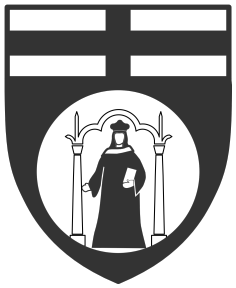


University of Genoa

Polytechnic School

Ph.D. in Maritime Science and Technologies -
Curriculum Logistics and Transportation

Ph.D. THESIS

Traffic Management System for the combined optimal routing, scheduling and motion planning of self-driving vehicles inside reserved smart road networks

Federico Gallo

Advisor:
Prof. D. Giglio
Prof. N. Sacco

Academic Year 2022/2023

Summary

The topic discussed in this thesis belongs to the field of automation of transport systems, which has grown in importance in the last decade, both in the innovation field (where different automation technologies have been gradually introduced in different sectors of road transport, in the promising view of making it more efficient, safer, and greener) and in the research field (where different research activities and publications have addressed the problem under different points of view).

More in detail, this work addresses the problem of autonomous vehicles coordination inside reserved road networks by proposing a novel Traffic Management System (TMS) for the combined routing, scheduling and motion planning of the vehicles. To this aim, the network is assumed to have a modular structure, which results from a certain number of roads and intersections assembled together. The way in which roads and intersections are put together defines the network layout. Within such a system architecture, the main tasks addressed by the TMS are: (1) at the higher level, the optimal routing of the vehicles in the network, exploiting the available information coming from the vehicles and the various elements of the network; (2) at a lower level, the modeling and optimization of the vehicle trajectories and speeds for each road and for each intersection in the network; (3) the coordination between the vehicles and the elements of the network, to ensure a combined approach that considers, in a recursive way, the scheduling and motion planning of the vehicles in the various elements when solving the routing problem.

In particular, the routing and the scheduling and motion planning problems are formulated as MILP optimization problems, aiming to maximize the performance of the entire network (routing model) and the performance of its single elements - roads and intersections (scheduling and motion planning model) while guaranteeing the requested level of safety and comfort for the passengers.

Besides, one of the main features of the proposed approach consists of the integration of the scheduling decisions and the motion planning computation by means of constraints regarding the speed limit, the acceleration, and the so-called safety dynamic constraints on incompatible positions of conflicting vehicles. In particular, thanks to these last constraints, it is possible to consider the real space occupancy of the vehicles avoiding collisions.

After the theoretical discussion of the proposed TMS and of its components and models, the thesis presents and discusses the results of different numerical experiments, aimed at testing the TMS in some specific scenarios. In particular, the routing model and the scheduling and motion planning model are tested on different scenarios, which demonstrate the effectiveness and the validity of such approach in performing the addressed tasks, also compared with more traditional methods.

Finally, the computational effort needed for the problem solution, which is a key element to take into account, is discussed both for the road element and the intersection element.

Contents

Chapter 1

Introduction

This chapter describes the actual framework where the work presented in this thesis is collocated. In particular, Sec. 1.1 discusses how the autonomous driving is, and is expected to be, part of the actual transportation framework, and in particular, what related problems and challenges autonomous vehicles are expected to solve. Sec. 1.2 presents a literature review of scientific papers addressing problems similar to the one faced in this work. Finally, Sec. 1.3 provides a summary of the main thesis contributions and of its structure.

1.1 Self-driving vehicles and the future transportation systems

By 2050, around 65% of the global population will live in cities [1]. A growing group of urban citizens poses a big challenge for already congested and densely populated urban areas. At the same time robotics technologies applied to cars are becoming more and more diffused, paving the way to fully autonomous vehicles, about which it is worldwide recognized that they will have a key role in the future road transport policies, as they can contribute to increase the efficiency of transport systems and, consequently, to reduce energy consumption and emissions ([2]). In addition, automation will address many important transport societal challenges, such as safety, decarbonization, accessibility, and social inclusion.

The automation of road transport is a theme in the transportation field that has grown in importance within the European Union in the last years, together with the aspects related to safety and sustainability. Different research and innovation calls dealing with these topics have been launched with Horizon 2020 both actually and in the recent years. The reason is that automated driving is seen as a possible way to achieve some goals such as the decarbonization, the reduction of emission, the reduction of incidents and, in particular, of fatalities, the improvement of accessibility, the introduction of systems for the management of traffic and the integration of different transport modes.

Transport automation is also mentioned by the Strategic Transport Research and Innovation Agenda (STRIA, [3, 4]), that is an initiative set out in the Energy Union strategy that provides a framework to achieve EU energy and climate goals,

supporting the vision of a clean, connected and competitive European transport system. STRIA builds on and integrates seven thematic transport research areas:

1. Cooperative, connected and automated transport;
2. Transport electrification;
3. Vehicle design and manufacturing;
4. Low-emission alternative energy for transport;
5. Network and traffic management systems;
6. Smart mobility and services;
7. Infrastructure.

The presence of a the road map specifically related to Cooperative, Connected and Automated Transport is a demonstration of the importance of research in the field of automation. This STRIA road map document addresses the Research and Innovation activities and other policy support measures required so that the concepts of connected and automated transport (CAT), for all transport modes, may contribute to the Energy Union 2050 goals in the domains of decarbonization, greater efficiency and competitiveness. Contributions to environmental and efficiency goals are linked: increasing transport system efficiency and reducing time in congested traffic will help decrease energy consumption and vehicle emissions. In particular, by using CAT technologies, these contributions can be achieved by: removing the human element from vehicle operation (the vehicle can achieve the optimum performance parameters, e.g. speed, acceleration, jerk, etc., contributing to environmental and efficiency objectives); minimizing the headways or spacing between vehicles through automated control and connectivity (not needing minimum safety distances based on human error and reaction times, it will be freed up space and the capacity of networks will be increased); through emerging innovative mobility concepts, enabled through CAT technologies (e.g. through modal shift to a greener mode and higher vehicle occupancy rates for passengers, less need for car-ownership through Mobility-as-a-Service (MaaS) or more efficient vehicle utilization for freight transport). In addition to environmental and efficiency benefits, it is overall claimed that automated driving will bring substantial contributions in the fields of safety, comfort, social inclusion and accessibility of the transport system. As far as safety is concerned, the element on which more attention is paid is the removal of the human element from vehicle operation, which can have large effects on lowering collision rates and severity. The comfort of the users will increase because automation can enable them freedom for other activities when automated systems are active, as well as social inclusion, ensuring mobility for all (including elderly and impaired users), and accessibility, facilitating access to city centers.

For these reasons many car and truck manufacturers are working on the development of a fully automated vehicle. The whole industrial sector needs to evolve and adapt in a fast pace to stay ahead in global competitiveness while including all stakeholders and addressing societal needs. By 2025 passenger cars with conditional

automated functions on highways will be available with fully autonomous vehicles, expected to be on the market by 2025-30. Various studies revealed the outstanding economic impact projected for automated driving for the years to come, realized through economic growth, new jobs across the automotive value chain, and wider economic impacts such as increased productivity, reduced time in congestion, reduced number of severe accidents (reduced number of fatalities), efficiency gains in the transport system (i.e. increased capacity and reduced fuel consumption), etc. But the list of challenges that need to be overcome for fully automated transport to become a reality is still long: driver-vehicle interface, system status, functional safety, cybersecurity, public acceptance, driver education, liability issues, policy, funding, and last but definitely not least, infrastructure. As a matter of fact, the progressive introduction of automation in (road) transport is a process strongly relying on ICT (Information and Communications Technologies) development and evolution, and coupled to electrification and alternative fuel propulsion, as asserted in other STRIA road maps like the ones for Transport Electrification, for Vehicle Design and Manufacturing, and for Low-emission Alternative Energy for Transport. The main technological challenges are to produce resilient, affordable sensors and CAT technology operational in all weather and harsh environmental conditions and variable road environments; to improve detection technology and perception intelligence; to enable optimized use of new ICT technologies (Internet of Things, Digitalization, Big data, Cloud computing, Connectivity, Deep Learning) to support the performance of automated transport technologies. A particular attention must be paid to the phase of mixed traffic situations (interaction of vehicles with different levels of automation and different connectivity features (e.g. short range vs, long range), but also with vulnerable road users (cyclists, pedestrians) and other non-automated road users. The safety impact caused by the gradual introduction of (increasingly) automated vehicles in a mixed environment in which, at least at the beginning, the majority of cars will be manually driven, needs to be assessed and consequently will force industry, policy makers and regulators to dynamically adapt to a continuously evolving scenario. It is evident that confined areas with restricted access control, such as terminal areas and ports, and dedicated lanes and parking areas are more suitable to test and introduce highly/fully automated vehicles, with respect to have them deployed to open roads, with mixed traffic flows.

This thesis is contextualized under this framework and it proposes a Traffic Management System (TMS) for the combined optimization and management of autonomous vehicle flows crossing road networks considering the ultimate form of Connected and Automated Vehicles (CAV); in particular, the circulation of fully automated vehicles in roads is analyzed. More in detail, the proposed TMS aims at optimizing in a combined way the vehicle paths in the network and the vehicle trajectories and speeds when crossing the intersections and traveling along the roads of the network.

Since vehicles are assumed to have a degree of automation corresponding to the levels 4 (High Automation) or 5 (Full Automation) of the SAE classification (see Sec. 1.1.1), they are able to perform almost all the driving activities, and in particular, they can safely cross the network with the speeds and the accelerations provided by the TMS, received through specific on-board V2I devices, without the

need of any human interaction, traffic light or signal. Such vehicles are provided with an autonomous driving system that is able to manage all aspects of the dynamic driving task, including the operational (steering, braking, accelerating, monitoring the vehicle and roadway) and tactical (responding to events, determining when to turn, use signals, etc.) features.

1.1.1 SAE classification

The continuous development of suitable technologies is indeed allowing a gradual introduction of growing levels of automation in different sectors of road transport. But a substantial work is still to be done on standardization. With the goal of providing common terminology for automated driving, in ([5]) SAE International delivers a harmonized classification system and supporting definitions (see Tab. 1.1.) that: identify six levels of driving automation from no automation to full automation; base definitions and levels on functional aspects of technology; describe categorical distinctions for a step-wise progression through the levels; are consistent with current industry practice; eliminate confusion and are useful across numerous disciplines (engineering, legal, media, and public discourse); educate a wider community by clarifying for each level what role (if any) drivers have in performing the dynamic driving task while a driving automation system is engaged. SAE specifies that these levels are descriptive rather than normative, technical rather than legal, and that they imply no particular order of market introduction. Elements indicate minimum rather than maximum system capabilities for each level: a particular vehicle may also have multiple driving automation features such that it could operate at different levels depending upon the features that are engaged. Key definitions in J3016 include (among others):

- Dynamic driving task: it includes the operational (steering, braking, accelerating, monitoring the vehicle and roadway) and tactical (responding to events, determining when to change lanes, turn, use signals, etc.) aspects of the driving task, but not the strategic (determining destinations and way-points) aspect of the driving task;
- Driving mode: it is a type of driving scenario with characteristic dynamic driving task requirements (e.g., highway merging, high speed cruising, low speed traffic jam, closed-campus operations, etc.);
- Request to intervene: it is notification by the automated driving system to a human driver that she/he should promptly begin or resume performance of the dynamic driving task.

Of course, such a classification applies to both passenger and freight vehicles, even though passenger cars are recognized to be the main driver of the development towards automated driving, because their high volume in the market makes it affordable to develop the necessary technologies. They evolve level by level with more sensors, connectivity and computing power and can be distinguished by parking and driving use cases. This evolutionary process already started with the development of

ABS, ESP and Advanced Driver Assistant Systems (ADAS) and will progressively apply to more functions and environments, up to reach the fully automated vehicle, which should be able to handle all driving from point A to B, without any input from the passenger, as discussed in another road-specific roadmap: ERTRAC (the European Road Transport Research Advisory Council) Road map on Automated Driving, May 2017, a policy document focused on cars and trucks in the framework of urban and inter-urban transport, which shows current and foreseeable developments from a mainly technology perspective.

Table 1.1: Levels of driving automation defined in SAE international standard J3016.

SAE level	**Name**	**Narrative definition**	**Execution of steering, acceleration and deceleration**	**Monitoring of driving environment**	**Fallback performance of dynamic driving task**	**System capability (driving modes)**
0	No Automation	The full-time performance by the human driver of all aspects of the dynamic driving task, even when enhanced by warning or intervention systems	Human driver	Human driver	Human driver	n/a
1	Driver Assistance	The driving mode-specific execution by a driver assistance system of either steering or acceleration/deceleration using information about the driving environment and with the expectation that the human driver performs all remaining aspects of the dynamic driving task	Human driver and system	Human driver	Human driver	Some driving modes
2	Partial Automation	The driving mode-specific execution by one or more driver assistance systems of both steering and acceleration/ deceleration using information about the driving environment and with the expectation that the human driver perform all remaining aspects of the dynamic driving task	System	Human driver	Human driver	Some driving modes
3	Conditional Automation	The driving mode-specific performance by an automated driving system of all aspects of the dynamic driving task with the expectation that the human driver will respond appropriately to a request to intervene	System	System	Human driver	Some driving modes
4	High Automation	The driving mode-specific performance by an automated driving system of all aspects of the dynamic driving task, even if a human driver does not respond appropriately to a request to intervene	System	System	System	Some driving modes
5	Full Automation	The full-time performance by an automated driving system of all aspects of the dynamic driving task under all roadway and environmental conditions that can be managed by a human driver	System	System	System	All driving modes

1.2 Literature review

Connected and Automated Vehicles (CAV), in their ultimate form (fully unmanned and autonomous), will enable completely new transport systems to be realized. The interest of the research community on this topic is very high: many events take place yearly worldwide (recently, [6], [7]) 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 thesis. As a matter of fact, such strategies bring improvements regarding roads and intersections performance while guaranteeing safety.

In particular, intersections are a very common studied topic in transportation engineering [8] since they arise congestion issues, as they tend to become bottlenecks of the traffic flow, and safety issues: for instance, about 40 percent of vehicle collisions in the United States occurs where two or more roads merge [9].

Therefore, the topic has received a great attention by the research community, as shown for example in [10], 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 [11] a signal control algorithm for an isolated intersection is proposed, analyzing various penetration rates of connected vehicles present in a traffic stream, and evaluating the benefits of this technology; in [12] a VTL (Virtual Traffic Light) algorithm aiming at defining the priorities within intersections for connected vehicles is studied; in [13] 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 [14], signal control strategies for an isolated intersection are determined assuming that three categories of vehicles populate the intersection (conventional vehicles, connected but non-automated vehicles, and automated vehicles), and simulations are conducted to analyze 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 [15], signalized 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 optimized, based

on advanced communication technology between approaching vehicles and signal controller. The research on signalized intersections populated by connected and automated vehicles is constantly growing: in [16], it is studied a joint optimization of CAV and traffic signal timing; in [17], it is developed an innovative intersection operation scheme for CAV which maximizes intersection capacity by dynamically optimizing green durations and lane assignment.

As for the use of automation for intersection control, [18] and [19] 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 [20]; 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 optimization 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: [21] where a genetic algorithm to find an optimal or a near-optimal vehicle passing sequence for adjacent intersections is defined; [22] that proposes an Ant Colony System (ACS) to solve the control problem for a large number of vehicles and lanes; [23] in which a control strategy aimed at minimizing the maximum exit time is obtained by applying dynamic programming; [24] where the coordination of multiple vehicles approaching an intersection is considered in a control-theoretical framework: a decentralized approach combining optimal control with model-based heuristics is proposed; [25] that presents a more general reservation protocol, named AIM*, in which the intersection manager assigns reservations to vehicles on the basis of the priority assigned to each vehicle; this new protocol makes it possible to optimize reservations in real-time using a conflict point separation model.

Going towards an optimization approach that is closer to the one proposed in this thesis, reference is made to the works [26]–[27], that are all relevant to scheduling problems. In particular, in [26] it is developed a Linear Programming formulation for Autonomous Intersection Control (LPAIC) accounting for traffic dynamics within a connected vehicle environment. In [28], the problem of coordinating the passage of vehicles through an intersection is studied, with the aim of minimizing 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 [29], an optimal intersection control scheme is designed under the scenario of an intersection completely modeled 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 [30], a MILP approach for scheduling the vehicle arrivals at an unsignalized intersection is defined with the aim of minimizing 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 [27], it is addressed the problem of optimally controlling CAV crossing an urban intersection without any explicit traffic signaling so as to minimize energy consumption subject to a throughput maximization requirement and to hard safety constraints, without making the vehicles stop at the intersection.

Finally, more recent works focus on methods based on mixed integer mathematical programming formulations, cellular automata models, reinforcement learning, risks functions and game theory to minimize the crossing time of vehicles while preventing collisions. [31] defines 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 optimization 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. [32] proposes an automatic optimization 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 [33], it is investigated a combination of two ideas (enabled by V2V communications and leant on a certain level of automation in the driving) for the optimization 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. [34] proposes a real-time optimization problem with static and dynamic constraints that minimizes 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. [35] proposes a cooperative scheduling mechanism for autonomous vehicles passing through an intersection, called TP-AIM, aimed at minimizing 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. [36] proposes 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 [37], a distributed algorithm for a graph-based intersection network is introduced, with the aim of controlling traffic at a macroscopic level; V2I and I2I 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 neighborhoods and to determine the control policy aimed at maximizing the throughput of each intersection as well as stabilizing the overall traffic in the network. In [38], it is formulated an optimization

problem with the goal of finding the sequence and times of arrival for autonomous vehicles, in order to minimize the maximum access time assigned to the subscribed vehicles and to avoid collisions. [39] addresses the problem of the optimization 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. [40] proposes a priority-based scheduling mechanisms for the management of autonomous vehicles crossing an intersection. [41] uses reinforcement learning to consider a centralized control scheme while reducing computational time; collision avoidance is addressed through the definition of a risk function to be minimized; however it is not guaranteed that conflicts are totally avoided. [42] proposes a coordination scheme with a multi-collision-set strategy to let more that one vehicle enter simultaneously the intersection avoiding conflicts. [43] proposes a conflict decision model to make intelligent vehicles select the optimal driving strategies based on game theory. Finally, [44] proposes an intersection management system based on deep reinforcement learning to schedule the vehicle crossings.

Recently, the attention of the research community started moving from single portions of networks (mainly intersections) to larger portions of networks composed by multiple intersections and roads connecting them, as in [45], which considers a corridor with intersections and roundabouts connected by road stretches and proposes a model to maximize the capacity and find the optimal control input for CAVs.

A network composed by many intersections and roads arises the challenging opportunity to route and coordinate the autonomous vehicles in it, so to dynamically adapt their paths to the variations in the demand and reduce travel times and congestion. While in literature specific routing algorithms for CAVs started recently to appear ([46]), the research on a combined traffic management system that specifically exploits the benefits of a network populated only by CAVs is still at its infancy: [47] proposes a cooperative route planning strategy for autonomous vehicles in a network, considering the real-time information of the system; similarly, [48] considers a routing scheme with current travel time information provided by the vehicles.

1.3 Thesis contributions and structure

Compared to the literature analyzed in Sec. 1.2, the main contributions of this work are the following.

First, this thesis proposes a Traffic Management System (TMS) that combines the routing of the CAVs in a network, their scheduling and motion planning on each element (road or intersection) and the coordination between the network and all its elements. In doing so, the routing problem is solved considering the actual information about the state of the network, which directly comes from the vehicles and the controllers of the various elements of the network. Thus, the real trajectories are considered when evaluating the network state. Additionally, the routing problem considers the real network topology, and in particular the actual layout of each of its elements. Finally, the proposed TMS has an high modularity thanks to the considered modular structure of the network, which is assumed to be composed by different elements, roads and intersections, connected together. As a result, the TMS can be adopted to different layouts of networks.

Secondly, for the elements of the network the scheduling and motion planning problem is formalized as a MILP problem whose solution generates the optimal speeds along the assigned trajectories (chosen among a predefined set according to the destinations for the intersection elements, determined by the problem together with the speeds for the road elements), for all the vehicles crossing the element, minimizing the total travel time while avoiding vehicle stops near or inside the elements. In doing so, the problem is formulated separately for a generic intersection layout (characterized by an arbitrary number of roads and lanes), and for a generic road layout. Note that, as it will be clarified, by only optimizing vehicle speeds, the sequence of vehicles crossing the element also results scheduled at the best. The key features of the MILP formalization are summarized as follows: vehicles are considered with their real space occupancies, positions, speeds and accelerations; vehicle trajectories are modeled to consider their geometrical characteristics such as the curvature radius and the consequent speed limits; conflicts are avoided by specific constraints that manage the incompatibilities of some relative positions of vehicles inside the element.

Third, the thesis proposes an approach, based on a rolling horizon framework, to solve large instances of the scheduling and motion planning problem in an acceptable amount of time.

Finally, the thesis presents the results of the application of the TMS to different case studies, with the aim of providing evidence of its effectiveness both at the routing level (network global performances) and at the elements level (conflict avoidance, maneuvers, improvement of local performance). In addition, for the intersection element the thesis provides also simulation results for the performance comparison between the proposed method and a traditional traffic light control with optimized phase sequences and lengths.

1.3.1 Thesis structure

The structure of the thesis reflects the one of its contributions. This work adopts and cites a variety of mathematical methods, the majority of them belonging to the field of mathematical optimization: they are briefly introduced and explained in chapter 2, together with the indication of suitable references to deepen the related topics.

Then, chapter 3 introduces the proposed Traffic Management System (TMS) for autonomous vehicles, which is based on the identification of three levels, each of them dedicated to specific tasks and operations. As it will be explained, at the higher level vehicles are routed in the network, while at the lower level the specific vehicle trajectories and speeds in each element of the network are determined. At the intermediate level the information and coordination between the higher and the lower level is performed.

After the general description of the TMS and of the actions performed at each level, the thesis enters into the details of the network modeling and of the optimization problems that are solved at each level: firstly, the routing model, which is solved at the higher level, is formalized and discussed in chapter 4, together with its features and the estimation of the network state, on the basis of which the vehicles

are routed. Secondly, the thesis describes the model of the single elements that are assumed to compose the network, which are the intersection element (chapter 5) and the road element (chapter 6). These two chapters have a similar structure, even if the two models have significant differences (which are underlined): they first introduce the notation and the assumptions, the key features of the proposed modeling framework (which are the spatial discretization of the vehicle trajectories, their analytical expression to find a speed reduction factor function on the curvature and the management of conflicts between vehicles) and the scheduling and motion planning model that is aimed at finding the optimal trajectory and speed for each vehicle in the element. In addition, chapter 5 discusses also some aspects which regard both the intersection and the road model: an heuristic method to solve the optimization problems while reducing the computational time, the continuous approximation of the trajectories and their analytical expression.

Subsequently, after the theoretical discussion of the TMS, its components and the optimization models, chapter 7 presents and discusses different numerical experiments, which are conducted assuming specific layouts of the network and of the intersections and roads composing it. The aim of this chapter is to show the possibilities given by the model and to test its performances: it first discusses the results of the application of the routing model to some scenarios; it then presents some applications about the scheduling and motion planning intersection and road models, together with a discussion of their parameters and of the computational time needed to solve them.

Lastly, the thesis ends with chapter 8, which reports the concluding remarks and possible future extensions of this work.

Chapter 2

Methodologies

This chapter provides a summary of the theoretical background of the models and methods used in this work. More in detail, Sec. 2.1 discusses the basic elements of mathematical optimization; Sec. 2.2 discusses the mathematical modeling of curves on the plane and Sec. 2.3 discusses the functions approximation with interpolation techniques (and, in particular, with the Hermite polynomial interpolation and the Hermite cubic spline interpolation) and with the least squares.

2.1 Mathematical optimization

Mathematical optimization deals with the problem of finding the minimum or the maximum of a function f, called *objective function*, subject to specific constraints:

$$\max f(\mathbf{x}) \tag{2.1}$$
$$\text{subject to} \quad \mathbf{x} \in \mathcal{U} \tag{2.2}$$

where $\mathcal{U}$ identifies the set of *feasible solutions* and $\mathbf{x}$ is the vector of the *decision variables*. $\mathcal{U}$ is usually defined by means of equations and disequations called *constraints*, which depends on the addressed problem and model. If a solution vector belongs to $\mathcal{U}$, it is called *feasible solution*. Otherwise it is called *unfeasible solution*. If it also maximizes the objective function, then it is called *optimal feasible solution*.

If both f and $\mathcal{U}$ can be defined by linear expressions, then the optimization problem is said to be *linear*. If at least one between the objective function f and $\mathcal{U}$ are defined by non linear expressions, then the optimization problem is said to be *non linear*. A linear problem can be formalized in the following way:

$$\max f(\mathbf{x}) = \mathbf{c}^T\mathbf{x} \tag{2.3}$$
$$\text{subject to} \quad A\mathbf{x} \leq \mathbf{b} \tag{2.4}$$
$$\mathbf{x} \geq \mathbf{0} \tag{2.5}$$

where the quantities $\mathbf{c}, \mathbf{B}, A$ are, respectively, two vectors and a matrix of constant coefficients representing the parameters of the problem.

If the decision variables in the vector $\mathbf{x}$ are real, then the optimization problem is referred to as Continuous Linear Programming (LP) problem. If they are all integer, then the optimization problem is referred to as Integer Linear Programming (IP)

problem. If some are real and some are integer, then the optimization problem is referred to as Mixed Integer Linear Programming (MILP) problem.

It is possible to demonstrate that LP problems are combinatorial problems, i.e., they correspond to the selection among a finite number of alternatives. However, explicit generation of all the possible solutions (brute force) is not convenient as their number grows exponentially with the size of A. Thus there are, in literature, several algorithms that allow to solve effectively LP problems, like the Simplex method and its variants, and the interior point method. As for IP problems, very often the Simplex algorithm does not generate a feasible optimal solution to IP problems; for these problems other methods are used, like the Cutting planes method, the Branch and Bound method and the Branch and Cut method, which is a combination of the first two.

For more details about the contents of this section, readers may refer to [49].

2.1.1 Multi-objective optimization

When the objective function to optimize is more than one, the problem is referred to as multi-objective optimization (MOO, see [50, 51, 52] for more details about the topic).

The MOO, introduced by Vilfredo Pareto, refers to finding the optimal solution values of more than one desired goals, allowing for a compromise (tradeoff) in case of contradictory objectives. Mathematically, the objective is a vector of scalar functions, and the solution is a vector too. In formulas the equations of the MOO problem can be written as follows:

$$\min/\max f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_n(\mathbf{x}) \tag{2.6}$$

$$\text{subject to} \quad \mathbf{x} \in \mathcal{U} \tag{2.7}$$

where $\mathbf{x}$ is the vector of decision variables, $n > 1$ is the number of objective functions, $\mathcal{U}$ is the set of feasible solutions and $f_n(\mathbf{x})$ is the n^{th} objective function. In general, there is the convention to call problems with large n not multi-objective optimization problems but many-objective optimization problems: these latter problems form a special, albeit important case of multi-objective optimization problems.

While in the single-objective optimization problem the superiority of a solution over other solutions is easily determined by comparing their objective function values, in multi-objective optimization problems the goodness of a solution is determined by the concept of *dominance*: a solution $\mathbf{x}_1$ dominates another solution $\mathbf{x}_2$ if $\mathbf{x}_1$ is not worse in each of the objectives and better in at least one objective than $\mathbf{x}_2$.

The concept of dominance is of fundamental importance in multi-objective optimization, and it allows to define the following concepts:

- **Non-dominated solution set**: given a set of solutions, it is the set of all the solutions that are not dominated by any member of the solution set.

- **Pareto-optimal set**: it is the non-dominated set of the entire feasible decision space. A solution is said to be Pareto-optimal if it belongs to the

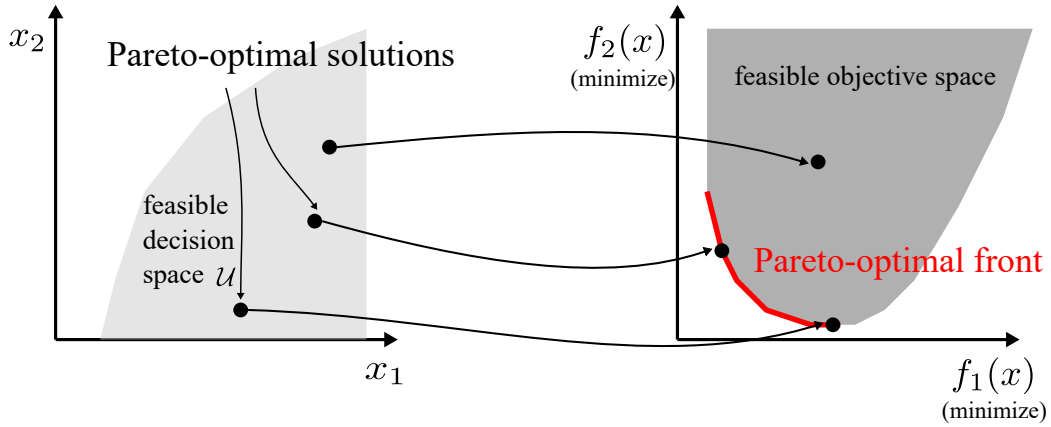


Figure 2.1: Mapping between feasible decision space and feasible objective space, and Pareto front.

Pareto-optimal set.

- **Pareto optimal front**: it is the boundary defined by the set of all points mapped in the objective space from the Pareto optimal set.

As it emerges from the above definitions, in multi-objective optimization there are two spaces to deal with: the decision space, which comprises all candidate solutions (feasible decision space $\mathcal{U}$), and the objective space which is identical to $\mathbb{R}^m$ and it is the space in which the objective function vectors are represented. The vector-valued function $f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_n(\mathbf{x})$ maps the decision feasible space $\mathcal{U}$ to the objective space $\mathbb{R}^m$. Fig. 2.1 shows graphically the mapping between the two spaces together with the Pareto front in the case of a bi-objective minimization problem.

As an example, with reference to Fig. 2.2, let consider a two-objective optimization where the goal is to minimize the functions $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$: red points are non-dominated solutions, where black points are dominated solutions. Other relevant points are the anchor Points, which can be obtained through the best of an objective function, and the utopia Point, that is obtained through the intersection of the maximum/minimum value of an objective function and the maximum/minimum value of another objective function.

The main goal of MOO is to find set of solutions as close as possible to the Pareto-optimal front (hereafter also indicated, for brevity, Pareto front). In literature there are two important families of solution approaches: the scalarization methods and the multi-objective evolutionary algorithms. In the following the most common scalarization method, the weighted sums, which is also used in this thesis, is described.

Scalarization methods and the weighted sums

Classically, multi-objective optimization problems are often solved using scalarization techniques, where the objective functions are aggregated (or reformulated as

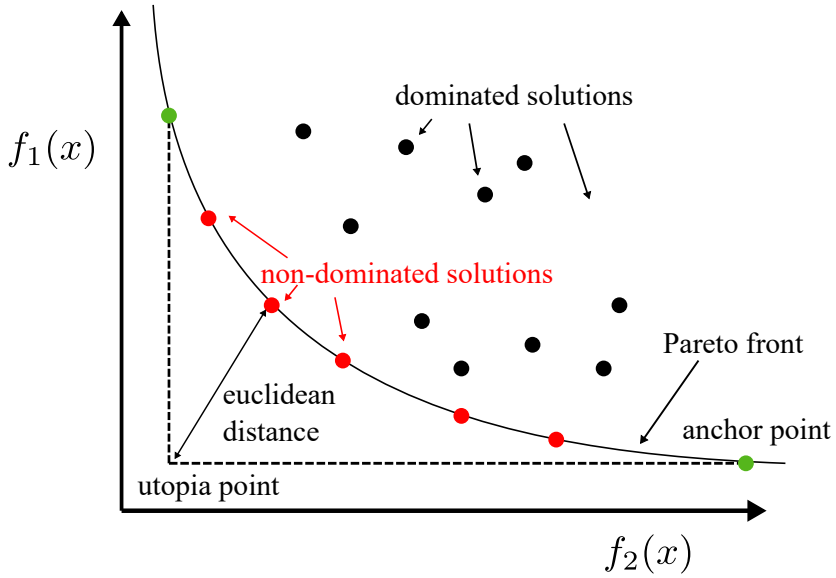


Figure 2.2: Example of dominated and non dominated solutions for a two-objective minimization problem.

constraints), and then a constrained single-objective problem is solved. By using different parameters of the constraints and aggregation function, it is possible to obtain different points on the Pareto front. Although these methods considerably facilitate the computational solution of the MOO problem, they have the disadvantage that for their application, the decision maker is forced to define the particular way of aggregation (including the weights assigned to the objectives) in advance. Basically, this means that (s)he has to specify her or his utility function in quantitative terms before the computational analysis can start.

A simple way to scalarize a problem is to attach non negative weights (at least one of them positive) to each objective function and then to minimize the weighted sum of objective functions. In formulas, the multi-objective optimization problem is reformulated to:

$$\min F(\mathbf{x}) = \sum_{i=1}^{n} w_i f_i(\mathbf{x}) \tag{2.8}$$

$$\text{subject to} \quad \mathbf{x} \in \mathcal{U} \tag{2.9}$$

which is known as linear scalarization problem (LSP). The weights w_i of the objectives are chosen in proportion to the relative importance of the objective.

It can be proved ([51]) that the solution of a LSP is on the Pareto front, no matter which weights are chosen. However, it cannot find certain Pareto-optimal solutions in the case of a non convex objective space. Practically speaking, in the case of concave Pareto fronts, the LSP will tend to give only extremal solutions, that is, solutions that are optimal in one of the objectives. This phenomenon is illustrated in Fig. 2.3, where the tangential points of the dashed lines indicate the solution obtained by minimizing an LSP for different weight choices (colors). In the case of the non-convex Pareto front (Fig. 2.3, right), even equal weights (dark green) cannot lead to a solution in the middle part of the Pareto front. Also, by

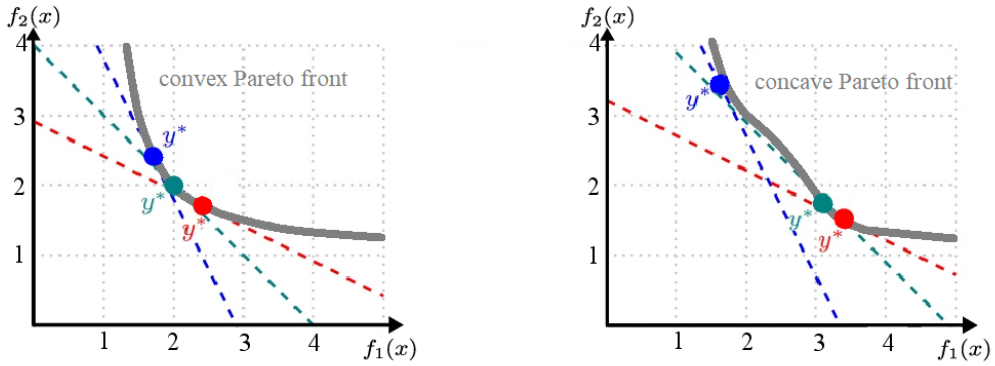


Figure 2.3: Linear scalarization problems with different weights for convex Pareto fronts (on the left), and concave Pareto front (on the right).

solving a series of LSPs with minimizing different weighted aggregation functions, it is not possible to obtain this interesting part of the Pareto front.

Another disadvantage of this method lies on the difficulty in setting the weights w_i to get a Pareto-optimal solution in a desired region in the objective space.

2.1.2 An overview of the available optimization tools

The theory and methods mentioned in the above sections are implemented in softwares and libraries (in the following, optimization tools) that allow to solve effectively an optimization problem. The aim of this section is to provide an overview of the most diffused and powerful optimization tools available on the market, discussing their main features, problem domain and capability of solving large scale optimization problems.

An optimization software is a tool specifically designed to solve complex optimization problems. Usually the process of finding a solution consists of five main stages:

1. problem formulation: definition of the decision variables, the objective function and the constraints;

2. model creation: translation of functions and constraints in a modeling language, usually in an algebraic modeling language that can be read by the solver;

3. model configuration: setting of input data and parameters of the model, according to the constraints and the objective function;

4. optimization of the objective function(s);

5. generation of the optimal solution.

To follow the above steps two main tools are needed, working in concert: an algebraic modeling system and an optimization solver. The optimization solver does the step 4, solving the optimization problem by means of specific algorithms.

The algebraic model generator does the steps 2 and 5, acting as an interface between the user and the solver. It provides an high-level structured language that allows to interact with the solver through text files instead of data structures and it allows to call different solvers. In addition it allows to keep separated the logical structure of the model (variables, objective function, constraints, ...) from numerical data so to avoid the need of rewriting the model each time a change in data occurs.

The most common algebraic modeling systems are ([53]) AMPL (A Mathematical Programming Language, http://www.ampl.com), GAMS (General Algebraic Modeling System, http://www.gams.com) and MPL (Mathematical Programming Language, http://www.maximal-usa.com/mpl/). These are commercial systems, but there exists also modeling languages and interfaces that integrates with most diffused programming languages (e.g. Pyomo and PuLP for Python, FLOPC++ for C++) and softwares (e.g. SolverStudio for spreadsheets, RSymphony for R, OPTI toolbox for Matlab).

Among the most common and diffused optimization solvers there are CPLEX, Gurobi, XPRESS, MOSEK, LINDO, Excel SOLVER (commercial) and CLP, GLPK, lp_solve and SOPLEX (open source).

Often modeling languages interface with solvers by writing out an intermediate file that is read by the solver; common file formats are MPS (not easily human-readable, it only supports linear modeling), LP (developed by CPLEX as a human-readable alternative to MPS) and .nl (AMPL intermediate format).

In most common optimization suites the algebraic modeling system and the solver are both provided, together with an Integrated Development Environment (IDE), as part of a unique stand-alone software (e.g. IBM ILOG CPLEX optimization Studio).

As for the solvers, it is worth saying that results in scientific literature (see, for instance, [54], [55], [56] and [57]) state that there is not a single best solver for all types of problems or for any quality measure, but the efficacy of an optimization software depends on the nature of the problem and on the computational time. However, while current market with such softwares is quite extensive, some of them are often considered ([54] and [55]), in general, more powerful: CPLEX, XPRESS and Gurobi. All this solvers come with an Integrated Development Environment (IDE) and work with some specific programming languages, but they can also be called from external environments or programming languages like, for instance, Matlab or Python.

CPLEX is one of the most known and advanced optimization solvers. It allows to deal with large scale problems and it offers features like heuristics, branch and cut methods and parallel optimization.

Algorithms available includes, for continuous problems: primal and dual simplex, interior-point (barrier); for integer problems: advanced branch-and-bound with presolve, feasibility heuristics, and cut generators. For continuous problems comprised mostly or entirely of linear network flow constraints: network simplex.

It can tackle with linear programming, mixed integer programming, convex and non-convex quadratic programming, second-order cone programming and non-linear programming. Under high dimensionality problems, it performs better that Gurobi ([54]). CPLEX is available to academic researchers and students for free in its full

professional version.

XPRESS can solve very large optimization problems, in particular mixed integers, and can often solve problems that cannot be solved by other solvers. Moreover, it performs better than CPLEX and Gurobi on complex and high scalability. It contains three main powerful solvers: simplex, barrier and integer. The most appropriate one is automatically chosen by the system on the basis of an analysis of the data set.

Available algorithms includes, for continuous problems: primal and dual simplex, interior-point (barrier); for integer problems: advanced branch-and-bound with presolve, feasibility heuristics, and cut generators.

It is not generally available for academic purposes (apart from the presence of agreements with specific institutions). As CPLEX, it can tackle with linear programming, mixed integer programming, convex and non-convex quadratic programming, second-order cone programming and non-linear programming.

Gurobi is a recently developed solver, it contains LP and MIP solver that reaches often better results than CPLEX. Algorithms available includes, for continuous problems: primal and dual simplex, interior-point (barrier); for integer problems: advanced branch-and-bound with presolve, feasibility heuristics and cut generators.

It is free in its full professional version for academic purposes, as CPLEX. It can tackle with linear programming, mixed integer programming, convex and non-convex quadratic programming and second-order cone programming.

Tab. 2.1 shows some details and information about the three optimization solvers CPLEX, XPRESS and Gurobi, like the website, the type of license, the modeling language used and other peculiarities.

Table 2.1: Different optimization solvers

	IBM ILOG CPLEX optimization Studio	FICO Xpress	Gurobi
Website	https://www.ibm.com/analytics/cplex-optimizer	https://www.fico.com/en/products/fico-xpress-optimization	https://www.gurobi.com
License	Proprietary (possible free access through academic programs)	Proprietary	Proprietary (possible free access through academic programs)
Language accepted	OPL	Mosel	Different languages accepted (GAMS, AMPL, ...)
GUI	Yes	Yes	No
Non-convex LP/MIQP	Yes	No	No
Support for multi objectives	Yes	No	Yes

The optimization solvers mentioned above can be used with Python through the

use of some dedicated libraries. One is ([58]) the Pulp library (https://pypi.org/project/PuLP/), an open source package that allows to describe mathematical programs in the Python computer programming language. Pulp supports linear and mixed integer programming and can generate MPS or LP files and call GLPK, COIN CLP/CBC, CPLEX, and Gurobi to solve linear problems.

To cope with non-linear models the Pyomo open-source package (http://www.pyomo.org, [59]) can be used: it supports also quadratic and stochastic programming as well as multi-objective optimization.

As a concluding remark, performance indicators and results from trials that are present in related (and above mentioned) literature show a lot of variability between problems and between runs of the same solver with different random data, so while it is not possible to indicate a best solver for any kind of problem, it is advisable to do some testing on the models of interest.

2.1.3 Heuristic methods

In general, IP and MILP problems are difficult to solve, thus many problem instances are intractable and the methods mentioned in the above sections (which are exact methods, since they are aimed at finding the exact optimal solution) struggle to find the optimal feasible solution. To cope with this, heuristic methods are used. Heuristic methods are a collection of techniques designed for solving a problem more quickly than the classic exact methods, and are particularly useful when these are too slow. Heuristic methods usually find an approximate solution (suboptimal solution) by trading optimality for speed. This thesis proposes an heuristic method for the solution of the proposed scheduling and motion planning problem, which is explained in Sec. 5.4.

2.2 Modeling of curves on the plane

This section provides a summary of the basic differential geometry concepts regarding the curves on the planes. Such concepts are used in Sec. 5.6 to derive the analytical expression of vehicle trajectories and subsequently the speed reduction factors as a function of the trajectory curvature.

A continuous parametrized curve on the plane ($\mathbb{R}^2$) is defined as a continuous map $\gamma : I \to \mathbb{R}^2$, where $I \subset \mathbb{R}$ is an interval:

$$\gamma \in C^0(I) \quad | \quad \gamma(t) = (x(t), y(t)), \quad t \in I \tag{2.10}$$

The image set $\mathcal{C} = \gamma(I) \subset \mathbb{R}^2$ is called the *trace* of the curve (Fig. 2.4).

Lines, circles and clothoids are example of curves that can be parametrized in the following way:

$$\gamma(t) = \xi, \quad t \in I, \quad p \in \mathbb{R}^2 \quad \text{fixed} \tag{2.11}$$

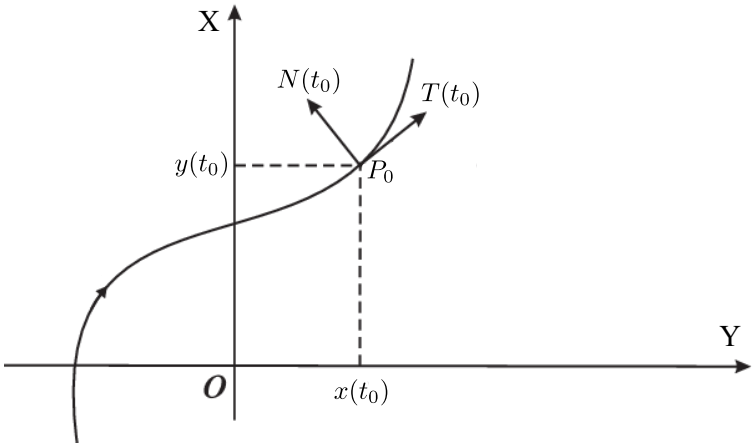


Figure 2.4: Trace of a simple curve on the plane.

for the line,

$$\gamma(t) = (R\cos(t), R\sin(t)), \quad t \in [0, 2\pi] \tag{2.12}$$

for the circle, where R is a constant value, and

$$\gamma(t) = \left(a\int_0^t \cos(t^2)dt, a\int_0^t \sin(t^2)dt\right) \tag{2.13}$$

for the clothoid, where a is a scale parameter.

A curve $\gamma : I \to \mathbb{R}^2$ is said to be *simple* if, $\forall t_0, t_1 \in I, t_0 \neq t_1$, with at least one between t_0 and t_1 internal to I, $\gamma(t_0) \neq \gamma(t_1)$. Geometrically, the trace of a simple curve does not intersect itself.

A curve $\gamma : I \to \mathbb{R}^2$ with parametrization $\gamma(t) = (x(t), y(t)), t \in [a, b]$ is said to be *regular* if $x, y : [a, b] \to \mathbb{R} \in C^1((a, b))$ and $\gamma'(t) = (x'(t), y'(t)) \neq (0, 0) \forall t \in (a, b)$.

In a regular curve there always exists, $\forall t_0 \in (a, b)$, the *tangent vector*

$$T(P_0) = \gamma'(t_0) = (x'(t_0), y'(t_0)) \tag{2.14}$$

to γ, at t_0, in $P_0 = (x(t_0), y(t_0))$.

In addition, it is also possible to define the *normal vector*

$$N(P_0) = (y'(t_0), -x'(t_0)) \tag{2.15}$$

to γ, at t_0, in P_0.

The length l of a simple regular curve $\gamma : [a, b] \to \mathbb{R}^2$ is

$$l(\gamma) = \int_a^b |\gamma'(t)| = \int_a^b \sqrt{x'^2(t) + y'^2(t)}dt \tag{2.16}$$

The last concept to be introduced is the curvature, which is a quantity describing the shape of a curve in a given point.

Given a curve $\gamma : I \to \mathbb{R}^2$, the curvature of γ at t is the real number

$$k(t) = \frac{||\gamma'(t) \wedge \gamma''(t)||}{||\gamma'(t)||^3} = \frac{\det[\gamma'(t)\gamma''(t)]}{||\gamma'(t)||^3} = \frac{|x'(t)y''(t) - x''(t)y'(t)|}{(x'(t)^2 + y'(t)^2)^{3/2}} \tag{2.17}$$

where $\wedge$ denotes the vectorial product and $[\gamma'(t)\gamma''(t)]$ is the 2 X 2 matrix with column $\gamma'(t)$ and $\gamma''(t)$.

Finally, the quantity

$$\rho(t) = \frac{1}{k(t)} \tag{2.18}$$

is called *curvature radius* at t.

According to the above definitions, for a circle with parametrization $\gamma(t) = (R\cos(t), R\sin(t))$ it can be easily found that $k(t) = \frac{1}{R}$ and thus that $\rho(t) = R$.

For more details about plane curves, the reader may refer to [60].

2.3 Functions approximation: interpolation and least squares

Interpolation is a mathematical technique to approximate a function when some of its points are known: it is used in this work (Sec. 5.5) to find the continuous speed function to provide to the vehicles once some of its points are known (because they are determined by the optimization problem, which discretizes the time and the space).

More in details, given $n + 1$ pairs of (x_i, y_i) values, interpolation deals with the problem of finding a function $\phi = \phi(x)$, dependent on $x_0, \ldots, x_n$, such that $\phi(x_i) = y_i \ \forall i = 0, \ldots, n$. It is then possible to say that ϕ interpolates the values $\{y_i\}$ in nodes $\{x_i\}$.

If ϕ is a polynomial, then the problem is referred to as *polynomial interpolation*.

It is possible to demonstrate that, given $n+1$ distinct pairs (x_i, y_i), there exists a unique polynomial of degree n that interpolates them. The unique interpolating polynomial can be written in two alternative (but equivalent) forms: the *Lagrange form* and the *Newton form*.

If, besides the values $\{y_i\}$, also the values of the derivatives of the unknown function in nodes $\{x_i\}$ are known, it is possible to generalize the Lagrange interpolation to the *Hermite polynomial interpolation*, which is said to be *simple* if only the first-order derivatives are known. In such case, if $n+1$ tuples $(x_i, f(x_i) = y_i, f'(x_i) = y'_i)$ are known, letting $N = 2(n + 1)$, it is possible to demonstrate that, if the nodes are all distinct, it exists a unique polynomial H_{N-1} of degree $N - 1$, called *Hermite interpolation polynomial*, such that

$$H_{N-1}(x_i) = y_i \quad \forall i = 0, \ldots, n \tag{2.19}$$

$$\left.\frac{dH_{N-1}(x)}{dx}\right|_{x=x_i} = H'_{N-1}(x_i) = y'_i \quad \forall i = 0, \ldots, n \tag{2.20}$$

in the form

$$H_{N-1}(x) = \sum_{i=0}^{n} y_i L_{i0}(x) + y_i' L_{i1}(x) \tag{2.21}$$

The functions L_{i0} and L_{i1} are called *Hermite characteristics polynomial* and are defined in the following way:

$$L_{i0}(x) = l_{i0}(x) - \left.\frac{dl_{i0}(x)}{dx}\right|_{x=x_i} L_{i,1}(x) \tag{2.22}$$

$$L_{i,1}(x) = l_{i,1}(x) \tag{2.23}$$

being

$$l_{i0}(x) = \prod_{k=0,k\neq i}^{n} \left(\frac{x - x_k}{x_i - x_k}\right)^2 \tag{2.24}$$

$$l_{i1}(x) = (x - x_i) \prod_{k=0,k\neq i}^{n} \left(\frac{x - x_k}{x_i - x_k}\right)^2 \tag{2.25}$$

The above formulas can be generalized to the case m_i derivatives are node for each node x_i.

A drawback of polynomial interpolations is that, as it emerges from the above definitions, when the number of data points increases also the degree of the polynomials increases; this can cause the interpolating function to have oscillations between the data points. This phenomenon is known as Runge's phenomenon, and can lead to high interpolation errors. To cope with this, polynomial interpolation is often substituted by the *spline interpolation*, which is a form of interpolation where the interpolant is a special type of piecewise polynomial called a spline. In other words, instead of fitting a single, high-degree polynomial to all of the data points at once, spline interpolation fits lower-degree polynomials to small subsets of the data points.

A particular type of spline interpolation is the Hermite cubic spline interpolation, which uses a third-degree spline where each polynomial is in the Hermite form defined by the Equs. (2.22) to (2.25).

The requested data are the desired function values and derivatives at each data point. For this reason cubic polynomial splines can be used in the modeling of motion trajectories that pass through specified points of the plane, once the position and the speed (which is the first time derivative of the position) are known. The Hermite formula is applied to each interval (x_k, x_{k+1}) separately and the resulting spline will be continuous and will have continuous first derivative.

Finally, another class of function approximation methods is based on least squares. Differently from interpolation, least squares method is aimed at finding a polynomial that approximate the unknown function by minimizing the mean square error between the data points and the polynomial itself. Thus, in least squares method the polynomial is not required to pass through all the data points. This method is particularly useful when addressing the problem of representing a relation with inputs and outputs when a large number of discrete observations is available (*polynomial data fitting*).

For more details about these and other numerical methods, the reader may refer to [61].

Chapter 3

The proposed Traffic Management System

The Traffic Management System (TMS) proposed in this thesis has the dual purpose of optimizing the performance of the network by determining and applying optimal routing strategies and of optimizing the trajectories of single self-driving vehicles traveling through the network.

In this connection, the TMS acts at three hierarchical levels, each of which associated to a proper controller performing specific tasks: the first level is the routing level, at which the routing controller (RC) determines, by solving a routing problem (whose model is formalized in Sec. 4), the optimal paths of vehicles entering the network. The paths take into account the origin and the destination of each vehicle, as well as the information regarding the current network state, which is provided by the network controller (NC), which acts at the second level, the network one.

The network controller manages the exchange of information among the various elements (intersections and arcs) which compose the network; it also collects from them the information regarding the network state and provides them to the routing controller, receiving the vehicle paths.

At a local level, the various local controllers (LC), related to both roads (LC-RO) and intersections (LC-IN), determine the motion (time trajectory over a predefined space trajectory) of each vehicle inside each element and send the solution to them. This is done by solving a scheduling and motion planning problem, which is mathematically formalized as an optimization problem, whose model is formalized and discussed in Sec. 5.3.1 for the intersections and in Sec. 6.3.1 for the roads.

Fig. 3.1 shows a sketch of the proposed TMS architecture for the case of a traffic network consisting of 4 intersections and 8 smart roads connecting the intersections. In such example, vehicles enter the network through the intersections, but it is worth pointing out that the proposed architecture can be applied to different layouts of networks; in addition, the proposed models for roads and intersections can be applied to different layouts of roads and intersections. However, for simplicity, this work considers only networks built by joining, in different combinations, the same layout of road (the one illustrated in Fig. 6.1) and the same layout of intersection (the one illustrated in Fig. 5.1 for intersections).

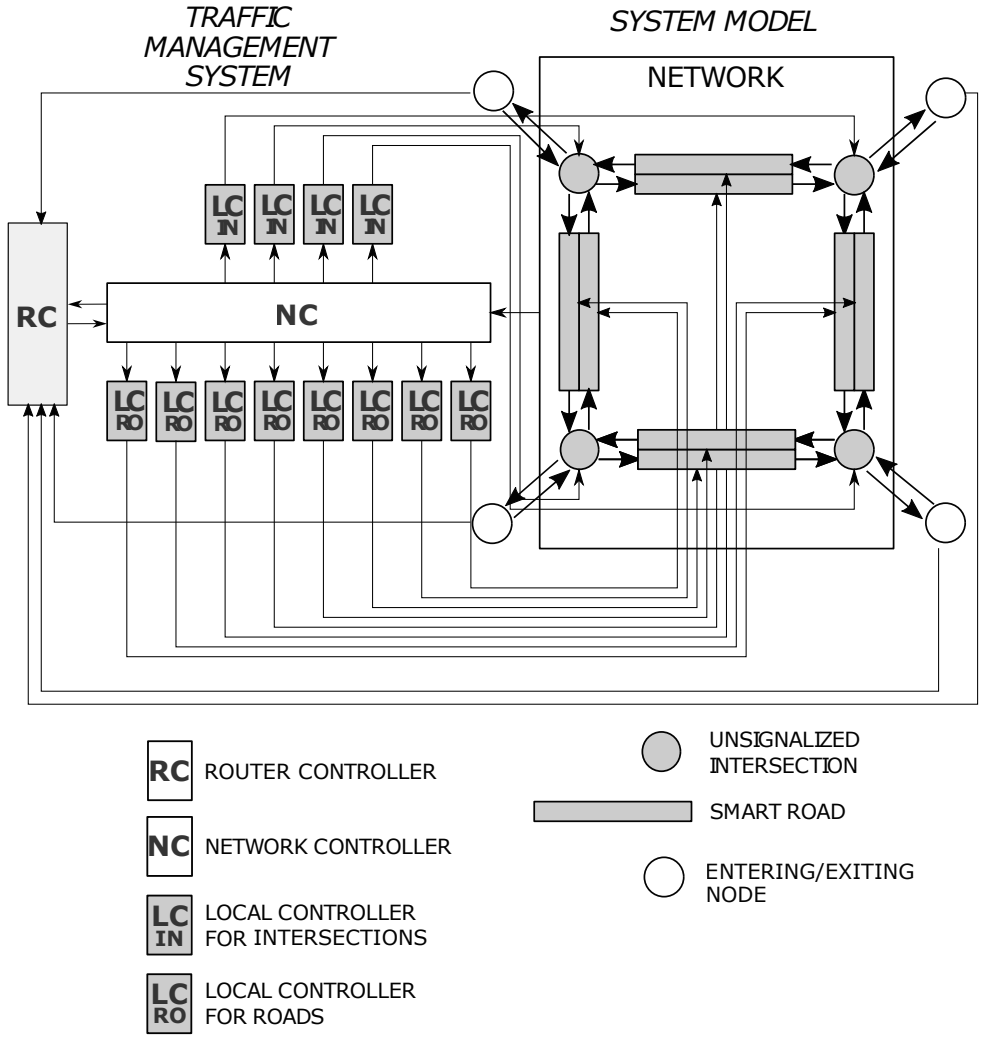


Figure 3.1: Architecture of the proposed 3-level Traffic Management System (TMS).

Figs. 3.2 and 3.3 shows two examples of network layouts that can be obtained by joining the intersections and the roads in two different ways: a square network with four intersections at the edges (Fig. 3.2) and an arterial road with an intersection in the middle (Fig. 3.3).

To summarize, the following general assumptions hold:

1. the traffic network can be decomposed in a set of basic elements, the "intersections" and the "road stretches" (in the following simply "roads");

2. vehicles have a SAE level 4 or 5 of automation, and so they can safely cross the intersection with the optimal speeds and accelerations determined by solving the proposed MILP problem without the need for any human interaction;

3. vehicles can communicate with the infrastructure through specific on-board vehicle-to-infrastructure (V2I) devices, in order to provide the information about their travel to the intersection controller and to receive from it the optimal speeds. Safety and comfort in crossing the intersection are ensured by specific constraints of the optimization problem.

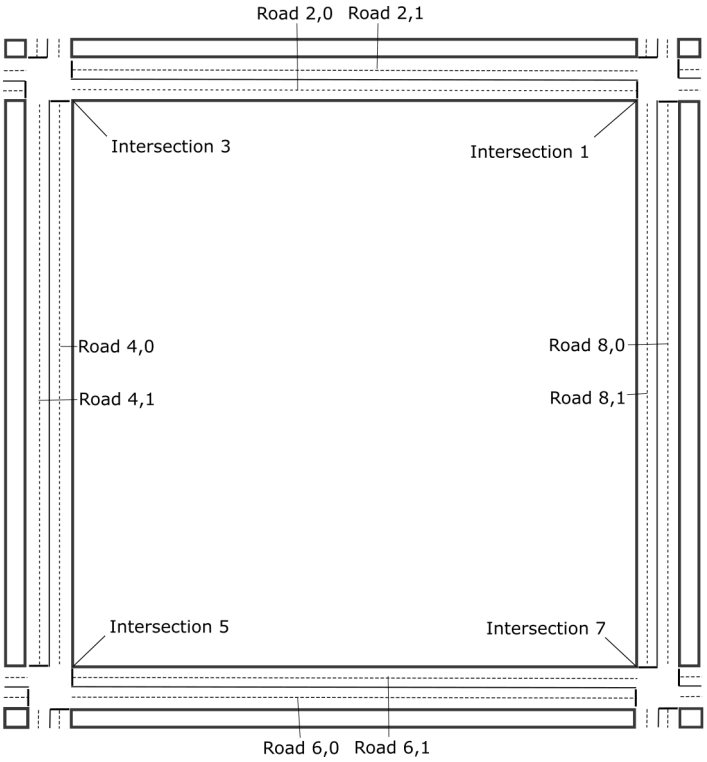


Figure 3.2: Example of network layout: a square network with four intersections at the edges connected by the road elements.

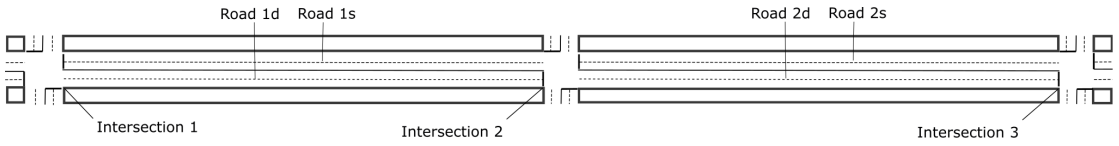


Figure 3.3: Example of network layout: an arterial road composed by three intersections and four roads.

In the following sections, the actions taking place at each of the three hierarchical levels characterizing the proposed TMS are described more in detail.

3.1 Three-level modeling

This section enters into the details of the logical steps that are performed at each of the three levels of the TMS: the routing level, where the optimal paths of the vehicles are determined, the network level, which manages the exchange of information among the various elements of the network and the information between the elements and the routing controller, and the local level, where the optimal trajectories and speeds of all vehicles are determined by solving, for each element, the related scheduling and motion planning problem.

3.1.1 Routing level

The first task performed by the TMS is the optimal routing of the vehicles that are going to enter the network. This task is performed by the RC component of the TMS, which adopts a discrete-time approach which allows to consider a relatively small number of vehicles in each routing problem (to reduce the computational times) and update the information regarding the network state. To be more precise, the routing problem solved at the generic time instant $\tau_k - \xi$ considers all the vehicles whose Estimated Arrival Time at the Network (EATN) is in the interval $\Delta\tau_k = [\tau_k, \tau_{k+1}) = [\tau_k, \tau_k + \Delta\tau)$, being $\Delta\tau$ the problem time window and ξ the maximum time allowed to the RC for solving the routing problem and providing the vehicle paths to the NC.

The logical steps accomplished by the system in the interval $[\tau_k - \xi, \tau_{k+1})$ are the following:

1. At $\tau_k - \xi$, all vehicles that are distant no more than a distance d_{n} from the network are queried by the RC about their EATN, origin and destination.

2. The RC defines the set $\mathcal{W}_k$ of active vehicles by gathering all vehicles with EATN $< \tau_{k+1}$.

3. The NC communicates to the RC the updated information regarding the network state.

4. The RC states, solves the routing problem, and communicates the solution in the interval $[\tau_k - \xi, \tau_k)$ to the NC.

Such scheme is applied repetitively interval-by-interval and the parameters ξ, d_{n} and $\Delta\tau$ can be suitably set in accordance with computational requirements.

3.1.2 Network level

The second task performed by the TMS is the management of the exchange of information between the RC and the various LCs, and among the LCs themselves. These tasks are performed by the NC. In particular, and with reference to the time discretization introduced in Sec. 3.1.1, at each interval $\Delta\tau_k$ the NC accomplishes the following steps:

1. In the interval $[\tau_k - \xi, \tau_k)$ it communicates to the RC the information regarding the state of the network, and receives from it the optimal paths of the vehicles entering the network.

2. At $\tau_k - \epsilon_i$ it defines, for each element e of the network, the set $\mathcal{V}_k$ of active vehicles, i.e., the vehicles that will enter e in the interval $[\tau_k, \tau_{k+1})$.

3. It determines:

   - in the case a road element, the entering lane of each vehicle belonging to $\mathcal{V}_k$;

- in the case of an intersection element, the exact stream of each vehicle belonging to $\mathcal{V}_k$ (for example, with reference to the intersection sketched in Fig. 5.1, one out of the admissible 24);

4. It communicates to the LCs all the data relevant to the considered set of vehicles.

5. In the interval $[\tau_k - \epsilon_i, \tau_k)$ it receives the optimal space-time trajectories of the vehicles from each LC and stores them.

6. In the interval $[\tau_k, \tau_{k+1})$ it receives from each LC the data (time instant, speed and exiting lane) about the vehicles exiting the related element of the network. This data will then be used to perform the step 2 in the interval $[\tau_{k+1}, \tau_{k+2})$, considering the network topology.

To summarize, the step 1 regards the communication with the RC; the steps 2 and 3 do dot require communication with the other levels of the TMS and is preformed entirely by the NC considering the network topology; the steps 4,5 and 6 are related to the communication with the various LCs and allows the exchange of information between the elements of the network.

To explain better the last point, consider a traffic network like those reported in Figs. 3.2 and 3.3, 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 element e) is assumed to provide the data only to the element that follows (let it be element f). The data to provide are the time instant at which the vehicles cross the last node of e and their speed. To guarantee enough spatial precision, in the proximity of the position of the last node, it is assumed the presence of a *trigger*; when a vehicle crosses the trigger, it sends the value of its current speed and the time instant to the LC. The LC adds the information regarding the exiting lane and provides the data to the NC. Before transmitting it to the LC of the element f, the NC collects this data from the LCs of the other adjacent elements of f. In doing so, the network topology is considered.

In particular, and with reference to the layouts depicted in Figs. 3.2 and 3.3, an intersection has four road elements exiting it (i.e., each intersection element has four following road elements); thus, the information regarding the exiting vehicles must be divided by the NC into four groups, one per each road connected to the intersection (obviously, in the case of other intersection layouts the number of exiting roads and thus of the groups may be less or greater than four). On the other hand, a road element has only one following element, the intersection, which receives vehicles by other three road elements; thus, the information regarding the exiting vehicles must be grouped by the NC together with the ones coming from the other three road elements, before being sent to the following intersection.

A graphical representation of the connection between two adjacent elements e and f is shown in Fig. 3.4, where a single lane of the two adjacent elements is reported (the notation adopted for variables s and nodes P is introduced in section 6.1.2). A buffer zone between the two elements (i.e., a road stretch not considered in the

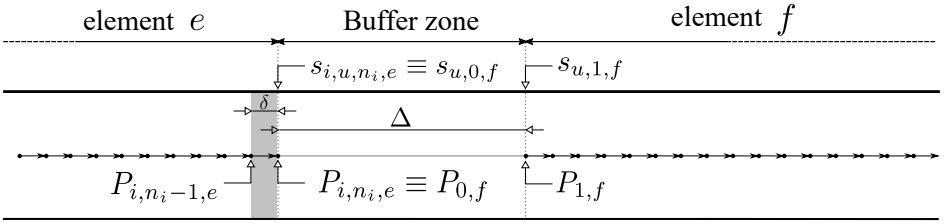


Figure 3.4: Graphical representation of the information that is exchanged between two adjacent elements e and f.

scheduling and motion planning problem where vehicles are assumed to control themselves) is considered, to allow the statement and the solution of the optimization problem before vehicles enter element f. As already mentioned, the position $P_{0,f}$ of the trigger of element f is assumed to coincide with the last node $P_{i,n_i,e}$ of e. After a distance Δ representing the length of the buffer there is the first node $P_{1,f}$ of element f.

Analogously, the time instant $s_{i,u,n_i,e}$ at which the vehicle u exits element e (which is determined by the LC of e by solving the related scheduling and motion planning model) is expected to be equal to $s_{u,0,f}$, which is the time instant at which u crosses the trigger.

Theoretically, thanks to the above mentioned architecture, the scheduling and motion planning model for element f could be solved in advance once known the value of $s_{i,u,n_i,e}$ (which comes from the solution of the scheduling and motion planning model for element e). However, using the value $s_{u,0,f}$ (transmitted by u at the trigger is in general preferable to guarantee precision and avoid possible small errors to propagate.

The proposed architecture 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 trigger of the road and enters the following intersection in the same optimization window (τ_h, τ_{h+1});
- each downstream element f is always able to receive all the vehicles coming from e, that is, 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 5.3.

The second assumption is required since the model assumes that a generic element exchanges information (through the network controller) related to the vehicle arrivals only with the next element, road or intersection, in the vehicle path. This hypothesis can be partially overcome by the RC, which could redirect vehicles (coming from e and going to f) to alternative routes.

In addition, a particular consideration must be done for the buffer zone, that 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 the speed required, avoiding excessive accelerations or decelerations. The buffer zone can be also used as a mean for recovering the possible errors related to the difference between the real crossing time of the vehicles at the trigger ($s_{u,0,f}$) and the expected time determined on the basis of the solution of the precedent element e ($s_{i,u,n_i,e}$). The error could be due to a difficulty of the vehicles in respecting the speeds that are assigned to them by the LC of the element. The presence of the buffer zones limits the propagation of the error.

Finally, it is worth pointing out that the above discussed network-level architecture has the advantage of guaranteeing an high modularity, since the scheme could be applied to different network layouts and also element layouts. Besides, vehicles can enter the network both from an intersection and from a road element.

3.1.3 Local level

At a local level, the TMS builds and solves a specific optimization problem for each element of the network (roads and intersections) by adopting a discrete-time approach that allows to consider a relatively small number of vehicles in each problem. To be more precise, the optimization problem that is solved at the generic time instant $\tau_k - \epsilon_i$ takes into consideration all the vehicles whose Estimated Arrival Time at Road (EATR) or Estimated Arrival Time at Intersection (EATI) is in the interval $\Delta\tau_k = [\tau_k, \tau_{k+1}) = [\tau_k, \tau_k + \Delta\tau)$, being $\Delta\tau$ the problem time window and ϵ_i the maximum time allowed to the local controller i for solving the optimization problem and providing the speed profiles to the vehicles.

The logical steps that are accomplished by the system in the interval $[\tau_k - \epsilon_i, \tau_{k+1})$ are the following:

1. 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 network controller about their EATR or EATI, origin, destination and shape (i.e., width, length, and barycenter position).

2. The network controller (NC):

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

   (b) determines:

   - in the case of smart road, the entering lane of each vehicle belonging to $\mathcal{V}_k$;
   - in the case of unsignalized intersection, the exact stream of each vehicle belonging to $\mathcal{V}_k$ (for example, with reference to the intersection sketched in Fig. 5.1, one out of the admissible 24);

   (c) communicates to the local controller all the data relevant to the considered set of vehicles.

3. The local controller states the optimization problem, solves it, and communicates the solution in the interval $[\tau_k - \epsilon_i, \tau_k)$ to each vehicle in the set $\mathcal{V}_k$.

4. The vehicles in set $\mathcal{V}_k$ safely travel the road or cross the intersection by following the assigned trajectories with the optimal speed values provided by the local controller at step 3.

It is worth pointing out that, while for an intersection element the trajectory each vehicle has to follow is predetermined at the step 2-b among a set of predefined ones depending on the origin and the destination (i.e., the entering lane and the direction to take), in the case of a road element the trajectory is determined at the step 3 together with the optimal speed values. The reason of such a difference will be clarified in Chapter 6.

Such an optimization scheme is applied repetitively interval-by-interval every $\Delta\tau$ seconds. The speeds received by vehicles (at τ_k) which are still inside the intersection at τ_{k+1} represent constraints for the problem instance referred to the interval $[\tau_{k+1}, \tau_{k+2})$, as discussed in Sec. 5.4.

The parameter d can be set to $2\hat{v}\Delta\tau$, being $\hat{v}$ the average speed of vehicles arriving at the intersection, that is, d is the space traveled, as average, in $2\Delta\tau$ by vehicles. The parameters ϵ and $\Delta\tau$ can be suitably set in accordance with computational requirements, as discussed in the case study described in Sec. 7. However, ϵ must be lower than $\Delta_i / v_{\text{rd}}^{\max}$, that is, vehicles must receive the speeds from the intersection controller before entering the intersection. In case this condition is not satisfied, a backup strategy must be used: this can consist of using the best admissible solution found by the intersection controller within ϵ. If neither an admissible solution is found, to preserve safety only vehicles traveling along non conflicting trajectories can be allowed to cross the intersection during $[\tau_k, \tau_k + \Delta\tau)$. However, in all the conducted experiments these situations have never occurred and should therefore considered very rare.

3.2 Software architecture

This section provides some details about the algorithm that implements the proposed TMS (in the following, *Optimal CAVs Traffic Management System* algorithm or, shortly, OCAV-TMS algorithm).

Firstly, let define the following terms:

- time interval: the time horizon is divided into different intervals $\Delta_{\tau k} = [\tau_k, \tau_{k+1})$ of length $\Delta\tau$ as already discussed in Secs. 3.1.3 and 5.4;

- entering element: first element vehicles cross when entering the network from the external;

- scenario: collection of data (entering time, entering lane, speed, weight, direction) related to the vehicles entering a given element in a given time interval, which is taken as input by the local controller;

- scheduling and motion planning model: implementation of the optimization model defined in Sec. 5.3 for the intersection and in Sec. 6.3 for the road.
- element slice: variable indicating the number of times the scheduling and motion planning model of a given element has been executed;
- routing model: implementation of the optimization model defined in Sec. 4.2;
- solution: output of the scheduling and motion planning model containing the space-time trajectories of each vehicle;
- vehicles_out: variable containing all the vehicles exiting a given element during a given time interval. It can be computed from the solution;
- network topology: collection of information regarding the structure of the network and, in particular, how the various elements are connected together;
- network state: collection of the vehicle flows in each element of the network.

Let now consider Fig. 3.5, which shows the flow diagram representation of the macro steps of the OCAV-TMS algorithm. At each interval $\Delta\tau_k$, all the blocks inside the bigger dashed rectangle are executed. Firstly, the code checks for any vehicle entering the network from the external in $\Delta\tau_k$. If there are no entering vehicles, than the blocks inside the smaller dashed rectangle are directly executed for each element. If there are entering vehicles, then the routing model is stated and solved, with the entering vehicles, the time instant and their origins and destinations as input. The output, which consists of the optimal vehicles paths computed considering the network state information, is saved and stored at the network level and the blocks inside the smaller dashed rectangle are executed for each element.

Then, a scenario is created for the current element e, containing all the vehicles entering e in $\Delta\tau_k$. If the element is an entering one, then the vehicles are the ones coming from the external and entering the network at e; if the element is not an entering one, then the vehicles entering e are already in the network and enter e from the adjacent elements. These data is taken from the variable vehicles_out, by looking for all data related to the vehicles exiting the elements adjacent to e in $\Delta\tau_k$.

Once the scenario with the new entering vehicles is created, the vehicles that at $\Delta\tau_k$ are still on e (if there are any) are added, and the related constraints are written in the scheduling and motion planning model. This procedure is necessary since the vehicles do not necessarily exit the element in the same time interval they enter it. Thus, it is very likely that in each time interval there are vehicles still inside the element: since these vehicles have already received the trajectory and the optimal speeds to follow in a previous time interval, they must be considered constraints in the current one, and the problem only optimizes the speeds of the new vehicles entering the element. This procedure allows to consider relatively short time intervals, with the advantage of reducing the computational time needed to solve the problems, as better explained in section 5.4.

The last step is skipped if the element is executed for the first time (i.e., if the element slice is equal to 1), and thus there cannot be vehicles still in it. Once the scenario and the model are created, this latter is executed and the related

optimization problem is solved. The solution containing the trajectories and the optimal speeds is then saved and stored. Data related to the exiting vehicles is saved in the variable vehicles_out in the position related to the element e and the exiting time interval $\Delta\tau_k+n\Delta t$, being n the distance, in terms of number of intervals, between the entering time interval and the exiting time interval. The value of n is computed separately for each vehicle depending on the time instant at which it exits e. Finally, the slice counter of e is updated, together with the network state information.

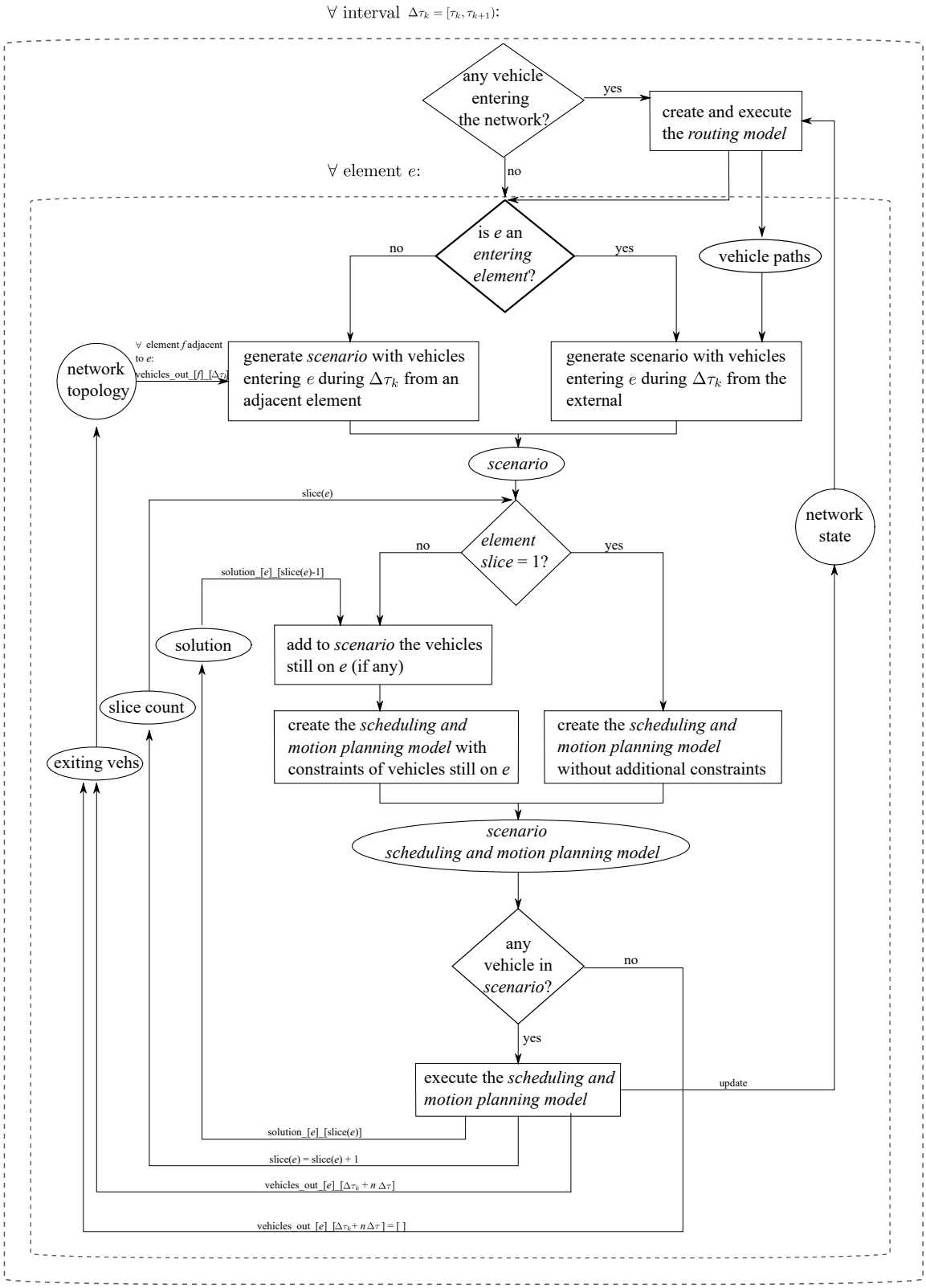


Figure 3.5: Flow diagram representation of the OCAV-TMS algorithm.

Chapter 4

The routing model

This chapter describes the routing model, which is aimed at determining the optimal path of the vehicles inside the network and is implemented and solved by the RC at each time interval $\Delta\tau_k$ if there are vehicles entering the network.

The optimality is defined on the basis of the minimization of the travel time along the network from the origin to the destination of each vehicle. To do so, the model takes as input the estimated travel times in the various elements of the network, which are assumed to be provided by the local controllers (see Sec. 4.3 for further details).

In addition, the following formalization assumes that the entering elements are always roads, that is, the first element vehicles cross once entered the network is always a road (and not an intersection). Obviously, the formalization may be easily adapted also to the case where entering elements are also intersections. Fig. 4.1 shows an example of network layout composed by 6 intersections, 10 origin/destination nodes and 34 roads. The origin/destination nodes are not modeled as intersections, but they are used only to model the entrance of the vehicles in the network (i.e., no scheduling and motion planning model is associated to them).

4.1 Notation

This section introduces the notation adopted in the routing model, together with the sets, the parameters, and the variables characterizing the model.

4.1.1 Sets

- $\mathcal{I}$ is the set of all the intersection elements that compose the network (for example, $\mathcal{I} = \{1, 2, 3, 4, 5, 6\}$ in the network shown in Fig. 4.1);
- $\mathcal{R}$ is the set of all the road elements that compose the network (for example, $\mathcal{R} = \{7, 8, \ldots, 40\}$ in the network shown in Fig. 4.1). Each road element is assumed here to be one-way and composed by two lanes;
- $\mathcal{I}(f)$ is the set of roads that enter into the intersection f (for example, $\mathcal{I}(1) = \{7, 9, 28, 30\}$ in the network shown in Fig. 4.1);

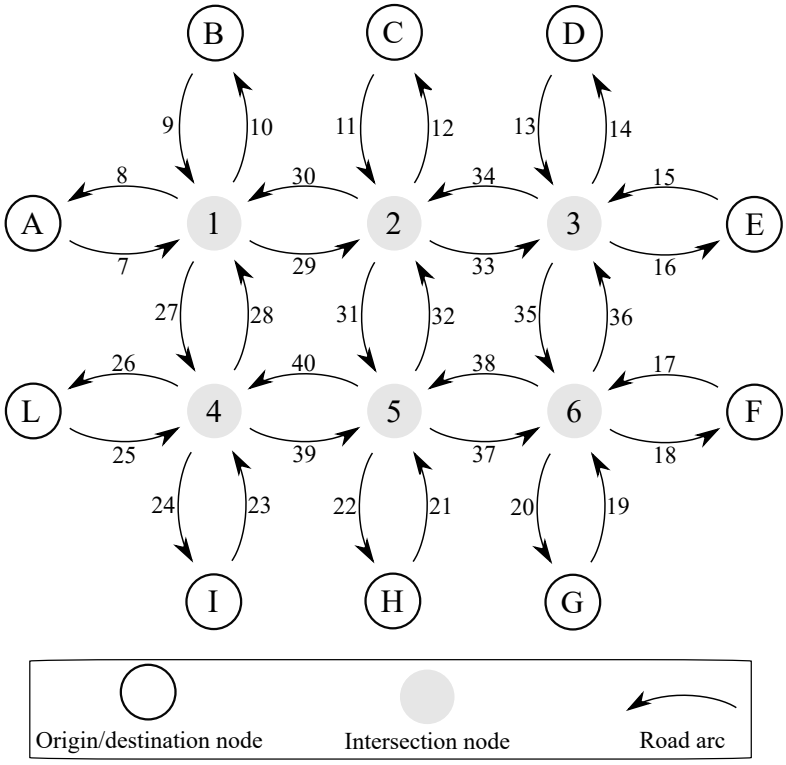


Figure 4.1: Graph representation of a network composed by intersections and roads. In this case the first element vehicles cross once in the network are the roads.

- $\mathcal{E}(f)$ is the set of roads that exit from the intersection f (for example, $\mathcal{E}(1) = \{8, 10, 27, 29\}$ in the network shown in Fig. 4.1);
- $\mathcal{RI}(f)$ is the set of ordered pairs $\{g, h\}$ of road elements adjacent to intersection f such that vehicles have to turn right (following any of the streams allowing this maneuver) at f in order to go from g to h;
- $\mathcal{SI}(f)$ is the set of ordered pairs $\{g, h\}$ of road elements adjacent to intersection f such that vehicles have to go straight (following any of the streams allowing this maneuver) at f in order to go from g to h;
- $\mathcal{LI}(f)$ is the set of ordered pairs $\{g, h\}$ of road elements adjacent to intersection f such that vehicles have to turn left (following any of the streams allowing this maneuver) at f in order to go from g to h;
- $\mathcal{UI}(f)$ is the set of ordered pairs $\{g, h\}$ of road elements adjacent to intersection f such that vehicles should have to make a U-turn at f in order to go from g to h;
- $\mathcal{V}$ is the set of $|\mathcal{V}|$ vehicles that are going to enter the network and must be routed;
- $\mathcal{H}_u$ is the set of prohibited roads for vehicle u;

4.1.2 Parameters

- w_u is a real-valued weight parameter representing the priority of vehicle u in the optimization problem;
- T_u^e is the expected travel time to cross road element e for vehicle u;
- $TT_{\mathrm{r},u}^f$ is the expected travel time to cross intersection element f for vehicle u in case it turns right at f;
- $TT_{\mathrm{s},u}^f$ is the expected travel time to cross intersection element f for vehicle u in case it goes straight at f;
- $TT_{\mathrm{l},u}^f$ is the expected travel time to cross intersection element f for vehicle u in case it turns left at f;
- o_u is the origin node of vehicle u;
- d_u is the destination node of vehicle u;
- E_{o_u} is the entering element connected to origin node o_u (i.e., the first element crossed by u after entering the network);
- F_{d_u} is the exiting element connected to destination node d_u (i.e., the last element crossed by u before exiting the network);
- Δ is the length of the buffer zone between each couple of elements;
- $v_{\mathrm{rd}}^{\max}$ is the maximum allowed speed in the road stretch (buffer zone) between two adjacent elements;

4.1.3 Variables

- σ_u^e is the estimated time instant at which vehicle u enters the element e (intersection or road);
- τ_u^e is the estimated time instant at which vehicle u exits the element e (intersection or road);
- $z_{u,1}^{e,c}$ is a Boolean variable that is set to 1 if vehicle u enters the road element e in lane c. Otherwise it is set to 0. $z_{u,1}^{e,c\star}$ is its optimal value;
- $z_{u,L}^{e,c}$ is a Boolean variable that is set to 1 if vehicle u exits the road element e in lane c. Otherwise it is set to 0. $z_{u,L}^{e,c\star}$ is its optimal value.

4.2 The routing MILP formulation

The routing optimization problem can be formalized as follows:

$$\mathbf{z}^\star = \arg\min_{\mathbf{z}} (\Theta + \zeta\Gamma) \tag{4.1}$$

being

$$\Theta = \sum_{u\in\mathcal{V}} w_u(\tau^{d_u} - \sigma_{u,1}^{o_u}) \tag{4.2}$$

the total weighted travel time and

$$\Gamma = \sum_{e\in\mathcal{R}} \sum_{f\in\mathcal{R}, f\neq e} \left| \sum_{u\in\mathcal{V}} \sum_{c\in\{0,1\}} \left(T_u^e z_{u,1}^{e,c} - T_u^f z_{u,1}^{f,c} \right) \right| \tag{4.3}$$

a second objective used to encourage vehicle to distribute equally on the available roads. In particular, Γ represents the total difference, for each pair (e, f) of road elements, between the number of vehicles traveling along e and the number of vehicles traveling along f, weighted by the travel time. In particular, for a given pair (e, f), if the travel time is equal for all the vehicles and the road elements e and f, then Γ is equal to 0 if the number of vehicles routed on e is equal to the number of vehicles routed on f. ζ is a weighting parameter.

The problem is subject to the following constraints:

$$\sum_{c\in\{0,1\}} z_{u,1}^{E_{o_u},c} = 1, \qquad \forall u \in \mathcal{V} \tag{4.4}$$

$$\sum_{c\in\{0,1\}} z_{u,L}^{F_{d_u},c} = 1, \qquad \forall u \in \mathcal{V} \tag{4.5}$$

Constraints (4.4) and (4.5) define, respectively, the entering and the exiting road element, ensuring that each vehicle u crosses them. It is worth pointing out that this kind of constraints does not limit the specific lane from which the vehicle enters and the one from which the vehicle exits the network.

$$\sum_{c\in\{0,1\}} z_{u,1}^{e,c} \leq 1, \qquad \forall u \in \mathcal{V}, \forall e \in \mathcal{R} \tag{4.6}$$

$$\sum_{c\in\{0,1\}} z_{u,L}^{e,c} \leq 1, \qquad \forall u \in \mathcal{V}, \forall e \in \mathcal{R} \tag{4.7}$$

Constraints (4.6) and (4.7) ensure that each vehicle enters and exits a road element from no more that one lane.

$$\sum_{c\in\{0,1\}} z_{u,1}^{e,c} = \sum_{c\in\{0,1\}} z_{u,L}^{e,c}, \qquad \forall u \in \mathcal{V}, \forall e \in \mathcal{R} \tag{4.8}$$

$$\sum_{e\in\mathcal{I}(f)} z_{u,L}^{e,c} = \sum_{e\in\mathcal{E}(f)} z_{u,1}^{e,c}, \qquad \forall u \in \mathcal{V}, \forall f \in \mathcal{I}, \forall c \in \{0,1\} \tag{4.9}$$

Constraints (4.8) and (4.9) are conservation equations for, respectively, the roads and the intersections. In particular, constraint (4.8) ensure that a vehicle exits a road

element if it entered it. Constraint (4.9) does the same for the intersection element: the left term in the equation is equal to 1 if vehicle u enters the intersection f (from any of the adjacent entering roads $\mathcal{I}(f)$), otherwise it is equal to 0. On the other hand, the right term is equal to 1 if vehicle u exits the intersection f (from any of the adjacent exiting roads $\mathcal{E}(f)$), otherwise it is equal to 0.

$$\sum_{e \in \mathcal{I}(f)} z_{u,L}^{e,c} \leq 1, \qquad \forall u \in \mathcal{V}, \forall f \in \mathcal{I}, \forall c \in \{0,1\} \tag{4.10}$$

Constraint (4.10) is used in order to ensure that a vehicle arrives at an intersection only from one road element.

$$\sum_{e \in \mathcal{I}(f)} z_{u,1}^{e,c} = 0, \qquad \forall u \in \mathcal{V}, \forall e \in \mathcal{H}_u \tag{4.11}$$

Constraint (4.11) is used to prevent vehicles from running a set of prohibited roads.

$$\tau_u^e - \sigma_u^e \geq T_u^e - M(1 - \sum_{c \in \{0,1\}} z_{u,1}^{e,c}), \qquad \forall u \in \mathcal{V}, \forall e \in \mathcal{R} \tag{4.12}$$

Constraint (4.12) defines the minimum travel time in the road elements: if the vehicle u crosses the road e, then $z_{u,1}^{e,c} = 1$ and so the running time must be greater or equal to the expected travel time T_u^e.

$$\tau_u^f - \sigma_u^f \geq TT_{\mathrm{r},u}^f - M(1 - \sum_{c \in \{0,1\}} z_{u,1}^{g,c}) - M(1 - \sum_{c \in \{0,1\}} z_{u,1}^{h,c}), \\ \forall u \in \mathcal{V}, \forall f \in \mathcal{I}, \forall \{g,h\} \in \mathcal{RI}(f) \tag{4.13}$$

Constraint (4.13) defines the minimum travel time in the intersections for the right-turning maneuver. In particular, the constraint is activated when both the terms $\sum_{c \in \{0,1\}} z_{u,1}^{g,c}$ and $\sum_{c \in \{0,1\}} z_{u,1}^{h,c}$ are equal to 1. This condition corresponds to the case vehicle u crosses the intersection, entering from any of the adjacent entering roads, and turning right at it.

$$\tau_u^f - \sigma_u^f \geq TT_{\mathrm{s},u}^f - M(1 - \sum_{c \in \{0,1\}} z_{u,1}^{g,c}) - M(1 - \sum_{c \in \{0,1\}} z_{u,1}^{h,c}), \\ \forall u \in \mathcal{V}, \forall f \in \mathcal{I}, \forall \{g,h\} \in \mathcal{SI}(f) \tag{4.14}$$

$$\tau_u^f - \sigma_u^f \geq TT_{\mathrm{l},u}^f - M(1 - \sum_{c \in \{0,1\}} z_{u,1}^{g,c}) - M(1 - \sum_{c \in \{0,1\}} z_{u,1}^{h,c}), \\ \forall u \in \mathcal{V}, \forall f \in \mathcal{I}, \forall \{g,h\} \in \mathcal{LI}(f) \tag{4.15}$$

Similarly to constraint (4.13), constraints (4.14) and (4.15) define the minimum travel time in the intersections for, respectively, the straight and the right-turning maneuver.

$$\tau_u^f - \sigma_u^f \geq 0.5M - M(1 - \sum_{c\in\{0,1\}} z_{u,1}^{g,c}) - M(1 - \sum_{c\in\{0,1\}} z_{u,1}^{h,c}),$$
$$\forall u \in \mathcal{V}, \forall f \in \mathcal{I}, \forall \{g,h\} \in \mathcal{UI}(f) \quad (4.16)$$

Constraint (4.16) prevents vehicles from doing a U-turn maneuver at the intersections, by assigning a very large travel time to cross the intersection in case a vehicle exit the intersection from the same direction of entrance.

$$\sigma_u^f - \tau_u^e \geq \frac{\Delta}{v_{rd}^{max}} - M(1 - \sum_{c\in\{0,1\}} z_{u,L}^{e,c}), \qquad \forall u \in \mathcal{V}, \forall f \in \mathcal{I}, \forall e \in \mathcal{I}(f) \qquad (4.17)$$

$$\sigma_u^g - \tau_u^f \geq \frac{\Delta}{v_{rd}^{max}} - M(1 - \sum_{c\in\{0,1\}} z_{u,1}^{g,c}), \qquad \forall u \in \mathcal{V}, \forall f \in \mathcal{I}, \forall g \in \mathcal{E}(f) \qquad (4.18)$$

Constraints (4.17) and (4.18) define the minimum travel time of the buffer zones between, respectively, a road and an intersection and an intersection and a road.

$$z_{u,1}^{e,c} \in \{0,1\}, z_{u,L}^{e,c} \in \{0,1\}, \qquad \forall u \in \mathcal{V}, \forall e \in \mathcal{R}, \forall c \in \{0,1\} \qquad (4.19)$$

$$\sigma_u^e \in \mathbb{R}_{\geq 0}, \tau_u^e \in \mathbb{R}_{\geq 0}, \qquad \forall u \in \mathcal{V}, \forall e \in \mathcal{R} \cup \mathcal{I} \qquad (4.20)$$

Finally, constraints (4.19) and (4.20) define the Boolean and non-negative real variables of the problem.

As Equ. (4.2) shows, the output of the routing model are the optimal values of the variables $z_{u,1}^{e,c}$ and $z_{u,L}^{e,c}$. Once they are known, the path of each vehicle u in the network (i.e., the sequence of elements it should cross to reach its destination in the minimum time) can be easily determined, together with the destination to take at each intersection. In addition, the solution provides also the lane at which u should enter or exit each road element.

4.3 Evaluating the network state and the expected travel times

A key element of the routing model defined in the previous section is the knowledge of the expected travel times of the vehicles in each element of the network, which are used to determine the optimal paths of the new vehicles entering the network. This section describes how they are computed once the network state is known, and how the network state is evaluated.

The state of the network $SN(\Delta\tau_k)$ at the time interval $\Delta\tau_k$ is assumed to be represented by the average vehicle flows in each element of the network at such time interval. This value is always known by the network controller of the TMS, and it is updated at each iteration of the OCAV-TMS algorithm considering the information

of the various LCs. Besides, at each $\Delta\tau_k$, once each new set of vehicles entering the network is routed, it is possible to exploit the determined optimal paths to make a prediction of the network state in the future time intervals, in the following 'expected network state' $ES(\Delta\tau_k, \Delta\tau_h)$ at $\Delta\tau_k$ for the interval $\Delta\tau_h$. The expected network state represents the minimum expected vehicle flows in each element of the network. It is worth pointing out that it represents a collection of *minimum* vehicle flows, updated at each time interval, because some vehicles may enter the network from the external after the interval $\Delta\tau_k$ (i.e., after the prediction is made) and reach element e in $\Delta\tau_h$.

More in detail, for each new routed vehicle u and for each element e crossed in its optimal path, the expected network state ES is updated according to the following procedure:

1. Find the expected time interval $\Delta\tau_h$ in which the vehicle is expected to reach e. This can be done by considering the expected times σ_u^e and τ_u^e, which come with the solution of the routing problem;

2. Update the expected network state $ES(\Delta\tau_k, \Delta\tau_h)$.

Then, the expected average flow $q_e(\Delta\tau_h)$ on a generic element e in $\Delta\tau_h$ is computed in the following way:

$$q_e(\Delta\tau_h) = \max\{SN_e(\Delta\tau_k); ES_e(\Delta\tau_k, \Delta\tau_h)\} \tag{4.21}$$

that is, the expected flow in time interval $\Delta\tau_h$ is the maximum value between the current flow and the minimum expected one.

Once known the expected flow on a given element, the expected travel times can be computed by considering a series of flow-travel time functions $f, f : [0, f_{\max}] \to \mathbb{R}, f \in C^0([0, f_{\max}])$, empirically derived from multiple executions, with different flows, of the intersection model and of the road model applied to the actual layouts of the network elements.

These functions must meet the following criteria:

- They must be defined and continuous over the entire interval $[0, f_{\max}]$, where $f_{\max}$ is the maximum admissible flow on the element. This is because f must provide a finite value of travel time for each flow value in the interval;

- They must be increasing functions, because the travel time is expected to increase with the flow;

- They must provide the minimum travel time in the case of a null flow, that is, $f(0) = t_{\min}$;

- They must represent a good approximation of the observed values.

As discussed in Secs. 7.2.3 and 7.3.3, in this work these functions have been estimated by a second-order polynomial fitting on multiple realizations of couples of flow-travel time values. In addition, in the case of the intersection element they have been estimated for each of the six typologies of trajectories (the four curved

trajectories of Fig. 5.3 plus the two ones going straight, one from the left lane and one from the right lane), since streams of different type differ per geometry, length and interferences with other streams. In the case of the road element a single flow/travel time function has been estimated, since in the case of the road the trajectories are geometrically more similar and do not significantly differ from each other.

In conclusion seven flow/travel time functions have been estimated:

- the flow-travel time road function fr;
- the flow-travel time intersection function of type 1 fi_1 (for the trajectories 1,7,14 and 21 shown in Fig. 5.1);
- the flow-travel time intersection function of type 2 fi_2 (for the trajectories 2,8,15 and 19 shown in Fig. 5.1);
- the flow-travel time intersection function of type 3 fi_3 (for the trajectories 3,6,13 and 20 shown in Fig. 5.1);
- the flow-travel time intersection function of type 4 fi_4 (for the trajectories 22,4,11 and 18 shown in Fig. 5.1);
- the flow-travel time intersection function of type 5 fi_5 (for the trajectories 23,5,12 and 16 shown in Fig. 5.1);
- the flow-travel time intersection function of type 6 fi_6 (for the trajectories 24,6,10 and 17 shown in Fig. 5.1);

Once they are known, it is possible to compute the values of the variables $TT^f_{\mathrm{r},u}$, $TT^f_{\mathrm{s},u}$, $TT^f_{\mathrm{l},u}$ and T^e_u for each vehicle u according to the following equations:

$$TT^f_{\mathrm{r},u} = \frac{fi_1(q_f(\Delta\tau_h)) + fi_4(q_f(\Delta\tau_h))}{2} \tag{4.22}$$

for the right turn maneuver at the intersections,

$$TT^f_{\mathrm{l},u} = \frac{fi_3(q_f(\Delta\tau_h)) + fi_6(q_f(\Delta\tau_h))}{2} \tag{4.23}$$

for the left turn maneuver at the intersections, and

$$TT^f_{\mathrm{s},u} = \frac{fi_2(q_f(\Delta\tau_h)) + fi_5(q_f(\Delta\tau_h))}{2} \tag{4.24}$$

for the go straight maneuver at the intersections. Finally,

$$T^e_u = fr(q_e(\Delta\tau_h)) \tag{4.25}$$

for the roads.

In Equs. (4.22)-(4.25), $q_f(\Delta\tau_h)$ is the expected flow on intersection f in $\Delta\tau_h$, and $\Delta\tau_h$ is the interval in which u is expected to reach f. For the intersections the travel time values for a given maneuver are assumed to be the mean between the travel times for the two stream typologies allowing such maneuver. This is because when solving the routing problem, the routing controller does not know yet which trajectory (between the two possible ones) will be followed.

Chapter 5

The isolated intersection model

This chapter describes the intersection model, which is aimed at determining the optimal speed profiles of the vehicles entering an intersection element.

5.1 Definitions, Assumptions and Notation

This section provides the definitions and the assumptions of the model, using the notation described in Sec. 5.1.3 and summarized in Tab. 5.1.

5.1.1 Definitions

In this thesis, some basic general assumptions are considered. To this aim, let:

- γ_i be the locus of admissible positions on the intersection plane for the barycenter of vehicles in the same stream i, hereafter addressed as *trajectory*. Each point $P \in \gamma_i$ is in one-to-one correspondence with its distance $\ell(P)$ from the the initial point of γ_i ($P \Leftrightarrow \ell(P)$, see Appendix 5.6);
- $\mathcal{N} = \{(i,j) | \gamma_i \cap \gamma_j = \emptyset, i \neq j\}$ be the set of indexes of *non-conflicting trajectory pairs*, that is, the trajectories which do not cross each other (e.g., γ_3 and γ_{18} in Fig. 5.1);
- $\mathcal{C} = \{(i,j) | (i,j) \notin \mathcal{N}, P_{i,1} \neq P_{j,1}, i \neq j\}$ be the set of indexes of *conflicting trajectory pairs*, that is, the trajectories which cross each other only in some points (e.g., γ_3 and γ_6 in Fig. 5.1);
- $\mathcal{O} = \{(i,j) | (i,j) \notin \mathcal{N}, P_{i,1} = P_{j,1}, i \neq j\}$ be the set of indexes of *partially overlapped trajectory pairs*, that is, the trajectories which are initially overlapped (e.g., γ_2 and γ_3 in Fig. 5.1).

5.1.2 Assumptions

In the model presented in this thesis it is assumed that:

1. the considered intersection is characterized by a set of trajectories γ_i, $i \in \mathcal{S}$, followed by streams of vehicles. Such trajectories, which depend only on the

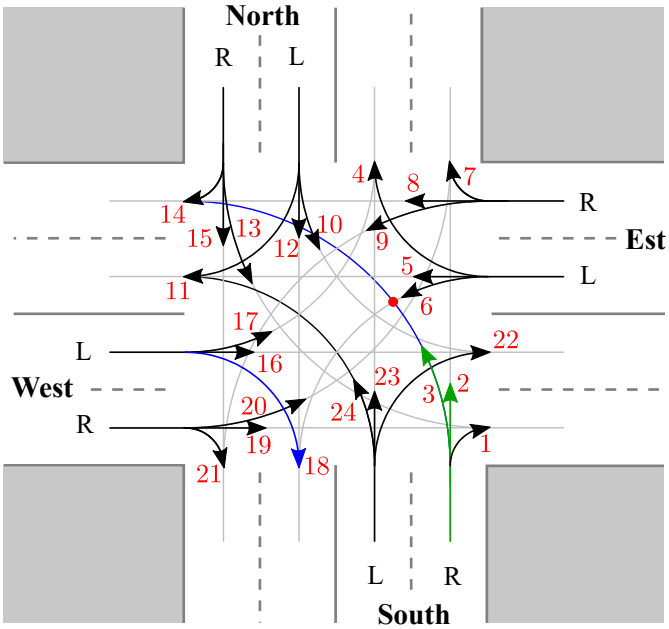


Figure 5.1: General sketch of an intersection with 24 admissible streams indicated by arrows. The gray dashed lines represent the related trajectories, the blue trajectories exemplify two non-conflicting pair ($\{3, 18\} \in \mathcal{N}$), the red dot identifies an example of conflict ($\{3, 6\} \in \mathcal{C}$), and the green ones exemplify two partially overlapped trajectories ($\{2, 3\} \in \mathcal{O}$).

intersection layout, lay in the intersection plane and are analytically predetermined as described in Appendix 5.6. An example is depicted in Fig. 5.1, which shows a generic 4-ways 2-lanes intersection and the predefined trajectories associated to the 24 possible streams. It is worth nothing that each trajectory is in a one-to-one relation with a vehicle stream;

2. vehicles are not allowed to change stream or to overtake the others in the same stream; therefore, vehicles are sorted in each stream;

3. vehicles are considered with their real space occupancy and it is possible to identify off-line the set of reciprocal incompatible positions, as described in Sec. 6.2.3;

4. when a vehicle u crosses a trigger located at a distance Δ_i from the intersection it sends the crossing time $s_{i,u,0}$ to the intersection controller. A sketch of the trigger is depicted in Fig. 5.2;

5. the speed limits along the incoming road stretches guarantee that all the vehicles can arrange their speed to reach the first point of the trajectory γ_i at the time instant determined by the optimization problem.

5.1.3 Notation

This section describes the notation used in the proposed model. The notation is also summarized in Tab. 5.1. Thanks to the generality of the model and for the

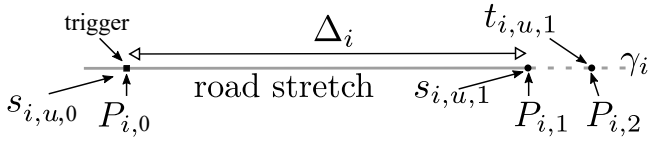


Figure 5.2: Trigger located at a distance Δ_i from the beginning of the trajectory γ_i in $P_{i,1}$.

sake of compactness, in the following the element index e is dropped.

Structures and sets

- $\mathcal{S}$ is the set of the streams crossing the intersection.
- $\mathcal{V}_i$ is the set of the vehicles on the stream i. Such a set is sorted, so the direct follower of the vehicle v is $(v+1)$
- γ_i, $\forall i \in \mathcal{S}$, is the trajectory followed by vehicles in the stream i (see Section 5.6).
- $\Gamma = \{\gamma_i | \forall i \in \mathcal{S}\}$ is the set of all trajectories.
- $P_{i,0}$ is the position of the trigger related to the trajectory γ_i.
- $P_{i,1}$ is the first point of the trajectory γ_i that is considered in the optimization problem, i.e., it is the first point at which the vehicles have the optimal value of speed to follow.
- $P_{i,h}$ is a generic point of the trajectory γ_i, $i \in \mathcal{S}$.
- $\mathcal{P}_i = \{P_{i,h}\}$ is the set of n_i points along the trajectory γ_i.
- $P_{i,h} \nleftrightarrow P_{j,k}$ means that the points $P_{i,h}$ and $P_{j,k}$ are incompatible, that is, they cannot be occupied, at the same time, by two vehicles because this situation may cause a collision. The concept of incompatibility between points and trajectories is better discussed in Sec. 6.2.3.
- $a_{i,h}$ is a generic arc of the trajectory γ_i between two consecutive points $P_{i,h}$ and $P_{i,h+1}$.
- δ is the fixed length of the arc $a_{i,h}$.
- $\ell(P_{i,h})$ is the distance of $P_{i,h}$ from $P_{i,1}$.

Parameters

- $v_{\text{rd}}^{\min}$ and $v_{\text{rd}}^{\max}$ are the minimum and the maximum admissible vehicle speeds in the road stretches between the triggers and the intersection.
- $v_{\text{int}}^{\max}$ is the maximum admissible vehicle speed inside the intersection.
- $\Phi_{\min} = \delta / v_{\text{int}}^{\max}$ is the minimum travel time on a generic arc of length δ.

- Φ_l and Φ_h are the maximum allowed relative vehicle travel time variations when the speed is low and high, respectively.
- $r_{i,h}$ is a coefficient, hereafter referred to as 'speed reduction factor', which is determined off-line and that reduces the maximum allowed speed on the arc $a_{i,h}$ (see Sec. 6.2.2) to guarantee the stability and the comfort on board along curves.
- w_u is the weight of vehicle u in the scheduling and motion planning model. It can be used to prioritize some vehicles, like emergency vehicles or public transport vehicles.
- α is a weighting coefficient in the objective function of the MILP problem.
- β is a safety parameter that increases the minimum permitted distance between vehicles.

Variables

- $s_{i,v,0}$ is the time instant at which vehicle v crosses the trigger and enters the road stretch between the trigger and the intersection.
- $s_{i,v,h}$ is a real variable indicating the time instant at which the vehicle v enters the arc $a_{i,h}$ of the trajectory γ_i.
- $t_{i,v,h}$ is a real variable indicating the time instant at which the vehicle v exits the arc $a_{i,h}$ of the trajectory γ_i.
- $x_{i,v,j,u}$ is a Boolean variable that is set to 1 if the vehicle v in the stream i has right of way with respect to the vehicle u in the stream j. Otherwise it is set to 0.
- $s^\star_{i,v,h}$, $t^\star_{i,v,h}$, $x^\star_{i,v,j,u}$ are the optimal values of the variables $s_{i,v,h}$, $t_{i,v,h}$, $x_{i,v,j,u}$.

5.2 Modeling and optimization approach

This section introduces and discusses three fundamental aspects of the proposed approach: the spatial discretization of trajectories, the speed reduction along curves, and the incompatibility constraints, which guarantee a safe and comfortable crossing of the intersection.

5.2.1 The spatial discretization

The considered optimization problem aims at minimizing the total travel time of a given set of vehicles in a given time interval, while guaranteeing vehicle safety. Such a problem is intrinsically a continuous-time optimization problem, as the state of each vehicle v is represented by differential state equations providing its position and speed for any time instant τ. In this thesis, such a problem is reformulated as a Mixed-Integer Linear Programming (MILP) problem, where trajectories have been

discretized into *sequences of arcs* of the same length δ. The search of a suitable discretization is an off-line problem to solve once per each intersection, the choice of $\delta = 0.5$ m has resulted to be sufficient for the intersection considered in this thesis.

Thanks to the spatial discretization, it is possible to introduce the sorted sequences $\mathcal{P}_i = \{P_{i,h}, h = 1, \ldots, n_i\}$, $\forall i \in \mathcal{S}$ of points of the trajectory γ_i such that $\ell(P_{i,h+1}) > \ell(P_{i,h})$ and $\ell(P_{i,h+1}) - \ell(P_{i,h}) = \delta$, being $\ell(P_{i,h}) = \ell(P)|_{P=P_{i,h}} = \delta \cdot (h-1)$ and $\ell(P)$ the function distance of P from $P_{i,1}$ (see Appendix 5.6). Thanks to this definition, $\ell(P_{i,h}) \Leftrightarrow P_{i,h}$, and n_i depends on the total length of trajectories and on the discretization parameter δ. It is assumed for simplicity that trajectories are adjusted so that $\ell_{i,n_i} \bmod \delta = 0$, $\forall \gamma_i \in \Gamma$.

5.2.2 Curved trajectories and speed limits

In this section, the maximum admissible speed of vehicles in any point of arcs is discussed with reference to its dependance on the road characteristics and, in particular, on the geometry of the trajectories. In fact, given the maximum admissible speed along a straight trajectory inside the intersection, namely $v_{\text{int}}^{\max}$, a 'local' reduction value must be determined, for each point $P_{i,h} \in \gamma_i$, as function of the trajectory curvature in $P_{i,h}$. To formalize such a requirement, a speed reduction factor $r(P)$ is first determined as a function of the point P, as described in Appendix 5.6. Then, a specific value is associated to each arc $a_{i,h}$ by choosing the minimum along the arc, that is $r_{i,h} = \min_{P \in a_{i,h}} r(P)$. Obviously, all the arcs corresponding to straight trajectory stretches have $r_{i,h} = 1$ (i.e., no speed reduction required).

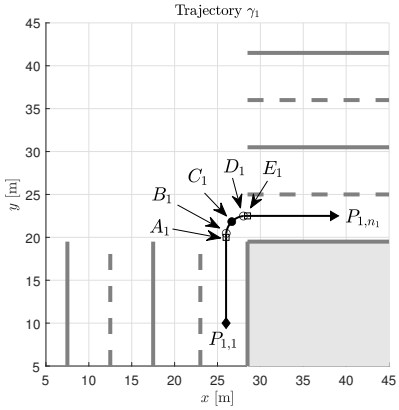
The functions $r(P)$ can be determined for any possible trajectory of an intersection. In this thesis, they have been determined for the four trajectories in Fig. 5.3 which are congruent to the other curved trajectories in the intersection in Fig. 5.1 (i.e., all the curved trajectories can be obtained by those in Fig. 5.3 via rotation and translation). All the trajectories in Fig. 5.3 are made up by:

1. a line from $P_{i,1}$ to A_i (infinite curvature radius);
2. a clothoid from A_i to B_i (variable curvature radius from infinite to a finite, non-null, value);
3. a circular segment from B_i to D_i (finite, non null, curvature radius);
4. a clothoid from D_i to E_i (variable curvature radius from a finite, non null, value to infinite);
5. a line from E_i to P_{i,n_i} (infinite curvature radius).

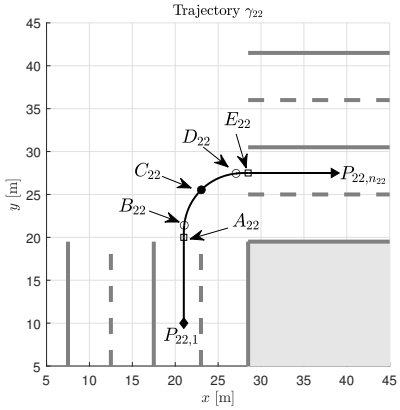
Therefore, $r(P)$ is equal to 1 (no speed reduction) for the initial and final parts, decreases from 1 to a minimum value in the clothoids, and has a constant minimum value for the circular segments, as the higher the curvature, the lower the value of $r(P)$.

A sketch of the reduction factors for the curves in Fig. 5.3 is depicted in Fig. 5.4 where, for the sake of clearness, the points A_i, B_i, C_i, D_i, and E_i are indicated.

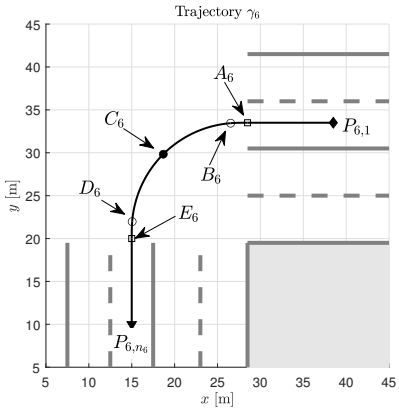
Note that the considerations described in this section are general since any trajectory along a real world road can be described as a sequence of lines, clothoids, and circular segments.



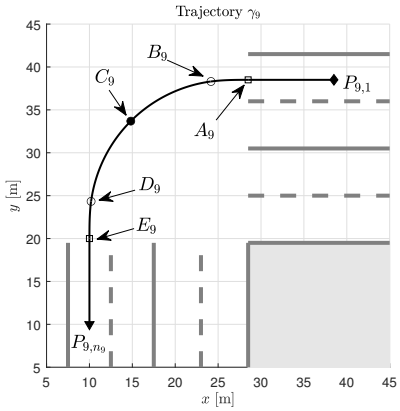
(a) Trajectory γ_1 (congruent to γ_7, γ_{14}, and γ_{21}).



(b) Trajectory γ_{22} (congruent to γ_4, γ_{11}, and γ_{18}).



(c) Trajectory γ_6 (congruent to γ_{10}, γ_{17}, and γ_{23}).



(d) Trajectory γ_9 (congruent to γ_3, γ_{13}, and γ_{20}).

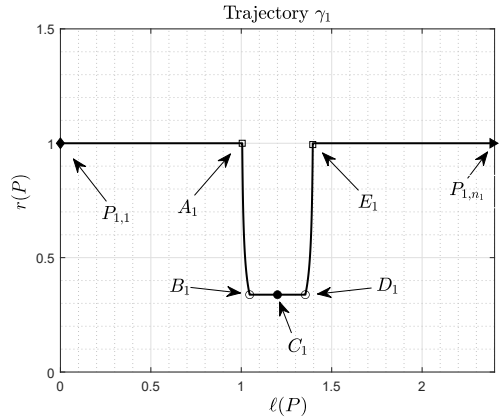
Figure 5.3: Representation of the four types of curve trajectories that are in the intersection shown in Fig. 5.1. The points A_i, B_i, C_i, D_i, and E_i correspond to those in Fig. 5.4.

5.2.3 Incompatibility constraints and conflicts management

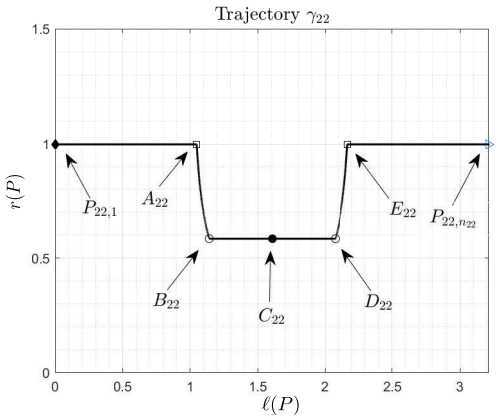
The vehicle collisions avoidance is ensured by MILP constraints preventing vehicles from reaching, at the same time, two incompatible points. This section describes in detail the concept of 'incompatibility' and discusses how the pairs of incompatible nodes are identified and used in the MILP.

First, let define the space occupancy of each vehicle in any point $P_{i,h} \in \mathcal{P}_i$, $\forall i \in \mathcal{S}$, as the space identified by a rectangular shape whose longer side is parallel to the unit tangent vector $T_{i,h}$ in $P_{i,h}$, assuming that the barycenter of the vehicle is in this point. An example of these *occupancies* is depicted in Fig. 5.5 where the empty rectangles indicate the sequence of the consecutive occupancies of a vehicle traveling along γ_i, having assumed the previously discussed discretization. Each point P belonging to the arc $a_{i,h}$ is associated to a shape coherently positioned among those associated to the points $P_{i,h}$ and $P_{i,h+1}$.

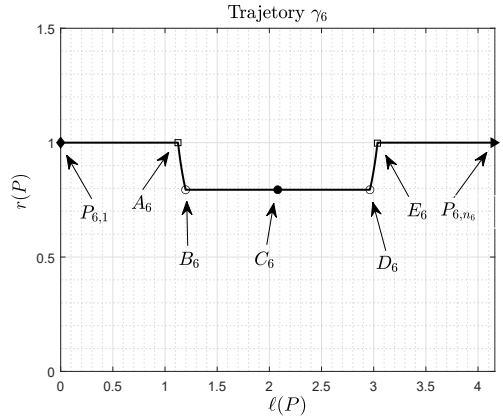
Then, two generic points P and Q are incompatible (i.e., there cannot be two



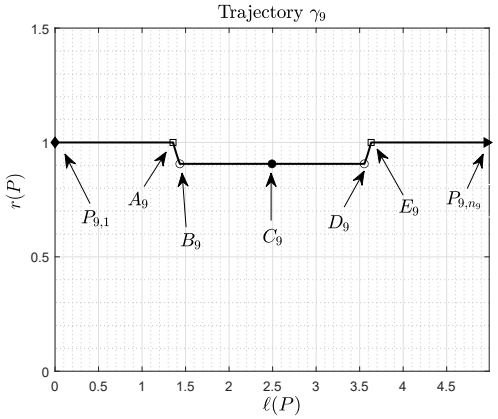
(a) Speed reduction factor along γ_1 (congruent to γ_7, γ_{14}, and γ_{21}).



(b) Speed reduction factor along γ_{22} (congruent to γ_4, γ_{11}, and γ_{18}).



(c) Speed reduction factor along γ_6 (congruent to γ_{10}, γ_{17}, and γ_{23}).



(d) Speed reduction factor along γ_9 (congruent to γ_3, γ_{13}, and γ_{20}).

Figure 5.4: Speed reduction factors of the trajectories depicted in Fig. 5.3: the speed is reduced more and more as the curvature of the trajectories increases . The points A_i, B_i, C_i, D_i, and E_i correspond to those in Fig. 5.3.

vehicles on P and Q at the same time), in short $P \nleftrightarrow Q$, if the relevant occupancies overlap.

The above definition is extended to the points $P_{i,h} \in \mathcal{P}_i, \forall i \in \mathcal{S}$, in the following way: two points $P_{i,h}$ and $P_{j,k}$ are incompatible, in short $P_{i,h} \nleftrightarrow P_{j,k}$, if $\exists P \in a_{i,h}, Q \in a_{j,k} | P \nleftrightarrow Q$. This definition is conservative, since, given a generic point $P \in a_{i,h}$ incompatible with a point Q, it considers incompatible with Q all the points of the arc $a_{i,h}$ before and after P. Moreover, this definition also allows to write safety constraints only with references to the initial points $P_{i,h}$ of arcs and it is not too restrictive thanks to the small value of δ.

Given the above definitions, it is possible to distinguish between two cases: points belonging to the same trajectory and points belonging to two different trajectories.

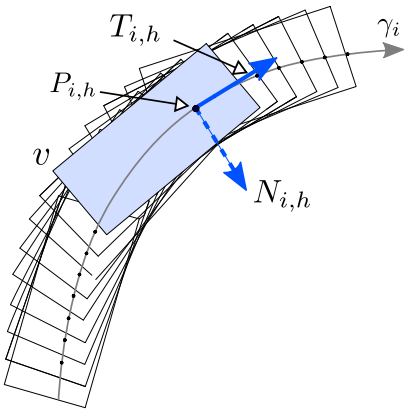


Figure 5.5: A generic vehicle v traveling along a curved trajectory γ_i. The continuous and dashed arrows represent, respectively, the tangent vector $T_{i,h}$ and the normal vector $N_{i,h}$ in $P_{i,h} \in \gamma_i$.

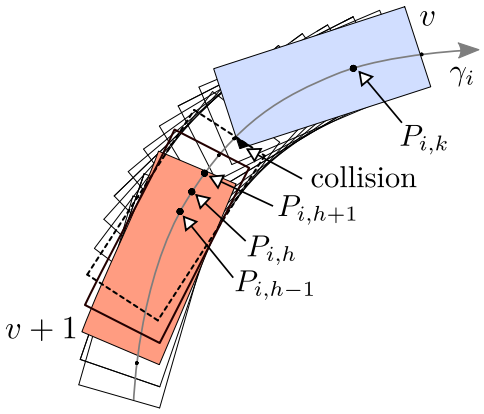


Figure 5.6: Example of incompatibility between two points of the same trajectory $P_{i,h}$ and $P_{i,k}$: the follower vehicle $(v+1)$ can reach $P_{i,h-1}$ but not $P_{i,h}$ while the leader vehicle v is on the arc $a_{i,k}$, otherwise a collision may occur. Therefore, $(v+1)$ must be delayed until v enters the arc $a_{i,k+1}$.

Incompatibility of points belonging to the same trajectory

Every pair of incompatible points $P_{i,h} \nleftrightarrow P_{i,k}$ belonging to the same trajectory γ_i must be associated with a constraint that guarantees the safe spacing between a leader vehicle v and a follower vehicle $(v+1)$. An example is shown in Fig. 5.6, where the occupancies of the vehicles v and $(v+1)$ are depicted with colored rectangles positioned in $P_{i,k}$ and $P_{i,h-1}$, respectively, and the positions of $(v+1)$ in $P_{i,h}$ and $P_{i,h+1}$ are represented with continuous and dashed rectangles, respectively. In such a case, $P_{i,h}$ and $P_{i,k}$ are incompatible since, by traveling along the arc $a_{i,h}$, the vehicle $v+1$ may collide with vehicle v, which is traveling on the arc $a_{i,k}$. Therefore, since vehicles cannot stop, the follower cannot even reach the point $P_{i,h}$, and has to be slowed down before reaching it.

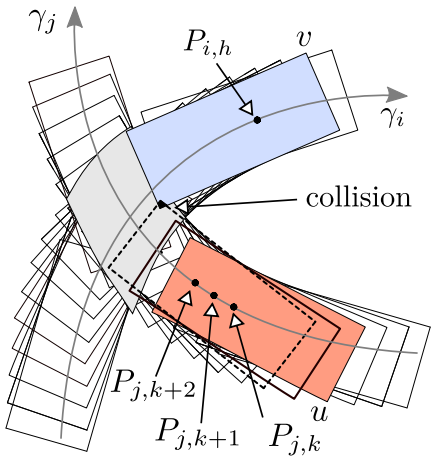


Figure 5.7: Example of incompatibility between points of two conflicting trajectories. The colored shapes indicate the positions of the vehicles u and v on $P_{j,k}$ and $P_{i,h}$: since $P_{j,k+1}$ and $P_{i,h}$ are not compatible, u can reach the point $P_{j,k}$ and travel along the arc $a_{j,k}$, but not along arc $a_{j,k+1}$.

Incompatibility of points belonging to different trajectories

Every pair of incompatible points $P_{i,h} \nleftrightarrow P_{j,k}$ belonging to the different trajectories γ_i and γ_j must be associated with a set of constraints that guarantees the assignment of the right of way to one and only one of the vehicles in each pair of vehicles v and u traveling along γ_i and γ_j, respectively. An example of this second type of conflict is depicted in Fig. 5.7 where the colored shapes indicate the positions of the vehicles on $P_{j,k}$ and $P_{i,h}$. In this specific frame, the vehicle v obtained right of way and is on the arc $a_{i,h}$; the vehicle u can travel along the arc $a_{j,k}$ but not along $a_{j,k+1}$ while v is on the arc $a_{i,h}$; therefore u must be slowed down before reaching the point $P_{j,k+1}$.

Incompatibility sets

In order to guarantee the safe crossing of vehicles, a key element of the proposed approach consists of the identification of the sets of incompatible points

$$\mathcal{I}^{\mathcal{S}} = \{\{P_{i,h}, P_{i,k}\} | P_{i,h} \nleftrightarrow P_{i,k}, P_{i,h}, P_{i,k} \in \mathcal{P}_i, i \in \mathcal{S}\} \tag{5.1}$$

$$\mathcal{I}^{\mathcal{C}} = \{\{P_{i,h}, P_{j,k}\} | P_{i,h} \nleftrightarrow P_{j,k}, P_{i,h} \in \mathcal{P}_i, P_{j,k} \in \mathcal{P}_j, (i,j) \in \mathcal{C}, j > i\} \tag{5.2}$$

$$\mathcal{I}^{\mathcal{O}} = \{\{P_{i,h}, P_{j,k}\} | P_{i,h} \nleftrightarrow P_{j,k}, P_{i,h} \in \mathcal{P}_i, P_{j,k} \in \mathcal{P}_j, (i,j) \in \mathcal{O}, j > i\} \tag{5.3}$$

which gather, respectively, the pairs on incompatible points in the same stream, in conflicting streams (e.g., streams 3 and 6 in Fig. 5.1), and in partially overlapped streams (e.g., streams 2 and 3 in Fig. 5.1). The condition $j > i$ avoids to duplicate the incompatibility pairs.

To conclude, it is worth pointing out that the incompatibilities of trajectories are analyzed off-line once to generate the sets $\mathcal{I}^{\mathcal{S}}$, $\mathcal{I}^{\mathcal{C}}$, and $\mathcal{I}^{\mathcal{O}}$. A simple algorithm for determining these sets consists of the comparison of all the occupancy rectangles

and has a computational complexity proportional to $(|\mathcal{S}| \cdot (\max_{i\in\mathcal{S}}\{n_i\})^2$, that is proportional to the squared total number of points of the longest trajectory. Note also that these definitions can be used to determine the incompatibility sets for any possible combination of vehicle classes by considering rectangles of the proper sizes.

5.3 The scheduling and motion planning model

In this section, the problem formulation is provided with the proposed heuristic solution approach, and a discussion about the analytical characteristics of the optimal solution.

The considered problem is formulated by means of the real variables $s_{i,v,h}$ and $t_{i,v,h}$ which represent the instants at which the generic vehicle v starts and ends its travel along the arc $a_{i,h}$. Note that, since all arcs have the same length δ, these two variables are related to the speed of the vehicle along the arcs, which is equal to $\delta/(t_{i,v,h} - s_{i,v,h})$. In addition, the Boolean variables $x_{i,v,j,u}$ allow to manage the incompatibilities defined in Sec. 6.2.3 and schedule whether the vehicle v in the stream i crosses the intersection before the vehicle u in the stream j ($x_{i,v,j,u} = 1$) or after ($x_{i,v,j,u} = 0$).

5.3.1 The intersection MILP formulation

Given the considerations and the modeling issues addressed in the previous section, the optimization problem can be formalized as follows:

$$[\boldsymbol{s}^\star, \boldsymbol{t}^\star, \boldsymbol{x}^\star] = \arg\min_{\boldsymbol{s},\boldsymbol{t},\boldsymbol{x}}(\Theta + \alpha SF) \tag{5.4}$$

being

$$\Theta = \sum_{i\in\mathcal{S}}\sum_{v\in\mathcal{V}_i} w_u \left(t_{i,v,n_i} - s_{i,v,0}\right) \tag{5.5}$$

the total weighted travel time; the choice to minimize the vehicle travel time follows from the structure of the optimization problem, whose main variables are the vehicle time instants, and by the routing model, which minimizes the vehicle travel times globally (i.e., at a network level).

$$SF = \sum_{i\in\mathcal{S}}\sum_{v\in\mathcal{V}_i}\sum_{h=1}^{n_i-3} |s_{i,v,h+3} - 2s_{i,v,h+2} + s_{i,v,h+1}| \tag{5.6}$$

is a smoothing filter aimed at getting smooth speeds, to avoid sudden changes of acceleration which otherwise could appear in the solution (as discussed in Sec. 7.2.2) in the form of speed peaks of the kind *high-low-high* or *low-high-low* (determined by *positive-negative-positive* or *negative-positive-negative* accelerations) in a generic sequence of three arcs.

The meaning of the absolute value term in the smoothing filter can be explained as follows: consider a sequence of three generic arcs $a_{i,h}$, $a_{i,h+1}$, and $a_{i,h+2}$ along the trajectory γ_i. If $(s_{i,v,h+3} - s_{i,v,h+2}) < (s_{i,v,h+2} - s_{i,v,h+1})$, then the average speed along the arcs $a_{i,h+1}$ and $a_{i,h+2}$ is increasing. Analogously, considering the whole sequence

of three arcs, if $(s_{i,v,h+3} - s_{i,v,h+2}) < (s_{i,v,h+1} - s_{i,v,h})$, then, averagely, the vehicle is accelerating. Let

$$\Delta\tau_{h,3} = (s_{i,v,h+1} - s_{i,v,h}) - (s_{i,v,h+3} - s_{i,v,h+2}) = \\ = -s_{i,v,h+3} + s_{i,v,h+2} + s_{i,v,h+1} - s_{i,v,h} \quad (5.7)$$

be the overall travel time variation from $a_{i,h}$ to $a_{i,h+2}$, that is, along the sequence $a_{i,h}$, $a_{i,h+1}$, and $a_{i,h+2}$, and let

$$\Delta\tau_{h,2} = (s_{i,v,h+1} - s_{i,v,h}) - (s_{i,v,h+2} - s_{i,v,h+1}) = -s_{i,v,h+2} + 2s_{i,v,h+1} - s_{i,v,h} \quad (5.8)$$

be the travel time variation along the sequence of arcs $a_{i,h}$ and $a_{i,h+1}$. If the vehicle is smoothly accelerating, than $\Delta\tau_{h,3}$ and $\Delta\tau_{h,2}$ are both positive. If $\Delta\tau_{h,3} > 0$ and $\Delta\tau_{h,2} < 0$, the vehicle accelerates globally but decelerates in the middle arc. Therefore, the term

$$|\Delta\tau_{h,3} - \Delta\tau_{h,2}| = |s_{i,v,h+3} - 2s_{i,v,h+2} + s_{i,v,h+1}| \quad (5.9)$$

is reduced if $\Delta\tau_{h,3}$ and $\Delta\tau_{h,2}$ are both positive or negative and, in particular, is null if $\Delta\tau_{h,3} = \Delta\tau_{h,2}$.

Problem constraints

The problem in (5.4) is subject to a set of constraints which guarantee some properties of vehicles' kinematics. In particular, the set of equations

$$s_{i,v,h+1} = t_{i,v,h}, \qquad \forall v \in \mathcal{V}_i, \forall h = 1, \ldots, n_i - 1, \forall i \in \mathcal{S} \quad (5.10)$$

guarantees the continuity of the time-space trajectories, whereas the set of equations

$$s_{i,v,0} + \frac{\Delta_i}{v_{\text{rd}}^{\max}} \leq s_{i,v,1} \leq s_{i,v,0} + \frac{\Delta_i}{v_{\text{rd}}^{\min}} \qquad \forall v \in \mathcal{V}_i, \forall i \in \mathcal{S} \quad (5.11)$$

guarantees that vehicles are not required to reach the first point of the trajectories too early or too late in compliance with the minimum speed $v_{\text{rd}}^{\min}$ and the maximum speed $v_{\text{rd}}^{\max}$ allowed in the road stretches between the triggers and $P_{i,1}$.

In addition, to avoid high, uncomfortable accelerations, the problem in (5.4) is subject to

$$\begin{cases} t_{i,v,h+1} - s_{i,v,h+1} \geq t_{i,v,h} - s_{i,v,h} - \Phi_l \\ t_{i,v,h+1} - s_{i,v,h+1} \leq t_{i,v,h} - s_{i,v,h} + \Phi_l \end{cases} \\ \forall v \in \mathcal{V}_i, \forall h = 1, \ldots, n_i - 1, \forall i \in \mathcal{S} \quad (5.12)$$

and

$$\begin{cases} t_{i,v,h+1} - s_{i,v,h+1} \geq (t_{i,v,h} - s_{i,v,h}) \cdot \left(1 - \dfrac{\Phi_h}{\Phi_{\min}}\right) \\ t_{i,v,h+1} - s_{i,v,h+1} \leq (t_{i,v,h} - s_{i,v,h}) \cdot \left(1 + \dfrac{\Phi_h}{\Phi_{\min}}\right) \end{cases} \\ \forall v \in \mathcal{V}_i, \forall h = 1, \ldots, n_i - 1, \forall i \in \mathcal{S} \quad (5.13)$$

These sets of equations guarantee that vehicles do not vary their speed too rapidly. In particular, the constraints in (5.12) guarantee that, if the travel time on the arc $a_{i,h}$ is long (that is, if the vehicle v is traveling with low speed), the maximum reduction of the travel time (that is the maximum speed increase) is limited by Φ_l. On the contrary, the constraints in (5.13) guarantee that, if the travel time on the arc $a_{i,h}$ is short (that is, if the vehicle v is traveling with high speed), the maximum reduction of the travel time (that is the maximum speed increase) is limited by $\Phi_h < \Phi_l$. This choice allows a bigger acceleration when a vehicle travels slowly, i.e., a more rapid reaction of vehicles that have been slow down to give way. It is worth nothing that if $(t_{i,v,h} - s_{i,v,h})\Phi_h < \Phi_l\Phi_{\min}$, i.e., if $t_{i,v,h} - s_{i,v,h}$ is small (high speed), constraints (5.12) are dominated by constraints (5.13). Suitable values of Φ_l and Φ_h can be achieved by considering the continuous-time accelerations of vehicles obtained by interpolation of the solution as described in Sec. 5.5.

A further set of constraints is considered to guarantee a safe crossing of the intersection. In particular,

$$t_{i,v,h} - s_{i,v,h} + \beta \geq \frac{\Phi_{\min}}{r_{i,h}} = \frac{\delta}{v_{\text{int}}^{\max} \cdot r_{i,h}} \qquad \forall v \in \mathcal{V}_i, \forall h = 1, \ldots, n_i, \forall i \in \mathcal{S} \tag{5.14}$$

guarantees that vehicles respect the maximum speed limit along the curved trajectories, being $r_{i,h}$ the speed reduction factor, whereas

$$s_{i,v+1,h} \geq t_{i,v,k}, \qquad \forall \{P_{i,h}, P_{i,k}\} \in \mathcal{I}^{\mathcal{S}}, \forall v \in \mathcal{V}_i \tag{5.15}$$

defines the minimum distance between two vehicles on the same stream. In particular, a generic follower vehicle $v+1$ cannot cross the point $P_{i,h}$ until the leader vehicle v in the stream has reached the point $P_{i,k+1}$, being $\{P_{i,h}, P_{i,k}\} \in \mathcal{I}^{\mathcal{S}}$.

Regarding vehicles traveling along conflicting trajectories $(i,j) \in \mathcal{C}$, the set of equations

$$\begin{cases} s_{i,v,h} \geq t_{j,u,k} + \beta - M\, x_{i,v,j,u} \\ s_{j,u,k} \geq t_{i,v,h} + \beta - M\,(1 - x_{i,v,j,u}) \end{cases} \qquad \forall \{P_{i,h}, P_{j,k}\} \in \mathcal{I}^{\mathcal{C}}, \forall v \in \mathcal{V}_i, u \in \mathcal{V}_j \tag{5.16}$$

manages the incompatibility between vehicles' relative positions; in particular if the binary variable $x_{i,v,j,u}$ is set to 0, the vehicle v in the stream i cannot leave the point $P_{i,h}$ until the vehicle u in the stream j has reached the point $P_{j,k+1}$ (and after a safety delay β), whereas if $x_{i,v,j,u} = 1$, then u cannot cross $P_{j,k}$ until v has reached $P_{i,h+1}$. Note that this constraints are set for all the conflicting points $\{P_{i,h}, P_{i,k}\}$ in the incompatibility set $\mathcal{I}^{\mathcal{C}}$.

In the end, the set of equations

$$\begin{cases} s_{i,v,h} \geq t_{j,u,k} + \beta - M\, x_{i,v,j,u} \\ s_{j,u,k} \geq t_{i,v,h} + \beta - M\,(1 - x_{i,v,j,u}) \\ s_{i,v,0} \geq s_{j,u,0} - M\, x_{i,v,j,u} \\ s_{j,u,0} \geq s_{i,v,0} - M(1 - x_{i,v,j,u}) \end{cases} \qquad \forall \{P_{i,h}, P_{j,k}\} \in \mathcal{I}^{\mathcal{O}}, \forall v \in \mathcal{V}_i, u \in \mathcal{V}_j \tag{5.17}$$

manages the incompatibility between vehicles' relative positions traveling along partially overlapped trajectories $(i,j) \in \mathcal{O}$. In particular, the third and the fourth

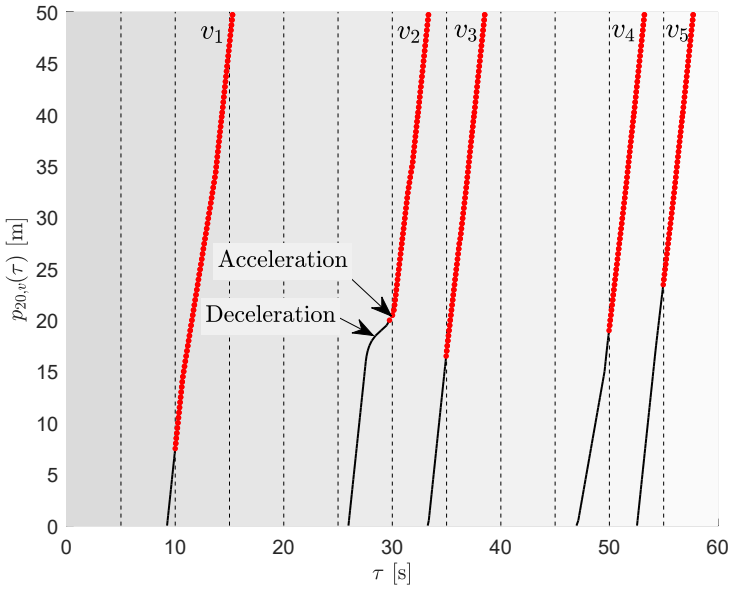


Figure 5.8: Sequence of intervals $\Delta\tau = 5$ s and relevant vehicles in the stream $i = 20$. The continuous black lines represent the time-space diagrams determined by the model in a certain time interval whereas the dotted red lines represent their portion that, being referred to vehicles still inside the intersection at the end of the time interval, constitutes additional constraints in such an interval.

equations identify which vehicle enters the intersection first, while the first and the second equations set the right-of-way in the conflicts as in constraints in (5.16).

5.4 Problem solution and the rolling horizon framework

The problem in (5.4)–(5.17) is computationally hard to solve and the solution time turns out to be not compatible with real-time applications, as discussed in Sec. 3.1. Therefore, in the following, an approach for reducing the computational time is proposed. As mentioned in Sec. 3.1, the considered architecture assumes to solve a temporal-discretized sequence of problems by considering, in each problem, only the vehicles arriving at the intersection in the interval $\Delta\tau_k$. Therefore, in order to reduce the computational complexity it is possible to reduce the number of vehicles in each problem instance by shortening the time window $\Delta\tau$. In doing so, a rolling horizon framework has to be adopted by applying the following steps:

1. consider the interval $\Delta\tau_k = [\tau_k, \tau_{k+1})$, $k = 1, 2, \ldots$ of length $\Delta\tau$ defined in Sec. 3.1;

2. the vehicles entered in the previous interval $[\tau_{k-1}, \tau_k)$ and still in the intersection in the interval $[\tau_k, \tau_{k+1})$ are considered in the relevant optimization problem as part of constraints, being the values of the relevant variables $s_{i,v,h}$, $t_{i,v,h}$, and $x_{i,v,j,u}$ fixed;

3. the value of the total travel time Θ related to wider intervals corresponding to the union of several time windows is the sum of the total travel times Θ_k in all the time windows, that is $\Theta = \sum_k \Theta_k$. Therefore, to avoid multiple counts of the same vehicles, in Θ_k only the vehicles whose speed is optimized in the interval $[\tau_k, \tau_{k+1})$ are considered, while those still in the intersection but arrived in previous intervals are neglected.

A graphical example is depicted in Fig. 5.8, which shows, for the stream $i = 20$, some optimal time-space trajectories over a minute assuming $\Delta\tau = 5$ s. For instance, the optimal time-space diagram of the vehicle v_1 is determined for the interval $[5, 10)$ and constitutes a constraint in the problem of the interval $[10, 15)$. In compliance with the above third step, the non-null terms Θ_k are: Θ_2, which considers only v_1; Θ_6, which considers only v_2; Θ_7, which considers only v_3; Θ_{10}, which considers only v_4; Θ_{11}, which considers only v_5.

5.5 Continuous approximation of the solution

The proposed MILP problem provides a discrete solution but the movement of vehicles along trajectories is continuous. Therefore, although such a solution could be sufficient for the vehicle controller, it is interesting to investigate the continuous approximation of the speed and acceleration of vehicles and analyze their properties. To this aim, it is worth saying that the MILP solution can be considered, for each v, as the sequences of pairs $(P_{i,h}, s^\star_{i,v,h})$, being $P_{i,h}$ a point of the trajectory and $s^\star_{i,v,h}$ the time at which the vehicle v reaches it. Such a sequence can be rewritten as the sorted set of pairs $(\ell(P_{i,h}), s^\star_{i,v,h})$ which express the times $s^\star_{i,v,h}$ at which the vehicle v is at a distance $\ell(P_{i,h})$ from $P_{i,1}$.

Therefore, by interpolating all the entries of the sequence it is possible to find a continuous function $f_{i,v}$ providing the time at which the vehicle v traveling along γ_i reaches any point P of its trajectory. In doing so, since $s^\star_{i,v,h+1} > s^\star_{i,v,h}$, the monotonicity of $f_{i,v}$ is a requirement that can be met by means of a monotone cubic Hermite spline interpolation [62] (see Sec. 2.3 for higher details). Note that, thanks to the chosen spline form, the function $f_{i,v}$ is bijective and, denoting with $p_{i,v}(\tau)$ the distance traveled by v from $P_{i,1}$ at τ, it is possible to find the inverse function $f^{-1}_{i,v} : p_{i,v}(\tau) = f^{-1}_{i,v}(\tau)$ expressing the distance traveled by the vehicle v at τ. In particular, being $f_{i,v}$ an interpolating function, it is guaranteed that $p_{i,v}(s_{i,v,h}) = f^{-1}_{i,v}(s_{i,v,h}) = \ell_{i,h}$. The resulting increasing monotonicity of $f^{-1}_{i,v}$ guarantees that the vehicle v never stops or go backward.

Then, it is immediate to verify if the safety constraint (5.14) is always met, that is, if

$$\dot{p}_{i,v}(\tau) \leq v^{\max}_{\text{int}} \cdot r\left(f^{-1}_{i,v}(\tau)\right) \tag{5.18}$$

Analogously, it is immediate to check if the module of instantaneous acceleration is within a given comfort threshold, that is, if

$$|\ddot{p}_{i,v}(\tau)| \leq a_{\max} \tag{5.19}$$

The analysis of (5.19) allows to calibrate the values of the problem parameters Φ_l and Φ_h with respect to the physical parameter $a_{\max}$.

To conclude, it is worth saying that in case these two constraints are not met, it is possible to cope with their infeasibility by increasing the parameter β in (5.14) or reduce the parameters Φ_l and Φ_h in (5.12) and (5.13), respectively.

5.6 Analytic forms of the trajectories

In this section, some details about the form of the trajectories and the procedure for determining the maximum speed constraint for a general curved trajectory are reported, making use of the plane curves concepts introduced in Sec. 2.2. To do so, consider the generic curve λ representing the trajectories depicted in Fig. 5.3. λ consists of the following parts:

- λ_1 and λ_5 (from P_1 to A and after the point E, respectively) are the initial and final straight trajectories;
- λ_3 (from B to D) is the central circular trajectory with fixed radius;
- λ_2 and λ_4 (from A to B and from D to E, respectively) are the clothoids that guarantee a smooth variation of the radius of curvature between λ_1 and λ_3 and between λ_3 and λ_5;

However, for the sake of conciseness and thanks to the symmetry property of the considered curve trajectories, it is possible to consider only the curves λ_1, λ_2, and the first half of λ_3, hereafter indicated as λ_{32}, that is until the point C. Therefore, with reference to such a figure, let $\bar{\lambda} = \lambda_1 \cup \lambda_2 \cup \lambda_{32}$ and P be a generic point belonging to $\bar{\lambda}$. It is possible to write a parametrization of P as

$$P \in \lambda_1 \Leftrightarrow \begin{cases} P = P_1 + (0, \xi) \\ \xi \in [0, \xi_M] \end{cases} \tag{5.20}$$

$$P \in \lambda_2 \Leftrightarrow \begin{cases} P = A + a \int_0^{\zeta} (\cos(\eta^2), \sin(\eta^2))\, \mathrm{d}\eta, \\ \zeta \in [0, \zeta_M] \end{cases} \tag{5.21}$$

$$P \in \lambda_{32} \Leftrightarrow \begin{cases} P = C_o + (R\cos(\vartheta), R\sin(\vartheta))\,, \\ \vartheta \in [\vartheta_m, \vartheta_M] \end{cases} \tag{5.22}$$

where: $P_1 = (x_1, y_1)$ and $A = (x_1, \xi_M)$ are the initial and the final point of λ_1; a is the scale parameter of the clothoid λ_2; C_o and R are the center and the radius of λ_3. The relevant tangent and normal vectors $T(P)$ and $N(P)$, are

$$P \in \lambda_1 \Leftrightarrow \begin{cases} T(P) = (0, 1) \\ \xi \in [0, \xi_M] \end{cases} \tag{5.23}$$

$$P \in \lambda_2 \Leftrightarrow \begin{cases} T(P) = a_i\,(\cos(\zeta^2), \sin(\zeta^2)) \\ \zeta \in [0, \zeta_M] \end{cases} \tag{5.24}$$

$$P \in \lambda_{32} \Leftrightarrow \begin{cases} T(P) = R_i\,(-\sin(\vartheta), \cos(\vartheta)) \\ \vartheta \in [\vartheta_m, \vartheta_M] \end{cases} \tag{5.25}$$

and

$$P \in \lambda_1 \Leftrightarrow \begin{cases} N(P) = (0,0) \\ \xi \in [0, \xi_M] \end{cases} \tag{5.26}$$

$$P \in \lambda_2 \Leftrightarrow \begin{cases} N(P) = 2a\zeta\,(-\sin(\zeta^2), \cos(\zeta^2)) \\ \zeta \in [0, \zeta_M] \end{cases} \tag{5.27}$$

$$P \in \lambda_{32} \Leftrightarrow \begin{cases} N(P) = R\,(-\cos(\vartheta), -\sin(\vartheta)) \\ \vartheta \in [\vartheta_m, \vartheta_M] \end{cases} \tag{5.28}$$

It is worth noting that, in these definitions, the physical meaning of the parameters ξ, ζ, and ϑ and their extreme values ξ_M, ζ_M, ϑ_m and ϑ_M are different, depend on the specific parametrization of the curves λ_1, and are chosen to guarantee the condition $\lambda \in C^2$ class. Note that, since the curve λ is oriented and simple, it is possible to identify any generic point P by means of its distance $\ell(P)$ from the point P_1, expressed as a function of the parametrization of λ itself. Given (5.20)–(5.28), it is possible to determine the radius of λ in any point P as

$$\rho(P) = \begin{cases} +\infty, & \text{if } P \in \lambda_1 \\ \dfrac{a}{2\zeta}, & \text{if } P \in \lambda_2 \\ R, & \text{if } P \in \lambda_{32} \end{cases} \tag{5.29}$$

As for the speed constraint, assuming that the friction force $\mu g M$ is the only relevant centripetal force $\frac{1}{2}Mv^2$ acting on the vehicle with speed v in the point P of a planar curve, being M the mass of the vehicle, μ the friction coefficient and g the gravity acceleration, the maximum speed $\dot{\ell}(P)$ that guarantees the stability of the vehicle can be determined as

$$\dot{\ell}(P) = \sqrt{\mu g \rho(P)}, \quad \forall P \in \lambda \tag{5.30}$$

and, by expressing the maximum speed as $\dot{\ell}(P) = r(P) \cdot v^{\max}$, it is possible to express the speed reduction factor for each point P as

$$r(P) = \begin{cases} 1 & \text{if } P \in \lambda_1 \\ \min\left\{1, \dfrac{\sqrt{\mu g \rho(P)}}{v^{\max}}\right\} & \text{if } P \in \lambda_2 \\ \min\left\{1, \dfrac{\sqrt{\mu g R}}{v^{\max}}\right\} & \text{if } P \in \lambda_{32} \end{cases} \tag{5.31}$$

$\dot{\ell}(P)$ is therefore a property of the curve, not dependent on the problem constraints. To conclude, it is worth remarking that the speed reduction factors have been computed per each trajectory in the intersection, once and off-line.

Table 5.1: Summary of notation for the isolated intersection element.

Symbol	Meaning
$\mathcal{S}$	set of streams crossing the intersection
$\mathcal{V}_i$	set of vehicles on the stream i. Such a set is sorted, so the direct follower of the vehicle v is $(v+1)$
γ_i, $\forall i \in \mathcal{S}$	trajectory followed by vehicles in the stream i (see Appendix 5.6)
$\Gamma = \{\gamma_i \vert \forall i \in \mathcal{S}\}$	set of all trajectories
$P_{i,0}$	position of the trigger related to γ_i
$P_{i,1}$	first point of γ_i considered in the optimization problem
$P_{i,h}$	generic point of γ_i, $i \in \mathcal{S}$
$\mathcal{P}_i = \{P_{i,h}\}$	set of n_i points along γ_i
$P_{i,h} \nleftrightarrow P_{j,k}$	the points $P_{i,h}$ and $P_{j,k}$ are incompatible
$a_{i,h}$	generic arc of the trajectory γ_i between two consecutive points $P_{i,h}$ and $P_{i,h+1}$
δ	fixed length of the arc $a_{i,h}$
$\ell(P_{i,h})$	distance of $P_{i,h}$ from $P_{i,1}$
$v_{\text{rd}}^{\min}$ and $v_{\text{rd}}^{\max}$	minimum and maximum admissible vehicle speed in the road stretches between the triggers and the intersection
$v_{\text{int}}^{\max}$	maximum admissible vehicle speed inside the intersection
$\Phi_{\min} = \delta / v_{\text{int}}^{\max}$	minimum travel time on an arc
Φ_l and Φ_h	maximum allowed relative vehicle travel time variations when the speed is low and high, respectively
$r_{i,h}$	coefficient determined off-line that reduces the maximum allowed speed on the arc $a_{i,h}$ (see Sec. 6.2.2)
w_u	weight of vehicle u
α	weighting coefficient in the objective function of the MILP problem
β	safety parameter that increases the minimum permitted distance between vehicles
$s_{i,v,0}$	time instant at which vehicle v crosses the trigger and enters the road stretch between the trigger and the intersection
$s_{i,v,h}$	real variable indicating the time instant at which the vehicle v enters the arc $a_{i,h}$ of the trajectory γ_i
$t_{i,v,h}$	real variable indicating the time instant at which the vehicle v exits the arc $a_{i,h}$ of the trajectory γ_i
$x_{i,v,j,u}$	Boolean variable set to 1 if the vehicle v in the stream i has right of way with respect to the vehicle u in the stream j
$s^\star_{i,v,h}$, $t^\star_{i,v,h}$, $x^\star_{i,v,j,u}$	optimal values of the variables $s_{i,v,h}$, $t_{i,v,h}$, $x_{i,v,j,u}$

Chapter 6

The isolated road model

This chapter describes the isolated road model, which is aimed at jointly determining the optimal trajectories and speeds of the vehicles entering a generic road element. While the general structure of this chapter follows the one of chapter 5, the similarities with the intersection model and the peculiarities of such model are highlighted and discussed. In particular, the two main peculiarities of the road model are that: (1) the road model jointly determines the trajectory and the speeds; thus, the trajectories are not assigned a priori depending on the direction of travel of the vehicle, as in the intersection model; (2) differently from the intersection model, vehicles can overtake themselves in the road and change lane if necessary.

6.1 Definitions, Assumptions and Notation

This section introduces the definitions and the specific assumptions and notation for the road model.

6.1.1 Assumptions

The following assumptions hold:

1. the road elements can be divided into "sections", each of them sub-divided into "lane change sub-sections" and "separated lanes sub-sections", being the first the locations at which vehicles can change lane, and the second the locations where vehicles cannot change lane (see Fig. 6.1 where a road element consisting of 4 sections is depicted); note that, thanks to the presence of the lane change sub-sections, vehicles can overtake each others inside the road elements. This is an important difference with respect to the intersection model, where vehicles are sorted in the assigned streams and cannot overtake each other;

2. along roads, 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 sub-sections and intersections; therefore, the proposed approach has to provide

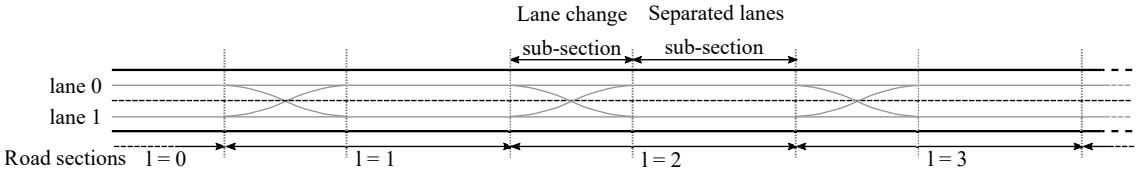


Figure 6.1: Example of road layout consisting of 4 sections (3 lane change sub-sections and 4 separated lanes sub-sections).

solutions that avoid collisions. Differently from the intersection model, in the road model trajectories are not assigned before the solution of the scheduling and motion planning model, but they are part of the solution itself;

3. 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 6.2.3.

4. as for the intersection, 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 6.2.2;

6.1.2 Notation

The notation adopted for the road model is similar to the one used for the intersection, and is introduced in the following, where the structures, the sets, the parameters, and the variables characterizing the model are reported. Besides, thanks to the generality of the model and for the sake of compactness, the element index e is dropped.

In the road model vehicles are allowed to change lane and to overcome themselves; thus, they cannot be a-priori sorted on a certain predefined trajectory. This requirement results in significant differences in the model and in the notation with respect to the intersection element.

A first difference is the absence of the concept of stream and of its index i. This is because vehicles are not assigned a-priori to a specific trajectory before the solution of the optimization problem, but the final trajectory of the vehicle comes as an output of the optimization problem itself together with the optimal values of speed to follow. In addition, the points discretizing the vehicle trajectories are no more referred to a specific trajectory and are characterized by a global index. As a result, the same point may belong to more than one trajectory. As for the variables, the binary variable x, which states the right of way for each couple of vehicles, is referred to a specific road section, and it can therefore change along the road.

Lastly, another Boolean variable z is introduced, which allows to determine the final trajectory of each vehicle inside the road.

Structures and sets

- $\mathcal{G} = \{\mathcal{N}, \mathcal{L}\}$ 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.

- $\mathcal{N} = \{P_h\}$ is the set of $|\mathcal{N}|$ 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 characterizing the spatial discretization of trajectories in Γ; nodes are not sorted and, in general P_k might follow P_{k+h}, with $h > 0$, in a trajectory.

- $\mathcal{L} = \{(P_h, P_k) | P_h, P_k \in \mathcal{N}\}$ is the set of $|\mathcal{L}|$ directed links connecting pairs of adjacent nodes P_h and P_k.

- for any generic road element, $\mathcal{Z}$ is the set of $|\mathcal{Z}|$ road sections, i.e., the set of distinct *lane change* and *separated lanes* sub-sections.

- $\mathcal{V}$ is the set of $|\mathcal{V}|$ vehicles.

- $\mathcal{V}_l$ is the set of vehicles turning left at the intersection after the road element. It is a subset of $\mathcal{V}$.

- $\mathcal{V}_r$ is the set of vehicles turning right at the intersection after the road element. It is a subset of $\mathcal{V}$.

- P_0^{out} is the exit node of the road in the right lane (lane 0).

- P_1^{out} is the exit node of the road in the left lane (lane 1).

- $_h \nleftrightarrow _k$ means that the points $_h$ and $_k$ are incompatible, that is, they cannot be occupied, at the same time, by two vehicles because this situation may cause a collision.

- for any road element, $\mathcal{LN}_l$ (respectively, $\mathcal{CH}_l$) is the set of nodes in the separated lane (resp., lane change) sub-section of the section l of the road; in this connection, considering a link $(P_h, P_k) \in \mathcal{L}$, note that:

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

- for any separated lane sub-section, $\mathcal{NL}_l^y \subset \mathcal{LN}_l$, $y \in \{0, 1\}$, is the set of nodes in lane y of section l.

- for any lane change sub-section, $\mathcal{NC}_l^{y_1,y_2,y_3,y_4} \subset \mathcal{CH}_l$ 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 6.2.3).

Parameters

- β is a safety parameter that prevent any vehicle to be too near to the others in proximity of a trajectory conflict.
- z_u^{in} is a binary parameter specifying the lane in which vehicle u enters the road (right lane when $z_u^{\text{in}} = 0$ and left lane when $z_u^{\text{in}} = 1$).
- $v_{\text{road}}^{\max}$ is the maximum speed of vehicles in the road elements; note that, in some cases, special vehicles (i.e., emergency vehicles) might be allowed to travel at speeds greater than $v_{\text{road}}^{\max}$.
- $v_{\text{rd}}^{\max}$ 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_l and Φ_h 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 (that is, not emergency brake); in addition, it is $\Phi_l > \Phi_h$ 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 6.2.2); note that, in presence of more than one link starting from P_h, r_h is set in a conservative way, that is, 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.

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 optimization problem described in section 6.3.1; note that, $s_{u,h}$ assumes the same value for all the links with 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 $t^{\star}_{u,h}$ indicates the relevant optimal value determined by solving the optimization problem described in section 6.3.1; in analogy to $s_{u,h}$, also the variable $t_{u,h}$ assumes the same value for all the links with origin at the node P_h, if more than one.

- s_u^{in} is the time instant at which the vehicle u crosses the trigger located at a distance Δ from the road.

- t_u^{out} is the time at which vehicle u exits the road.

- $x_{u,v,l}$ is a Boolean variable that is set to 1 if vehicle u has right of way with respect to vehicle v in the road section l; as before, let $x^{\star}_{u,v,l}$ be its optimal value.

- $z_{u,l}$ is a Boolean variable that is set to 0 (respectively, 1) if vehicle u in the road section l is on the right lane (resp., left lane) and $z^{\star}_{u,l}$ indicates its optimal value; it is worth noting that the value of these variables is the only information which is needed for determining the trajectory of each vehicle, since the sub-trajectory of a vehicle in a lane change sub-section is in one-to-one correspondence with the values assumed by z in the previous and subsequent separated lanes sub-sections.

6.2 Modeling and optimization approach

This section discusses for the road model the three fundamental aspects of the proposed approach that have already been introduced in Sec. 5.2 for the intersection model: the spatial discretization of the trajectories, the speed reduction along curves and the incompatibility constraints. Focus is, in particular, on the differences with the intersection model.

6.2.1 The spatial discretization

As for the intersection model, the dynamics of vehicles traveling along the trajectories are formulated as a set of constraints of a Mixed-Integer Linear Programming (MILP) problem. To this aim, the trajectories are spatially discretized into sequences of arcs in order to define an equivalent graph $\mathcal{G}$ whose arcs have an a-priori defined constant length δ. The search of a suitable discretization is an off-line problem to solve once per each element: in the experiments conducted in this thesis the choice of $\delta = 1.0$ m resulted to be sufficient. This value turns out to be higher than the one for the intersection, due to the lower density of conflicts between different trajectories in the road layout.

Thanks to this spatial discretization it is possible to define the set $\mathcal{N} = P_h\}$ of all the points of the road. This set is, in general, not sorted, and P_{h+1} may not follow P_h. Thus, it is necessary to consider the set $\mathcal{L}$ of all the directed arcs (P_h, P_k) connecting P_h to P_k.

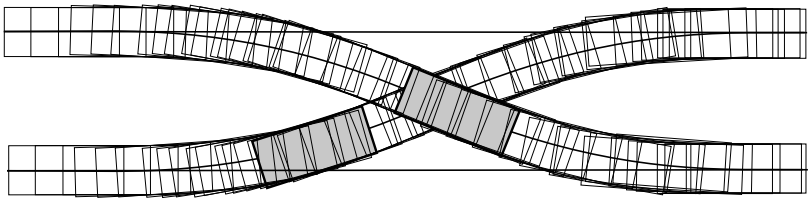


Figure 6.2: Example of vehicles occupancies in a lane change sub-section.

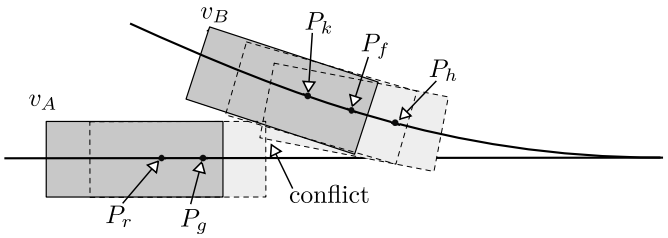


Figure 6.3: Example of vehicles occupancies in a lane change sub-section: P_r and P_g are not compatible with P_k, P_f, and P_h.

6.2.2 Curved trajectories and speed limits

The maximum admissible speed of vehicles in any arc is determined, as for the intersection, by multiplying the speed limit on the road by a speed reduction factor r, which is no more related to a stream but to the arc starting from a given point P_h. The reduction factor considers the geometry of the arc to which it is referred, and ensure, in order, an acceptable level of comfort and safety along curves.

In particular the speed limit on the road arc with origin in P_h can be defined as $r_h \cdot v_{\text{road}}^{\max}$, being

$$r_h = \min \left\{ 1, \min_{k|(P_h,P_k)\in\mathcal{L}} \frac{\sqrt{\mu g \rho(P_h, P_k)}}{v_{\text{road}}^{\max}} \right\} \tag{6.1}$$

a speed reduction factor determined as described in Sec. 5.6. In (6.1), 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. Finally note that the definition in (6.1) is conservative, as it considers the minimum radius in case of multiple links starting from the same node P_h, and that all the links of straight trajectories have $r_h = 1$.

6.2.3 Incompatibility constraints and conflicts management

As done for the intersection, vehicles collisions are avoided by specific constraints preventing vehicles from reaching, at the same time, two incompatible points. The concepts of incompatibility and space occupancy are analogous to the ones introduced in Sec. 6.2.3; an example of vehicle occupancies in the road is depicted in Fig. 6.2, whereas Fig. 6.3 shows an example of incompatible points for two vehicles traveling within a lane change sub-section.

In such figure, darker-gray rectangles indicate the current position of two vehicles and the lighter-gray ones indicate successive occupancies of the vehicles due to their

movements, having assumed the previously mentioned discretization. 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 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 can not 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.

By considering these concepts, it is possible 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, that is, the incompatibility sets. These latter can be related to the separated lanes sub-sections and to the lane change sub-sections. For what concerns the incompatibilities in the separated lanes sub-sections, 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 sub-sections, there are conflicts both between vehicles traveling on the same trajectory and between vehicles traveling on different trajectories (at least partially).

The following incompatibility sets are defined for the separated lanes sub-sections and the lane change sub-section, respectively:

$$\mathcal{I}^{\text{lane}}_{y,l} = \Big\{(P_h, P_k) | P_h, P_k \in \mathcal{NL}^y_l, P_h \nleftrightarrow \mathrm{P}_k\Big\}, \qquad \forall l \in \mathcal{Z}, \forall y \in \{0,1\} \quad (6.2)$$

$$\mathcal{I}^{\text{change}}_{y_1,y_2,y_3,y_4,l} = \Big\{(P_h, P_k) | P_h, P_k \in \mathcal{NC}^{y_1,y_2,y_3,y_4}_l, \mathrm{P}_h \nleftrightarrow \mathrm{P}_k \Big\}, \\ \forall l \in \mathcal{Z}, \forall y_1, \forall y_2, \forall y_3, \forall y_4 \in \{0,1\}, \\ \text{with } [y_1, y_2, y_3, y_4] \neq \Big\{[0,0,1,1], [1,1,0,0]\Big\} \quad (6.3)$$

The above sets define the pairs of incompatible nodes, both relative to the same trajectory and to different trajectories. (6.2) and (6.3) are $16\,|\mathcal{Z}| + 2$ incompatibility sets. In (6.3), 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 sub-section l and ends in the last node of lane y_2 always in the lane change sub-section 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.

6.3 The scheduling and motion planning model

As previously stated, the optimal trajectories and speeds of the self-driving vehicles traveling the road elements are determined by solving a scheduling and motion planning problem, which is mathematically formalized as a MILP problem; in the following it is described in detail.

6.3.1 The road MILP formulation

The optimization problem for a generic road element is formalized as follows.

$$[\boldsymbol{s}^\star, \boldsymbol{t}^\star, \boldsymbol{x}^\star, \boldsymbol{z}^\star] = \arg\min_{\boldsymbol{s},\boldsymbol{t},\boldsymbol{x},\boldsymbol{z}} \Theta + \Omega \tag{6.4}$$

being

$$\Theta = \sum_{u\in\mathcal{V}} w_u \left(t_u^{\text{out}} - s_u^{\text{in}}\right) \tag{6.5}$$

the total weighted travel time and

$$\Omega = \sum_{u\in\mathcal{V}_r} w_u z_{u,|\mathcal{Z}|} - 1000 \sum_{u\in\mathcal{V}_l} w_u z_{u,|\mathcal{Z}|} \tag{6.6}$$

a second objective that encourages vehicles to exit the road from the the right lane if they are turning right or to exit from the left lane if they are turning left at the following intersection.

The problem is subject to the following constraints.

$$t_u^{\text{out}} \geq t_{u,k_0^{\text{out}}} - M z_{u,|\mathcal{Z}|}, \quad \forall u \in \mathcal{V} \tag{6.7}$$

$$t_u^{\text{out}} \geq t_{u,k_1^{\text{out}}} - M(1 - z_{u,|\mathcal{Z}|}), \quad \forall u \in \mathcal{V} \tag{6.8}$$

Constraints (6.7) and (6.8) 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 sub-section. In (6.7) and (6.8), 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).

$$s_{u,k} = t_{u,h}, \quad \forall u \in \mathcal{V}, \forall (P_h, P_k) \in \mathcal{L} \tag{6.9}$$

Constraint (6.9) guarantees the time continuity of the trajectories.

$$\begin{cases} t_{u,k} - s_{u,k} \geq t_{u,h} - s_{u,h} - \Phi_l \\ t_{u,k} - s_{u,k} \leq t_{u,h} - s_{u,h} + \Phi_l \end{cases} \qquad \forall u \in \mathcal{V}, \forall (P_h, P_k) \in \mathcal{L} \tag{6.10}$$

$$\begin{cases} t_{u,k} - s_{u,k} \geq (t_{u,h} - s_{u,h}) \cdot \left(1 - \dfrac{\Phi_h}{\Phi_{\min}}\right) \\ t_{u,k} - s_{u,k} \leq (t_{u,h} - s_{u,h}) \cdot \left(1 + \dfrac{\Phi_h}{\Phi_{\min}}\right) \end{cases} \qquad \forall u \in \mathcal{V}, \forall (P_h, P_k) \in \mathcal{L} \tag{6.11}$$

Constraints (6.10) and (6.11) are "comfort" constraints which guarantee that vehicles do not vary their speed too rapidly; in particular, constraints (6.10) 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_l; on the contrary, constraints (6.11) guarantee that, if the travel time on the link (P_h, P_k) is short (high speed), it can only vary of the value $\Phi_h < \Phi_l$.

$$t_{u,h} - s_{u,h} \geq \frac{\delta}{r_h v_{\text{road}}^{\max}} \quad \forall u \in \mathcal{V}, \forall h = 1, \ldots, |\mathcal{N}| \tag{6.12}$$

$$s_{u,h} \geq t_u^{\text{in}} + \frac{\Delta}{v_{\text{rd}}^{\max}} \quad \forall u \in \mathcal{V}, \forall h = 1, \ldots, |\mathcal{N}| \tag{6.13}$$

Constraints (6.12) and (6.13) 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.

$$z_{u,0} = z_u^{\text{in}} \quad \forall u \in \mathcal{V} \tag{6.14}$$

$$t_u^{\text{in}} - t_v^{\text{in}} \leq M(1 - x_{u,v,0}) \quad \forall u, v \in \mathcal{V} \tag{6.15}$$

$$t_v^{\text{in}} - t_u^{\text{in}} \leq M x_{u,v,0} \quad \forall u, v \in \mathcal{V} \tag{6.16}$$

For section 0 (that consists of the first separated lanes sub-section only), constraint (6.14) initializes variable $z_{u,0}$ by setting it to the value that specifies the given lane of arrivals, and constraints (6.15) and (6.16) initialize 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).

$$s_{u,h}\mathbf{1}_2 \geq [t_{v,k} + \beta - M \cdot x_{u,v,l}]\mathbf{1}_2 - M[\mathbf{A}\boldsymbol{z}' + \mathbf{b}]$$
$$\mathbf{A} = \begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \quad \boldsymbol{z}' = \begin{pmatrix} z_{u,l} \\ z_{v,l} \end{pmatrix}$$
$$\forall u, v \in \mathcal{V}, u \neq v, \forall (P_h, P_k) \in \mathcal{I}_{y,l}^{\text{lane}}, \forall l \in \mathcal{Z} \tag{6.17}$$

Equ. (6.17) is a compact representation of the "incompatibility constraints" for a generic separated lanes sub-section, being $\mathbf{1}_n$ a column vector with dimension n whose entries are all equal to 1. As a matter of fact, (6.17) corresponds to 2 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 enters the link (P_h, P_m) until v exits the link (P_k, P_n), being $(P_h, P_k) \in \mathcal{I}_{0,l}^{\text{lane}}$ or $(P_h, P_k) \in \mathcal{I}_{1,l}^{\text{lane}}$, and $(P_h, P_m), (P_k, P_n) \in \mathcal{L}$; it is worth noting that the constraints in (6.17) are always satisfied (hence, they are not significant) if $z_{u,l} \neq z_{v,l}$, that is, when the two vehicles are not in the same lane.

$$\begin{cases} x_{u,v,l} \geq x_{u,v,l-1} - M z_{u,l-1} - M z_{v,l-1} \\ x_{u,v,l} \leq x_{u,v,l-1} + M z_{u,l-1} + M z_{v,l-1} \\ x_{u,v,l} \geq x_{u,v,l-1} - M(1 - z_{u,l-1}) - M(1 - z_{v,l-1}) \\ x_{u,v,l} \leq x_{u,v,l-1} + M(1 - z_{u,l-1}) + M(1 - z_{v,l-1}) \end{cases}$$
$$\forall u, v \in \mathcal{V}, u \neq v, \forall l = 1, \ldots, |\mathcal{Z}| \tag{6.18}$$

For a generic section $l > 1$ of the road (that is, not the first one), constraints (6.18) 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 sub-section of section l in the same order they leave section $l-1$.

$$s_{u,h}\mathbf{1}_{14} \geq [t_{v,k} + \beta - Mx_{u,v,l}]\mathbf{1}_{14} - M[\mathbf{C}\boldsymbol{z}'' + \mathbf{d}]$$

$$\mathbf{C} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & -1 \\ 1 & 1 & -1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & -1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ -1 & 1 & -1 & -1 \\ -1 & -1 & 1 & -1 \\ -1 & -1 & -1 & 1 \\ -1 & -1 & -1 & -1 \end{pmatrix}, \quad \mathbf{d} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 2 \\ 2 \\ 3 \\ 1 \\ 2 \\ 2 \\ 3 \\ 3 \\ 3 \\ 4 \end{pmatrix},$$

$$\boldsymbol{z}'' = \begin{pmatrix} z_{u,l-1} \\ z_{v,l-1} \\ z_{u,l} \\ z_{v,l} \end{pmatrix}$$

$$\forall u, v \in \mathcal{V}, u \neq v, \forall (P_h, P_k) \in \mathcal{I}^{\text{change}}_{y_1,y_2,y_3,y_4,l}, \forall l = 1, \ldots, |\mathcal{Z}| \quad (6.19)$$

Equ. (6.19) is again a compact representation of the "incompatibility constraints" for a generic lane change sub-section: as a matter of fact, (6.19) corresponds to 14 constraints each of them defining the incompatibilities between two vehicles in two specific trajectories of the lane change sub-section (there are 4 trajectories in each lane change sub-section and then 16 possible pair of trajectories, but two of them are non in conflict); for example the fourth constraint (obtained by the fourth row of matrix $\mathbf{C}$ and vector $\mathbf{b}$) 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. 6.3); 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 $z_{v,l}$.

$$x_{v,u,l} = 1 - x_{u,v,l} \quad \forall u, v \in \mathcal{V}, \forall l \in \mathcal{Z} \quad (6.20)$$

Constraint (6.20) defines the relationship between $x_{v,u,l}$ and $x_{u,v,l}$.

$$x_{u,v,l} \in \{0, 1\} \quad \forall u, v \in \mathcal{V}, u \neq j, \forall l \in \mathcal{Z} \quad (6.21)$$

$$z_{u,l} \in \{0, 1\} \quad \forall u \in \mathcal{V}, \forall l \in \mathcal{Z} \quad (6.22)$$

$$s_{u,h} \in \mathbb{R}_{\geq 0}, t_{u,h} \in \mathbb{R}_{\geq 0}, \quad \forall u \in \mathcal{V}, \forall h \in \mathcal{N} \quad (6.23)$$

Lastly, constraints (6.21)÷(6.23) define the Boolean and non-negative real variables of the problem.

Chapter 7

Case studies

This chapter presents and discusses the main characteristics of the solution of the proposed TMS, with a focus both on the routing level and on the local level. At a routing level results are aimed at providing evidence of the effectiveness of the proposed routing model, which considers the information regarding the actual network state (provided by the local controllers) and the specific element layouts and vehicle trajectories.

At a local level, results are aimed at showing the vehicle scheduling, motion planning and collision avoidance, the effect of some model parameters, the local performance of the elements and the computational time.

7.1 Vehicle routing and network global performance

This section presents and discusses the effectiveness of the routing model in different scenarios, all considering the network layout depicted in Fig. 7.14, consisting of 6 intersections connected together with the road elements. This layout is also schematically represented in Fig. 4.1.

More in detail, the main goals are:

1. Showing how the routing model allows the vehicles entering the network to avoid the most congested elements;
2. Discussing the role of the distribution objective Γ of Equ. (4.3) in distributing the vehicles on the available paths;
3. Discussing the network state transitions, the TMS dynamics and the advantages of the proposed routing strategy with respect to a reference one.

To discuss the above mentioned points, 9 different scenarios are considered and defined in the following. In addition, each of them considers the following two different types of vehicle flows:

- routing flow: this is the flow on which performance parameters are considered. It is always assumed to enter the network from node L and to be directed towards node E;

- disturbance flow: this is the flow used to modify the network state in such a way that travel times increase on the elements crossed by such flow.

7.1.1 Congestion avoidance

The following scenarios are aimed at investigating how the TMS reacts, at the routing level, to the disturbance flows that worsen locally the performance of the network:

- Scenario 1: the disturbance flow crosses the network from node I to node B. The distribution objective Γ of Equ. (4.3) is activated ($\zeta \neq 0$).
- Scenario 2: the disturbance flow crosses the network from node H to node C. The distribution objective Γ of Equ. (4.3) is activated ($\zeta \neq 0$).
- Scenario 3: the disturbance flow crosses the network from node G to node D. The distribution objective Γ of Equ. (4.3) is activated ($\zeta \neq 0$).

Tab. 7.1 reports, for the different scenarios, the values of the routing model parameters and the vehicle flows rates for the conducted experiments.

Table 7.1: Routing scenarios parameters.

Scenario	Routing flow rate [veh·h^{-1}]	Disturbance flow rate [veh·h^{-1}]	Time horizon [s]	ζ
1	360	3600	120	10
2	360	3600	120	10
3	360	3600	120	10
4	1500	3600	300	0
5	1500	3600	300	10
6	1500	3600	300	1000
7	1500	3600	300	10
8	1500	3600	300	10
9	1500	3600	300	10

Related to the experiments performed in this section and in the following one, Figs. from 7.1 to 7.6 show the road usage (number of vehicles traveling in each road element, represented by the thickness of the bars, for the entire time horizon) for the six considered scenarios: it is possible to see how the flows distribute on the available paths and, in particular, how the routing flow avoids the congested elements crossed by the disturbance flows.

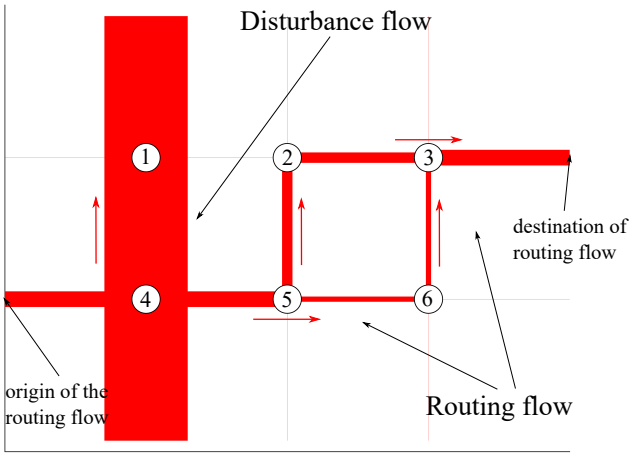


Figure 7.1: Road usage in Scenario 1: the routing flow avoids the left part of the network, which is congested by the disturbance flow.

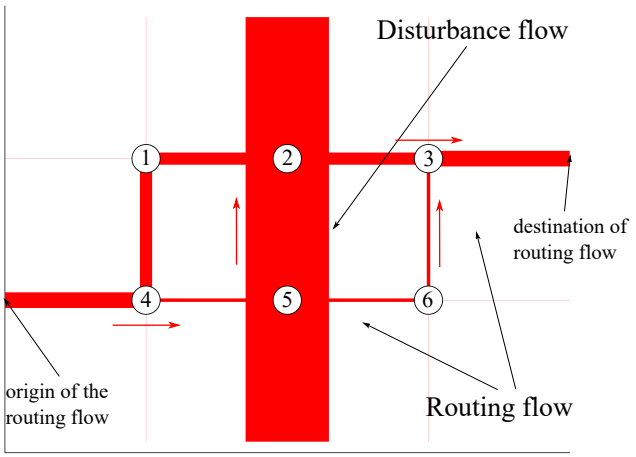


Figure 7.2: Road usage in Scenario 2: the routing flow avoids the central part of the network, which is congested by the disturbance flow.

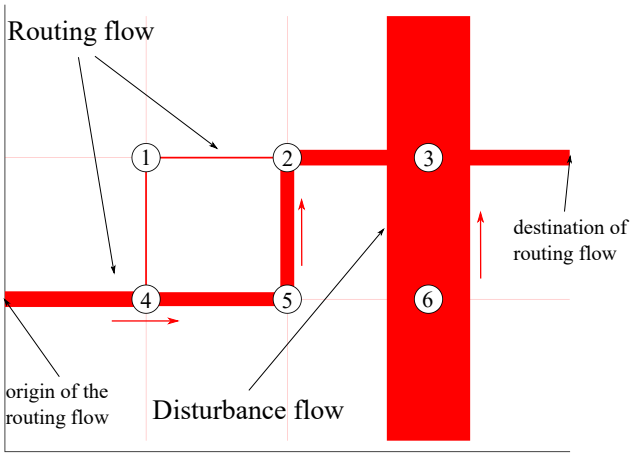


Figure 7.3: Road usage in Scenario 3: the routing flow avoids the right part of the network, which is congested by the disturbance flow.

7.1.2 Vehicle distribution and road usage

This section describes the role of the distribution objective of Γ of Equ. (4.3).

To this aim, let define the following scenarios:

- Scenario 4: the disturbance flow crosses the network from node H to node C. The distribution objective Γ of Equ. (4.3) is deactivated ($\zeta = 0$);
- Scenario 5: the disturbance flow crosses the network from node H to node C. The distribution objective Γ of Equ. (4.3) is activated ($\zeta = 10$) with a low weight and only for the routing flow vehicles;
- Scenario 6: the disturbance flow crosses the network from node H to node C. The distribution objective Γ of Equ. (4.3) is activated ($\zeta = 1000$) with an higher weight and only for the routing flow vehicles.

Figs. 7.4, 7.5 and 7.6 show the road usage for Scenarios 4, 5 and 6, respectively. When the distribution objective is not used, if the travel times of the non congested alternatives are equal, vehicles are mainly routed on one of the two possible path (Fig. 7.4). In this case the average travel time of the vehicles in the routing flow is 170 s.

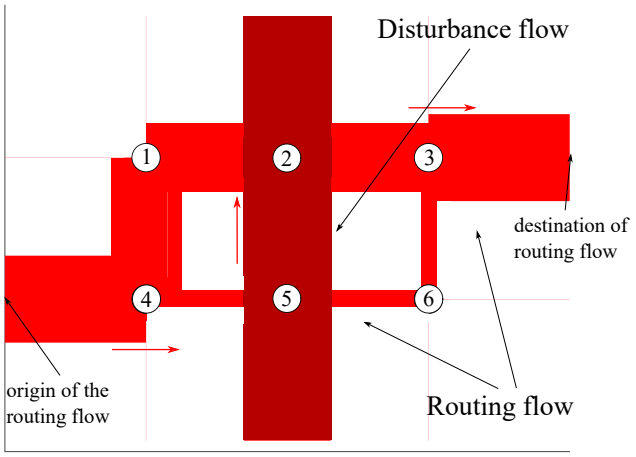


Figure 7.4: Road usage (percentage of vehicle distributions) in Scenario 4: the routing flow avoids the congested part of the network, but it is mainly routed on the left road.

On the contrary, if the distribution objective is activated vehicles are routed equally on the two non congested alternatives, which provide the same travel times (Fig. 7.5). Also in this case the average travel time of the vehicles in the routing flow is 170 s.

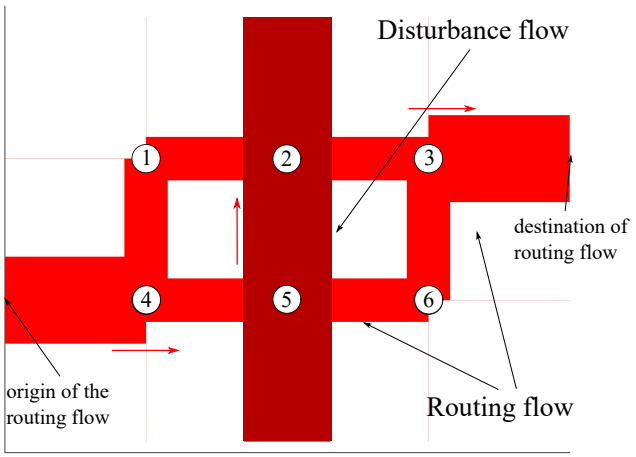


Figure 7.5: Road usage (percentage of vehicle distributions) in Scenario 5: the routing flow avoids the congested part of the network, and equally distributes between the two non congested alternatives (left and right road).

In the case the distribution objective is activated with an higher weight, that is, if it is given a comparable/higher importance than the first objective (the total travel time), then vehicles are routed among the three available paths even if one of them is congested. In this solution the average travel time of the vehicles in the

routing flow that are routed across the congested elements is 6% higher than the one for vehicles that are routed along the non congested alternatives.

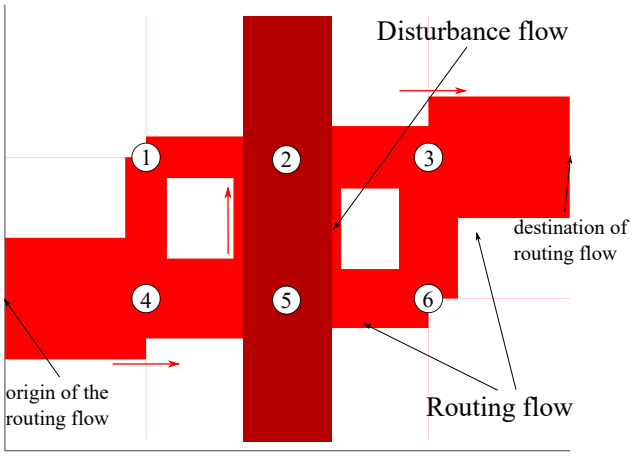


Figure 7.6: Road usage (percentage of vehicle distributions) in Scenario 6: the routing flow avoids the congested part of the network, and distributes over the three alternatives, even if one of them (central road) is congested.

7.1.3 Routing with network state prediction

This section describes the advantages of making routing decisions with both the current network state and the expected one, and discusses the TMS reaction time in the case a portion of the network moves from a non congested to a congested state.

To this aim, the following scenarios are considered:

- Scenario 7: the disturbance flow crosses the network from node H to node C. The routing flow is not optimally routed by the RC, but vehicles are supposed to choose randomly one of the three available paths between the origin and the destination;
- Scenario 8: the disturbance flow crosses the network from node H to node C. The routing flow is optimally routed by the RC, but taking into account only the current state of the network;
- Scenario 9: the disturbance flow crosses the network from node H to node C. The routing flow is optimally routed by the RC, considering also the expected network state on the basis of the vehicles that have already received their path.

Figs. 7.7 and 7.8 show the linear density of vehicles per kilometer in the road elements 25 and 21, respectively. In both the roads there is a loading phase where the density increase linearly with time, then a phase where the density remains constant (on a average value of 950 veh/km in road 25 and 1600 veh/km in road 21) and then a final phase when the density start decreasing. There are not significant

differences between the 3 scenarios, since the vehicles always cross these elements no matter their destination. In particular, the density on road 25 (which is the entrance element of the routing flow) is related only to the routing flow and the density on road 21 (which is the entrance of the disturbance flow) is related only to the disturbance flow.

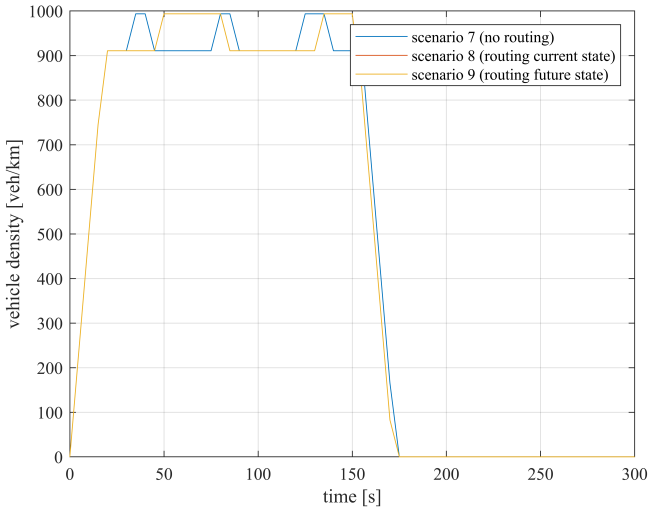


Figure 7.7: Vehicle density on road element 25 as a function of time.

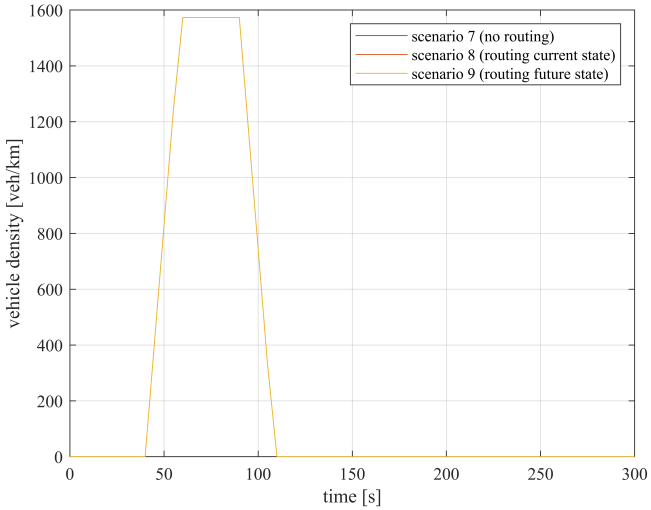


Figure 7.8: Vehicle density on road element 21 as a function of time.

As regards the central road elements over which the routing flow distributes to reach its destination, Figs. 7.11, 7.9 and 7.10 are related, respectively, to the road elements 28, 32 and 36. In particular, Fig. 7.9 show the vehicle density on the road 32, which is crossed both by the routing flow and the disturbance flow. As for roads 25 and 21, there is a loading phase where the density increases, followed by a peak and a decreasing phase. The highest density is observed in scenario 7, where no

optimal path is provided to the vehicles in the routing flow, but they distributes randomly, with equal probability, among the three alternatives (roads 28, 32 or 36). Thus, the difference between the blue curve and the orange one represents the excess of vehicle density (which raises from 200 to 500 veh/km) that can be avoided by routing them along the less congested alternatives. In the case of scenario 9, this difference is even higher (yellow curve). The peak in the density is due to the coexistence of the routing and the disturbance flows; when this latter ends (at around $t = 130s$), the density behavior differ significantly depending on the scenario. In scenario 7 the density does not decrease to 0, due to the portion of vehicles of the routing flow that decide to cross road 32 no matter its level of congestion, but remains between the values of 200 and 500 veh/h, which corresponds, on average, to one third of the density caused by the routing flow (Fig. 7.7).

Differently from scenario 7, in scenario 8 the density decreases to 0 and remains constant at this value for a period of around 30 s; this is because the TMS does not route the vehicles in the routing flow along this road anymore, since, on the basis of the information coming from the LC, it has realized that the road is congested. After, the constant phase at 0, vehicles are routed also on road 32 again, and the density increases and reaches a value similar to the one in scenario 7. In other words, the constant phase with null density corresponds to the reaction time of the TMS, which in scenario 8 takes the routing decisions on the basis of the current network state.

On the other hand, in scenario 9 the reaction time of the TMS is close to 0, since (yellow curve), the density start increasing just after the end of the disturbance flow. This is because in this scenario the RC computes the optimal paths of the vehicles considering not only the current network state, but also the future expected one; it is therefore able to anticipate the end of the disturbance flow and thus to route vehicles also on road 32.

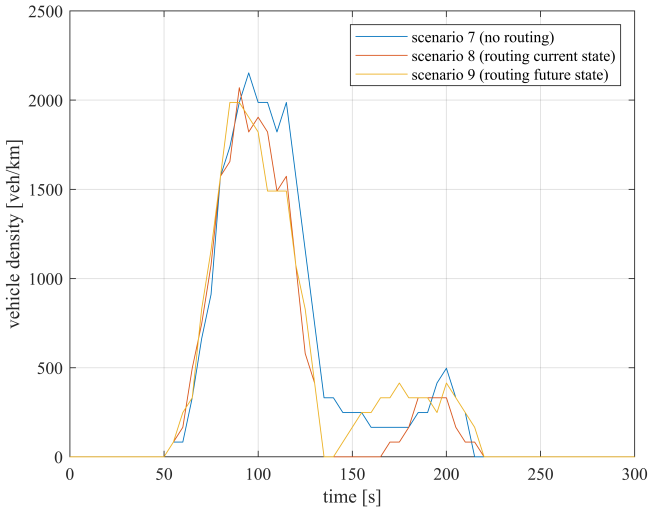


Figure 7.9: Vehicle density on road element 32 as a function of time.

Similar observations can be done also for road 36 (Fig. 7.10). In case of no routing (blue curve) the density remains quite constant around a value of 300 veh/h,

which corresponds to one third of the total density due to the routing flow (Fig. 7.7). On the other hand, in case of routing the density reaches a peak due to the vehicles that are not routed along road 32 because it is congested. However, the peak is observed at different times depending on the scenario; in particular, in case of routing with current network state (orange curve) the peak is delayed with respect to the case of routing with current and future network state (yellow curve). This is because when the TMS considers only the current network state it has an higher reaction time, and thus it starts routing more vehicles on road 36 after the road element 32 is congested. If the TMS also considers the expected future network state, then its reaction time is significantly lower since it starts routing more vehicles on road 36 immediately after it realizes that the road 32 is going to be congested in the next time intervals.

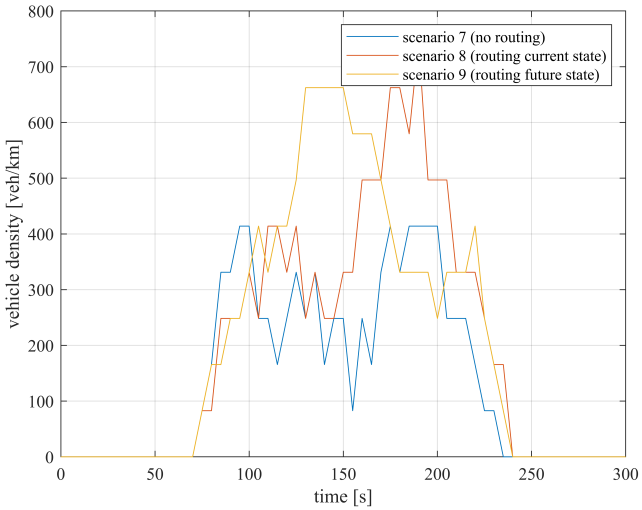


Figure 7.10: Vehicle density on road element 36 as a function of time.

Fig. 7.11 show the vehicle density for road 28. The behavior of the curves is similar to the one of road 36 apart from scenario 9. For this scenario there is not a peak of density, which instead remains constant over time. This can be explained by the capability of the TMS of routing vehicles by considering the expected future state. In particular, in scenario 9 the TMS routes more vehicles on road 36 than on road 28 to partly avoid the congestion on intersection 2, which is expected to increase in the next time intervals due to the vehicle flows that will cross it (both from South to North and from South to West, Fig. 7.12).

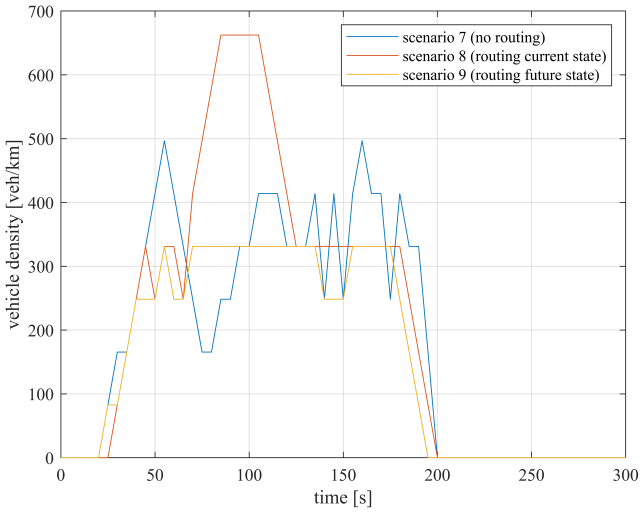


Figure 7.11: Vehicle density on road element 28 as a function of time.

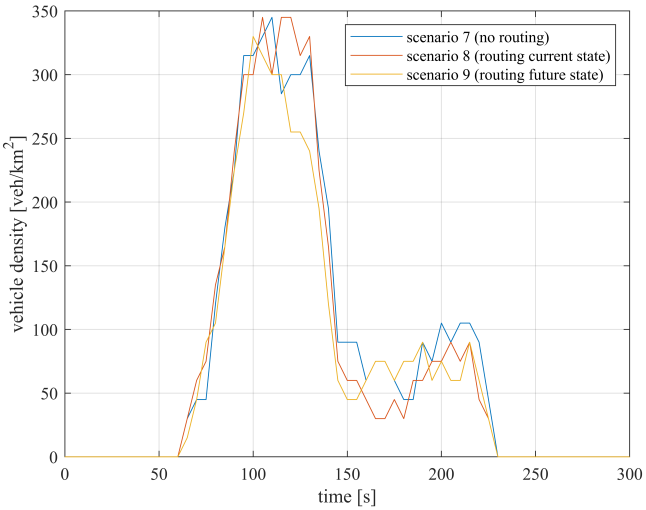


Figure 7.12: Vehicle density on intersection element 2 as a function of time.

Finally, Fig. 7.13 show, for the 3 scenarios, the average travel time on intersection 2, for the left turn maneuver, as a function of time. The increase due the congestion on the intersection starts at $T = 50s$, and it is possible to see how its value is lower when the vehicles are routed by the TMS (orange and yellow curves) than when the vehicles chooses their path randomly (blue curve) and, in particular, significantly lower in the scenario 9 (yellow curve).

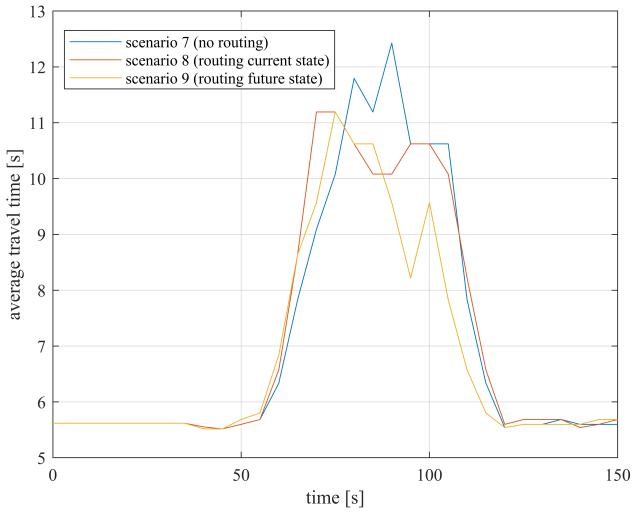


Figure 7.13: Average travel time for the left turn maneuver, on intersection element 2, as a function of time.

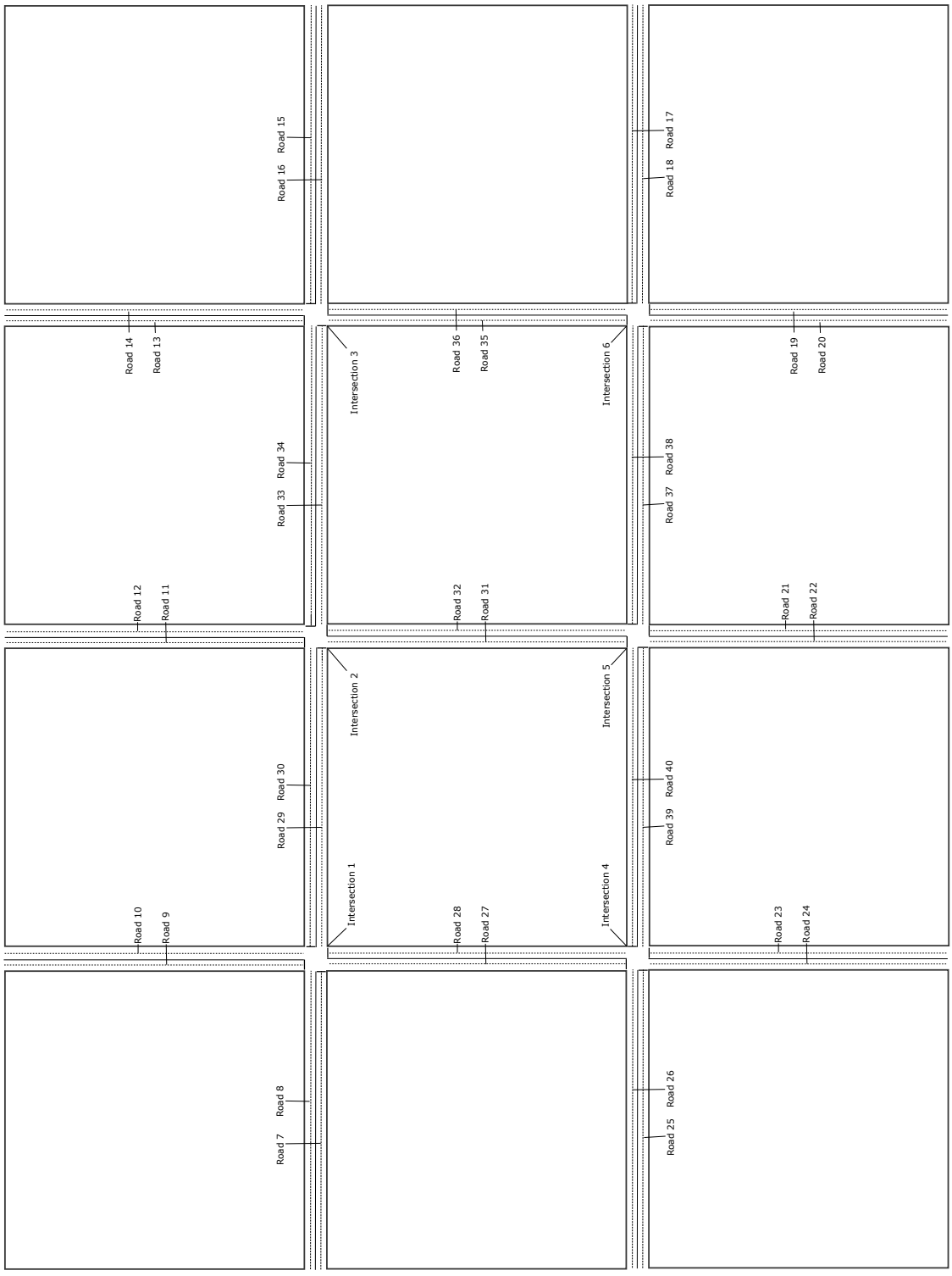


Figure 7.14: Layout of the network considered to test the routing model

7.2 Vehicle scheduling and motion planning in the intersection

In this section, the application of the proposed MILP problem to a case study related to an isolated intersection is discussed with the aim of:

1. showing the vehicles motion planning, and in particular, how the model can safely avoid collisions between vehicles inside the intersection;
2. discussing the effects of the smoothing filter SF;
3. showing the travel time as a function of the vehicle flow;
4. comparing the performances of the described model with those obtainable with a traditional signalized intersection.

In all the analyses, the intersection depicted in Fig. 5.1 is considered, with the model parameters reported in Tab. 7.7. In addition, to conduct some performance analyses, different randomly generated demand scenarios are considered (in the following called 'Low', 'Medium', and 'High'). In doing so, vehicles are assumed to be characterized by Poissonian arrivals (with rate equal to the incoming hourly flows), with the arrival speeds (at triggers) uniformly distributed in $[4, 10]$ ms^{-1}, and with the lane flows and the turning rates reported in Tab. 7.8. As shown in Fig. 5.1, each lane is identified by a cardinal point (North, South, West and East) and by a letter addressing the Left (L) and the Right (R) lane. Given a lane, a cardinal point and a turning direction, it is possible to identify the relevant stream (e.g. vehicles turning left at the South-Right lane are associated to stream 3 shown in Fig. 5.1).

Table 7.2: Intersection model parameters.

Parameter	Value
$v_{\text{int}}^{\max}$	10 ms^{-1}
$v_{\text{rd}}^{\max}$	10 ms^{-1}
Φ_l	0.05 s
$\Phi_{\min}$	0.05 s
Φ_h	0.025 s
Δ_i	50 m
w_u	1
β	1.5 s
α	$\alpha = \hat{\Theta} 2 \hat{SF}$ (unless otherwise stated)

7.2.1 Vehicles motion planning and conflicts management

In this section, the characteristics of feasible solutions are analyzed and discussed. To this aim, consider the frames shown in Figs. 7.15 and 7.16, where consecutive

Table 7.3: Incoming flows for the three demand scenarios.

Lane	Lane flow [veh·h^{-1}]			Turning rates [%]		
	Low	Medium	High	Left	Straight	Right
South L	600	720	1000	20	50	30
South R	600	720	1000	30	50	20
East L	420	560	680	50	25	25
East R	630	840	1020	42	33	25
North L	450	700	825	30	40	30
North R	450	700	825	30	40	30
West L	400	532	960	25	25	50
West R	600	798	1440	25	33	42

positions of two vehicles are depicted with different shades of color. In particular, the darker rectangles represent the current positions of the vehicles in the last frame, whereas the lighter rectangles represent the previous ones.

Looking at the two frames in Fig. 7.15, it is possible to note that the vehicle v in the stream 5 has right-of-way, and so it crosses the intersection before the vehicle u in the stream 2, always traveling at the maximum speed. On the contrary, u decelerates and reaches the point $P_{2,36}$ at $\tau = 1.49$ s, when v is already on the point $P_{5,41}$.

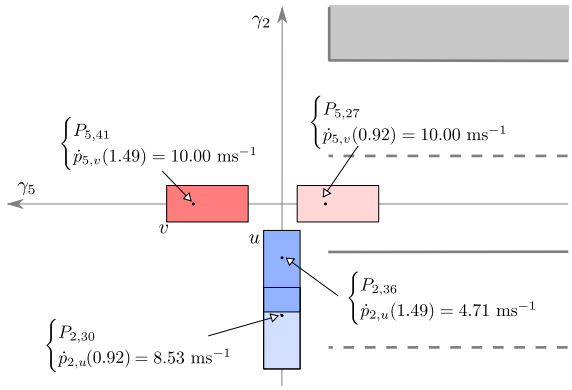


Figure 7.15: First example of vehicles motion planning.

A second example is depicted in Fig. 7.16, where it is possible to note that the vehicle u in the stream 13, traveling at a low speed along the curve trajectory γ_{13} due to the vehicle stability constraint in (5.14), can accelerate as the trajectory curvature decreases, and pass in front of the vehicle v in the stream 3: in other words, u has right of way. At the same time, the vehicle v was initially traveling at a low speed to give way to u but can eventually accelerate to reach the point $P_{3,18}$ when u is on $P_{13,84}$.

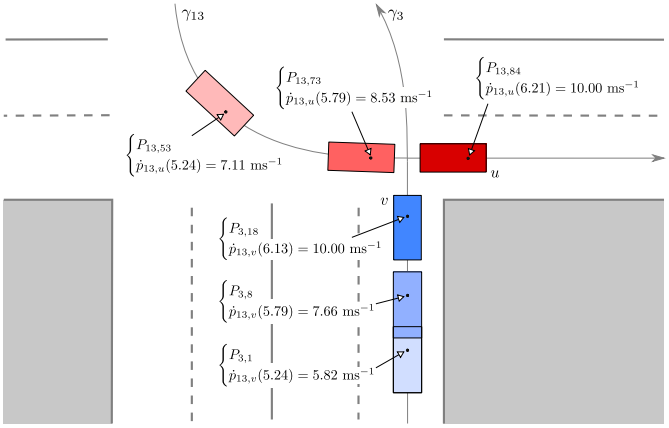


Figure 7.16: Second example of vehicles motion planning: the vehicle u has the right of way and, after traveling at a reduced speed along the trajectory γ_{13}, it accelerates crossing γ_3 area in front of the vehicle v.

7.2.2 Effects of the smoothing filter

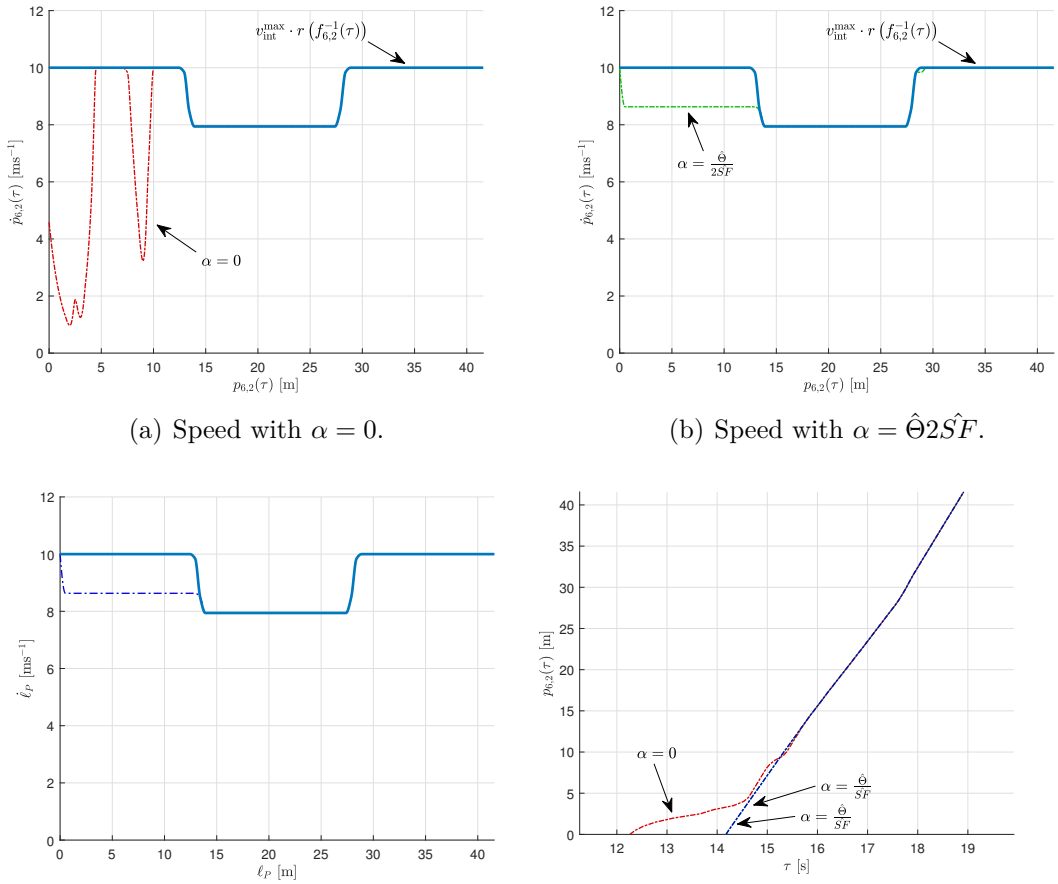
In this section, the effect of the smoothing filter and the solution characteristics for different values of α are discussed. To this aim, three values of such a parameter are considered: $\alpha = 0$, which represent the case without filter, $\alpha = \hat{\Theta}2\hat{SF}$ and $\alpha = \hat{\Theta}\hat{SF}$, which represent the cases where SF is made comparable to half of the average total travel time and to the average total travel time, respectively. In this framework, the values $\hat{\Theta}$ and $\hat{SF}$ have been estimated by averaging the values of the cost function computed by considering only the first term and the second terms in (5.4), respectively.

The results are summarized in Fig. 7.17 and in Fig. 7.18, where the speed of vehicles and the time-space diagrams are reported for the considered values of α. In such figures, it is possible to note that the speeds tend to be smoother and more comfortable when α increases. Obviously, if $\alpha > 0$ the number of accelerations/decelerations decreases although they cannot be totally avoided due to the traffic conditions and the safety constraints. In any case, no significant differences can be identified between the cases $\alpha = \hat{\Theta}2\hat{SF}$ and $\alpha = \hat{\Theta}\hat{SF}$, proving a negligible sensitivity to this parameter.

As for the global effect of the smoothing filter, the results reported in Tab. 7.4 show that it affects negligibly the average Θ. Such a behavior is easily explained by analyzing the time-space diagrams in Fig. 7.17.d and Fig. 7.18.d: with $\alpha = 0$ vehicles reach the first point $P_{i,1}$ of the trajectories with high speed and then decelerate to fulfill the conflict constraints; with $\alpha \neq 0$ vehicles reach $P_{i,1}$ with lower speed, decelerating after passing the trigger. While this second behavior allows avoiding a sudden deceleration of the vehicles inside the intersection, it does not affect the total travel time. In fact, since the travel time of a vehicle is given by the time elapsed from the passage on the trigger and the time the vehicle leaves the intersection, it is not relevant if the vehicles are delayed between the trigger and $P_{i,1}$ or between $P_{i,1}$ and P_{i,n_i}.

Table 7.4: Average total travel time Θ for different values of the parameter α and for the three demand scenarios. Averages are computed over 10 runs.

Demand scenario	$\alpha = 0$	$\alpha = \hat{\Theta}2\hat{SF}$	$\alpha = \hat{\Theta}\hat{SF}$
Low	533.93 s	533.99 s	535.34 s
Medium	891.65 s	891.98 s	890.99 s
High	1357.43 s	1356.10 s	1356.53 s



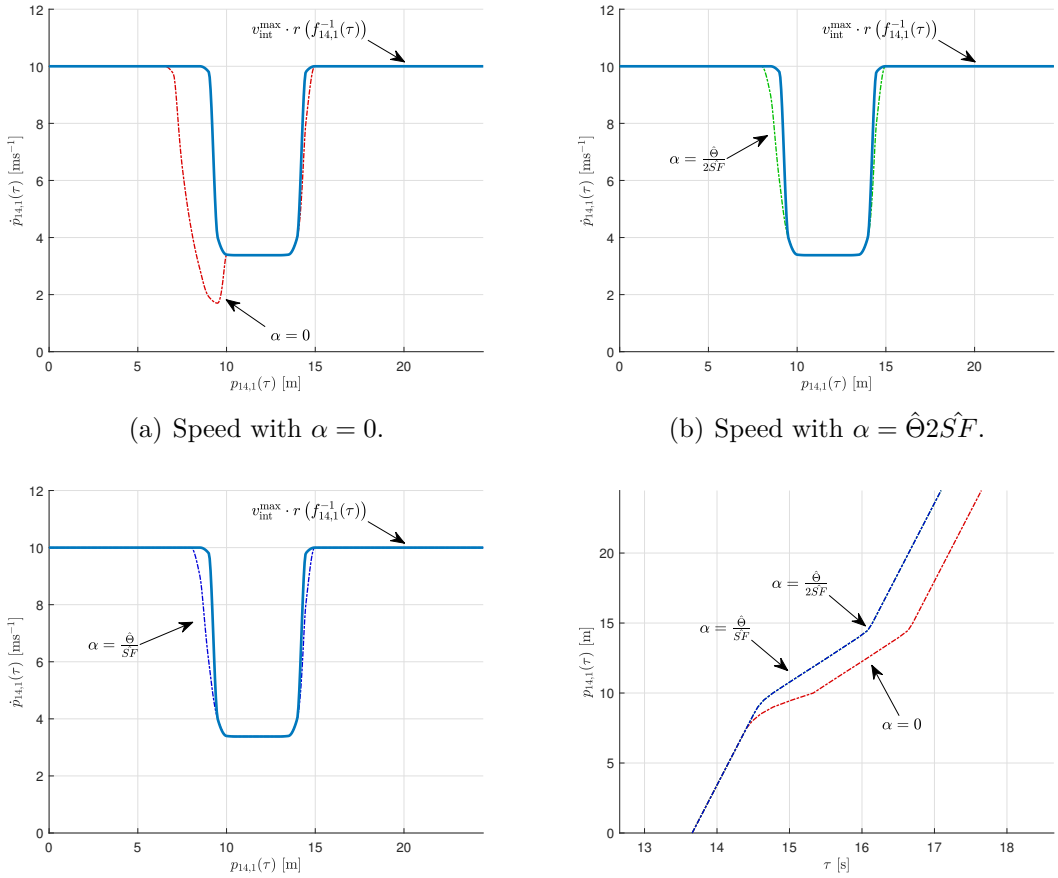
(a) Speed with $\alpha = 0$.

(b) Speed with $\alpha = \hat{\Theta}2\hat{SF}$.

(c) Speed with $\alpha = \hat{\Theta}\hat{SF}$.

(d) Time-space diagram.

Figure 7.17: Example of optimal time-space trajectory and speed (dotted lines) of the second vehicle in the stream 6. The continuous blue lines represent the speed limit. The speeds and the time-space diagrams of the cases $\alpha = \hat{\Theta}2\hat{SF}$ and $\alpha = \hat{\Theta}\hat{SF}$ are indistinguishable.



(a) Speed with $\alpha = 0$.

(b) Speed with $\alpha = \hat{\Theta}2\hat{S}\hat{F}$.

(c) Speed with $\alpha = \hat{\Theta}\hat{S}\hat{F}$.

(d) Time space trajectories.

Figure 7.18: Example of optimal time-space trajectory and speed (dotted lines) of first vehicle in the stream 14. The continuous blue lines represent the speed limit. The speeds and the time-space diagrams of the cases $\alpha = \hat{\Theta}2\hat{S}\hat{F}$ and $\alpha = \hat{\Theta}\hat{S}\hat{F}$ are indistinguishable.

7.2.3 Intersection flow-travel time functions

As mentioned in Sec. 4.3, the expected travel times of the vehicles at the various elements of the network, used to determine the optimal paths, are computed in two steps: firstly, the expected flows on each element at the future intervals is computed by exploiting both the current flows and the vehicle paths already determined; secondly, the expected travel times for each element are computed by means of a series of flow-travel time functions. These functions are computed by executing a second-order polynomial fitting of a set of observed pairs flow-travel time. In the case of the intersection, the observations have been generated by executing the scheduling and motion planning model of the intersection several times, assuming random Poissonian arrivals of the vehicles with interarrival rates ranging from 1 s/veh to 15 s/veh for each of the four directions, with different random combinations.

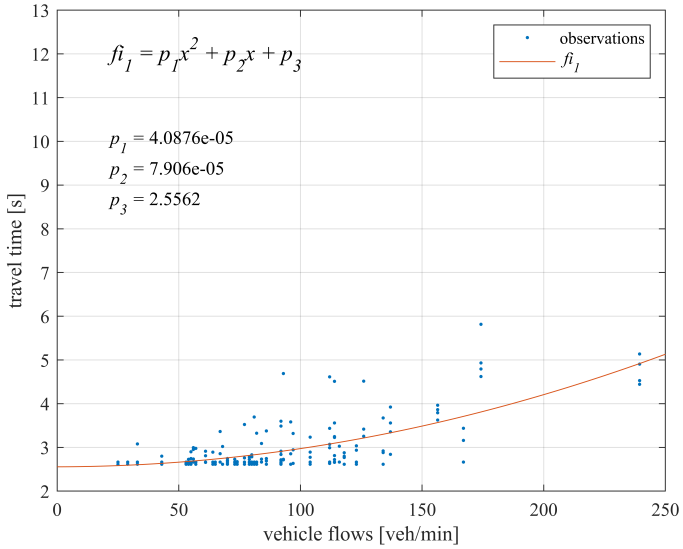


Figure 7.19: Intersection flow-travel time function of type 1.

Fig. 7.19 shows the generated observations and the estimated flow-travel time function of type 1 (fi_1), that is the one related to the right turn maneuver from the right lane. It is possible to note the traditional behavior of the flow-travel time functions, with a constant initial phase where the travel time remains constant to its minimum value, independently from the flow, and a second phase (approximately starting from a flow of 70 veh/min) where the travel time start increasing with the flow.

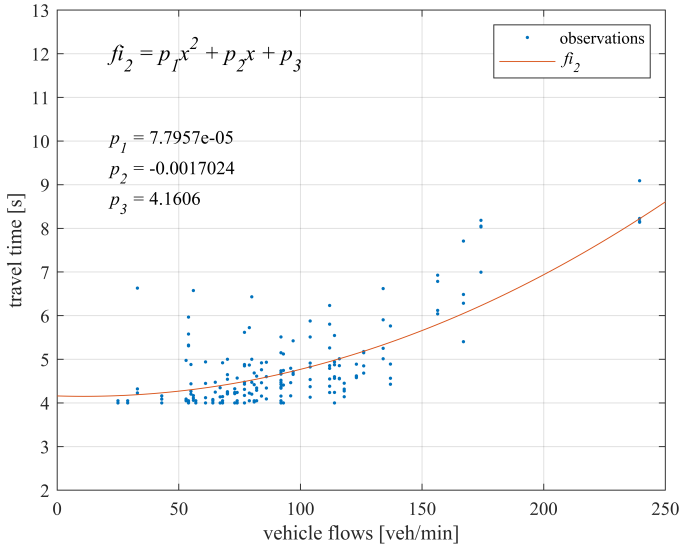


Figure 7.20: Intersection flow-travel time function of type 2.

Fig. 7.20 shows the generated observations and the estimated flow-travel time function of type 2 (fi_2), that is the one related to the go straight maneuver from the right lane. In this case the function is more sensitive to the flow increase, since

this maneuver crosses the entire intersection and thus more streams than the right turn maneuver on function fi_1.

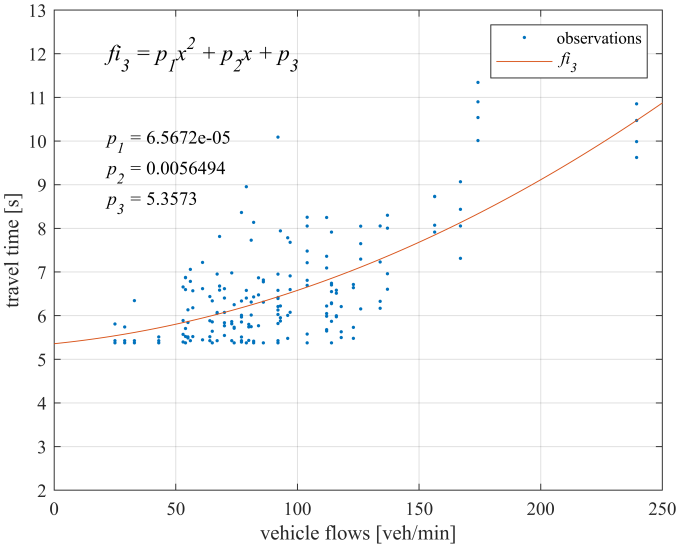


Figure 7.21: Intersection flow-travel time function of type 3.

Fig. 7.21 shows the generated observations and the estimated flow-travel time function of type 3 (fi_3), that is the one related to the left turn maneuver, from the right lane. This kind of maneuver, usually not allowed in real intersections, is made possible by the proposed model thanks to the coordination of the self-driving vehicles. As expected, as for the go straight maneuver the travel times are more sensitive to the increase of flows, since also this maneuver crosses the entire intersection and the majority of the other streams. For the same reason, it is also interesting to note that the constant phase is significantly shorter that the one of the right turn maneuver, and that the travel time start soon to increase with the flow.

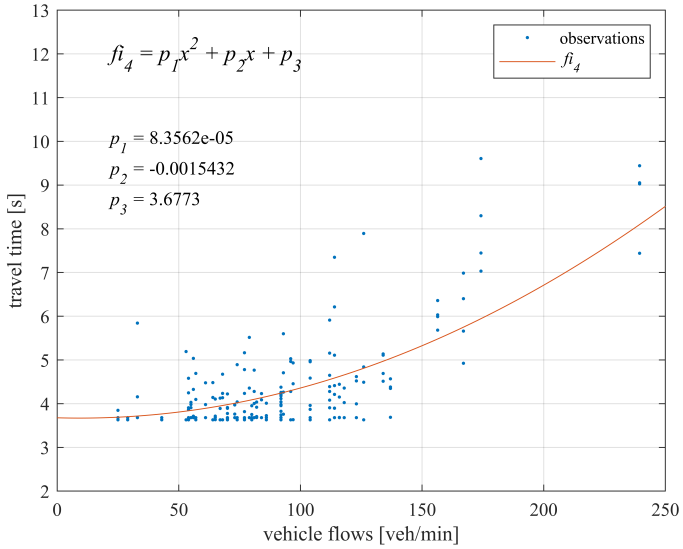


Figure 7.22: Intersection flow-travel time function of type 4.

Fig. 7.22 shows the generated observations and the estimated flow-travel time function of type 4 (fi_4), that is the one related to the turn right maneuver from the left lane. As the left turn from the right lane, this maneuver is made possible thanks to the proposed model.

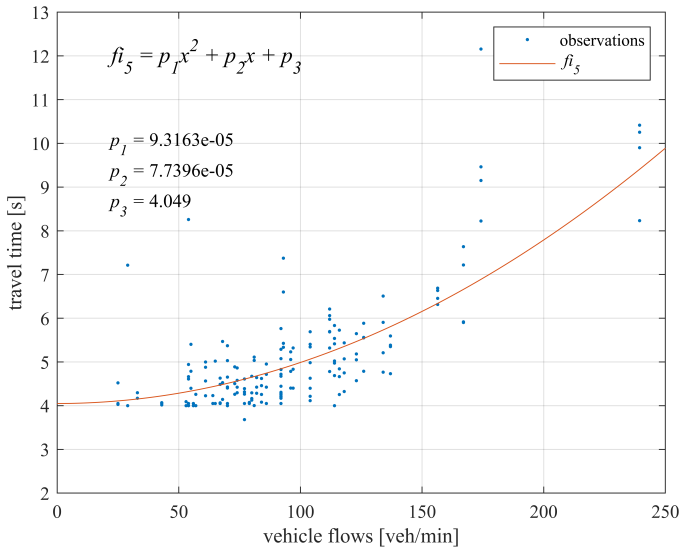


Figure 7.23: Intersection flow-travel time function of type 5.

Fig. 7.23 shows the generated observations and the estimated flow-travel time function of type 5 (fi_5), that is the one related to the go straight maneuver from the left lane.

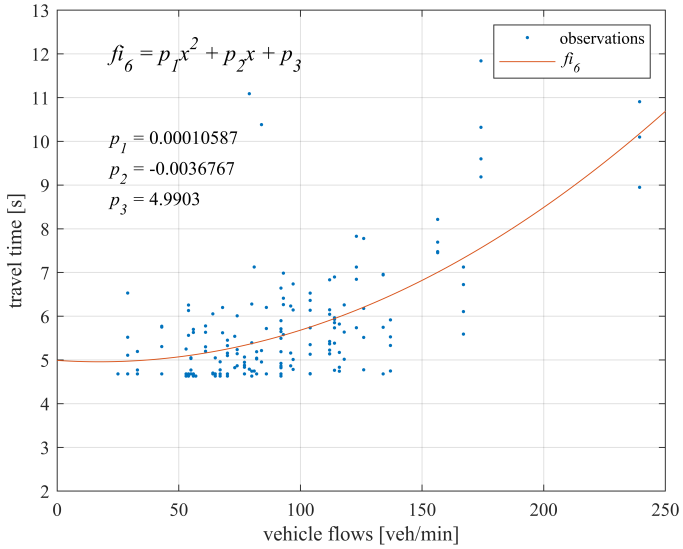


Figure 7.24: Intersection flow-travel time function of type 6.

Finally, Fig. 7.24 shows the generated observations and the estimated flow-travel time function of type 6 (fi_6), that is the one related to the left turn maneuver from the laft lane.

7.2.4 Simulation results

To analyze the performance of the proposed approach, the results obtainable with a traditional, traffic light-based, solution have been computed and compared on the basis of:

1. *Total travel time* computed over all the vehicles entering the intersection area in 60 s;

2. *Number of vehicles* crossing the intersection in 60 s.

In both cases the intersection of Fig. 5.1 is considered. The values of the above performance metrics are estimated both for the proposed method (directly from the model output), and for a more traditional control policy simulated via PTV Vissim™ (10 independent runs of 660 s; for each run the performance metrics have been computed every 60 s; results are the averages over the runs of the averages over the simulation time). As regards the traditional control policy, a traffic light with 8 movements and 4 phases has been considered. In fact, in a traditional intersection with traffic light the left lanes allow only the left turnings, and the right lanes allow only going straight or turning right, as shown in Fig. 7.25. Concerning the traffic scenario, the incoming flows are those reported in Tab. 7.8, although, for each of the four directions, they have been redistributed between the left and the right lanes to consider the new layout.

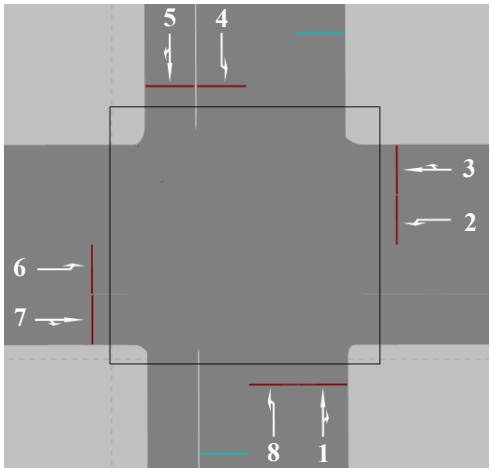


Figure 7.25: Layout of the intersection considered in the simulation and movement numbers.

The traffic light setup has been designed by means of the optimization model proposed in [63], which, starting from the definition of the movements and their incompatibilities, determines the phase sequences and lengths that maximize the intersection capacity.

Tab. 7.5 shows the results, together with the saturation flow of each movement which has been evaluated with PTV Vissim™. The movement numbering refers to Fig. 7.25. Furthermore, the cycle time is set to 120 s (a common value for intersections with this layout and traffic volume), the lost time is null, and the traffic light has only green/red lights. It is worth pointing out that these last assumptions favor the traffic light intersection performance.

Total travel time

The results in Tab. 7.6 show that the proposed method sensibly reduces the travel time inside the intersection area: the greatest advantage is in the low demand scenario, where the delaying effect of the traffic light is more evident, while the proposed method allows vehicles to cross the intersection without stops.

Vehicle crossings

The proposed method always performs much better than the traffic light regulation described above also for the vehicle crossings, especially in the medium and high demand scenario, with a particularly interesting increase ratio of 1.85 in this last case.

7.3 Vehicle scheduling and motion planning in the road element and in the network

In this section, the considered model is applied to the road elements with the aim of showing and discussing the optimal solution provided by the solver. In particular:

Table 7.5: Traffic light setup.

Phase	Movements activated (sat. flow [veh·h^{-1}])	Green time interval [s]		
		Low	Medium	High
1	4 (2500) - 8 (2500)	[110, 120)	[108, 120)	[0, 10)
2	2 (2500) - 6 (2500)	[46, 64)	[0, 19)	[10, 25)
3	1 (2000) - 5 (1700)	[0, 46)	[66, 108)	[25, 63)
4	3 (1800) - 7 (1600)	[64, 110)	[19, 66)	[63, 120)

Table 7.6: Simulation results. Averages computed over 10 runs.

	Travel time [s]		Vehicle crossings	
Demand scenario	Traffic light	Problem in (5.4)–(5.17)	Traffic light	Problem in (5.4)–(5.17)
Low	2371	549	63	70
Medium	2714	897	72	92
High	2895	1271	72	133

- two specific scenarios (vehicles traveling on conflicting trajectories and overtaking maneuvers) are reported in subsections 7.3.1 and 7.3.2 to show how the road model can avoid collisions between vehicles and allow the overtakings;
- subsection 7.3.3 shows the travel time as a function of the flow;
- subsection 7.3.4 describes the application of the model to the layouts of traffic networks illustrated in Figs. 3.2 and 3.3.

The parameters of the model are those reported in Tab. 7.7, where it is possible to note that both the maximum speed and the spatial discretization of trajectories are different for road and intersection elements. The MILP model was coded and solved with IBM^{TM}Cplex^{TM}12 Optimizer and the experiments were performed on a standard Intel^{TM}Xeon^{TM}CPU @3.4GHz processor and 8 Gb RAM laptop.

7.3.1 Vehicles conflict management

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

With reference to the trajectories illustrated in Fig. 7.26, 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 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

Table 7.7: Road model parameters.

Parameter	Value
$v_{\text{road}}^{\max}$	14 ms^{-1}
$v_{\text{rd}}^{\max}$	10 ms^{-1}
Φ_l	0.05 s
$\Phi_{\min}$	0.05 s
Φ_h	0.025 s
δ	1.0 m
Δ	50 m
w_u	1 (unless otherwise stated)
β	0.0 s

Table 7.8: Arrival rates

Entering lane	Rate ($\text{s}\cdot\text{h}^{-1}$)
right lane	2900
left lane	2150

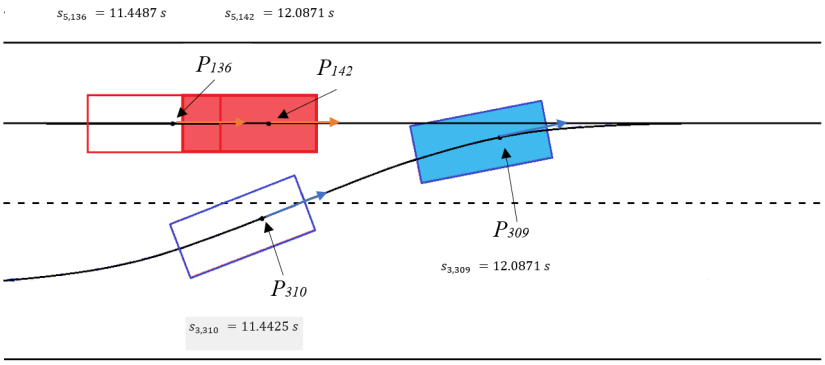


Figure 7.26: Example of timings in the road.

and reported in Fig. 7.26; 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. 7.27, 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).

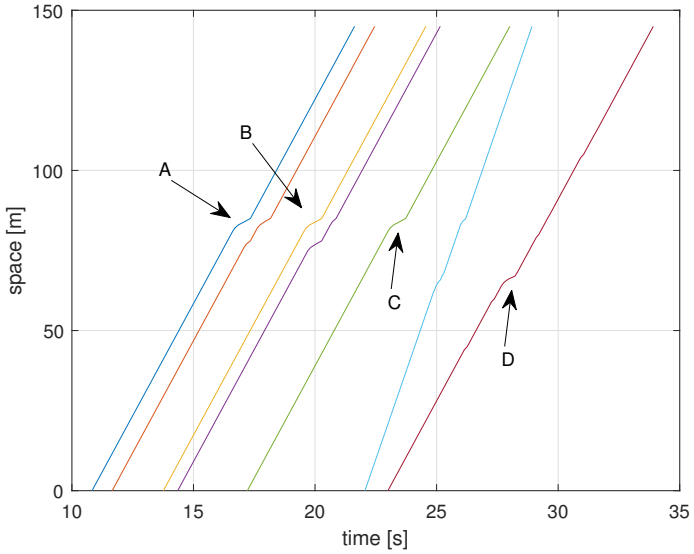


Figure 7.27: Optimal time-space trajectories of a set of vehicles in a generic path inside a road. Each letter points out the vehicle deceleration.

7.3.2 The overtaking

An example of overtaking maneuver 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 $w_B = 2\,w_A$, while the maximum speeds allowed are 14 ms^{-1} 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. 7.28 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. 7.29 reports the optimal space-time trajectories of the two vehicles traveling 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, approximately 50 meters from the beginning of the road. In Fig. 7.29, 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.

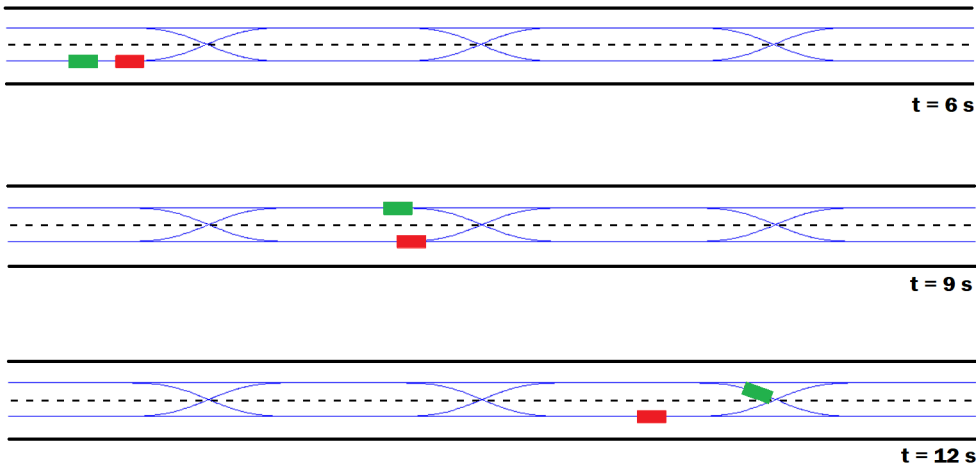


Figure 7.28: Snapshot of the road at $t = 6$ s, $t = 8$ s and $t = 10$ s. The leader is red and the follower is green.

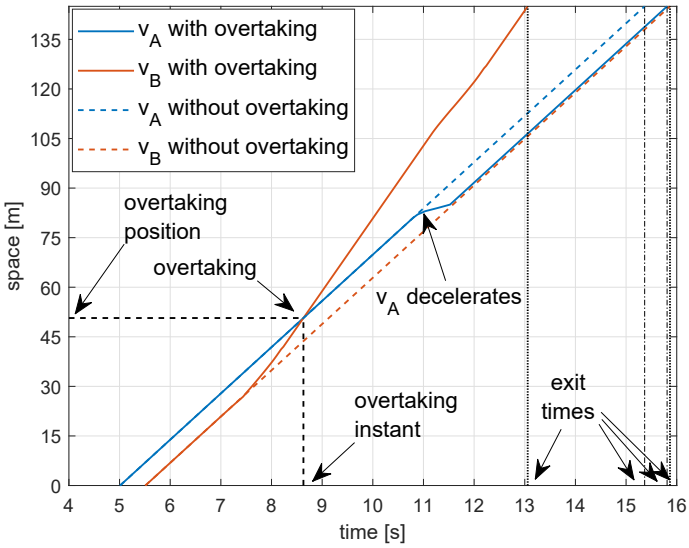


Figure 7.29: Optimal space-time position of the two vehicles in the road (solid curves in case of overtaking allowed and dashed curved with no overtaking).

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. 7.29 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 penalized (2.9 s delay) and the value of the objective function significantly increases.

7.3.3 Road flow-travel time function

In the case of the road element, the flow-travel time function has been estimated without distinguishing among the possible maneuvers of the vehicles inside the element, because these latter basically consist of lane changes that do not affect significantly the travel time inside the element. For the road the flow-travel time function has been estimated with a cubic spline interpolation, which turned out to approximate data in a better way than polynomial fitting. The observations in this case are the mean between the flow observed in each of 10 executions of the road scheduling and motion planning model with a certain value of interarrival rates ranging from 0.5 s/veh to 15 s/veh, and in each execution the random samples have been generated by assuming random Poissonian arrivals of the vehicles.

From Fig. 7.30 it is possible to see that the amount of travel time increase is lower the the ones observed in the intersections, due to the reduced number of interferences in the road element. In particular, the minimum travel time in the road is 10.9s, which can increase up to 12.5s for high flows.

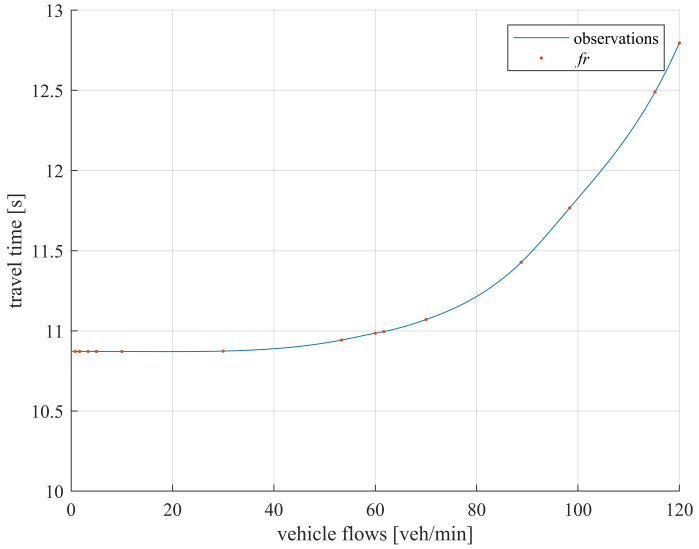


Figure 7.30: Road flow-travel time function.

7.3.4 The crossing of a network

The global performance of the network in terms of speed and travel times of vehicles traveling the network has been analyzed. The layouts considered in this numerical experiment are those depicted in Figs. 3.2 and 3.3.

In Fig. 7.31, the space-time trajectory of a vehicle which enters the network of Fig. 3.2 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).

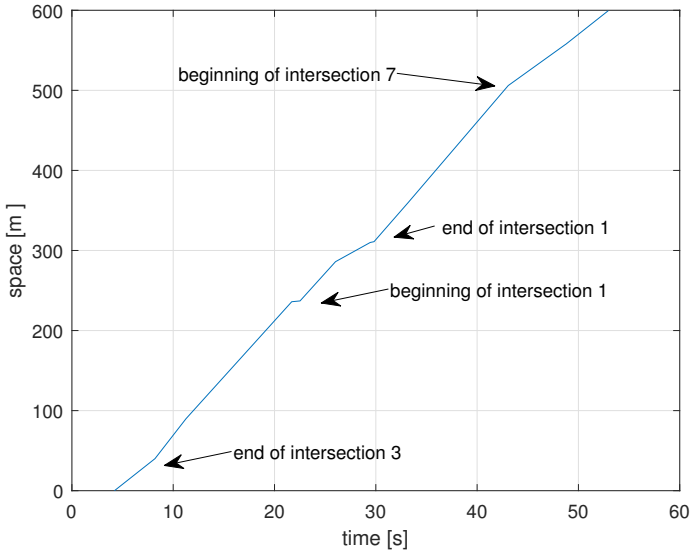


Figure 7.31: Space-time trajectory of a vehicle in the network in Fig. 3.2.

A second example is reported in Fig. 7.32, where it is possible to see the time-space representation of the optimal trajectories of three vehicles which enter the network of Fig. 3.3 at intersection 1 and exit at intersection 3. In this case, it is possible to note that 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. 7.32 show that the average speed of vehicle 3 is always greater than the speed of the other two vehicles.

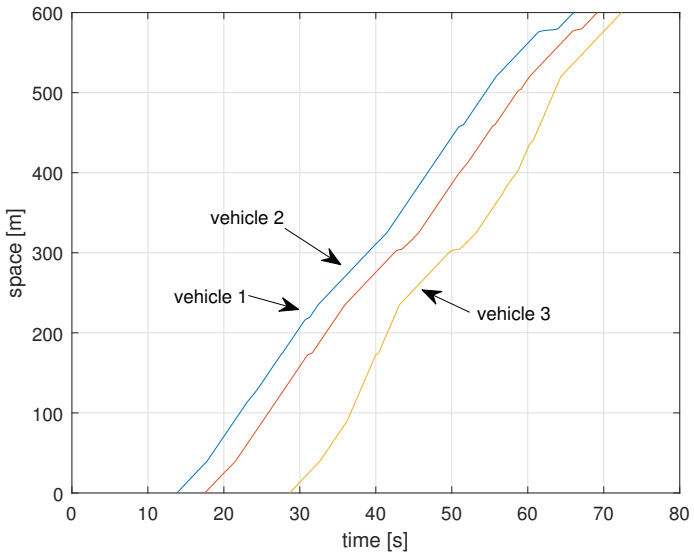


Figure 7.32: Time-space representation of the optimal trajectories of three vehicles in the network in Fig. 3.3.

7.4 Computational complexity

The aim of this section is to discuss the computational effort needed to solve the scheduling and motion planning problem for the intersection and for the road, considering case studies characterized by a time horizon of 60 seconds. In this regards, three solution approaches are compared:

a) the time horizon is solved as a whole, that is, considering the problem with all the vehicles that arrive at the intersection in the interval $[0, 60]$;

b) the time horizon is divided into twelve time windows with $\Delta\tau = 5$ s;

c) the time horizon is divided into twenty time windows with $\Delta\tau = 3$ s.

In the second and in the third case, the rolling horizon framework described in Sec. 5.4 is adopted.

Table 7.9: Comparison between the three solution approaches in terms of the average minimized total travel time Θ and the average computational time CT for the intersection. Averages are computed over 10 runs.

Demand scenario		Low	Medium	High
Average number of vehicles		57	92	133
a	Θ_a	536.15 s	887.43 s	1326.92 s
	CT_a	100.94 s	11.4 hours s	2 days[1]
b	Θ_b	548.98 s	896.73 s	1271.25 s
	CT_b	0.278 s	1.09 s	2.37 s
c	Θ_c	556.72 s	936.17 s	1294.19 s
	CT_c	0.40 s	0.475 s	0.67 s

The solutions have been computed by means of the solver IBM™ Cplex™ on a Intel™ Xeon™ CPU @ 3.4GHz and 8 Gb RAM. Tab. 7.9 shows, for the intersection, the total travel time Θ and the computational time CT for the three solution approaches and for the three demand scenarios. With approach a, the required computational time for all the three demand scenarios is too high and not compatible with real world applications. In particular, the search for the solution in the high demand scenario was stopped after two days and the found solution is suboptimal, as it is possible to see in comparison with those found with the approaches b and c. On the contrary, with the other approaches the average CT is always largely less than 5 s (b) and 3 s (c); this means that the optimal solution can be found within the relevant $\Delta\tau$ also providing an estimate of the maximum allowed time for solution ϵ. Note that the CT in the approach c is averagely less than those in the approach b, as with $\Delta\tau = 3$ s few vehicles are considered in any instance. However, it is interesting to note that in the scenario with low demand, $CT_b < CT_c$ due to

[1]Best solution value after two days of processing time.

the overhead required to 'prepare' and solve a higher number of sub-problems, even if they are simpler.

As for the performances, the total travel time Θ results to be better with the approach a, as expected, since the problem considers all the vehicles at a time (apart from the high demand scenario where only a suboptimal solution was found). However, the differences between Θ_b and Θ_c with respect to Θ_a are negligible. The approach b provides better (lower) values of the total travel time than the approach c thanks to the higher number of vehicles that are considered in any instance of the approach b. Therefore, a balance between the acceptable sub-optimality of the solution and the relevant computational time has to be a-priori chosen, mostly depending on the computational resources that can be involved in the problem solution.

To summarize, the computational complexity depends on the time windows $\Delta\tau$ and on the traffic flows, since the more vehicles are traveling in the element, the more variables and constraints are considered in the related MILP problem.

Experimental results, obtained by randomly generating the arrival times of vehicles in accordance with a Poissonian processes characterized by different rates, show that the solution time is in general less than the chosen time horizon $\Delta\tau = 5$s, even with high vehicle flows (almost 3000 vehicles per hours).

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. 7.33 and 7.34, 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. 6.1 and the intersection of Fig. 5.1, respectively. Fig. 7.33 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.

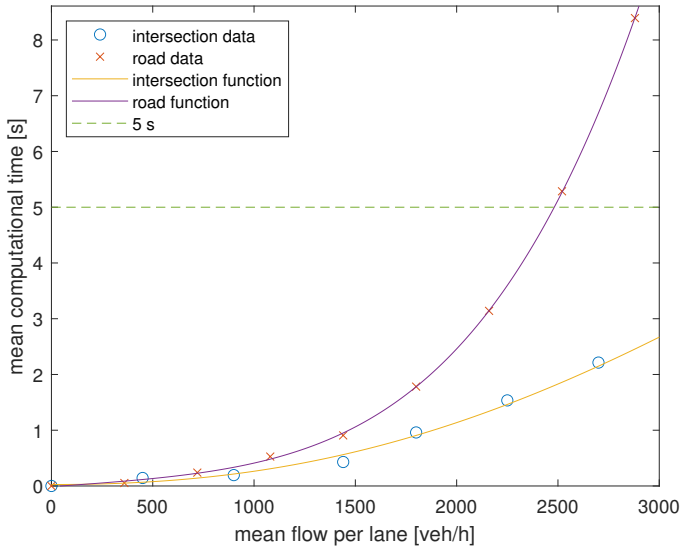


Figure 7.33: Mean computational times (blu and red symbols) and computational complexity trends (orange and purple curves) as a function of the arrival rates, with $\Delta\tau = 5$s.

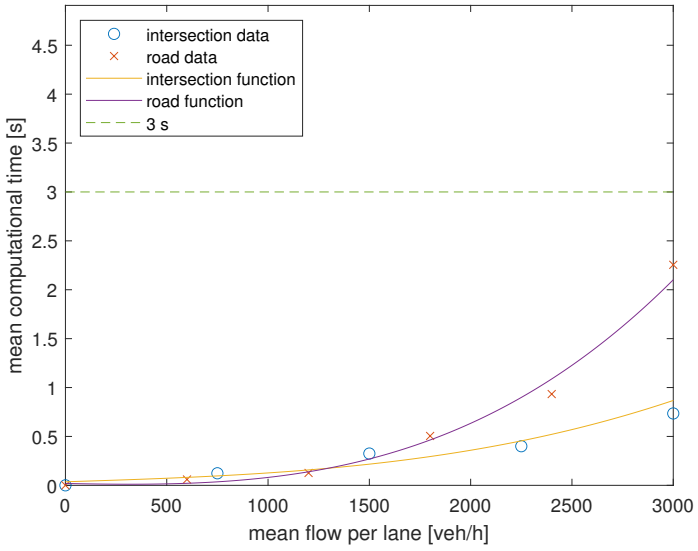


Figure 7.34: Mean computational times (blu and red symbols) and computational complexity trends (orange and purple curves) as a function of the arrival rates, with $\Delta\tau = 3$s.

Chapter 8

Conclusions

This thesis proposes a novel Traffic Management System (TMS) for the combined routing, scheduling and motion planning of self-driving vehicles in reserved road networks. The TMS architecture is based on three different levels, where at each of them a controller performs specific tasks. In particular, at the routing level the vehicles that are going to enter the considered road network are optimally routed by solving a proper routing model, which is formalized as an optimization model that takes into account the information regarding the network state coming from the system. At the local level, the controller of each element (road or intersection) composing the network builds and solves a scheduling and motion planning problem, formalized as an optimization model, that finds the optimal trajectories and speeds of each vehicle in the element that ensure minimum travel times while avoiding collisions and guaranteeing safety and comfort. Between the routing and the local level, the network level manages the exchange of information and the actions between them.

The results of the application of the proposed TMS to a series of case studies allow to show the good performance achieved at the various levels. In particular, at the routing level vehicles are effectively distributed on the network and routed on the less congested paths. This allow a reduction of the total travel times and an optimal distribution of the vehicles among the available alternatives. The continuous exchange of information between the various levels and components of the TMS allows to consider always up-to-date network state information, and thus to allow vehicles to avoid the most congested portions of the network as soon as the vehicle density start increasing or is expected to increase in the next time intervals (as an effect of the vehicles already in the network). At the local level, results show how the TMS is able to optimally schedule the vehicle crossings in each element while guaranteeing enough safety and comfort, and provide evidence of its effectiveness also compared with traditional control methods.

The optimization problem solved at the local level is computationally hard to solve, but the thesis proposes an heuristic method to find a suboptimal solution in a time compatible with the considered application.

The proposed TMS may find application, in a real-world scenario, in portions of urban or suburban road networks that could be isolated by the rest of the network and that could be entirely reserved to self-driving vehicles (e.g., fast roads). In such

case, the self-driving vehicles may perform autonomous driving tasks in the mixed-flow roads (crossed also by human-driven vehicles, bicycles and pedestrians), while they would be controlled by the TMS only in the portions of network reserved to them. In this way the existence of vehicles with a lower level of automation may be taken into account during the transition phase that is expected from human-driven vehicles to self-driving ones.

Moreover, the TMS easily allows to put into action and test different transport policies, like the prioritization of public transport vehicles (both at the intersections, where they may be given the right of way over all the other vehicles, and along the roads, where they may be allowed to travel at higher speeds and to overtake the other vehicles). Another example may be to penalize or prioritize vehicles depending on the number of occupants, so to encourage car sharing or car pooling practices, or depending on their dimension and propulsion system (so to foster, for instance, the use of small-size vehicles with low-emission propulsion systems).

The TMS may also be applied on the highways, where, for their characteristics, an exclusive flow of self-driving vehicles may be expected sooner than in urban roads. In such roads the TMS would be particularly effective in managing the entrance or the exit of the vehicles in the highway (that could be considered as a particular type of intersection) as well as the overtake between vehicles.

As a matter of fact, even if the thesis proposes and tests some applications, the TMS architecture and the optimization models are general enough to be adapted and applied, with few modifications, to cope with different scenarios, applications and vehicle maneuvers.

Possible future extensions of this work may include different activities: firstly, the models for different more complex layouts of roads, intersections and networks could be tested; secondly, another future development could be the analysis of interactions of self-driving vehicles with other kinds of less-advanced CAVs.

List of Figures

List of Tables

Bibliography

[1] "World urbanization prospects: The 2018 revision," United Nations, Department of Economic and Social Affairs, Population Division, Tech. Rep.

[2] ERTRAC Working Group "Connectivity and Automated Driving", *Connected Automated Driving Roadmap - Version 8*, 2019.

[3] "The Strategic Transport Research and Innovation Agenda (STRIA) - Roadmap for Cooperative, Connected and Automated Transport, 2016."

[4] "The Strategic Transport Research and Innovation Agenda (STRIA) - Roadmap for Low-emission Alternative Energy for Transport, 2016."

[5] SAE International, *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*, apr 2021.

[6] "1st European Conference on Connected and Automated Driving, 2017, Bruxelles, Belgium."

[7] "Automated Vehicles Symposium 2017, 2017, San Francisco, CA,USA."

[8] D. J. Findley, *Traffic Engineering Studies*. John Wiley & Sons, Ltd, 2015, ch. 4, pp. 109–148.

[9] E. H. Choi, "Crash Factors in Intersection-Related Crashes: An On-Scene Perspective," NHTSA's National Center for Statistics and Analysis, Tech. Rep., Sep. 2010.

[10] L. Li, D. Wen, and D. Yao, "A survey of traffic control with vehicular communications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 1, pp. 425–432, Feb 2014.

[11] S. I. Guler, M. Menendez, and L. Meier, "Using connected vehicle technology to improve the efficiency of intersections," *Transportation Research Part C: Emerging Technologies*, vol. 46, pp. 121 – 131, 2014.

[12] A. Bazzi, A. Zanella, and B. M. Masini, "A distributed virtual traffic light algorithm exploiting short range v2v communications," *Ad Hoc Networks*, vol. 49, pp. 42 – 57, 2016.

[13] P. Gora and I. Rub, "Traffic models for self-driving connected cars," *Transportation Research Procedia*, vol. 14, pp. 2207 – 2216, 2016, transport Research Arena TRA2016.

[14] K. Yang, S. I. Guler, and M. Menendez, "Isolated intersection control for various levels of vehicle technology: Conventional, connected, and automated vehicles," *Transportation Research Part C: Emerging Technologies*, vol. 72, pp. 109–129, 2016.

[15] Z. Li, L. Elefteriadou, and S. Ranka, "Signal control optimization for automated vehicles at isolated signalized intersections," *Transportation Research Part C: Emerging Technologies*, vol. 49, pp. 1 – 18, 2014.

[16] X. Zhou, "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," *Transportation Research Part B: Methodological*, vol. 105, no. C, pp. 479–506, 2017.

[17] W. Sun, J. Zheng, and H. X. Liu, "A capacity maximization scheme for intersection management with automated vehicles," *Transportation Research Procedia*, vol. 23, pp. 121–136, 2017, papers Selected for the 22nd International Symposium on Transportation and Traffic Theory Chicago, Illinois, USA, 24-26 July, 2017.

[18] M. A. S. Kamal, J. Imura, A. Ohata, T. Hayakawa, and K. Aihara, "Coordination of automated vehicles at a traffic-lightless intersection," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, 2013, pp. 922–927.

[19] M. A. S. Kamal, J. i. Imura, T. Hayakawa, A. Ohata, and K. Aihara, "A vehicle-intersection coordination scheme for smooth flows of traffic without using traffic lights," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 3, pp. 1136–1147, June 2015.

[20] J. Lee and B. Park, "Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 1, pp. 81–90, March 2012.

[21] F. Yan, M. Dridi, and A. El Moudni, "An autonomous vehicle sequencing problem at intersections: a genetic algorithm approach," *International Journal of Applied Mathematics and Computer Science*, vol. 23, no. 1, pp. 183–200, 2013.

[22] J. Wu, A. Abbas-Turki, and A. El Moudni, "Cooperative driving: an ant colony system for autonomous intersection management," *Applied Intelligence*, vol. 37, no. 2, pp. 207–222, 2012.

[23] J. Wu, A. Abbas-Turki, and F. Perronnet, "Cooperative driving at isolated intersections based on the optimal minimization of the maximum exit time," *International Journal of Applied Mathematics and Computer Science*, vol. 23, no. 4, pp. 773–785, 2013.

[24] G. Rodrigues de Campos, P. Falcone, R. Hult, H. Wymeersch, and J. Sj oberg, "Traffic coordination at road intersections: Autonomous decision-making algorithms using model-based heuristics," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 1, pp. 8–21, 2017.

[25] A. Katriniok, P. Kleibaum, and M. Josevski, "Distributed model predictive control for intersection automation using a parallelized optimization approach," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 5940–5946, 2017, 20th IFAC World Congress.

[26] F. Zhu and S. V. Ukkusuri, "A linear programming formulation for autonomous intersection control within a dynamic traffic assignment and connected vehicle environment," *Transportation Research Part C: Emerging Technologies*, vol. 55, pp. 363 – 378, 2015.

[27] A. A. Malikopoulos, C. G. Cassandras, and Y. J. Zhang, "A decentralized energy-optimal control framework for connected automated vehicles at signal-free intersections," *Automatica*, vol. 93, pp. 244–256, 2018.

[28] P. Tallapragada and J. Cortes, "Coordinated intersection traffic management," *IFAC-PapersOnLine*, vol. 48, no. 22, pp. 233 – 239, 2015, 5th IFAC Workshop on Distributed Estimation and Control in Networked Systems NecSys 2015.

[29] B. Yang and C. Monterola, "Efficient intersection control for minimally guided vehicles: A self-organised and decentralised approach," *Transportation Research Part C: Emerging Technologies*, vol. 72, pp. 283 – 305, 2016.

[30] E. R. Muller, R. C. Carlson, and W. K. Junior, "Intersection control for automated vehicles with MILP," *IFAC-PapersOnLine*, vol. 49, no. 3, pp. 37 – 42, 2016.

[31] W. Wu, Y. Liu, Y. Xu, Q. Wei, and Y. Zhang, "Traffic control models based on cellular automata for at-grade intersections in autonomous vehicle environment," *Journal of Sensors*, vol. 2017, 2017, cited By 4.

[32] L. Cruz-Piris, M. Lopez-Carmona, and I. Marsa-Maestre, "Automated optimization of intersections using a genetic algorithm," *IEEE Access*, vol. 7, pp. 15 452–15 468, 2019.

[33] J. V. Saians-Vazquez, E. F. Ordonez-Morales, M. Lopez-Nores, Y. Blanco-Fernandez, J. F. Bravo-Torres, J. J. Pazos-Arias, A. Gil-Solla, and M. Ramos-Cabrer, "Intersection intelligence: Supporting urban platooning with virtual traffic lights over virtualized intersection-based routing," *Sensors (Switzerland)*, vol. 18, no. 11, 2018.

[34] Y. Bichiou and H. Rakha, "Real-time optimal intersection control system for automated/cooperative vehicles," *International Journal of Transportation Science and Technology*, vol. 8, no. 1, pp. 1–12, 2019.

[35] B. Liu, Q. Shi, Z. Song, and A. El Kamel, "Trajectory planning for autonomous intersection management of connected vehicles," *Simulation Modelling Practice and Theory*, vol. 90, pp. 16–30, 2019.

[36] G. Lu, Y. Nie, X. Liu, and D. Li, "Trajectory-based traffic management inside an autonomous vehicle zone," *Transportation Research Part B: Methodological*, vol. 120, pp. 76 – 98, 2019.

[37] C. Wuthishuwong and A. Traechtler, "Consensus-based local information coordination for the networked control of the autonomous intersection management," *Complex and Intelligent Systems*, vol. 3, no. 1, pp. 17–32, 2017.

[38] S. Fayazi and A. Vahidi, "Mixed-integer linear programming for optimal scheduling of autonomous vehicle intersection crossing," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 3, pp. 287–299, 2018.

[39] L. Li and X. Li, "Parsimonious trajectory design of connected automated traffic," *Transportation Research Part B: Methodological*, vol. 119, pp. 1 – 21, 2019.

[40] K. Zhang, D. Zhang, A. de La Fortelle, X. Wu, and J. Gregoire, "State-driven priority scheduling mechanisms for driverless vehicles approaching intersections," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2487–2500, Oct 2015.

[41] Y. Guan, Y. Ren, S. E. Li, Q. Sun, L. Luo, and K. Li, "Centralized Cooperation for Connected and Automated Vehicles at Intersections by Proximal Policy Optimization," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 12 597–12 608, Nov. 2020.

[42] C. Liu, Y. Zhang, T. Zhang, X. Wu, L. Gao, and Q. Zhang, "High Throughput Vehicle Coordination Strategies at Road Intersections," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 14 341–14 354, Dec. 2020.

[43] X. Chen, Y. Sun, Y. Ou, X. Zheng, Z. Wang, and M. Li, "A Conflict Decision Model Based on Game Theory for Intelligent Vehicles at Urban Unsignalized Intersections," *IEEE Access*, vol. 8, pp. 189 546–189 555, 2020.

[44] P. Karthikeyan, W.-L. Chen, and P.-A. Hsiung, "Autonomous Intersection Management by Using Reinforcement Learning," *Algorithms*, vol. 15, no. 9, p. 326, Sep. 2022.

[45] A. I. Mahbub, A. A. Malikopoulos, and L. Zhao, "Decentralized optimal coordination of connected and automated vehicles for multiple traffic scenarios," *Automatica*, vol. 117, p. 108958, Jul. 2020.

[46] A. Mostafizi, C. Koll, and H. Wang, "A Decentralized and Coordinated Routing Algorithm for Connected and Autonomous Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 11 505–11 517, Aug. 2022.

[47] Y. Wang, P. Cai, and G. Lu, "Cooperative autonomous traffic organization method for connected automated vehicles in multi-intersection road networks," *Transportation Research Part C: Emerging Technologies*, vol. 111, pp. 458–476, Feb. 2020.

[48] H. Bang, B. Chalaki, and A. A. Malikopoulos, "Combined Optimal Routing and Coordination of Connected and Automated Vehicles," *IEEE Control Systems Letters*, vol. 6, pp. 2749–2754, 2022.

[49] F. S. Hillier and G. J. Lieberman, *Introduction to operations research*, 7th ed. Boston: McGraw-Hill, 2001.

[50] N. Gunantara, "A review of multi-objective optimization: Methods and its applications," *Cogent Engineering*, vol. 5, no. 1, p. 1502242, 2018.

[51] M. Emmerich and A. Deutz, "A tutorial on multiobjective optimization: fundamentals and evolutionary methods," *Natural Computing*, no. 17, pp. 585–609, 2018.

[52] W. Gutjahr and A. Pichler, "Stochastic multi-objective optimization: a survey on non-scalarizing methods," *Annals of Operations Research*, vol. 236, pp. 475–499, 2016.

[53] J. Kallrath, P. M. Pardalos, and D. W. Hearn, Eds., *Modeling Languages in Mathematical Optimization*. Springer US, 2004, vol. 88.

[54] R. Anand, D. Aggarwal, and V. Kumar, "A comparative analysis of optimization solvers," *Journal of Statistics and Management Systems*, vol. 20, no. 4, pp. 623–635, Jul. 2017.

[55] J. Jablonsky, "Benchmarks for Current Linear and Mixed Integer Optimization Solvers," *Acta Universitatis Agriculturae et Silviculturae Mendelianae Brunensis*, vol. 63, no. 6, pp. 1923–1928, 2015.

[56] A. Neumaier, O. Shcherbina, W. Huyer, and T. Vinko, "A comparison of complete global optimization solvers," *Mathematical Programming*, vol. 103, no. 2, pp. 335–356, Jun. 2005.

[57] E. D. Dolan and J. J. More, "Benchmarking optimization software with performance profiles," *Mathematical Programming*, vol. 91, no. 2, pp. 201–213, Jan. 2002.

[58] S. Mitchell, M. O Sullivan, and I. Dunning, "Pulp: A linear programming toolkit for python," 2011.

[59] W. E. Hart, C. Laird, J.-P. Watson, and D. L. Woodruff, Eds., *Pyomo - optimization modelling in Python*, ser. Springer optimization and its applications. New York: Springer, 2012, no. 67.

[60] K. F. Riley, M. P. Hobson, and S. J. Bence, *Mathematical methods for physics and engineering*, 3rd ed. Cambridge; New York: Cambridge University Press, 2006.

[61] R. L. Burden and J. D. Faires, *Numerical analysis*, 9th ed. Boston, MA: Brooks/Cole, Cengage Learning, 2011.

[62] F. N. Fritsch and R. E. Carlson, "Monotone piecewise cubic interpolation," *SIAM Journal on Numerical Analysis*, vol. 17, 04 1980.

[63] G. Improta and G. Cantarella, "Control system design for an individual signalized junction," *Transportation research.*, vol. 18, no. 2, pp. 147–167, 1984.