IET Intelligent Transport Systems

Special Section: Scientific Seminar of the Italian Association of Transport Academicians (SIDT) 2019

Traffic management system for smart road networks reserved for self-driving cars

ISSN 1751-956X
Received on 16th October 2019
Revised 20th April 2020
Accepted on 27th April 2020
E-First on 20th July 2020
doi: 10.1049/iet-its.2019.0675
www.ietdl.org

Angela Di Febbraro[1,2], Federico Gallo[1,2], Davide Giglio[1,2] ✉, Nicola Sacco[1,2]

[1]*Department of Mechanical, Energy, Management, and Transport Engineering (DIME), University of Genoa, Via Montallegro 1, 16145 Genova, Italy*

[2]*Italian Centre of Excellence for Logistics, Transport, and Infrastructures (CIELI), University of Genoa, Via Vivaldi 5, 16126 Genoa, Italy*

✉ *E-mail: davide.giglio@unige.it*

Abstract: A model of a smart road network consisting of unsignalised intersections and smart roads connecting them is considered in this work with the aim of presenting a traffic management system for self-driving cars (or, more generally, autonomous vehicles) which travel the network. The proposed system repeatedly solves a set of mathematical programming problems (each of them relative to a single intersection or to a single road stretch of the network) within a decentralised control scheme in which each local intersection controller and each local road controller communicates with the fully autonomous vehicles in order to receive travel data from vehicles and to provide speed profiles to them once determined the optimal solution of the problem. In order to reduce the computational effort required to provide the optimal solution, a discrete-time approach is adopted so that, in each time interval, a limited number of vehicles are taken into consideration; in this way, solutions can be determined in a very short time thus making the proposed model compatible with a practical application to real traffic systems. The proposed model is general enough, and can be adapted to different scenarios of smart road networks reserved for self-driving cars.

1 Introduction

By 2050, around 65% of the world population will live in cities, and such a growing number of urban citizens poses a big challenge for already congested and densely populated urban areas [1]. At the same time, automation technologies applied to cars are becoming more and more diffused, opening the roads to fully autonomous vehicles about which it is worldwide recognised that they will have a key role in the future road transport policies as they can contribute to increase the efficiency and safety of transport systems and to reduce energy consumption and emissions [2]. In this paper, in order to exploit these potentials, it is addressed a model of a smart road network consisting of unsignalised intersections and smart roads connecting them, with the aim of presenting a traffic management system (TMS) for self-driving cars and autonomous vehicles which travel the network.

The traffic scenarios considered in this paper are relative to a road network travelled by several fully connected and automated vehicles. An SAE level 4 or 5 is required for vehicles so that they can safely travel the smart roads and cross the unsignalised intersections with the speeds (and the accelerations) determined by solving the optimisation problems here proposed, without the need of any human interaction. Each vehicle is assumed to be able to communicate with the infrastructure (through specific on-board vehicle-to-infrastructure (V2I) devices), in order to provide the information about its travel to the network controller (NC), and to receive the speed profiles from the local controllers associated with the roads and the intersections.

The TMS has the dual purpose of optimising the performance of the whole system (network-level optimisation) and of determining the trajectories of vehicles that ensure a safe and fast travel through the network (local-level optimisation). To achieve such targets, a set of mathematical programming problems (each of them relative to a single intersection or to a single road stretch of the network) are repeatedly solved within a decentralised control scheme in which each local intersection controller and each local road controller communicates with the fully autonomous vehicles in order to receive travel data from vehicles and to provide speed profiles to them. A key element of the proposed approach is relevant to the mathematical programming problems which model the behaviour of self-driving vehicles through the roads and the intersections, and provide, once solved, the optimal trajectories of vehicles. Such problems take into consideration the physical space actually occupied by vehicles during their journey, in order to provide solutions that is safe. In addition, specific comfort constraints are also introduced to avoid strong accelerations and decelerations. In this paper, the mathematical model for smart roads is described in details and the one relative to unsignalised intersections is also discussed. Besides, the whole network is also considered and the exchange of information among nodes (intersections) and links (roads) is defined, so that it is possible to execute sequentially the optimisation problems relative to the single elements of the network.

The paper is organised as follows. In the following subsection, a literature review on methods and models for the optimisation of traffic systems with autonomous vehicles is reported. In Section 2, the architecture of the proposed traffic management system is presented and the network-level modelling and the local-level one are described. The assumptions considered in this work and some geometric considerations about trajectories to be followed by self-driving cars are reported in Sections 3 and 4, respectively. The mathematical programming problem for smart roads is presented in Section 5 and, in the same section, some specific features of the model for unsignalised intersections are briefly introduced. The paper ends with the description of some experiments (Section 6) and some concluding remarks (Section 7).

1.1 Literature review

Connected and automated vehicles (CAVs), in their ultimate form (fully unmanned and autonomous), will enable completely new transport systems to be realised. The interest of the research community on this topic is very high: many events take place yearly worldwide (recently, [3, 4]) and the relevant literature is growing faster. In this section, a brief literature review on different control strategies for road intersections and networks is reported, mainly considering the works that assume the presence of only autonomous vehicles, as in this paper. As a matter of fact, such strategies bring improvements regarding roads and intersections performance while guaranteeing safety. Therefore, this topic has received great attention from the research community, as shown, for example in [5], a survey of some research frontiers in this trend.

Research streams in the field of CAV are in many cases based on preceding studies on the technology of connected vehicles. Connectivity is actually seen as an enabling technology for vehicle automation. Although connected vehicles do not carry out automated driving, connectivity enables and fosters the expansion of autonomous vehicles by making distributed information and big data accessible. To cite a few: in [6] a signal control algorithm for an isolated intersection is proposed, analysing various penetration rates of connected vehicles present in a traffic stream, and evaluating the benefits of this technology; in [7] a VTL (virtual traffic light) algorithm aiming at defining the priorities within intersections for connected vehicles is studied; in [8] a microscopic traffic simulation model for such vehicles is designed.

Going beyond the research closely related to connected vehicles, various authors have focused their research on different themes relative to CAV. In [9], signal control strategies for an isolated intersection are determined to assume that three categories of vehicles populate the intersection (conventional vehicles, connected but non-automated vehicles, and automated vehicles), and simulations are conducted to analyse the impact of technology on the considered problem; in that work, attention is focused on the transient condition of technological evolution with a gradual introduction of growing levels of automation, assuming that the actual traffic signal system is maintained. In [10], signalised intersections populated only by automated vehicles are taken into consideration: the automated vehicles use connectivity and mechatronics to gather information and autonomously perform driving functions, they handle situations that call for an immediate response, but the driver must still be prepared to intervene when called upon by the vehicle to do so; the class of vehicles considered in that paper is different from autonomous vehicles which sense the environment, navigate and perform driving functions all by the vehicle themselves, and it is also different from connected vehicles which are connected with the surrounding vehicles and roadside infrastructure but still need the driver to control the steering, acceleration, and braking; the authors developed a signal control algorithm that allows vehicle paths and signal control to be jointly optimised, based on advanced communication technology between approaching vehicles and signal controller. The research on signalised intersections populated by CAVs is constantly growing: in [11], it is studied a joint optimisation of CAV and traffic signal timing; in [12], it is developed an innovative intersection operation scheme for CAV which maximises intersection capacity by dynamically optimising green durations and lane assignment.

For what concerns the use of automation for intersection control, Kamal *et al.* [13, 14] propose a coordination scheme of automated vehicles at an intersection, without using any traffic lights, thus overcoming the limitations of a cooperative vehicle intersection control (CVIC) system proposed in [15]; in those works, an intersection coordination unit uses two-way communication to receive basic driving information from the approaching vehicles (e.g. current position, speed and destination) and to send guidance instructions; to do that, a constrained non-linear optimisation problem that includes a risk function is solved for all vehicles in a model predictive control framework, in order to evaluate the optimal trajectories to be followed by vehicles to cross the intersection safely without much drop in their velocities. Other solutions under the framework of Autonomous Intersection Management (AIM) are [16] where a genetic algorithm to find an optimal or a near-optimal vehicle passing sequence for adjacent intersections is defined; [17] that proposes an ant colony system (ACS) to solve the control problem for a large number of vehicles and lanes; [18] in which a control strategy aimed at minimising the maximum exit time is obtained by applying dynamic programming; [19] where the coordination of multiple vehicles approaching an intersection is considered in a control-theoretical framework: a decentralised approach combining optimal control with model-based heuristics is proposed; [20] that presents a more general reservation protocol, named AIM*, in which the intersection manager assigns reservations to vehicles based on the priority assigned to each vehicle; this new protocol makes it possible to optimise reservations in real-time using a conflict point separation model.

Going towards an optimisation approach that is closer to the one proposed in this paper, reference is made to the works [21–25], that are all relevant to scheduling problems. In particular, in [21] it is developed a Linear Programming formulation for Autonomous Intersection Control (LPAIC) accounting for traffic dynamics within a connected vehicle environment. In [22], the problem of coordinating the passage of vehicles through an intersection is studied, with the aim of minimising the total travel time and the energy consumption; an intersection manager communicates with vehicles heading towards the intersection, groups them into clusters, and determines an optimal order of passage and the average speed profiles. In [23], an optimal intersection control scheme is designed under the scenario of an intersection completely modelled by interactions between vehicles; in this case, it is required that a majority of vehicles on the road are equipped with a simple driver assistance system. In [24], a mixed-integer linear programming (MILP) approach for scheduling the vehicle arrivals at an unsignalised intersection is defined with the aim of minimising the delays, in a scenario where all the vehicles are highly automated; the constraints of the MILP are obtained by finding the intervals of feasible arrival times for each vehicle; the obtained schedules are then used as the input for a motion planning problem which is solved analytically by using linear motion equations; in particular, vehicles obey all commands received from an intersection controller while inside the control region without driver interference. In [25], it is addressed the problem of optimally controlling CAV crossing an urban intersection without any explicit traffic signalling so as to minimise energy consumption subject to a throughput maximisation requirement and to hard safety constraints, without making the vehicles stop at the intersection.

Moving on more recent papers, the focus is made on cellular automata models and linear programming formulations of intersections that are populated only by autonomous vehicles. Wu *et al.* [26] define a cellular automata model with a greedy algorithm for the traffic control of intersections in an autonomous vehicle environment, being a platoon of autonomous vehicles the optimisation object; in that work, conflicts between vehicles are solved by giving the right of way to the longest platoon of vehicles that is close to the intersection. Cruz-Piris *et al.* [27] propose an automatic optimisation system based on three models: the first one labels automatically and univocally all the elements that compose an intersection, the second model proposes a process to calculate the shortest paths with minimum conflict points between them in a cellular automata scenario, and the last one defines an algorithm that obtains the patterns or the frequency of the entry of vehicles into the intersection (using the previously calculated paths), to achieve maximum performance. In [28], it is investigated a combination of two ideas (enabled by vehicle-to-vehicle (V2V) communications and leant on a certain level of automation in the driving) for the optimisation of traffic flow and fuel consumption: the virtual traffic lights (that allow vehicles in the proximity of an intersection to create and coordinate the traffic signals by themselves) and the platooning. Bichiou and Rakha [29] propose a real-time optimisation problem with static and dynamic constraints that minimises vehicle trips in an intersection; conflicts are avoided by dividing the intersection into four zones, each of them assigned to at most one vehicle per time. Liu *et al.* [30] propose a cooperative scheduling mechanism for autonomous vehicles passing through an intersection, called TP-AIM, aimed at minimising travel delays in the intersection; such a mechanism is based on three phases: firstly, an intersection management system assigns reasonable priorities for all present vehicles and hence plans their trajectories; secondly, a window searching algorithm is performed to find an entering window, which can produce a collision-free trajectory with minimal delay (collisions are avoided by considering conflict zones); finally, vehicles are left free to arrange their trajectory individually, by applying dynamic programming to compute the speed profile. Lu *et al.* [31] propose a

MIP formulation for the solution of a trajectory-based traffic management problem for the purpose of managing traffic in a road facility reserved exclusively for autonomous vehicles; the basic aim of that model is to find the optimal trajectories for multiple autonomous vehicles resolving conflicts between vehicles. In [32], a distributed algorithm for a graph-based intersection network is introduced, with the aim of controlling traffic at a macroscopic level; V2I and between infrastructures communications are used to exchange the traffic information between a single autonomous vehicle and the network of autonomous intersections; a discrete-time consensus algorithm is proposed to coordinate the traffic density of an intersection with its neighbourhoods and to determine the control policy aimed at maximising the throughput of each intersection as well as stabilising the overall traffic in the network. In [33], it is formulated an optimisation problem with the goal of finding the sequence and times of arrival for autonomous vehicles, in order to minimise the maximum access time assigned to the subscribed vehicles and to avoid collisions. Li and Li [34] address the problem of the optimisation of vehicle trajectories by considering firstly the problem of planning continuous-time trajectories and secondly a discrete-time model with a more general objective function. Finally, Zhang *et al.* [35] propose priority-based scheduling mechanisms for the management of autonomous vehicles crossing an intersection.

From this literature review, it emerges that the majority of the works focus on optimising the traffic flows for a single element of infrastructure, mainly an intersection. Then, the first contribution of this paper is to consider explicitly roads at the same detail level of intersections, and to propose a methodology that, starting from the analysis of a single element (smart road or unsignalised intersection), works at a network level for which coordination between all elements is developed. A second important contribution is that the proposed model jointly considers the planning of the vehicle trajectories and the scheduling of vehicles passages, with a set of original mathematical programming problems whose solutions provide the time-space trajectories of vehicles travelling through the network.

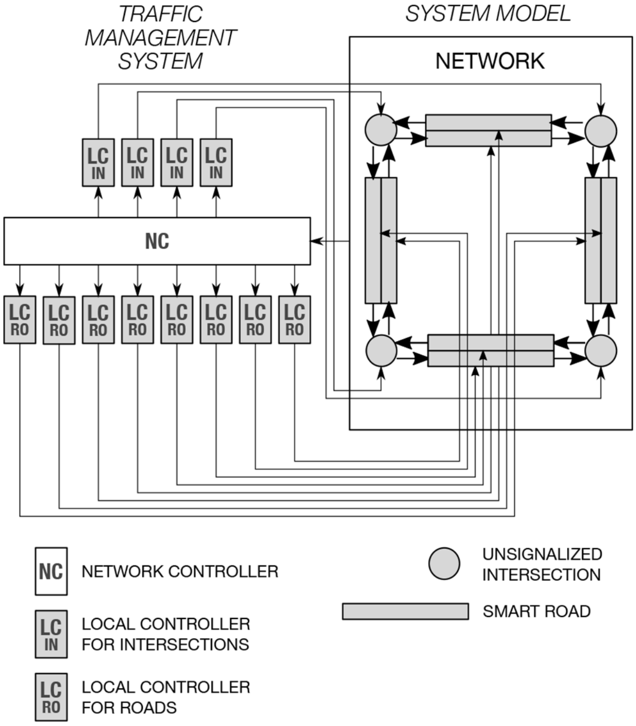

Fig. 1 *Architecture of the TMS*

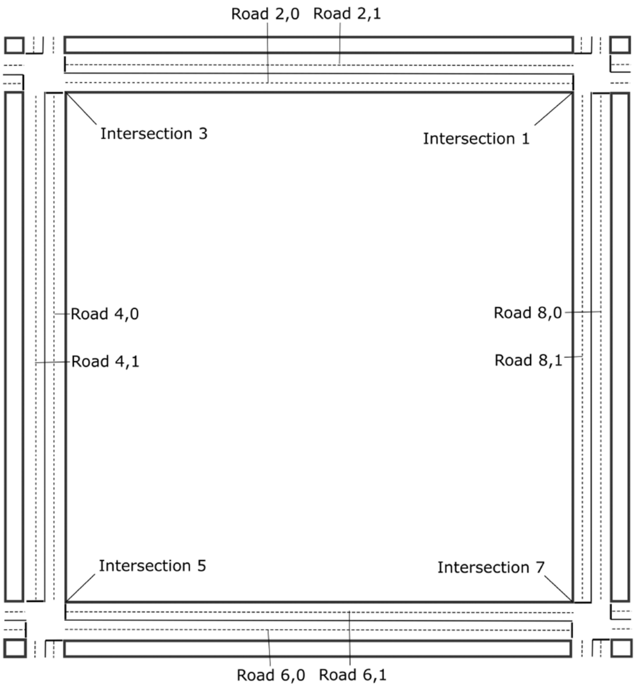

Fig. 2 *Layout of the network used for the numerical experiment*

2 System architecture

The Traffic Management System (TMS) proposed in this paper has the dual purpose of optimising the performance of the network by determining and applying optimal routing strategies and of optimising the trajectories of single self-driving vehicles travelling through the network. In this connection, the TMS acts at two hierarchical levels: at network-level, the NC computes the optimal routes for vehicles entering the network (on the basis of the network state and of the origin and the destination of vehicles) and manages the exchange of information among the various elements (nodes and links) which compose the network; at local-level, the various local controllers (LCs), relevant to both roads (LC-RO) and intersections (LC-IN), determine the motion (time trajectory over predefined space trajectory) of each vehicle inside each element.

A sketch of the TMS architecture for the case of a traffic network consisting of four unsignalised intersections and eight smart roads is reported in Fig. 1. It is obvious that the proposed architecture can be applied to different layouts of network and, also, the proposed models for roads and intersections can be applied to different layouts of roads and intersections. However, for simplicity, two specific networks will be considered in this paper (those reported in Figs. 2 and 3), that are built by joining roads and intersections that have two specific layouts (the one illustrated in Fig. 4 for roads and the one reported in Fig. 5 for intersections).

2.1 Local-level modelling

The TMS builds and solves a specific optimisation problem for each element of the network (smart roads and unsignalised intersections) by adopting a discrete-time approach that allows considering a relatively small number of vehicles in each problem. To be more precise, the optimisation problem which is solved at the generic time instant $\tau_k - \epsilon_i$ takes into consideration all the

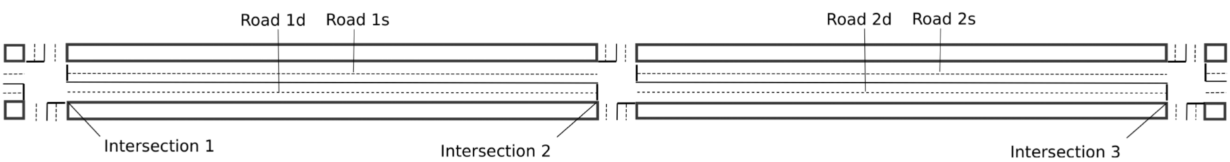

Fig. 3 *Layout of the arterial road considered in the numerical example; the main road is composed of three intersections and four roads*

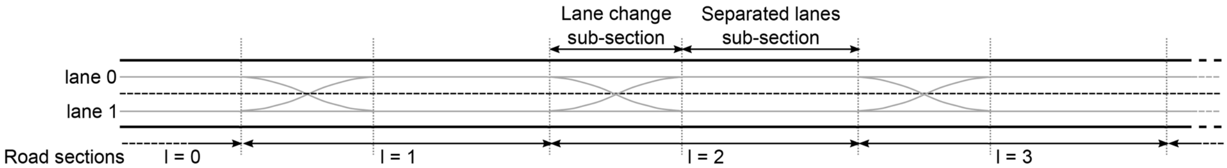

Fig. 4 *Road stretch example consisting of four sections (three-lane change subsections and four separated lanes subsections)*

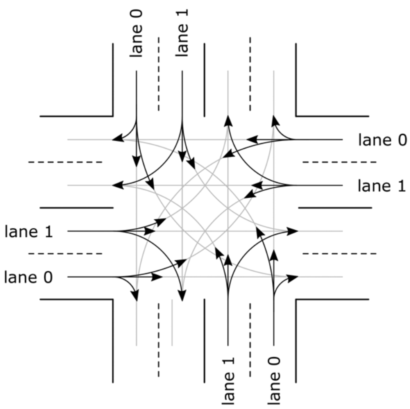

Fig. 5 *General sketch of an intersection with 24 admissible trajectories. The grey dashed lines represent the relevant trajectories*

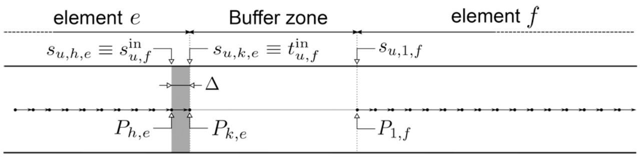

Fig. 6 *Graphical representation of the information that is exchanged between two adjacent elements e and f*

vehicles whose estimated arrival time at road (EATR) or estimated arrival time at intersection (EATI) is in the interval $[\tau_k, \tau_{k+1}) = [\tau_k, \tau_k + \Delta\tau)$, being $\Delta\tau$ the problem time window and ϵ_i the maximum time allowed to the local controller i for solving the optimisation problem and providing the speed profiles to the vehicles.

The logical steps that are accomplished by the system are the following:

i. At $\tau_k - \epsilon_i$, all vehicles that are distant no more than a distance d from the road or the intersection are queried by the NC about their EATR or EATI, origin, destination, shape (i.e. width, length, and barycentre position).
ii. The NC:

(a) defines the set $\mathscr{V}_k$ of active vehicles by gathering all vehicles with EATR or EATI $< \tau_{k+1}$;
(b) determines:

- in the case of a smart road, the entering lane of each vehicle belonging to $\mathscr{V}_k$;
- in the case of unsignalised intersection, the exact stream of each vehicle belonging to $\mathscr{V}_k$ (e.g. with reference to the intersection sketched in Fig. 5, one out of the admissible 24);

(c) communicates to the local controller all the data relevant to the considered set of vehicles.
iii. The local controller builds the optimisation problem, solves it, and communicates the solution in the interval $[\tau_k - \epsilon_i, \tau_k)$.
iv. The vehicles in set $\mathscr{V}_k$ safely travel the road or cross the intersection by following the speed profiles provided by the local controller.

Such an optimisation scheme is applied repetitively interval-by-interval and the parameters ϵ_i and d as well as the interval length $\Delta\tau$ can be suitably set in accordance with computational requirements. Obviously, the speed profiles (transmitted at τ_k) of vehicles which are still inside the intersection in τ_{k+1} represent constraints for the problem instance relevant to the interval $[\tau_{k+1}, \tau_{k+2})$. Finally, to the aim of simplifying the notation, in the following, the problem is formulated for $\tau_k = 0$. Thanks to this assumption, the index k can be dropped.

2.2 Network-level modelling

Consider a traffic network like those reported in Figs. 2 and 3, and let two generic elements of the network (both road and intersection) be specified by indexes e and f.

The main problem is to define in what way a generic element can exchange information with the previous and the next element. In the proposed scheme, a generic element (let it be an element e) is assumed to provide the data only to the element that follows (let it be an element f). The data to provide are the time instants at which the vehicles will cross the last two nodes of e. Then, in the proximity of the position of the last two nodes, it is assumed the presence of a *speed trap*, possibly virtual, whose function is to measure the time instants at which the vehicles cross it and their speed (that can be computed by knowing the length Δ of the two sensors of the trap, that is assumed to be 0.5 m in the case that f is an intersection and 1 m in the case that f is a road).

A graphical representation of the information that is exchanged between two adjacent elements e and f is shown in Fig. 6, where a single lane of the two adjacent elements is reported (note that the notation adopted for variables s and nodes P is a simplification of the formal notation that will be introduced in Section 3.2). The element e, which can be either an intersection or a node, determines (by solving the mathematical programming problem) for each vehicle u the time instants $s_{u,h,e}$ and $s_{u,k,e}$ at which u crosses the last two nodes of the intersection or road; besides, the speed trap of the following element f (which is a road if e is an intersection or an intersection when e is a road) measures for the same points the values $s_{u,f}^{\text{in}}$ and $t_{u,f}^{\text{in}}$ (time instants at which the vehicle u crosses the speed trap of element f), that are expected to be equal to $s_{u,h,e}$ and $s_{u,k,e}$. These values are then transmitted to the local controller of the element f.

In this way, it is possible to compare the values measured by the speed trap of the element f, namely $s_{u,f}^{\text{in}}$ and $t_{u,f}^{\text{in}}$ (real-time instants), with the values provided by the element e, namely $s_{u,h,e}$ and $s_{u,k,e}$ (that are known before the measurement of the two real-time instants, since, once the optimisation problem is solved, all the time instants at which vehicles will cross the various nodes are known). So, the scheme is based on the idea that each element has a speed trap that measure input data and a local controller that solves the optimisation problem, and the speed trap coincides with the last two nodes of the previous elements. The use of the two values measured by the speed traps is in general preferable; however, in case of the absence of a speed trap or due to the need or the desire of solving in advance the problem relevant to downstream elements, it is possible to use the two values computed by the local controller of the element e as the measurements of a 'virtual' speed trap for the downstream element f.

The architecture proposed is based on the following assumptions:

- The length of the road connecting two intersections must be enough to guarantee that the minimum time required to travel the road by vehicles is sufficiently high to avoid that a vehicle crosses the speed trap of the road and enters the following intersection in the same optimisation window (τ_h, τ_{h+1}).

- Each downstream element f is always able to receive all the vehicles coming from e, i.e. it is never saturated.

As a consequence of the first assumption, the minimum length allowed for a road is set to 50 m; however, in the case there are two intersections separated by a road with a length lower than this value, the two intersections and the road stretch between them can be considered as a unique complex element of the network, for which it can be applied the model discussed in Section 5.2. The second assumption is required since the model assumes that a generic element exchanges information related to the vehicle arrivals only to the next element, road or intersection, in the vehicle path. This hypothesis can be partially overcome by the NC, which can redirect vehicles (coming from e and going to f) to alternative routes.

Finally, a particular consideration must be done for the buffer zone, which is the road section between two consecutive elements. In this zone, vehicle trajectories are not controlled and vehicles cannot change the lane, but they are assumed to autonomously adjust their speed in order to arrive at the entrance of the following element at the time instant and with speed required, avoiding excessive accelerations or decelerations. The buffer zone can also be used as a mean for recovering the possible errors related to the difference between the real entering time of the vehicles at an element f ($s^{\text{in}}_{u,f}$ and $t^{\text{in}}_{u,f}$) and the expected time determined on the basis of the solution of the precedent element e ($s_{u,h,e}$ and $s_{u,k,e}$). The error could be due to a difficulty of the vehicles in respecting the speeds that are assigned to them by the local controller of the element. The presence of the buffer zones limits the propagation of the error.

3 Assumptions, notation, and spatial discretisation

3.1 Assumptions

In the model presented, in this paper, it is assumed that

i. The traffic network can be decomposed into a set of basic elements, the 'intersections' and the 'road stretches'.
ii. Road stretches can be divided into 'sections', each of them subdivided into 'lane change subsections' and 'separated lanes subsections', being the first the locations at which vehicles can change lane, and the second the locations where vehicles cannot change lane (see Fig. 4 where a road stretch consisting of 4 sections is depicted); note that, thanks to the presence of the lane change subsections, vehicles can overtake each other's inside road stretches.
iii. Along roads and inside intersections, vehicles are constrained to follow a-priori determined paths, hereafter indicated as trajectories, whose geometrical characteristics are a-priori analytically determined with the aim of guaranteeing vehicles stability; note that two or more trajectories can exit or enter in some nodes, for instance to allow lane changes, and also they cross each other inside lane change subsections and intersections; therefore, the proposed approach has to provide solutions that avoid collisions.
iv. The geometric trajectories are addressed as simple curves and, based on the common road design framework, it is possible to determine the maximum allowed speed in each of their point, as discussed in Section 4.
v. At the beginning of each element of the network, a 'speed trap' determines the actual vehicle speed; each speed trap is followed by a suitable 'buffer zone' where vehicles can decelerate or accelerate to reach the optimal speed determined by the optimisation problem for the first node; note that, given the distance between the points $P_{h,e}$ and $P_{k,e}$ (see Fig. 6 again), the vehicle speed is determined by simply measuring the time instants at which the vehicle crosses them.
vi. Vehicles are considered with their real space occupancy; therefore, it is possible to identify off-line the set of reciprocal incompatible positions, as described in Section 4.1.

3.2 Notation

The notation adopted in this paper is introduced in the following, where the structures, the sets, the parameters, and the variables characterising the model are reported. Besides, thanks to the generality of the model and for the sake of compactness, the element index e is dropped.

3.2.1 Structures and sets:

- $\mathscr{G} = \{\mathscr{N}, \mathscr{L}\}$ is the graph underlying the geometrically pre-defined trajectories Γ (that are followed by vehicles) in an element of the considered network, either road or intersections. Each trajectory is defined by a sequence of nodes of the graph.
- $\mathscr{N} = \{P_h\}$ is the set of $|\mathscr{N}|$ points, hereafter also indicated as nodes, of all the trajectories $\in \Gamma$; P_h is a generic point of the graph and it might belong to more than one trajectory. Points P_h are defined so that the length of all links connecting two adjacent nodes is equal to a suitably chosen parameter δ characterising the spatial discretisation of trajectories in Γ; nodes are not sorted and, in general P_k might follow P_{k+h}, with $h > 0$, in a trajectory.
- $\mathscr{L} = \{(P_h, P_k) | P_h, P_k \in \mathscr{N}\}$ is the set of $|\mathscr{L}|$ directed links connecting pairs of adjacent nodes P_h and P_k.
- For any generic road stretch, $\mathscr{Z}$ is the set of $|\mathscr{Z}|$ road sections, i.e. the set of distinct *lane change* and *separated lanes* subsections.
- $\mathscr{V}$ is the set of $|\mathscr{V}|$ vehicles.
- P^{out}_0 is the exit node of the road in the right lane (lane 0).
- P^{out}_1 is the exit node of the road in the left lane (lane 1).
- For any road stretch, $\mathscr{LN}_l$ (respectively, $\mathscr{CH}_l$) is the set of nodes in the separated lane (resp., lane change) subsection of the section l of the road; in this connection, considering a link $(P_h, P_k) \in \mathscr{L}$, note that

    ○ if both P_h and $P_k \in \mathscr{LN}_l$, then (P_h, P_k) is a link connecting nodes of the separated lanes subsection of section l; besides, P_h and P_k are on the same lane;
    ○ if both P_h and $P_k \in \mathscr{CH}_l$, then (P_h, P_k) is a link connecting nodes of the change lane subsection of section l;
    ○ if $P_h \in \mathscr{LN}_l$ and $P_k \in \mathscr{CH}_{l+1}$, then (P_h, P_k) is a link connecting the last node P_h of the separated lanes subsection of section l and the first node P_k of the change lane subsection of section $l + 1$;
    ○ if $P_h \in \mathscr{CH}_l$ and $P_k \in \mathscr{LN}_l$, then (P_h, P_k) is a link connecting the last node P_h of the change lane subsection of section l and the first node P_k of the separated lanes subsection of section l.

- For any separated lane subsection, $\mathscr{NL}^y_l \subset \mathscr{LN}_l$, $y \in \{0, 1\}$, is the set of nodes in the lane y of section l.
- For any lane change subsection, $\mathscr{NC}^{y_1, y_2, y_3, y_4}_l \subset \mathscr{CH}_l$ is the set of nodes which belong to the trajectory which goes from lane y_1 to lane y_2 or to the trajectory which goes from lane y_3 to lane y_4 (further details are provided in Section 4.1).

3.2.2 Parameters:

- β is a safety parameter that prevents any vehicle to be too near to the others in proximity of a trajectory conflict.
- z^{in}_u is a binary parameter specifying the lane in which vehicle u enters the road (right lane when $z^{\text{in}}_u = 0$ and left lane when $z^{\text{in}}_u = 1$).
- w_u is a real-valued weight parameter representing the priority of vehicle u in the optimisation problem; priorities are assigned to

vehicles by the NC on the basis of the vehicle class (emergency vehicles have always higher priority) and based on the routing strategy determined at network-level.

- $v_{\text{road}}^{\max}$ (respectively, $v_{\text{int}}^{\max}$) is the maximum speed of vehicles in the road stretches (resp., intersections); note that, in some cases, special vehicles (i.e. emergency vehicles) are allowed to travel at speeds greater than $v_{\text{road}}^{\max}$.
- $v^{\max}$ is the maximum speed of vehicles in the buffer zones.
- $\Phi_{\min} = \delta / v_{\text{road}}^{\max}$ is the minimum travel time on a link (determined by the maximum admissible speed).
- Φ_{low} and Φ_{high} are the maximum allowed relative vehicle travel time variations between two consecutive links when the speed of the vehicle is low and high, respectively; such values are calibrated by considering the maximum admissible acceleration/deceleration in comfortable driving conditions (i.e. not emergency brake); in addition, it is $\Phi_{\text{low}} > \Phi_{\text{high}}$ to allow higher accelerations changes when the speed is low.
- $r_h \leq 1$ is a coefficient ('speed reduction factor') that is used to reduce the maximum allowed speed on the links with origin in P_h to a value that guarantees the vehicle stability on curved trajectories; values r_h depend only on the trajectories and can be determined off-line once (as described in Section 4); note that, in the presence of more than one link starting from P_h, r_h is set in a conservative way, i.e. it is set to the lowest possible value; in any case, this is not too restrictive since all the links with the same origin have approximately the same curvature radius, and hence approximately the same speed limitation.

3.2.3 Variables:

- $s_{u,h}$ is the time instant at which the vehicle u leaves the node P_h and $s_{u,h}^{\star}$ indicates the relevant optimal value determined by solving the optimisation problem described in Section 5.1; note that, $s_{u,h}$ assumes the same value for all the links with an origin P_h, if more than one; this is not a limitation since the values associated to the links not used by the vehicle u do not affect the travel time of u and then do not influence the cost function; on the contrary, it simplifies the problem formulation, as it reduces the number of variables to consider.
- $t_{u,h}$ is the time instant at which the vehicle u arrives at the end of the links with origin in P_h and $t_{u,h}^{\star}$ indicates the relevant optimal value determined by solving the optimisation problem described in Section 5.1; in analogy to $s_{u,h}$, also the variable $t_{u,h}$ assumes the same value for all the links with an origin at the node P_h, if more than one.
- s_u^{in} and t_u^{in} are the time instants at which the vehicle u crosses the speed trap detectors.
- t_u^{out} is the time at which the vehicle u exits the road.
- $x_{u,v,l}$ is a Boolean variable that is set to 1 if the vehicle u has right of way with respect to vehicle v in the road section l; as before, let $x_{u,v,l}^{\star}$ be its optimal value.
- $z_{u,l}$ is a Boolean variable that is set to 0 (respectively, 1) if the vehicle u in the road section l is on the right lane (resp., left lane) and $z_{u,l}^{\star}$ indicates its optimal value; it is worth noting that the value of these variables is the only information which is needed for determining the path of each vehicle, since the sub-path of a vehicle in a lane change subsection is in one-to-one correspondence with the values assumed by z in the previous and subsequent separated lanes subsections.

3.3 Problem spatial discretisation

In this paper, the dynamics of vehicles travelling along the trajectories in road and intersections are formulated as a set of constraints of a MILP problem. To this aim, the geometrical trajectories are spatially discretised in order to define an equivalent graph $\mathscr{G}$ whose links have an a-priori defined constant length δ.

Thanks to this discretisation, the speed $v_{u,h}$ of the generic vehicle u along with the link (or the links) with origin in P_h can be expressed in terms of time instants at which the vehicle departs from the node P_h and at which the vehicle arrives at the destination node, i.e.

$$t_{u,h} - s_{u,h} = \frac{\delta}{v_{u,h}} \tag{1}$$

While such an equivalence allows to write the whole MILP in terms of the time variables $t_{u,h}$ and $s_{u,h}$, it turns out that the optimal speed $v_{u,h}^{\star}$ is a piecewise constant function that has to be interpolated prior to be transmitted to the vehicle u. In this regards, note that if a piecewise cubic Hermite interpolating polynomial [36] is considered, a good behaviour (continuity and smoothness) of the function expressing the positions and the accelerations of u is guaranteed. Nevertheless, the continuous approximation might not satisfy all the MILP constraints, with particular reference to the safety ones: to tackle with this problem, some test may be performed off-line to verify if a different trajectory discretisation (for instance by considering $\delta' < \delta$) has to be considered. In the experiments described in Section 6, choosing $\delta = 1\,\text{m}$ for the discretisation of road trajectories and $\delta = 0.5\,\text{m}$ for the discretisation of intersection trajectories resulted to be always sufficient. To conclude, it is worth observing that, since the definition of $t_{u,h}$ and $s_{u,h}$, the speed resulted to be defined for all the links with origin in P_h, although only the one associated to the link actually used by vehicle u is used to build the continuous approximation of the optimal solution.

4 Curve trajectories and safety constraints

In this section, the maximum admissible speed of vehicles in any link is discussed and provided. To do so, let start observing that the speed of vehicles travelling along a road is limited to specific values depending on the road characteristics (i.e. geometry, pavement, etc.) and on the surrounding environment (i.e. weather condition). For instance, the maximum admissible speed on urban roads can be limited to $v^{\max} = 35\,\text{km/h}$ but it must be further reduced along curves to guarantee the stability of the vehicle itself. This reduction is a function of the friction coefficient and of the curvature. With the aim of formalising the concept of speed reduction within the MILP problems proposed in the following section, the speed limit on the road link with origin in P_h can be defined as $r_h \cdot v_{\text{road}}^{\max}$, being

$$r_h = \min \left\{ 1, \min_{k \mid (P_h, P_k) \in \mathscr{L}} \frac{\sqrt{\mu g \rho(P_h, P_k)}}{v_{\text{road}}^{\max}} \right\} \tag{2}$$

a speed reduction factor determined on the basis of the well-known road design methodologies. In (2), μ is the friction coefficient, g is the gravitational acceleration, and $\rho(P_h, P_k)$ is the curvature radius in P_h of the trajectory between the nodes P_h and P_k. An analogous model is applied to the links which belong to an intersection; in that case, $v_{\text{int}}^{\max}$ is used in place of $v_{\text{road}}^{\max}$. Finally, note that the definition in (2) is conservative, as it considers the minimum radius in the case of multiple links starting from the same node P_h, and that all the links of straight trajectories have $r_h = 1$.

4.1 Incompatibility of trajectories

As mentioned, to get a MILP formalisation of the problem addressed in this paper and in order to avoid vehicles conflicts, it is of worth importance to formalise the concept of 'space occupancy' for each vehicle in any point P_h of the graph. This can be done by determining the unit tangent vector $\hat{t}(P_h)$ in such a point and drawing a rectangular shape assuming that the barycentre of the vehicle is in P_h and that the longer side of the rectangle is parallel to the tangent. In Fig. 7, it is depicted as an example of vehicle occupancy in a generic curve trajectory, whereas in Fig. 8, the case

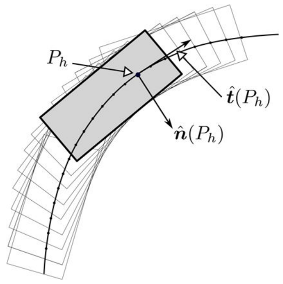

Fig. 7 *Generic vehicle travelling along a curved trajectory. The two dashed arrows represent the tangent vector $\hat{\boldsymbol{t}}(P_h)$ and normal vector $\hat{\boldsymbol{n}}(P_h)$ in the point $P_h \in \mathcal{N}$ corresponding to the barycentre of the vehicle*

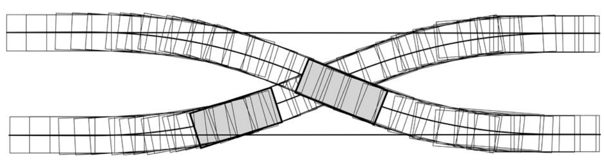

Fig. 8 *Example of vehicles occupancies in a lane change subsection*

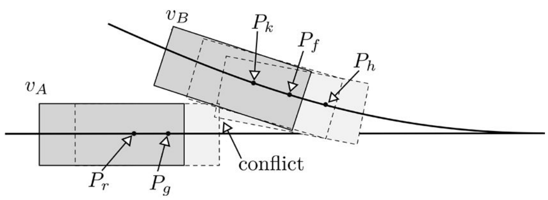

Fig. 9 *Example of vehicles occupancies in a lane change subsection: P_r and P_g are not compatible with P_k, P_f, and P_h*

of two vehicles travelling within a lane change subsection is reported.

By exploiting this concept, it is easy to identify all the node pairs (P_h, P_k), $\forall h, k \in \mathcal{N}$, that cannot be reached by two vehicles at the same time, since the relevant occupancies overlap.

An example of these occupancies is depicted in Fig. 9 where the darker-grey rectangles indicate the current position of two vehicles and the lighter-grey ones indicate successive occupancies of the vehicles due to their movements, having assumed the previously mentioned discretisation. The vehicle v_A cannot leave the node P_r since there are positions along the link (P_r, P_g) that are not compatible with the vehicle v_B when it is on the nodes P_k, P_f, and P_h. This means that the node P_r is not compatible with nodes P_k, P_f, and P_h (they are conflict nodes). It is worth noting that, since vehicles are not allowed to stop in nodes, v_A cannot even reach P_r and has to be slowed down, and delayed, in the previous links of its path. Besides, it is evident that also P_g is not compatible with P_k, P_f, and P_h.

Therefore, a key element of the proposed approach consists in the definition and identification of the incompatibility sets, which can be related to the separated lanes subsections and to the lane change subsections. For what concerns the incompatibilities in the separated lanes subsections, only conflicts between vehicles on the same lane (either 0 or 1) are present. Instead, for what concerns the incompatibilities related to the lane change subsections, there are conflicts both between vehicles travelling on the same trajectory and between vehicles travelling on different trajectories (at least partially).

The following incompatibility sets are defined for the separated lanes subsections and the lane change subsection, respectively,

$$\begin{aligned}\mathscr{I}_{y,l}^{\text{lane}} = \{(P_h, P_k) \big| P_h, P_k \in \mathcal{NL}_l^y,\\ P_h \text{ is not compatible with } P_k\}, \quad \forall l \in \mathscr{Z}, \forall y \in \{0,1\}\end{aligned} \tag{3}$$

$$\begin{aligned}\mathscr{I}_{y_1,y_2,y_3,y_4,l}^{\text{change}} = \{(P_h, P_k) \big| P_h, P_k \in \mathcal{NC}_l^{y_1,y_2,y_3,y_4},\\ P_h \text{ is not compatible with } P_k\},\\ \forall l \in \mathscr{Z}, \forall y_1, \forall y_2, \forall y_3, \forall y_4 \in \{0,1\},\\ [y_1, y_2, y_3, y_4] \neq \{[0,0,1,1],[1,1,0,0]\}\end{aligned} \tag{4}$$

These sets define the pairs of incompatible nodes, both relative to the same trajectory and to different trajectories. Equations (3) and (4) are $16|\mathscr{Z}| + 2$ incompatibility sets. In (4), subscripts y_1, y_2, y_3, y_4 indicate two specific trajectories: the first one is specified by y_1 and y_2 and it is the trajectory which starts in the first node of lane y_1 in the lane change subsection l and ends in the last node of lane y_2 always in the lane change subsection l; similarly, the second one is specified by y_3 and y_4 and it is the trajectory which goes from the first node of lane y_3 to the last node of lane y_4. It is worth noting that no incompatibilities have to be defined when $[y_1, y_2, y_3, y_4] \neq [0,0,1,1]$ or when $[y_1, y_2, y_3, y_4] \neq [1,1,0,0]$, since in this case the two trajectories are distinct and parallel and therefore do not have conflict zones.

As regards intersections, the trajectories are similar to those in the road stretches but vehicles cannot change lane. Therefore, there is no distinction between separated lanes subsections and change lane subsections.

Then, it is possible to define the set

$$\mathscr{I}^{\text{int}} = \{(P_h, P_k) \big| P_h \text{ is not compatible with } P_k\} \tag{5}$$

that gathers all the pairs of incompatible nodes. Note that the definition of $\mathscr{I}^{\text{int}}$ is more general than the definition of $\mathscr{I}_{y,l}^{\text{lane}}$ since the nodes P_h and P_k can belong to the same or to different trajectories.

5 Road and intersection models

As previously stated, the optimal trajectories of self-driving cars travelling the traffic network are determined by solving a set of mathematical programming problems; in the following, the MILP model for smart roads is described in detail, and the specific features characterising the MILP model for unsignalised intersections are also introduced.

5.1 Road MILP formulation

The optimisation problem can be formalised as follows

$$[\boldsymbol{s}^\star, \boldsymbol{t}^\star, \boldsymbol{x}^\star, \boldsymbol{z}^\star] = \arg \min_{s,t,x,z} \Theta \tag{6}$$

being

$$\Theta = \sum_{u \in \mathscr{V}} w_u \left(t_u^{\text{out}} - s_u^{\text{in}}\right) \tag{7}$$

the total weighted travel time, subject to the following constraints:

$$t_u^{\text{out}} \geq t_{u,k_0^{\text{out}}} - M z_{u,|\mathscr{Z}|}, \quad \forall u \in \mathscr{V} \tag{8}$$

$$t_u^{\text{out}} \geq t_{u,k_1^{\text{out}}} - M(1 - z_{u,|\mathscr{Z}|}), \quad \forall u \in \mathscr{V} \tag{9}$$

Constraints (8) and (9) define t_u^{out} taking into account the lane from which vehicle u exits the road: if the right (resp., left) lane is used, then $z_{u,L} = 0$ (resp., $z_{u,L} = 1$) and the two constraints correspond to $t_u^{\text{out}} \geq t_{u,k_0^{\text{out}}}$ (resp. $t_u^{\text{out}} \geq t_{u,k_1^{\text{out}}}$) being k_0^{out} (resp., k_1^{out}) the index of the last node in the right lane (resp., left) of the last separated lanes subsection. In (8) and (9), M is a positive large coefficient whose value is carefully chosen (not too large but in any case larger than any reasonable value that continuous variables $s_{u,h}$ and $t_{u,h}$ may take).

$$s_{u,k} = t_{u,h}, \quad \forall u \in \mathscr{V}, \forall (P_h, P_k) \in \mathscr{L} \tag{10}$$

Constraint (10) guarantees the time continuity of the trajectories.

$$\begin{cases} t_{u,k} - s_{u,k} \geq t_{u,h} - s_{u,h} - \Phi_{\text{low}} \\ t_{u,k} - s_{u,k} \leq t_{u,h} - s_{u,h} + \Phi_{\text{low}} \end{cases} \tag{11}$$
$$\forall u \in \mathscr{V}, \forall (P_h, P_k) \in \mathscr{L}$$

$$\begin{cases} t_{u,k} - s_{u,k} \geq (t_{u,h} - s_{u,h}) \cdot \left(1 - \dfrac{\Phi_{\text{high}}}{\Phi_{\min}}\right) \\ t_{u,k} - s_{u,k} \leq (t_{u,h} - s_{u,h}) \cdot \left(1 + \dfrac{\Phi_{\text{high}}}{\Phi_{\min}}\right) \end{cases} \tag{12}$$
$$\forall u \in \mathscr{V}, \forall (P_h, P_k) \in \mathscr{L}$$

Constraints (11) and (12) are 'comfort' constraints which guarantee that vehicles do not vary their speed too rapidly; in particular, constraints (11) guarantee that, when the travel time on the link (P_h, P_k) is long (low speed), it can vary less than the value Φ_{low}; on the contrary, constraints (12) guarantee that, if the travel time on the link (P_h, P_k) is short (high speed), it can only vary of the value $\Phi_{\text{high}} < \Phi_{\text{low}}$.

$$t_{u,h} - s_{u,h} \geq \frac{\delta}{r_h\, v_{\text{road}}^{\max}} \quad \forall u \in \mathscr{V}, \forall h \in \mathscr{N} \tag{13}$$

$$s_{u,h} \geq t_u^{\text{in}} + \frac{\Delta}{v^{\max}} \quad \forall u \in \mathscr{V}, \forall h \in \mathscr{N} \tag{14}$$

Constraints (13) and (14) guarantee that vehicles respect the maximum speed limit, properly reduced in curves, both in the various links of the road and in the road segment between the speed trap and the entry in the road.

$$z_{u,0} = z_u^{\text{in}} \quad \forall u \in \mathscr{V} \tag{15}$$

$$t_u^{\text{in}} - t_v^{\text{in}} \leq M(1 - x_{u,v,0}) \quad \forall u, v \in \mathscr{V} \tag{16}$$

$$t_v^{\text{in}} - t_u^{\text{in}} \leq M x_{u,v,0} \quad \forall u, v \in \mathscr{V} \tag{17}$$

For section 0 (that consists of the first separated lanes subsection only), constraint (15) initialises variable $z_{u,0}$ by setting it to the value that specifies the given lane of arrivals, and constraints (16) and (17) initialise a variable $x_{u,v,0}$ by setting it equal to 1 if $t_u^{\text{in}} \leq t_v^{\text{in}}$ and 0 otherwise (in other words, precedence is given to the vehicle that arrives first).

$$s_{u,h}\mathbf{1}_2 \geq [t_{v,k} + \beta - M \cdot x_{u,v,l}]\mathbf{1}_2 - M[\boldsymbol{A}\boldsymbol{z}' + \boldsymbol{b}]$$
$$\boldsymbol{A} = \begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \quad \boldsymbol{z}' = \begin{pmatrix} z_{u,l} \\ z_{v,l} \end{pmatrix} \tag{18}$$
$$\forall u, v \in \mathscr{V}, u \neq v, \forall (P_h, P_k) \in \mathscr{I}_{y,l}^{\text{lane}}, \forall l \in \mathscr{Z}$$

Constraint (18) is a compact representation of the 'incompatibility constraints' for a generic separated lanes subsection, being $\mathbf{1}_n$ a column vector with dimension n whose entries are all equal to 1. As a matter of fact, (18) corresponds to two constraints which define the minimum distance between two vehicles on the same lane; in particular, they state that if two vehicles u and v are in the same lane ($z_{u,l} = z_{v,l}$) and v precedes u ($x_{u,v,l} = 0$), then u cannot enter the link (P_h, P_m) until v exits the link (P_k, P_n), being $(P_h, P_k) \in \mathscr{I}_{0,l}^{\text{lane}}$ or $(P_h, P_k) \in \mathscr{I}_{1,l}^{\text{lane}}$, and $(P_h, P_m), (P_k, P_n) \in \mathscr{L}$; it is worth noting that the constraints in (18) are always satisfied (hence, they are not significant) if $z_{u,l} \neq z_{v,l}$, i.e. when the two vehicles are not in the same lane.

$$\begin{cases} x_{u,v,l} \geq x_{u,v,l-1} - Mz_{u,l-1} - Mz_{v,l-1} \\ x_{u,v,l} \leq x_{u,v,l-1} + Mz_{u,l-1} + Mz_{v,l-1} \\ x_{u,v,l} \geq x_{u,v,l-1} - M(1 - z_{u,l-1}) - M(1 - z_{v,l-1}) \\ x_{u,v,l} \leq x_{u,v,l-1} + M(1 - z_{u,l-1}) + M(1 - z_{v,l-1}) \end{cases} \tag{19}$$
$$\forall u, v \in \mathscr{V}, u \neq v, \forall l \in \mathscr{Z}$$

For a generic section $l > 1$ of the road (i.e. not the first one), constraints (19) guarantee that the right of way of two vehicles does not change if the two vehicles were in the same lane in the previous section $l - 1$; this is because the two vehicles must approach the lane change subsection of section l in the same order they leave section $l - 1$.

$$s_{u,h}\mathbf{1}_{14} \geq [t_{v,k} + \beta - Mx_{u,v,l}]\mathbf{1}_{14} - M[\mathbf{C}\mathbf{z}'' + \mathbf{d}]$$
$$\mathbf{C} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & -1 \\ 1 & 1 & -1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & -1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ -1 & 1 & -1 & -1 \\ -1 & -1 & 1 & -1 \\ -1 & -1 & -1 & 1 \\ -1 & -1 & -1 & -1 \end{pmatrix}, \quad \mathbf{d} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 2 \\ 2 \\ 3 \\ 1 \\ 2 \\ 2 \\ 3 \\ 3 \\ 3 \\ 4 \end{pmatrix}, \quad \mathbf{z}'' = \begin{pmatrix} z_{u,l-1} \\ z_{v,l-1} \\ z_{u,l} \\ z_{v,l} \end{pmatrix} \tag{20}$$
$$\forall u, v \in \mathscr{V}, u \neq v, \forall (P_h, P_k) \in \mathscr{I}_{y_1,y_2,y_3,y_4,l}^{\text{change}}, \forall l \in \mathscr{Z}$$

Constraint (20) is again a compact representation of the 'incompatibility constraints' for a generic lane change subsection: as a matter of fact, (20) corresponds to 14 constraints each of them defining the incompatibilities between two vehicles in two specific trajectories of the lane change subsection (there are 4 trajectories in each lane change subsection and then 16 possible pair of trajectories, but 2 of them are not in conflict); for example the fourth constraint (obtained by the fourth row of matrix $\boldsymbol{C}$ and vector $\boldsymbol{b}$) is relevant to the case of two vehicles u and v that arrive from different lanes and join the right lane (as the case illustrated in Fig. 9); in all the 14 cases of conflicting trajectories, the incompatibility is defined as follows: if two vehicles u and v are in conflicting trajectories and v precedes vehicle u, then u cannot enters link (P_h, P_m) until v exits link (P_k, P_n); finally note that, each case of conflicting trajectories can be expressed as a specific combination of the values of variables $z_{u,l-1}$, $z_{u,l}$, $z_{v,l-1}$, and $z_{v,l}$.

$$x_{v,u,l} = 1 - x_{u,v,l} \quad \forall u, v \in \mathscr{V}, \forall l \in \mathscr{Z} \tag{21}$$

Constraint (21) defines the relationship between $x_{v,u,l}$ and $x_{u,v,l}$.

$$x_{u,v,l} \in \{0, 1\} \quad \forall u, v \in \mathscr{V}, u \neq j, \forall l \in \mathscr{Z} \tag{22}$$

$$z_{u,l} \in \{0, 1\} \quad \forall u \in \mathscr{V}, \forall l \in \mathscr{Z} \tag{23}$$

$$s_{u,h} \in \mathbb{R}_{\geq 0}, t_{u,h} \in \mathbb{R}_{\geq 0}, \quad \forall u \in \mathscr{V}, \forall h \in \mathscr{N} \tag{24}$$

Lastly, constraints (22)–(24) define the Boolean and non-negative real variables of the problem.

5.2 Intersection MILP model

The model of intersections, that is, junctions between two or more road stretches, is briefly described in this section. Since the intersection model is similar to the road one, only the differences

Table 1 Model parameters

Parameter	δ_{road}	δ_{int}	$v_{\text{int}}^{\max}$	$v_{\text{road}}^{\max}$	Δ	β
value	1 m	0.5 m	14 ms^{-2}	10 ms^{-2}	50 m	0 s

Table 2 Arrival rates

Entering lane	Rate, s·h^{-1}
right lane	2900
left lane	2150

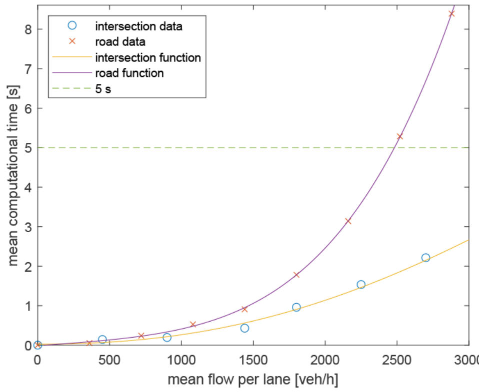

Fig. 10 *Mean computational times (blue and red symbols) and computational complexity trends (orange and purple curves) as a function of the arrival rates, with* $\Delta\tau = 5$s

will be discussed; in doing so, reference is made to a single isolated and unsignalised intersection, as the one depicted in Fig. 5.

As already mentioned, the main difference between an intersection and a road stretch consists of the reduced degrees of freedom in the vehicles trajectories, which are considered to be fixed. In other words, inside intersections, there are not lane change subsections and vehicles are constrained to travel along fixed trajectories (that can be defined in advance); in the example of Fig. 5, there are 24 fixed trajectories and each vehicle is assigned to one of them on the basis of its origin and destination. Such a simplified behaviour, which is reasonable for the passengers' comfort, reduces the complexity of the problem described in Section 5.1 which can be employed also for intersections after having assumed that

i. The arriving lane and the exiting lane of each vehicle approaching an intersection are known and fixed: the values of the variables z are fixed (at a network level, they are determined by the road section problems).
ii. Constraints (18) are applied to all the vehicles on node pairs belonging to the set $\mathcal{F}^{\text{int}}$, with the above mentioned fixed values of the variables z.
iii. Constraints (19), (20), and (23) are not considered.

6 Computational experiments

In this section, the considered model is applied to roads, intersections, and networks with the aims of showing and discussing the optimal solution provided by the solver. In particular

- Section 6.1 includes some considerations about the computational effort required by the model.
- Two specific scenarios (vehicles travelling on conflicting trajectories and overtaking manoeuvres) are reported in Sections 6.2 and 6.3.
- Section 6.4 describes the application of the model to the traffic networks illustrated in Figs. 2 and 3.

The parameters of the model are those reported in Table 1, where it is possible to note that both the maximum speed and the spatial discretisation of trajectories are different for road and intersection

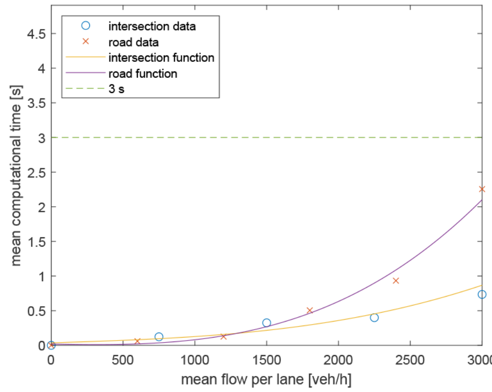

Fig. 11 *Mean computational times (blue and red symbols) and computational complexity trends (orange and purple curves) as a function of the arrival rates, with* $\Delta\tau = 3$s

elements. The MILP model was coded and solved with IBM$^{\text{TM}}$Cplex$^{\text{TM}}$12 Optimiser and the experiments were performed on a standard Intel$^{\text{TM}}$Xeon$^{\text{TM}}$CPU @3.4 GHz processor and 8 Gb RAM laptop.

6.1 Performance of the model

The performance of the model in terms of computational time has been assessed by solving some basic scenarios in which both the road stretch illustrated in Fig. 4 and the intersection sketched in Fig. 5 are taken into consideration. As a matter of fact, the optimisation problem formulated in Section 5.1 is computationally hard to solve. The computational complexity depends on the time windows $\Delta\tau$ and on the traffic flows, since the more vehicles are travelling in the network, the more variables and constraints are considered in the MILP problem defined by (6)–(24).

Experimental results, obtained by randomly generating the arrival times of vehicles in accordance with a Poissonian processes characterised by the rates reported in Table 2, showed that the solution time ξ is in general less than the chosen time horizon $\Delta\tau = 5$ s, even with high vehicle flows (almost 3000 vehicles/h).

A more detailed analysis has been carried out by considering a variable arrival rate, in the range (0, 3000) veh/h/lane; the results of such analysis are reported in Figs. 10 and 11, where the two curves represent the computational complexity trend obtained from the data relative to the mean computational time required to obtain the optimal solution for the road of Fig. 4 and the intersection of Fig. 5, respectively. Fig. 10 is for the case $\Delta\tau = 5$ s; it can be observed that in this case, for high flow values, it is not possible to get the optimal solution for the road within the adopted interval $\Delta\tau$ (the model of the road is actually computationally harder since the presence of Boolean variable $z_{u,l}$); on the contrary, in the case $\Delta\tau = 3$ s the mean computational time to provide the optimal solution is always less than $\Delta\tau$. Since a wider $\Delta\tau$ is preferable, in the case of roads, it can be chosen $\Delta\tau = 5$ s for low-medium arrival rates and $\Delta\tau = 3$ s for high arrival rates.

6.2 Vehicles conflict management

In this section, the conflicts-avoidance system of the model is discussed.

With reference to the trajectories illustrated in Fig. 12, which are initially separated and then merge, consider the two vehicles (indicated as vehicles 3 and 5) which travel on different trajectories; it is evident that there is a conflict situation that must be solved by the MILP model. Two consecutive positions of the two vehicles are represented: blue shapes are relative to vehicle 3, whereas red ones are relative to vehicle 5. The solution proposed

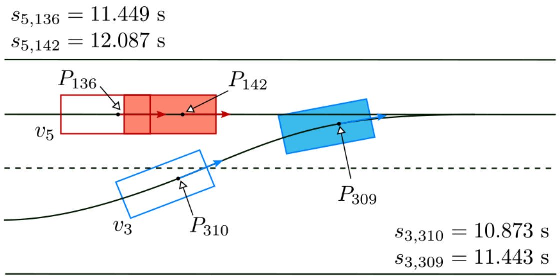

Fig. 12 *Example of timings in the road*

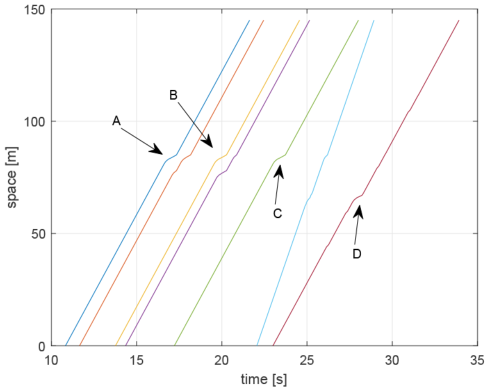

Fig. 13 *Optimal time-space trajectories of a set of vehicles in a generic path inside a road. Each letter points out the vehicle deceleration*

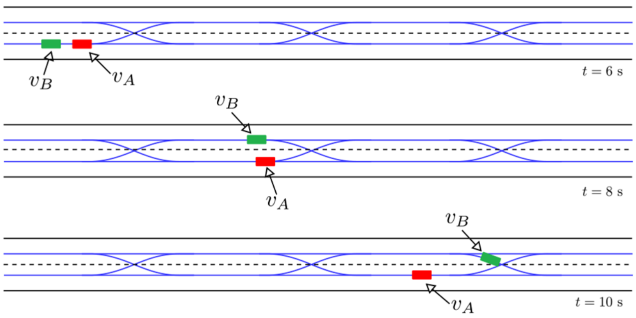

Fig. 14 *Snapshot of the road at $t = 6$ s, $t = 8$ s, and $t = 10$ s. The leader is red and the follower is green*

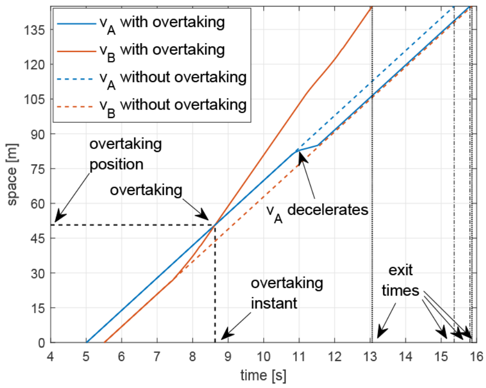

Fig. 15 *Optimal space-time position of the two vehicles in the road (solid curves in case of overtaking allowed and dashed curved with no overtaking)*

by the model gives right of way to vehicle 3 which can travel the conflict zone before vehicle 5: this can be seen by considering the time instants $s_{3,309}$, $s_{3,310}$, $s_{5,136}$, and $s_{5,142}$ provided by the solution and reported in Fig. 12; besides, vehicle 5 brakes in order to avoid to collide with vehicle 3 which, in the end, pass ahead vehicle 5.

A second example is reported in Fig. 13, where the optimal time-space trajectories of a set of vehicles crossing a road are depicted. In such a figure, it possible to note that some vehicles have to decelerate to give way to others that travel along different trajectories (the first vehicle decelerates at point A, the third at point B, the fifth at point C, and the seventh at point D) as put into evidence by the arrows. It is interesting to note that the second and the fourth vehicles decelerate to maintain a sufficient distance from, respectively, the first and the third vehicles (see again points A and B in the figure).

6.3 Overtaking

An example of overtaking manoeuvre that can take place in the road sections is described in this section. In particular, it is shown the effect of allowing a higher maximum speed and a higher priority to follower vehicle v_B approaching its leader v_A. It is here assumed $w_B = 2w_A$, while the maximum speeds allowed are 14 and 20 ms^{-1}, respectively for v_A and v_B. A real-world example of this scenario consists of an emergency vehicle that enters after a normal vehicle.

In Fig. 14, it is illustrated a snapshot of the road at $t = 6$ s, $t = 8$ s, and $t = 10$ s, in which it is possible to observe the overtaking of v_B on v_A. In addition, Fig. 15 reports the optimal space-time trajectories of the two vehicles travelling along the road (solid curves). In such a figure, it is possible to see that v_B overtakes the leader v_A at $t = 8.64$ s, ~50 m from the beginning of the road stretch. In Fig. 15, it is also possible to see that v_B, at first, adapts its speed to the red vehicle (the two time-space trajectories are parallel), then it accelerates while changing the lane to overtake. Around $t = 11$ s, vehicle v_A reduces its speed in order to give way to v_B, which has a higher priority, that must return to the initial lane.

The importance of allowing overtaking stands in the possibility of giving right of way to vehicles with higher priority (in the considered model, priorities take a very important role since the travel times of vehicles in the objective function are weighted by them). In Fig. 15, it is also reported the case in which no overtaking is allowed (dashed curves). By comparing the two scenarios, it is evident that with no overtaking, even if there is a short reduction of the travel time of v_A (0.5 s), the high-priority vehicle v_B is strongly penalised (2.9 s delay) and the value of the objective function significantly increases.

6.4 Network global performance

The global performance of the network in terms of speed and travel times of vehicles travelling the network has been analysed. The layouts considered in this numerical experiment are those depicted in Figs. 2 and 3.

In Fig. 16, the space-time trajectory of a vehicle which enters the network of Fig. 2 at the intersection 3, travels along the road lane (2,0), the intersection 1, the road lane (8,1), and exits the network at the intersection 7 is shown. In such a figure, it is possible to note the speed variations due to the reduction factors r_h in all the intersections, and those due to the presence of other vehicles in the intersection 1 (where more variations are necessary).

A second example is reported in Fig. 17, where it is possible to see the time-space representation of the optimal trajectories of three vehicles which enter the network of Fig. 3 at intersection 1 and exit at intersection 3. In this case, it is possible to note that

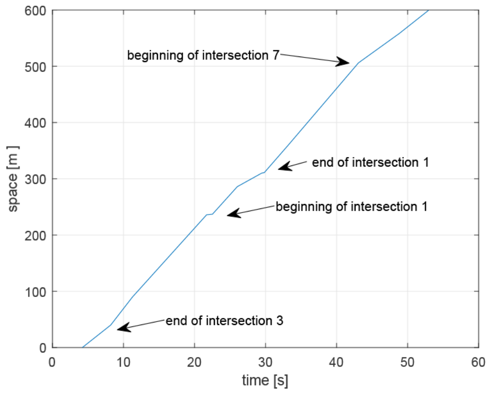

Fig. 16 *Space-time trajectory of a vehicle in the network in Fig. 2*

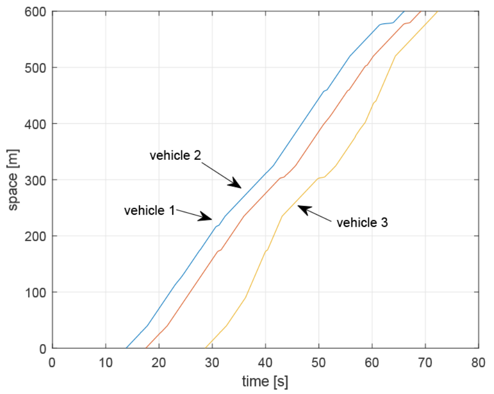

Fig. 17 *Time-space representation of the optimal trajectories of three vehicles in the network in Fig. 3*

vehicle 3 (which is allowed to travel at a higher maximum speed) enters the network around 11 s after vehicle 2 but it exits around 3 s only after vehicle 2. In fact, the shapes reported in Fig. 16 show that the average speed of vehicle 3 is always greater than the speed of the other two vehicles.

7 Conclusions

In this paper, the architecture of a TMS aimed at guaranteeing safety and optimising performance in road networks reserved for self-driving cars has been presented and discussed. At network-level, the TMS determines optimal routing strategies, whereas at local-level the TMS determines optimal trajectories that the autonomous vehicles travelling the network must follow to avoid collisions and to complete the journey in the shortest possible time. This represents a first added value of the proposed research as the works appeared in the literature usually consider a single element of road infrastructure (in general, a single intersection). The optimisation is achieved by solving in sequence a set of mathematical programming problems that model the travel of vehicles within roads and intersections and force safety with some specific constraints. The formalisations of such problems, which are original, represent a second important innovative contribution to the research in this field, as they can be used in real-time to set the trajectories of self-driving cars that approach the traffic network.

The practical application of the proposed methodology has been discussed on the basis of the results of some experiments that have been defined both to analyse the computational effort required by the model and to test some specific scenarios. The outcomes of such experiments are encouraging.

As regards the future perspective of this research, different activities could be carried out to improve the model: firstly, a higher-level macroscopic model, compatible with the several degrees of freedom allowed by the considered models of smart roads and unsignalised intersections, can be integrated into the existing optimisation approach with the aim of determining the best routes between the origin and the destination of vehicles; secondly, the models for different more complex layouts of roads and intersections can be tested; finally, another future development can be the analysis of interactions of self-driving vehicles with other kinds of less-evoluted CAV.

8 References

[1] World Urbanization Prospects: The 2018 Revision, United Nations, Department of Economic and Social Affairs, Population Division, 2018

[2] Connected Automated Driving Roadmap, European Road Transport Research Advisory Council (ERTRAC), 2019

[3] 2nd European Conference on Connected and Automated Driving (EUCAD 2019), 2-3 April 2019, Bruxelles, Belgium

[4] Automated Vehicles Symposium (AVS 2019), 15–18 July 2019, Orlando, FL, USA

[5] Li, L., Wen, D., Yao, D.: 'A survey of traffic control with vehicular communications', *IEEE Trans. Intell. Transp. Syst.*, 2014, **15**, (1), pp. 425–432

[6] Guler, S.I., Menendez, M., Meier, L.: 'Using connected vehicle technology to improve the efficiency of intersections', *Transp. Res. C, Emerg. Technol.*, 2014, **46**, pp. 121–131

[7] Bazzi, A., Zanella, A., Masini, B.M.: 'A distributed virtual traffic light algorithm exploiting short range V2V communications', *Ad Hoc Netw.*, 2016, **49**, pp. 42–57

[8] Gora, P., Rüb, I.: 'Traffic models for self-driving connected cars', *Transp. Res. Procedia*, 2016, **14**, pp. 2207–2216

[9] Yang, K., Guler, S.I., Menendez, M.: 'Isolated intersection control for various levels of vehicle technology: conventional, connected, and automated vehicles', *Transp. Res. C, Emerg. Technol.*, 2016, **72**, pp. 109–129

[10] Li, Z., Elefteriadou, L., Ranka, S.: 'Signal control optimization for automated vehicles at isolated signalized intersections', *Transp. Res. C, Emerg. Technol.*, 2014, **49**, pp. 1–18

[11] Li, P., Zhou, X.: 'Recasting and optimizing intersection automation as a connected-and-automated-vehicle (CAV) scheduling problem: a sequential branch-and-bound search approach in phase-time-traffic hypernetwork', *Transp. Res. B*, 2017, **105**, pp. 479–506

[12] Sun, W., Zheng, J., Liu, H. X.: 'A capacity maximization scheme for intersection management with automated vehicles', *Transp. Res. Procedia*, 2017, **23**, pp. 121–136

[13] Kamal, M.A.S., Imura, J., Ohata, A., *et al.*: 'Coordination of automated vehicles at a traffic-lightless intersection'. Proc. of the 16th Int. IEEE Conf. on Intelligent Transportation Systems (ITSC 2013), The Hague, The Netherlands, 2013, pp. 922–927

[14] Kamal, M.A.S., Imura, J., Ohata, A., *et al.*: 'A vehicle-intersection coordination scheme for smooth flows of traffic without using traffic lights', *IEEE Trans. Intell. Transp. Syst.*, 2015, **16**, (3), pp. 1136–1147

[15] Lee, J., Park, B.: 'Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment', *IEEE Trans. Intell. Transp. Syst.*, 2012, **13**, (1), pp. 81–90

[16] Yan, F., Dridi, M., El Moudni, A.: 'An autonomous vehicle sequencing problem at intersections: a genetic algorithm approach', *Int. J. Appl. Math. Comput. Sci.*, 2013, **23**, (1), pp. 183–200

[17] Wu, J., Abbas-Turki, A., El Moudni, A.: 'Cooperative driving: an ant colony system for autonomous intersection management', *Appl. Intell.*, 2012, **37**, (2), pp. 207–222

[18] Wu, J., Abbas-Turki, A., Perronnet, F.: 'Cooperative driving at isolated intersections based on the optimal minimization of the maximum exit time', *Int. J. Appl. Math. Comput. Sci.*, 2013, **23**, (4), pp. 773–785

[19] de Campos, G.R., Falcone, P., Hult, R., *et al.*: 'Traffic coordination at road intersections: autonomous decision-making algorithms using model-based heuristics', *IEEE Intell. Transp. Syst. Mag.*, 2017, **9**, (1), pp. 8–21

[20] Katriniok, A., Kleibaum, P., Josevski, M.: 'Distributed model predictive control for intersection automation using a parallelized optimization approach', *IFAC-PapersOnLine*, 2017, **50**, (1), pp. 5940–5946

[21] Zhu, F., Ukkusuri, S.V.: 'A linear programming formulation for autonomous intersection control within a dynamic traffic assignment and connected vehicle environment', *Transp. Res. C, Emerg. Technol.*, 2015, **55**, pp. 363–378

[22] Tallapragada, P., Cortés, J.: 'Coordinated intersection traffic management', *IFAC-PapersOnLine*, 2015, **48**, (22), pp. 233–239

[23] Yang, B., Monterola, C.: 'Efficient intersection control for minimally guided vehicles: a self-organised and decentralised approach', *Transp. Res. C, Emerg. Technol.*, 2016, **72**, pp. 283–305

[24] Muller, E.R., Carlson, R.C., Junior, W.K.: 'Intersection control for automated vehicles with MILP', *IFAC-PapersOnLine*, 2016, **49**, (3), pp. 37–42

[25] Malikopoulos, A.A., Cassandras, C.G., Zhang, Y.J.: 'A decentralized energy-optimal control framework for connected automated vehicles at signal-free intersections', 2017, arXiv:1602.03786v7 [math.OC]

[26] Wu, W., Liu, Y., Xu, Y., *et al.*: 'Traffic control models based on cellular automata for at-grade intersections in autonomous vehicle environment', *J. Sens.*, 2017, **2017**, pp. 1–6

[27] Cruz-Piris, L., Lopez-Carmona, M.A., Marsa-Maestre, I.: 'Automated optimization of intersections using a genetic algorithm', *IEEE Access*, 2019, **7**, pp. 15452–15468

[28] Saians-Vazquez, J.V., Ordonez-Morales, E.F., Lopez-Nores, M., *et al.*: 'Intersection intelligence: supporting urban platooning with virtual traffic lights over virtualized intersection-based routing', *Sensors*, 2018, **18**, (11), pp. 1–16

[29] Bichiou, Y., Rakha, H.A.: 'Real-time optimal intersection control system for automated/cooperative vehicles', *Int. J. Transp. Sci. Technol.*, 2018, **8**, pp. 1–12

[30] Liu, B., Shi, Q., Song, Z., *et al.*: 'Trajectory planning for autonomous intersection management of connected vehicles', *Simul. Modelling Pract. Theory*, 2019, **90**, pp. 16–30

[31] Lu, G., Nie, Y.M., Liu, X., *et al.*: 'Trajectory-based traffic management inside an autonomous vehicle zone', *Transp. Res. B*, 2019, **120**, pp. 76–98

[32] Wuthishuwong, C., Traechtler, A.: 'Consensus-based local information coordination for the networked control of the autonomous intersection management', *Complex Intell. Syst.*, 2017, **3**, pp. 17–32

[33] Alireza Fayazi, S.A., Vahidi, A.: 'Mixed integer linear programming for optimal scheduling of autonomous vehicle intersection crossing', *IEEE Trans. Intell. Veh.*, 2018, **3**, (3), pp. 287–299

[34] Li, L., Li, X.: 'Parsimonious trajectory design of connected automated traffic', *Transp. Res. B*, 2018, **119**, pp. 1–21

[35] Zhang, K., Zhang, D., de La Fortelle, A., *et al.*: 'State-driven priority scheduling mechanisms for driverless vehicles approaching intersections', *IEEE Trans. Intell. Transp. Syst.*, 2015, **16**, (5), pp. 2487–2500

[36] Kahaner, D., Moler, C., Nash, S.: '*Numerical methods and software*' (Prentice-Hall, Inc., Englewood Cliffs, NJ, USA, 1989)