + 24.5 \cdot v_\mathrm{n}. \tag{4}$$

Flow rate (2) and headway time (3) are in a relation: $h_\mathrm{s}(t) = 3600/q_\mathrm{s}(t)$, which we can use to get the minimum (saturation) headway time as:

$$h_\mathrm{n} = 3600/q_\mathrm{n}. \tag{5}$$

The parameter m_q in the discharge volume model is obtained from equation 8 (Akçelik Rahmi, 2002):

$$m_q = 1000 \cdot m_v \frac{v_\mathrm{n}}{q_\mathrm{n}\, L_\mathrm{hj}}, \tag{6}$$

where L_hj is the average jam spacing of vehicles in meters computed as the sum of average vehicle length L_v and minimal gap L_s. We used $L_\mathrm{v} = 4.3$ and $L_\mathrm{s} = 2.5$ in the simulation, resulting in $L_\mathrm{hj} = 6.8$. The model equations (2) and (3) with these parameters are plotted in Fig. 1 (bottom left and right respectively).

Now we are interested in average queue departure response time or average reaction time of a driver to start moving after a vehicle ahead starts moving. According to equation 10 (Akçelik Rahmi, 2002), we can compute the time as:

$$t_\mathrm{x} = h_\mathrm{n} - \frac{3.6 \cdot L_\mathrm{hj}}{v_\mathrm{n}}. \tag{7}$$

Average spacing of vehicles at maximum queue discharge flow can be obtained as follows:

$$L_\mathrm{hn} = \frac{v_\mathrm{n} \cdot h_\mathrm{n}}{3.6} \tag{8}$$

Using equations 11-18 (Akçelik Rahmi, 2002) we get average acceleration time t_a of a single vehicle to accelerate from zero to saturation speed. The parameters and their values which we used are listed in Table 1.

Moreover, assigning $n(t)$ as the number of vehicles which are still in front of a stop line of the intersection, then the following first order differential equation holds:

Table 1. Parameters of the model used in algorithm.

v_n	q_n	h_n	m_v	m_q	t_x	t_a	L_hn
35.25	1875	1.92	0.22	0.62	1.22	5.82	18.79

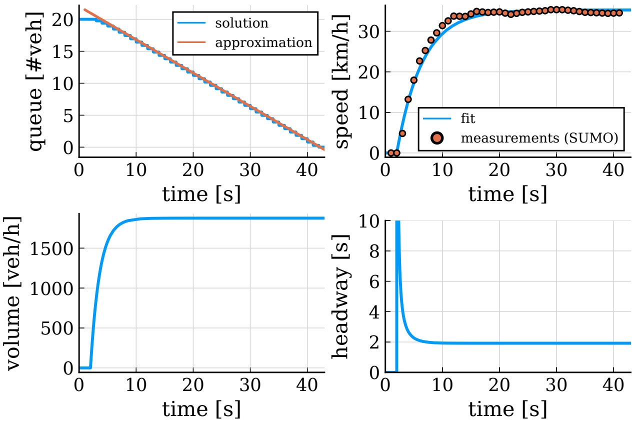


Fig. 1. Queue discharge model characteristics.

$$\dot{n}(t) = -q_\mathrm{s}(t), \tag{9}$$

with the initial condition a number of queued vehicles before the start of green phase n_0. A numerical solution for $n_0 = 20$ is plotted in the Fig. 1 (top left) together with an approximation of the solution. The approximation is used in section 3:

$$n_\mathrm{app}(n_0, t) = n_0 - \frac{q_\mathrm{n} \cdot t}{3600} + c, \tag{10}$$

where c is used for more precise fit.

To sum up this section, we exploited the model of a discharging queue to get a set of parameters: average reaction time t_x, average acceleration time t_a, average saturation flow spacing of vehicles L_hn and a function approximating the discharging of vehicles $n_\mathrm{app}(n_0, t)$.

3. DESCRIPTION OF ALGORITHM

This section describes the computing process which is done on a unit of an intersection controller. The parameters identified in section 2 are saved in a memory of the controller. The operational speed (or desired speed) of the EV: v_op, might be saved as well, but an actual speed of the EV can be also be used. We use constant operational speed in our work. There are two inputs the controller needs which must be provided as frequently as possible:

(1) GPS position (and optionally driving speed) of the EV approaching the intersection,
(2) number of vehicles queued in front of a stop line of the intersection.

While the former is easily supplied to the controller with a help of V2X system and an On-Board Unit (OBU) with GPS capabilities, the latter is more challenging and is not developed in this paper.

The output of the computation is the time after which the controller sends the command to request preemption phase: T_P.

Let's consider a situation in which N cars stand stopped on red lights at an intersection when a traffic lights controller receives a beacon, containing information about a position of an approaching EV. Then the following steps are executed:

(1) Travel distance D is taken from the received position and expected arrival time T_A of the EV is computed based on saved operational speed:

$$T_\mathrm{A} = \frac{D}{v_\mathrm{op}}. \tag{11}$$

(2) Time after which the very last vehicle of the queue hits the saturation speed v_n is computed:

$$T_\mathrm{L} = N \cdot t_\mathrm{x} + t_\mathrm{a}. \tag{12}$$

(3) A number of vehicles in front of the intersection stop line at time T_L is taken from $n_\mathrm{app}(t)$. Knowing the average spacing L_hn of vehicles in saturation speed v_n, we can compute the distance between the last moving vehicle and the intersection. This allows to compute the time needed by EV to pass the distance of the trailing queue tail:

$$T_\mathrm{X} = \frac{n_\mathrm{app}(T_\mathrm{L}) \cdot L_\mathrm{hn}}{v_\mathrm{op}}. \tag{13}$$

(4) Time to start preemption, T_P, is computed as:

$$T_\mathrm{P} = T_\mathrm{A} - T_\mathrm{L} - T_\mathrm{X}. \tag{14}$$

Figure 2 shows how far from an intersection stop line would be a request for preemption sent, depending on v_op and number of vehicles n_0.

Note that this approach resembles the work of Wang et al. (2013), where authors also compute time when to start preemption by first computing expected arrival time of EV and then subtracting time needed for a queue to dissipate based on a model. They use a queue tail travel speed model identified from probe vehicles. However, we think that their approach lacks the time needed for the last vehicle to start moving. Their approach would work neatly if all the vehicles in the queue start moving simultaneously right after the green signal appears on traffic lights.

One more remark is necessary: n_app can be negative at time T_L for a small initial number of vehicles in the queue. A negative number means that some vehicles are already behind the stop line of the intersection when they hit the saturation speed. This would cause T_X to be negative, effectively delaying the time to start preemption. This is correct behaviour because it preserves the premise that the EV meets the queue when even the last vehicle is at saturation speed. With T_X negative, the EV meets other vehicles at some downstream distance (in the direction of travel) from the stop line of the intersection.

To deploy the algorithm in a realistic setting, some more complexity needs to be added:

- Firstly, traffic lights can rarely switch from regular operation to preemption immediately. More often, some phases need to finish (e.g. in-progress pedestrian phase or left turn arrow) before the preemption might even start. To cope with the problem of switching, the output of the algorithm must be adjusted by additional parameter holding a time needed to do the switch t_sw.
- Secondly, we need to deal with the situation in which the intersection controller receives the beacon in time when traffic light signal state is green for the approach

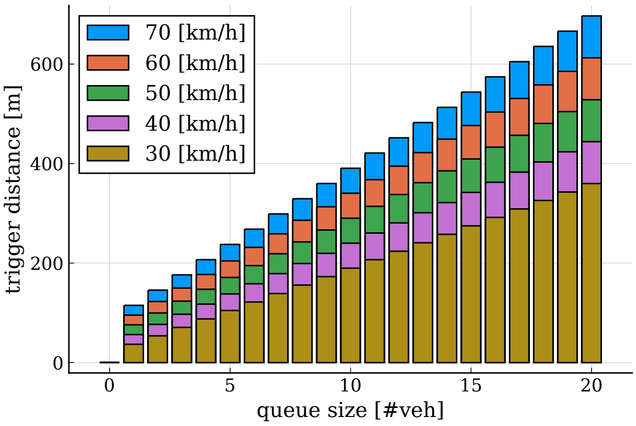


Fig. 2. Preemption trigger distance based on number of vehicles and operation speed of the EV.

to the intersection that the EV is using. To do this, the algorithm needs to decide whether it is necessary to request preemption immediately or whether we can let the controller continue in normal operation for some more time.

To tackle the second problem, we came up with the following solution. We can expect that the queue would start to discharge in the same way as with the preemption, hence the traffic controller should keep the number of vehicles that were standing in the queue before the green light began. Assuming that the traffic light controller provides an actual time of current green of the approach of EV: T_G, we can add T_G to the computation of T_P until it cancels out with the delay effect of the queue that was standing there (lines 8 and 10 of Algorithm 1). Moreover, there is a minimal time guaranteed for each phase to remain according to a regular signal plan: t_min. We can use it so the algorithm does not request preemption before this time. This would block other road users for more time than is necessary. We used $t_\mathrm{min} = 10s$ for the experiment. Consequently, the traffic light controller must be able to give us the time of the current phase: T_ph, to compare it with t_min. Note that T_G and T_ph might not be equal, because a next phase might be different (e.g. by adding green for pedestrian crossing) but the signal state for EV remains the same. For a situation when the traffic light controller changes the signal to red, we also need an estimation of how many vehicles might queue up before we can get green again. To do the estimate, we chose (rather arbitrarily but conservatively enough) the saturation flow of the intersection q_n. The parameter t_sw is useful here again because it is the minimal time necessary to switch to the preemption phase. Using these values, we compute the delay caused by the expected queue (lines 14-18 of Algorithm 1). To recapitulate, when the signal state is green for the approach of EV, the algorithm uses a combination of factors described above (line 11 of the Algorithm 1). The preemption request is sent only if the current phase is active for a longer time than the minimal phase time (t_min) and the delay effect of sudden red phase would be longer than the EV needs for preemption.

A pseudocode of the algorithm which we implemented in both simulation and real experiment is shown in Algorithm 1.

Algorithm 1 Traffic light controller

```
 1: procedure processInputs($N$, pos)
 2:     $T_\mathrm{A} \leftarrow \mathrm{compute}T_\mathrm{A}(\mathrm{pos})$
 3:     $T_\mathrm{L} \leftarrow \mathrm{compute}T_\mathrm{L}(N)$
 4:     $T_\mathrm{X} \leftarrow \mathrm{compute}T_\mathrm{X}(N, T_\mathrm{L})$
 5:     if isEvApproachWayRed then
 6:         $T_\mathrm{P} \leftarrow T_\mathrm{A} - T_\mathrm{L} - T_\mathrm{X}$
 7:     else
 8:         $T_\mathrm{G} \leftarrow \mathrm{getCurrentGreenTime}()$
 9:         $T_\mathrm{ph} \leftarrow \mathrm{getCurrentPhaseTime}()$
10:         $T_\mathrm{P} \leftarrow T_\mathrm{A} - \max(0,\ T_\mathrm{L} + T_\mathrm{X} - T_\mathrm{G})$
11:         if $T_\mathrm{ph} > t_\mathrm{min}$ & $T_\mathrm{P} < t_\mathrm{sw} + \mathrm{expDelay}(t_\mathrm{sw})$ then
12:             $T_\mathrm{P} \leftarrow 0$
13:     requestPreemptionInTime($T_\mathrm{P}$)
14: procedure expDelay($t$)
15:     $N = q_\mathrm{n}/3600 \cdot t$
16:     $T_\mathrm{L} \leftarrow \mathrm{compute}T_\mathrm{L}(N)$
17:     $T_\mathrm{X} \leftarrow \mathrm{compute}T_\mathrm{X}(N, T_\mathrm{L})$
18:     return $T_\mathrm{L} + T_\mathrm{X}$
```

4. SIMULATION

The algorithm is implemented in Veins (Vehicles in Network Simulation) framework by Sommer et al. (2011). Veins couples OMNeT++ (Discrete Event Simulator primarily used for building network simulations) to SUMO, which enables to simulate not only traffic mobility but also the aspect of V2X communication. There are two simulation scenarios together with a manual of how to run them available on GitHub (see `https://github.com/aa4cc/evp`).

The first scenario is called *discharge* and simulates only a straight road blocked by traffic lights. The traffic light signal is changed only after by request from arriving EV. A visualisation of SUMO outputs from one run of this scenario with 20 vehicles is plotted in Fig. 3.

The second scenario called *brno_por* is more realistic and covers the intersection 206 from Brno city (more information in Section 5). The traffic demand was determined using the measurements from induction loops (from Tuesday, November 20, 2018) placed at the site. A static traffic lights signal plan designed for the real intersection is also a part of the simulation.

5. FIELD EXPERIMENTS

The algorithm was tested on two distinct intersections in Brno city (Czech Republic). The first, isolated T-shaped intersection of Olomoucká and Černovická streets[1], was used to experiment and tune the algorithm. The second intersection of Vídeňská and Poříčí streets[2] is a cross-shaped intersection with usually heavy traffic throughout the day. We will refer to the second intersection as intersection 206. The implementation was done in cooperation with Herman electronics company[3].

We used an ordinary passenger car equipped with V2X unit using the IEEE 802.11p protocol. The algorithm was

[1] `https://goo.gl/maps/UymSn5k9k9QAc1Qe9`

[2] `https://goo.gl/maps/vnYRSda3bWX3zjnN8`

[3] `http://www.herman.cz/en`

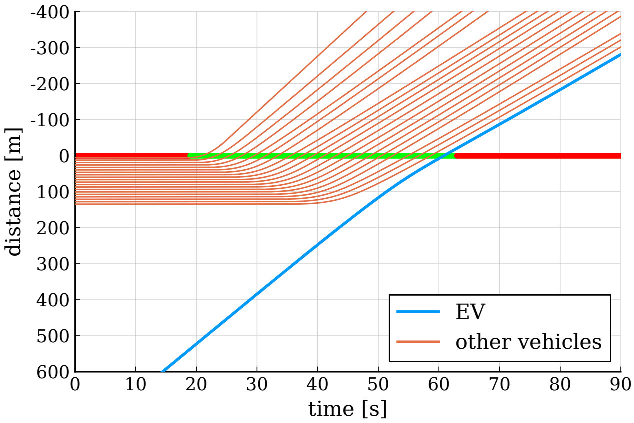


Fig. 3. Time-distance plot of the EV approaching the intersection in simplistic simulation scenario. Vertical line shows the distance to the stop line of the intersection. Horizontal line at distance 0 shows the signal state of traffic lights, the green state is triggered by the algorithm based on the distance of the EV from the stop line and on the number of vehicles in the queue (20 in this case).

implemented according to Section 3 on the traffic light controller unit that is also capable of V2X communication. The maximal range of communication between the vehicle and traffic light controller is around 600m, which was measured beforehand by the Herman company. To further increase the reach of beacon messages, forwarding via V2X-enabled public transport vehicles is possible. The public transport vehicle that successfully receives the beacon can resend it immediately. In this case, a hop count variable should be used to prevent flooding of messages.

We estimated the current queue length by (person) observation of the intersection. The number indicating the count of vehicles was then uploaded to the traffic light controller via mobile phone application every few seconds.

The parameters used were the same as listed in Table 1. The test car drove in constant operation speed 50 km/h.

The distance-based emergency vehicles preemption system is currently operational on the intersection 206, Fig. 4b shows the comparison of trigger distance of the distance-based system and our system which is based on the number of vehicles and the model of a queue.

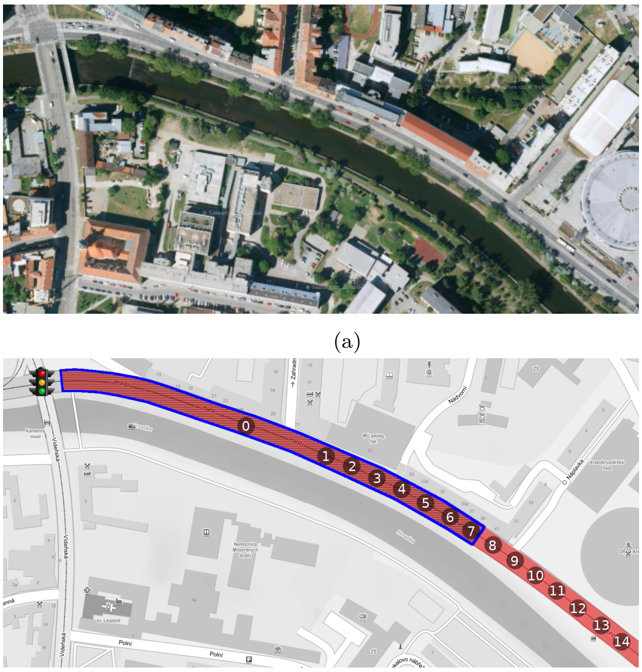
The results are promising. The time computed by the algorithm is enough for all vehicles in front of the EV to reach saturation speed. We made a video of two transits. First, we encountered one vehicle standing in front of the intersection. Second, we had a queue of thirteen vehicles ahead. The preemption was requested at a distance of 240m and 517m respectively. Total time from preemption request until the end of preemption was 21s and 41s respectively. A preview of the video can be found in Fig. 5 and the video can be found on YouTube (see `https://youtu.be/WS80hyG2-rM`).

6. CONCLUSION AND FUTURE WORK

In this paper, we presented a novel method for emergency vehicle traffic signal preemption. We aimed at improving the class of existing methods that trigger the preemption upon the emergency vehicle passing through some *apriori specified distance* from the intersection. The claimed improvement consists in *adaptation of this triggering distance* to the number of vehicles in the queue, parameters of the (model of) dynamics of the queue, the current phase of the traffic signal, and the communicated position and speed of the emergency vehicle.

We not only analyzed the method analytically and through simulations but we also conducted experiments in real traffic, which were documented through a video.

The method needs a mathematical model of the dynamics of discharging of the queue. The parameters of such a model do certainly depend on the type of the particular



(a)

(b)

Fig. 4. (a) Aerial picture taken from `www.mapy.cz` of the east approach of the intersection 206. (b) Greyscale map of the east approach of the intersection 206 with a static polygon area (blue) and areas (black circles) where the preemption would be triggered by the EV based on a number of vehicles (numbers in black circles) by our approach for $v_{\mathrm{op}} = 50$ km/h.

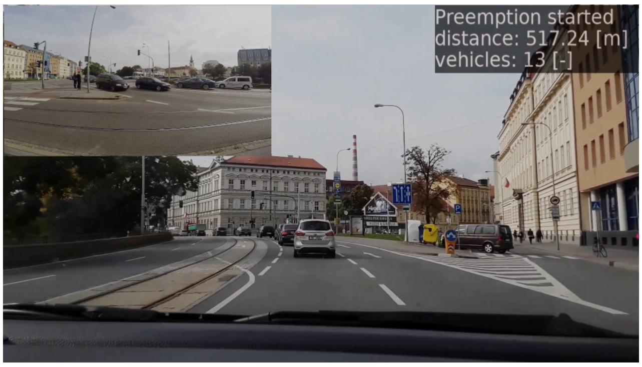


Fig. 5. Preview of the video from the experiment described in the main text.

intersection. In our experiments, we used a model tailored to straight lanes without turning but other situations such as straight lanes with right-turning exhibit different queue discharging dynamics, hence appropriate parameters need to be determined. Developing automatic *parameter identification* procedures is one of our future tasks.

The method also relies on knowing the current phase of the traffic signal cycle. In principle, not only the current but also the near-future phases could be made available to the algorithm, which could help to improve the performance. Due to some (certainly solvable) technical issues in our project these were not available. Extending the method so that it could incorporate the traffic signal plans then constitutes an immediate extension of the presented method.

The method is helpless in case of heavy congestions, when the queues are very long, not to speak of grid locks, when the cars have nowhere to leave the intersection for. Special actions need to be taken under such conditions.

Being model-based, the method is sensitive to deviations of reality from the model. Presence of bikes or heavy trucks can present such a deviation, hence robustness of the method must be analyzed and possibly improved.

The presented method constitutes a model-based feedback control problem. However, the model is not given in any format popular in the control systems domain such as state equations or transfer function. This makes applications of both basic and advanced computational control design difficult if not impossible. Exploring the possibility to model the system consisting of an EV and a platoon of vehicles in the queue as a single dynamic system described by differential equations constitutes another future work of ours. If successful, it could enable formulating the problem of generating the emergency vehicle preemption as the *minimum time control problem* (the EV reaching the tail vehicle in the minimum time but as smoothly as possible).

ACKNOWLEDGEMENTS

We acknowledge Loi Do for significant help during the experiments. We also express our gratitude to Jiří Zukal for expert consulting, to Jan Přikryl for suggesting the queue discharge model, to Laura Bieker-Walz for her work on modelling emergency vehicles in SUMO and to the engineers from BKOM Co. Inc. (Brněnské komunikace a. s.) for providing data.

REFERENCES

Akcelik, R., Besley, M., and Roper, R. (1999). Fundamental Relationships for Traffic Flows at Signalised Intersections. *Research Report*, (ARR 340). URL `https://trid.trb.org/view/655848`.

Akçelik Rahmi (2002). Queue Discharge Flow and Speed Models for Signalised Intersections. In Besley Mark and Michael A. P. Taylor (eds.), *Transportation and Traffic Theory in the 21st Century*, 99–118. Emerald Group Publishing Limited. doi:10.1108/9780585474601-006.

Bieker-Walz, L. and Behrisch, M. (2019). Modelling green waves for emergency vehicles using connected traffic data. In *Proceedings of SUMO User Conference 2019*, 10–2. doi:10.29007/sj1m. URL `https://easychair.org/publications/paper/6KGt`.

Humagain, S., Sinha, R., Lai, E., and Ranjitkar, P. (2019). A systematic review of route optimisation and preemption methods for emergency vehicles. *Transport Reviews*, 0(0), 1–19. doi:10.1080/01441647.2019.1649319.

Li, J.Q., Zhou, K., Shladover, S.E., and Skabardonis, A. (2013). Estimating Queue Length under Connected Vehicle Technology: Using Probe Vehicle, Loop Detector, and Fused Data. *Transportation Research Record*, 2356(1), 17–22. doi:10.1177/0361198113235600103.

Liu, H.X., Wu, X., Ma, W., and Hu, H. (2009). Real-time queue length estimation for congested signalized intersections. *Transportation Research Part C: Emerging Technologies*, 17(4), 412–427. doi:10.1016/j.trc.2009.02.003.

Lopez, P.A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., and WieBner, E. (2018). Microscopic Traffic Simulation using SUMO. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2575–2582. doi:10.1109/ITSC.2018.8569938. ISSN: 2153-0017, 2153-0009.

Noori, H., Fu, L., and Shiravi, S. (2016). Connected-Vehicle-Based Traffic Signal Control Strategy for Emergency Vehicle Preemption. URL `https://trid.trb.org/view.aspx?id=1394483`.

Sommer, C., German, R., and Dressler, F. (2011). Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis. *IEEE Transactions on Mobile Computing*, 10(1), 3–15. doi:10.1109/TMC.2010.133.

Tiaprasert, K., Zhang, Y., Wang, X.B., and Zeng, X. (2015). Queue Length Estimation Using Connected Vehicle Technology for Adaptive Signal Control. *IEEE Transactions on Intelligent Transportation Systems*, 16(4), 2129–2140. doi:10.1109/TITS.2015.2401007.

Treiber, M. and Kesting, A. (2013). Car-Following Models Based on Driving Strategies. In M. Treiber and A. Kesting (eds.), *Traffic Flow Dynamics: Data, Models and Simulation*, 181–204. Springer Berlin Heidelberg, Berlin, Heidelberg. doi:10.1007/978-3-642-32460-4_11.

U.S. Department of Transportation (2006). Traffic signal preemption for emergency vehicles: a cross-cutting study. Technical report, U.S. Department of Transportation, Intelligent Transportation Systems, Washington, DC. URL `https://www.hsdl.org/?abstract&did=17589`. OCLC: 70657844.

Wang, Y., Wu, Z., Yang, X., and Huang, L. (2013). Design and Implementation of an Emergency Vehicle Signal Preemption System Based on Cooperative Vehicle-Infrastructure Technology. *Advances in Mechanical Engineering*, 5, 834976. doi:10.1155/2013/834976.