SHOCKWAVE DAMPING ON FREEWAYS USING MAGNETOMETERS AND PROBE VEHICLE DATA

MASTER THESIS

ALBERT GARRIGA PORQUERAS

Facultat d'Informàtica de Barcelona
Universitat Politècnica de Catalunya

Advisor	Mari Paz Linares Herreros - inLab FIB
Tutor	Josep Casanovas Garcia
Department	Departament of Statistics and Operations Research
Master	Master in Innovation and Research in Informatics
Speciality	Data Science

5th July 2019

Abstract

Currently, connected cars are uncommon on our streets, but their percentage is expected to grow uninterruptedly. These devices would provide a lot of data in real time that can be used for road operators to improve the traffic. We focus on one of these applications, which is shockwave damping on freeways. Shockwaves are congestions that appear spontaneously and reduce considerably the flow of freeways. The use of these data together with fixed sensors data in real time can be useful to mitigate them.

This work was developed with an education cooperation agreement at the inLab FIB (UPC) in the context of C-Roads, a European project. Our objective is to solve this problem. We implement two detection methods and one mitigation algorithm that uses variable speed limits to resolve shockwaves. All algorithms are proposed in the literature and exploit probe vehicle data in real time.

We analyze with microsimulation their performance and how their parameters affect them on several test scenarios. Also, the effect of the penetration rate of connected vehicles is analyzed.

Finally, we apply the best algorithm with the best parameters in a model of the AP7, a freeway in Girona. The results obtained show that the algorithms applied greatly improve the total travel time in this network.

Keywords: shockwave damping, shockwave detection, traffic simulation, variable speed limits, probe vehicle data.

Acknowledgements

En primer lloc, voldria agrair a la meva família el seu suport durant la realització d'aquest treball de final de màster i durant tot el màster en general.

He d'agrair a la MªPaz la seva ajuda i la seva dedicació durant tot aquest temps. Ha tingut molta paciència corregint les diverses versions dels capítols i discutint els diversos camins per on avançar en molts moments. També vull agrair al Josep liderar un projecte com l'inLab, que m'ha permès realitzar-hi aquest treball.

Al Ricard li dono les gràcies pel seu suport constant durant tot aquest temps i els bons moments passats durant l'últim any, però en especial els últims mesos. Sempre s'agraeix tenir algú que està en la mateixa situació que tu en qui puguis recolzar-te.

Al Juan li he d'agrair que sempre ha estat allà per resoldre tots els meus dubtes informàtics i que sempre el trobes disposat a donar un cop de mà.

A més a més, voldria donar les gràcies a tota la gent de l'inLab pel seu suport constant i pel bon ambient generat, cosa que sempre és d'agrair. No els anomenaré a tots perquè segur que em deixaria algú, però us estic molt agraït a tots.

Moltes gràcies a tots vosaltres!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we present what this master thesis is about. We explain the context in which it has been done and the problem that we have worked on. It also contains the objectives and motivations of the thesis, its temporal planning and its outline.

1.1 Master thesis context

This master thesis has been done while working in the inLab FIB. The inLab FIB[1] is a laboratory of research and innovation of the Barcelona School of Informatics (FIB) at the Universitat Politècnica de Catalunya (UPC).

The work developed here is part of the C-Roads[2] project. This project is a European collaboration between states, companies, and universities with the objective of deploying interoperable and harmonized cooperative Intelligent Transport System (ITS) in several test sites (pilots) across Europe. In this project, 17 countries participate as core members and 6 others as associated members. Some of its core members are Spain, France, Austria, Netherlands, and Italy.

In Spain, there are five different pilots: DGT 3.0, SISCOGA Extended, Madrid Calle 30, Cantabrian pilot and Mediterranean pilot. The UPC, through the inLab FIB, is a partner in the Mediterranean pilot and in the Madrid Calle 30 pilot. The deployment of the Mediterranean pilot spreads along 125 km of the freeway AP-7, split between Catalonia and Andalusia. The other partners in the Catalan sub pilot are Autopistas (the leader of the sub pilot), Aimsun, Sensefields, Kapsch, ITS-España, Servei Català del Trànsit, and RACC.

This master thesis contains the work done related to the service of shockwave damping. This is a service that the UPC has to implement in the Catalan sub pilot.

This sub pilot had to deploy fixed sensors in the Girona section of the AP-7 together with road-side units (RSU) that collect data from connected vehicles, known as probe vehicle data (PVD) or floating car data. The intention was to use the data from all these sources for the services deployed. However, due to several delays in the installation of these sensors and the fact that the number of connected vehicles was too small, this work only has used simulated data. In addition, the use of simulation has allowed us to test several scenarios and the effects of actions on the road that we could not have tested otherwise.

[1] `https://inlab.fib.upc.edu/es`, Last accessed: 14/03/2019

[2] `https://www.c-roads.eu/platform.html`, Last accessed: 12/03/2019.

1.2 The shockwave damping problem

As we mention before, the inLab FIB has to work on the shockwave damping (mitigation) service, which also involves detecting the shockwaves. In this section, we explain in what consists shockwave damping. To this purpose, first, we explain what is a shockwave and its relevance for traffic management. This problem contains two subproblems: the detection of the shockwaves and the mitigation itself. This is because shockwaves must be detected to be mitigated. We also describe the data sources that are available for tackling this problem. Finally, we discuss how usually the performance of the proposed methods are evaluated because there are not big datasets of shockwaves available.

1.2.1 Background

One of the problems of shockwaves is that they are difficult to define. There are several types of shockwaves. The focus of this report are the ones known as *phantom* or *moving jams*. First, we must know that shockwaves occur in long roads or freeways, thus in interurban traffic, it is not a phenomenon observed in urban traffic.

According to Hegyi et al. [2008] there are two types of jams in a road: *fixed jams* (which we will just call *jams*) and *moving jams* (shockwaves). A fixed jam is a retention of the vehicles produced by a bottleneck on the road. In a fixed jam, its head is fixed in the bottleneck and its tail moves upstream (in the direction opposed to the traffic) since more vehicles are incorporated in the jam. In contrast, a moving jam is a retention of the vehicles that has a moving head and a moving tail. These moving jams are also known as *traffic waves*, *stop-and-go waves* or *shockwaves*. According to Treiber and Kesting [2013], their origin is:

> "[...] *traffic waves*, also termed *stop-and-go waves*, are caused by the delays in adapting the speed to the actual conditions. [...] If traffic density is sufficiently high, this delay leads to a positive feedback on density and speed perturbations."

A shockwave is a traffic phenomenon that happens in high-density roads when a perturbation occurs. A perturbation can be an abrupt change of lanes or an accidental breaking of a vehicle. The driver behind the perturbation is forced to reduce its speed abruptly. Then, the driver behind does the same and so on, until eventually vehicles have a very low speed or they even stop. This propagates upstream of the traffic flow. Once it is generated, from the point of view of drivers entering in it, they see that they have congestion in front and they have to reduce its speeds, and after a while they reach the head of the shockwave and can recover their speed.

One key difference between shockwaves and jams is that although shockwaves can be produced by bottlenecks that reduce the capacity of a road (or as a consequence of a jam), they could remain in the road much after this bottleneck has been resolved. But it is remarkable the fact that shockwaves do not require a physical bottleneck to appear.

According to the annual report of the Observatorio del Transporte y la Logística en España [2017], during 2016, 94.0% of the freight interior traffic in Spain was transported using roads. More than 60% of it was transported through high capacity roads such as freeways or multi-lane roads. These numbers show the importance of interurban roads for freight transport, although it is also very relevant for the mobility of people. The presence of a shockwave can reduce the maximum flow of a road by 30% (Hegyi et al. [2008]). In the Netherlands, more than 20% of the congestion is considered a consequence of traffic jams (Wismans et al. [2015]).

Thus, improving the efficiency of freeways would provide great benefits from an economical and environmental point of view.

1.2.2 Shockwave detection

This problem consists on detecting that a shockwave appears and obtaining its main parameters. We consider that these main parameters are the following ones: the position of the upstream front, the position of the downstream front, the speed of the upstream front, the speed of the downstream front and the density of vehicles inside the shockwave.

The key point here is to distinguish a shockwave from a regular traffic jam. Their main difference is that a traffic jam is formed by a bottleneck on the road, so the head of a traffic jam does not move.

There are many ways to detect shockwaves, depending on the data available. The main available sources of data are video cameras, fixed detectors and probe vehicle data (data coming from connected vehicles). Although connected vehicles do not represent a relevant fraction of the total number of vehicles, its number is growing and it is expected to keep doing so.

If there are connected cars present, we also can distinguish between two types of detection. One is performed by a centralized system that collects all the data available and processes the information and the other is a decentralized system in which each vehicle receives communications from its surrounding vehicles and uses this information to detect that it is entering a shockwave by itself, without the need for a centralized system to perform the detection.

It is difficult to evaluate the detection of a shockwave. There is not an analytical procedure that given enough information gives as an output the characteristics of a shockwave.

1.2.3 Shockwave damping

This is the main purpose of this project, to dissipate shockwaves. In order to decide which algorithm to use, we must know which actions are available. The main method studied to damp shockwaves consists of applying vehicle speed limitations (VSL) upstream of the wave in order to reduce the inflow, which leads to the dissipation of the shockwave. This can be done in the following ways:

- To use variable message signals (VMS) in the previous sections upstream the shockwave to regulate the maximum allowed speed. This new maximum speed can be given as a recommendation or as mandatory. In this case, VSL are applied discretely, by sections. They can also be used to define different speeds for each lane.

- To send a new speed to OBUs (onboard units) of the vehicles. This can also be mandatory or recommended. This would allow providing more personalized speeds that are more continuous than the ones provided by VMS (which are constant in a section). If the location of vehicles is known with enough precision, the speeds can also be different for each lane.

- To fix the speed of every vehicle on the affected road by a centralized system. This is only feasible if cars are autonomous. Another similar version of this consists on, instead of fixing the speed of these vehicles, to define a new driving model in such situations.

In addition to the methods that set vehicle speed limitations, another possible actuation is ramp metering. Ramp metering consists on restricting the access to the main road with a

barrier or a light signal. There are also some proposals to change the behaviour of drivers by showing them a traffic jam ahead warning, for instance, so they can pay more attention and reduce their reaction time.

The evaluation of the mitigation of a shockwave is more direct. Key Performance Indicators (KPI) that measure the overall performance of the freeway can be used, for example, the total time spent on the road by all vehicles. Other possible KPIs are the average speed, the flow, the density, the emission and the damping time (time to mitigate the shockwave since it is detected). The methodology used usually is to compare with simulations what happens without doing any action and what would have happened with the actions proposed by a shockwave damping (SWD) algorithm. In literature it is usual to compare the effect of the shockwave damping algorithm with what whould have happened if the shockwave is not mitigated.

1.3 Objectives and motivations

The main objective of this master thesis is to perform shockwave damping on freeways using data from fixed sensors and probe vehicle data. To do this, first, we perform a detection of the shockwaves and then, their mitigation. We study the detection because it is necessary for the mitigation.

This master thesis uses simulated data because, as commented before, we do not have enough real data of the freeway where we want to test the developed algorithms. Also, it allows us to test several scenarios that are not possible yet, specifically scenarios with a high penetration of connected vehicles.

This project is of interest to the inLab FIB for the C-Roads project. Thus, we want to evaluate the methods implemented in the AP-7 in Girona. The inLab FIB research group is interested in this because they are very focused on traffic and they have a lot of experience working with traffic simulation, though they had not treated the shockwave damping problem before.

In addition, the research questions that we want to answer are the following ones:

- how well the ASM detection algorithm performs and how this performance is affected by its parameters?
- how Izadpanah's detection performs and also how it is affected by its parameters?
- how does the penetration rate affect the performance of the previous algorithms?
- does the mitigation algorithm actually mitigate shockwaves in different networks?
- does this algorithm improve the network with respect to doing nothing?
- does this algorithm work in a realistic scenario?

1.4 Temporal planning

We defined the following tasks to be performed:

- learn Aimsun: we first had to learn how the simulator that we decided to use (Aimsun Next) works.

- state of the art: the elaboration of the state of the art.
- create dummy networks: we created some dummy networks to test our algorithms.
- generate shockwaves: we had to perform several simulations to find how to generate shockwaves in Aimsun.
- implementation of classes: we had to implement all the classes needed to apply the algorithms to Aimsun.
- implementation of ASM detection
- implementation of Izadpanah's detection
- implementation of SPECIALIST
- design of experiments of ASM detection
- design of experiments of Izadpanah's detection
- design of experiments of SPECIALIST: this design of experiments requires all the experiments of the detection algorithms to be done because it only uses the best parameters obtained.
- execute ASM simulations
- execute Izadpanah's simulations
- execute SPECIALIST simulations
- analyze ASM results
- analyze Izadpanah's result
- analyze SPECIALIST results
- Apply SPECIALIST to AP7: we had to define the classes needed for this freeway and generate again new shockwaves in this scenario.
- write final conclusions: we wrote the conclusions obtained after analyzing the results of the experiments.
- write master thesis: this is a transversal task that was done during the whole elaboration of the master thesis.

The Gantt diagram of Figure 1.1 shows the duration assigned to each task and its dependencies.

1.5 Master thesis outline

This master thesis is structured as follows:

- **Chapter 2** contains a state of the art of shockwave damping and detection methods. We present some of the algorithms proposed in the literature and the different strategies available.
- **Chapter 3** introduces the algorithms selected. It explains the algorithms that we have implemented. These are variations of algorithms proposed on the literature. We also go further and explain some of the implementation details.
- **Chapter 4** presents the experiments performed. It explains the design of experiments, which factors we decided to analyze, and the obtained results.
- **Chapter 5** includes the conclusions and contributions of the performed work and proposes further research.

Figure 1.1: Gantt diagram of this master thesis.

Chapter 2

State of the Art

The purpose of this state of the art is to present the research that has been done in recent years about shockwaves damping on freeway. In particular, we want to analyze, not only the mitigation algorithms, but also the detection ones. This is necessary because in order to mitigate shockwaves they have to be previously detected.

This state of the art is organized as follows. In Section 2.1, we summarizes the most relevant detection algorithms found in the literature. Next, Section 2.2 describes the algorithms to mitigate shockwaves. Finally, Section 2.3 presents the main conclusions obtained.

2.1 Shockwave detection

The previous step for suppressing shockwaves is to detect them. The algorithm of detection can be independent from the method used to mitigate a shockwave. However, many papers present an algorithm that covers both. In this section we explain the algorithms available just for detecting shockwaves.

One of the key factors for choosing a method or another one is the data available. In most of the older papers authors consider algorithms based on fixed sensors (such as magnetometers or induction loops). Nowadays almost all authors also consider that Prove Vehicle Data (PVD) is available. PVD is the data that a connected vehicle sends, either to the infrastructure (V2I) or to other vehicles (V2V). These data usually consists on speed and position, but it may also contain additional information such as the distance to the vehicle in front.

Both types of data, fixed sensors and PVD are expensive as they require infrastructure. Fixed sensors have to be installed on the road, so they require works. PVD do not require to install anything on the road, but they need road-side units (RSU) that collect the data. The accuracy that can be achieved with algorithms that use PVD depends a lot on the penetration rate, which is the proportion of connected vehicles (the ones that send PVD) over the total number of vehicles. Most papers that use PVD present their results for diverse penetration ratios.

There are mainly two ways to detect shockwaves. The first one consists on using the data available to estimate the state of the flow (speed, density and flow) and then search shockwaves in the state obtained. The two measures more used for this purpose are the speed and the density. Shockwaves are characterized by having significantly lower speeds and higher densities than the rest of the road. This can be used to identify them, and this is

what these methods do. They search in the estimated state of the road for regions with these characteristics.

Other methods use directly the data available, particularly PVD to detect the head and the tail of shockwaves. The boundaries of a shockwave (head and tail) are a transition from a high density and a low speed (inside the shockwave) to a zone with a lower density and a higher speed (outside the shockwave). The methods that detect directly the shockwave without estimating the state of the road try to find regions in the flow with these characteristics.

2.1.1 Estimate the flow characteristics

One way of using the information available is to use it to obtain estimations of the characteristics of the traffic. This is what Adaptive Smoothing Method (ASM) does. This method was presented in Treiber and Helbing [2002]. With this algorithm we can estimate the speed (or any other measure) in any point of the road given measures from fixed sensors. Usually, it is estimated in some discretized points. What this algorithm does is to compute a weighted mean of the available measures for each point. The weight depends on the distance from the measure to the point (in space and time). This method was later extended (Van Lint and Hoogendoorn [2010]) to merge data from different sources (PVD and fixed sensors, for instance). They estimate the traffic state for every source independently and then they are combined with a weighted mean also. In this case, the weight depends on the accuracy of the source. This allows to have the speed at each discretization point and this can be provided as input to many other methods that intend to dissolve shockwaves. For example, the SPECIALIST algorithm that is presented later, uses this method to estimate the flow and detect the shockwaves.

ASM is not the only way to estimate the state of the traffic. Grumert and Tapani [2018] propose a method that also estimate the state of the traffic combining PVD and sensor data. They estimate the speed profile of the road using the PVD data, and the density is estimated with number of connected vehicles (those that provide PVD) divided by the penetration ratio that is estimated with the fixed sensors. Also Seo et al. [2015] propose an algorithm to estimate flow and density if connected cars can give information of the distances to the vehicle in front.

Herrera and Bayen [2010] propose two methods to use mobile GPS positioning (that is very similar to PVD of connected cars) to estimate the traffic. First, they use modified the Lighthill-Whitham-Richards traffic equation with a Newtonian relaxation. The second one is based in the Kalman filter.

2.1.2 Detect head and tail of shockwaves

The head and the tail of a shockwave are a boundary between two traffic states. There is a sudden change of speed in the tail, when vehicles have to slow down or stop, and then in the head when they can recover a high speed. Some methods aim at detecting these changes of state to identify shockwaves. One example is the method presented in Izadpanah et al. [2009]. In this paper the authors detect shockwaves using only PVD data. They find the inflection points in the trajectories of vehicles that indicate a sudden change of speed using two phase regression. Then, the inflection points found are filtered, because not all of them are associated with a shockwave. Finally, the remaining ones are clustered to group them into fronts that indicate a change of state in the flow. These fronts are heads and tails of

shockwaves. This method also allows to estimate the propagation speed of these fronts (the propagation speed of the head and the tail can be different).

Li et al. [2017] propose another algorithm for detecting shockwaves. This method is also based in PVD. They reconstruct the shockwave speed extending the PVD measures to all other vehicles. This gives an undeterminated system of equations, that they solve with the orthogonal matching pursuit (OMP) algorithm. They make a strong assumption of a triangular fundamental diagram. The authors of this paper do not provide many details, but, at least, they comment how their method could be extended to a freeway with several lanes.

One example that merges PVD and fixed sensors is Rempe et al. [2017]. This paper tries to forecast the position of the shockwaves fronts in a short time horizon (up to 10 min). It estimates the propagation speed with the well known formula for it from the Lighthill and Whitham theory. To apply it, it estimates the flows using a weighted mean of the measures available similar to the ASM. It also needs the densities, which they estimate using three different formulas proposed. They achieve a good performance for short prediction horizons, but then it decreases quickly.

2.2 Shockwave damping

Once shockwaves have been detected, there are several algorithms and strategies in the literature to damp them. The first thing that we must consider are what actions has a road manager to modify the flow. The most common actuations available are the following ones:

- set speed obligation/recommendation (they are called variable speed limits, VSL): it can be done using variable message signals (VMS) on the roadside. This allows to set a driving speed that is slower than the maximum speed of the road. Connected cars also allow to receive the new speed directly into the driver cockpit (or in their mobile phones). This is called in-car actuation. These two methods have two main differences between them. First, VMS allow to give the new speeds to all vehicles in the freeway, while with in-car actuation only connected cars receive them. Secondly, VMS set the same speed limit in regions that must be quite large while with in-car actuation this speed limit to be personalized for each receiver.

- ramp metering: it consists on restricting the access to the road with a barrier or a light signal in the entrances. This allows to regulate the influx of vehicles in the road. This actuation has the disadvantage that it can transfer the problems of the road to its entrances.

Here we explain SPECIALIST, an algorithm that has been developed and tested in a real scenario. It was developed with VSL, but it was extended in a posterior paper that allows it to use ramp metering as well. We also explain model predictive control (MPC), a class of methods that are being widely studied nowadays. They are used with VSL.

We present other algorithms that propose other types of actuations. Some papers explore a change in the driving behaviour. This can happen because a human driver receives a warning or an autonomous vehicle that receives a message indicating to change the driving model. Another approach consists on using pace cars, that are vehicles that cannot be overtaken with a fixed speed.

2.2.1 SPECIALIST

SPECIALIST (SPEed ControllIng ALgorIthm using Shockwave Theory) was proposed by Hegyi et al. [2008]. The state of the traffic is estimated with the ASM (explained before). If a region of the road has a speed lower than a fixed threshold it considers that there is a shockwave there. Then, using the density estimates inside and outside of the shockwave it performs a solvability assessment. If it considers it solvable, it triggers variable speed limits (VSL) upstream of the shockwave. The length of the region affected by them is computed such that it is enough to mitigate the shockwave but not too large because it would affect traffic more than necessary. The solution proposed and why it should solve the shockwave are very analytical based on the shockwave theory (as its name indicates) developed by Lighthill and Whitham in 1955.

This algorithm has been tested in a road by Hegyi and Hoogendoorn [2010]. SPECIALIST had a good performance in the field test, it solved 80% of the shockwaves that the algorithm considered as solvable. The problem that it had is that it was activated half of the times by shockwaves and the other half by other types of jams. In the latter cases, the effect of the algorithm was not as good. It even could reduce the efficiency of the network, depending on the parameter configuration.

SPECIALIST has been extended to consider ramp metering in Schelling et al. [2011]. Two years later is was further extended to take into account PVD and video-based monitoring in Hegyi et al. [2013]. This last extension also included in-car actuation. This consists on providing the recommended or mandatory speed to connected cars directly, that is displayed in the cockpit, instead that in variable message signals on the road.

2.2.2 Model Predictive Control

One of the most studied methods is model predictive control (MPC). MPC consists in finding the most appropriate configuration of variable speed limits (VSL) by solving an optimization problem. An objective function that is intended to reflect the performance of the network in a fixed horizon of time is optimized. To evaluate it, a simulation that begins at the current state of the road with a given set of speed limits (the variables that are optimized) is executed. This is the general framework of this kind of methods. The different papers basically propose different objective functions and simulation models to evaluate it. They also consider several types of restrictions on the VSL, such as avoiding high differences of speed between consecutive speed limited regions.

Since the simulation must be executed at each evaluation of the objective function and the problem must be solved in real-time when the shockwave is detected to apply the configuration of VSL found, this simulation must be very fast. Thus, macroscopic simulation models are used, that are much faster than microsimulation models. The problem is that macrosimulation models do not represent well shockwaves. Thus, authors have to add modification allow these models to model shockwaves better, but also to model the effect of speed limits. We can find one example of this in Hegyi et al. [2005], where the authors implement it with *METANET*, a well known macrosimulation model with some modifications. In this paper the model for simulating the network is the same used for the evaluation of the objective function. This can lead to overestimate the performance. Thus, in a later paper (Hegyi et al. [2007]) they implemented the same MPC but the to simulate the network (the simulation of the scenario, not the evaluation of the objective function) they use a microsimulation model called Paramics 5.1. A similar approach take Csikós et al. [2013], their MPC is also based on

METANET. Another macroscopic model, the *cell transmission model*, is used in Han et al. [2017].

2.2.3 Other methods

Some of the explored methods that do not use VSL or ramp metering are algorithms that propose a change in the driving behaviour. This is the approach that Motamedidehkordi et al. [2016] take. They consider a system in which a (human) driver receives a Traffic Jam Ahead Warning (TJAW) when they have a shockwave in front. Since they have received a warning, drivers change their driving behaviour, they pay more attention on the road. This is modelled by changing the driving model in microsimulations, from Wiedemann 74 to Wiedemann 99. This affects how shockwaves are propagated, because they are generated due to the delay in the response time of drivers. Another approach in a similar direction is proposed by Horn [2013]. He studies a driving model for autonomous vehicles that takes into account the distance to the vehicle in front and the distance to the vehicle behind (bilateral control). This is new, because nowadays most driving models just take into account the leading vehicle (so they are called car-following models). He suggests that this model could reduce the number of instabilities, such as stop-and-go waves.

A very different solution is explained in Behl and Mangharam [2010]. They explain the use of pace cars. A pace car is a vehicle that drives at a fixed speed and cannot be overtaken. They have the advantage that they do not require any new infrastructure. It is a simple solution and they estimate that it can improve the traffic flow a 8-12%. The paper only deals with the mitigation, it does not cover how shockwaves are detected.

2.3 Conclusions

We observe that almost all papers simulate their proposal but do not test it in a freeway. SPECIALIST on the contrary has a field test. There are other algorithms for shockwave damping that have been implemented, like the ZOOF App[1] in the A58 in Holland. This app provides speed advice so that users can drive cooperatively. The problem is that there are not papers that explain the algorithm used, there are just reports that describes the project generally (Spookfiles [2016]).

We must consider that SPECIALIST is quite old and that many authors, including the author of the SPECIALIST algorithm, A. Hegyi, are moving towards MPC since the increase on computational power facilitates the implementation of these methods. However, these models still do not have any field test that reflects its simulated performance on a real scenario.

With respect to the detection algorithms, although the ones based in PVD show good results, we are far from having high penetration rates in our roads. So, nowadays algorithms that combine both, measures from PVD and fixed sensors are more feasible to be deployed in a real freeway. In particular the ASM algorithm that allows to merge any number of sources seems particularly useful.

We also observed that most algorithms were just implemented for one lane and do not consider how to deal with multiple lanes and lane changing behaviour. This is something that should be considered before deploying any method to a real freeway (SPECIALIST has been deployed but they do no mention how they deal with multiple lanes).

[1] `http://www.zoof.nu/home.html`, last accessed: 26/06/2019

It is also remarkable that many of the algorithms proposed do not make use of machine learning. Currently, many papers about traffic use machine learning algorithms, such as those that do traffic forecasting, where applying machine learning is quite common. In contrast, probably because the study of shockwaves is not a topic so popular, the use of these methods is very uncommon in this field.

Chapter 3

Selected Algorithms

In this chapter, we examine the algorithms that we have implemented. First, we explain their original version, as their authors presented them. Then, we expose our implementation, which may include some modifications. We also present the implementation details of these algorithms. Finally, we explain the general structure used to implement the algorithms.

3.1 ASM detection algorithm

This detection method uses an algorithm called Adaptive Smoothing Method (ASM). We use it to obtain measures of the speed in some fixed discretized positions on the freeway. Then, we search for congested regions (regions with a low speed) that move over time. These regions are the shockwaves. In the following subsections, we explain in details this adaptive smoothing method: how we merge the measures of fixed sensors with PVD and how these congested regions are detected from the discretized measures.

3.1.1 Adaptive Smoothing Method

The ASM is described in Treiber and Helbing [2002]. It is an algorithm that allows to estimate the value of a traffic measure (speed, density or flow) in any location of a one lane section using one source of data. We only use it to estimate the speed, so from now on we focus on speed, but any of the other two measures could be estimated instead. In Van Lint and Hoogendoorn [2010], there is an extension of the algorithm that allows to merge several sources of data to obtain a more accurate estimation.

We start explaining the case of PVD as the only source of data. The same procedure is used to estimate the speed with fixed sensors measures. These are the two sources that we will use. Then, we explain the general algorithm for combining several sources, though we only use it with these two sources.

Speed computation using PVD

The input is a set of n measures $X = \{(x_1, t_1, v_1), ..., (x_n, t_n, v_n)\}$, where a triplet (x_i, t_i, v_i) indicates that vehicle i is at position x_i with speed v_i at time t_i. We assume that these measures are PVD, but their origin does not matter.

The main idea of the ASM is to obtain the estimation of the speed in a point (x, t) as a weighted mean of the measures in a neighbourhood of this point. The weight function used, based on the exponential is shown in equation 3.1.

$$\phi_0(x,t) = exp\left(-\frac{|x|}{\sigma} - \frac{|t - x/c|}{\tau}\right) \tag{3.1}$$

Where c, σ and τ are the parameters that must be calibrated. This is the base function used, but the weight of data (x_i, t_i, v_i) is actually given by its distance (in space and time) to the point where we are computing the speed (x, t), as shown in equation 3.2.

$$\phi(x_i, t_i) = \phi_0(x_i - x, t_i - t) \tag{3.2}$$

In order to estimate the speed, two speeds are computed using the exact same procedure: $z^{free}(x,t)$ and $z^{cong}(x,t)$, one considering that the section is in free flow and the other considering the section congested. The difference between them are the parameters τ, σ and c used to compute them. Thus, two sets of parameters are needed, one for each of these estimations.

Then, these two speeds are combined into the actual estimation speed for that point $z(x,t)$:

$$z(x,t) = w(x,t) \cdot z^{cong}(x,t) + (1 - w(x,t)) \cdot z^{free}(x,t) \tag{3.3}$$

Where function $w(x,t)$ is:

$$w(x,t) = \frac{1}{2}\left(1 + \tanh \frac{V_c + \min\left(z^{cong}(x,t), z^{free}(x,t)\right)}{\Delta V}\right) \tag{3.4}$$

V_c and ΔV are also parameters that have to be configured.

Equation 3.5 is used to compute $z^{cong}(x,t)$. The procedure for $z^{free}(x,t)$ is analogous but with its own parameters. We call the neighbourhood of a point $N(x,t)$, which are the values taken into account for that point. We use the notation ϕ^{cong} to indicate the weight function that uses the parameters for the congested case.

$$z^{cong}(x,t) = \frac{\sum_{i \in N(x,t)} \phi^{cong}(x_i, t_i) \cdot v_i}{\sum_{i \in N(x,t)} \phi^{cong}(x_i, t_i)} \tag{3.5}$$

There are several ways to define the used neighbourhood. We will consider neighbourhoods of the type:

$$N(x,t) = \left\{(x_i, t_i, v_i) \in X \mid (x_i - x)^2 + (t_i - t)^2 \leq r^2\right\}^{1} \tag{3.6}$$

Where $r^2 = dl_{max}^2 + dt_{max}^2$ is the maximum radius considered. These are user-defined parameters.

Implementation details

We implement this algorithm by discretizing the road with points equally separated by distance dx. The speed is estimated at each of these points. Thus, to compute equation 3.5 we have a double loop, for each point we compute z and to compute each one we need the add over all other points inside the neighbourhood. If for each point we have to check if all measures are in his neighbourhood or not, it would be very slow, it would have a cost of $\mathcal{O}(m \cdot n)$, where n is the number of dots and m the number of measures. This is too much, it

[1] We define N(x,t) as a set of points. For simplicity, we have written equation 3.5 considering that N(x,t) is the set of the indexes i of these points.

is unfeasible if the number of vehicles grows, it would take too long to compute ASM (that must be done often to update the state of the freeway).

To avoid this, we sort the measures and create a list containing an index that indicates the positions in the vector of speeds that are separated by at most dl_{max}. This way, for each point that we want to compute its speed, we look for the interval in which it falls (which is done with a modified dichotomic search). Then, the measures in this interval and in the adjacent ones are used in equation 3.5. Thus, we considerable reduce the computation time, specially for large m, since we do not have to check for all measures if they are in the neighbourhood of each point. For each point we directly obtain a small set of measures that can be in its neighbourhood.

If some discretized point does not have any data in its neighbourhood, we assign to the estimation at that point a *None* instead of a 0. This allows to distinguish the case in which vehicles are stopped (we assign the value 0) from the case in which there are no vehicles in the neighbourhood (we assign the value *None*). This distinction is relevant for merging PVD data and sensor data, as will be explained in section 3.1.2.

Speeds computation with fixed sensors

Sensor data is treated in such a way that it has the same format than PVD. When a vehicle gets through a sensor, we obtain a triplet of data $(position, time, speed)$ that allows us to treat this data as if it was PVD.

Implementation details

For each time step of the simulation, we obtain for each sensor the vehicles that have cross it and for each of them, we obtain the triplet $(position, time, speed)$. This data is stored during a fixed interval of time. This is because storing all the data collected would slow down the computation considerably until eventually it would become unfeasible. Another interval time is defined, which determines when the speed estimation obtained with the fixed sensors data is recomputed using the sensor data available at that time.

3.1.2 Estimations merge

The algorithm of merge (also called fusion) allows to obtain an estimation of the speed from many different sources. The two sources of data that we use are just PVD and sensors. However, the algorithm is more general and can be applied to any number of sources.

Equation 3.7 shows the expression used for fusing the different sources (S):

$$z_{fus}(x,t) = \frac{\sum_{s\in S} \alpha^s(x,t) \sum_{i\in N_A^s(t,x)} \phi^s(x_i,t_i) z^s(x_i,t_i)}{\sum_{s\in S} \alpha^s(x,t) \sum_{i\in N_A^s(t,x)} \phi^s(x_i,t_i)} \tag{3.7}$$

Here $z^s(t_i,x_i)$ is the i-th measure of the neighbourhood of (x,t) of the source s. The superindex s in the base function and the neighbourhood indicates the source. We could use different parameters for the base function of different sources, for instance. The neighbourhoods can also be defined differently for each source. We must also note that the positions and times x_i and t_i are the ones where we estimated $z^s(x,t)$, not the ones from the original data. Finally, the weights α^s are associated to each source. They take into account the accuracy/reliability of them. They are defined by equation 3.8.

$$\alpha^s(x,t) = \frac{1}{\theta_0^s[1+\mu^s(1-w^s(x,t))]} \tag{3.8}$$

The parameters θ_0^s and μ^s must be defined by the user. θ_0^s represents the error variance of source s and μ^s how this error changes with the congestion of the freeway. It is important to note also that to compute z_{fus} the variable w has to be stored for each source.

Implementation details

For the fusioned speed the use of *None* is the same that in the standard ASM. If all the values $z^s(x_i,t_i)$ are *None* for all sources for a given point, we assign to it the value *None*. We must also take into account that if an estimation has value *None*, the associated weight is not taken into account in the denominator as well, otherwise the obtained speed would not be correctly weighted. For instance, if from the sensor measures we estimate a speed $50km/h$ and we have no PVD data nearby, the fused speed is $50km/h$ instead of a weighted mean between 50 and 0 that would give a much smaller value.

3.1.3 Shockwave detection

The definition that we are using of shockwave is a region of more than d_{min} length where vehicles have a speed smaller than v_{min} that moves on the road with a certain speed. This movement is measured in its boundaries, the head and the tail. The parameters d_{min} and v_{min} must be calibrated to obtain a quick detection but avoid having false detections.

We directly apply the definition to detect shockwaves by getting through the vector of speed estimations and counting the length of regions where the speed is smaller than the speed threshold. This vector of estimated speeds is computed as follows. At each step, the method checks if the PVD is updated or not according to some fixed message frequency of the PVD, that we call sending time and we calibrate. It does the same with sensor data, that are updated every certain time. If at least one of the two sources has new speed estimations, the detector performs the ASM fusion algorithm to merge both estimations into a single speed estimation at each discrete point. The method implemented should allow to detect several shockwaves happening in the same freeway at the same time, but we did not test this.

We also extend this method to ensure that shockwaves must move. We maintain a list of the detected shockwaves that is updated when new measurements arrive. We apply the previous algorithm to the new data to retrieve possible new shockwaves (we do not know if they are shockwaves or just jams). We update the position of the jams previously detected to an estimation of where they should be now, knowing the interval of time between measurements. Then, we find the new detected shockwaves that are close to these estimated positions. If a shockwave has a new measurement close, we consider that it has not vanished and we update its position. If it does not, it is vanished and removed from the list of shockwaves. The measurements that do not match any previously detected shockwave are added to the list of detected shockwaves. If in the next step another measurement confirms them as shockwaves, they are maintained. Otherwise, they are dropped. This way, a region with a low density must be detected by two consecutive measures (in a displaced position the second time) in order to be considered a shockwave. This allows to distinguish shockwaves from jams.

The distance between the measured and the estimated position of the shockwave is taken as the maximum distance between distance between tails (expected and measured) and the distance between heads (also expected and measured).

Dealing with several lanes

In order to deal with roads that have several lanes, the data of PVD and sensors available is splitted into the data that belong to every lane. Thus, we have PVD and fixed sensors data individually for each lane. Then, we apply to every lane the algorithm of fusion (ASM), that we have explained in Section 3.1.1, to obtain the speed in each one and we detect shockwaves separately for each lane.

3.2 Izadpanah's detection algorithm

This section explains the detection algorithm proposed by Izadpanah et al. [2009]. This algorithm only requires PVD measures, it does not need fixed sensors. It consists of analysing the trajectory data of each individual connected vehicle. The idea is to detect what the authors call inflection points, that are great changes in the speed of vehicles that appear in their trajectories. These points can be visually seen in the position-time plot as the changes of slope of the line. The idea behind the paper is that many of these points that represent a significant change of speed can indicate that there is a shockwave front there. Thus, after detecting inflection points for all vehicles, these are clustered to identify shockwave fronts.

3.2.1 Detection of inflection points

The algorithm used for detecting the inflection points is based on two-piecewise regression. To compute a two-piecewise regression, we test all middle points and compute two regressions, one on the left and other on the right. Then, we check if the difference in their errors, compared with a linear regression, is significant or not. If it is not, then that point is not an inflection points. If it is, then that point is considered an inflection point.

To detect all inflection points, we use this two-piecewise algorithm with different chunks of the data. These chunks are updated with new data and start on the last inflection point detected. The pseudocode of this is shown in algorithm 1.

3.2.2 Clustering algorithm

The clustering used is not one of the usual clusterings, where points are grouped by proximity. In this case, we expect the inflection points to follow linear patterns (because shockwaves propagate with constant speed). Thus, we perform a clustering in which we aim at putting in the same cluster points that follow a linear pattern. This is called clusterwise linear regression. In order to obtain this, the algorithm used aims at minimizing the sum of the squared error of linear regression in each cluster.

To do this clusterwise linear regression, we implement the method explained in Gitman et al. [2018]. It is not the same one proposed in the original paper that proposed this detection, but the main idea is the same. The algorithm is the following. First, each point is assigned to a cluster randomly (the number of clusters is fixed, several values are tried). Then, we compute the linear regression for all clusters. Once we have all the models, we compute for each point the error with each model and assign it to the cluster that has the lowest error (with a normalization factor). We iterate this step until we achieve converge in the total error. This method is quite slow and we have to take into account that, since the starting solution is random, resampling may be used.

Algorithm 1 Schema for finding inflection points.

function GETINFLECTIONPOINTS(x, y, K) ▷ x and y are the coordinates, K the number of chunks
  $inflectionPoints \leftarrow []$ ▷ Empty list
  $coefL \leftarrow []$
  $coefR \leftarrow []$
  $size \leftarrow \lceil len(x)/K) \rceil$
  $indexL \leftarrow 0$
  for k in 1..K **do**
    $indexR \leftarrow \min(len(x) - 1, (k + 1) \cdot size)$
    $range \leftarrow indexL : indexR$
    $fitL, fitR, p, rss \leftarrow twoPiecewiseRegression(x[range], y[range])$
    $rssl \leftarrow linearRegressionError(x[range], y[range])$
    if $significant(rssl, rss)$ **then**
      $inflectionPoints.append(p, fitL.predict(p))$
      $coefL.append(fitL.slope)$
      $coefR.append(fitR.slope)$
      $indexL \leftarrow getIndex(p) + 1$
    end if
  end for
  return $inflectionPoints, coefL, coefR$
end function

The expression that we compute to assign each point to a cluster is the one shown in equation 3.9. It computes the error to assign point (x_i, y_i) to cluster j. This includes the model error and also a normalization error, which is the same term used in k-means clustering. This is the second term of the expression and its weight is controlled by the parameter γ, that in general is lower than 1. The values w_j and b_j are the slope and intercept respectively of the linear regression of cluster j.

$$E_j(x_i) = (y_i - x_i w_j - b_j)^2 + \gamma \left(x_i - \frac{1}{|C_j|} \sum_{t \in C_j} x_t \right)^2 \tag{3.9}$$

As we said before, this algorithm requires to know the number of clusters. Thus, we apply it with several clusters. We consider that a cluster should have at least three points, thus, we set the maximum number of clusters to the number of points divided by 3. However, since each cluster corresponds to a shockwave and we do not expect many shockwaves, we limit this method to use at most 20 clusters to speed it up. The best cluster is chosen as the one with the biggest marginal gain. This is, the number of clusters n that when changing from $n - 1$ to n gives the maximum reduction in the total error. We apply this method because otherwise increasing the number of clusters should always reduce the error and if we only reduce the error we would overfit the clusters. We also add a tolerance such that if the difference between the error in the chosen number of clusters and the error obtained with one cluster is not significant we consider that we only have one cluster. We have to add this because the maximum marginal gain criteria cannot tell that we have just one cluster.

3.3 SPECIALIST

This is the algorithm proposed in Hegyi et al. [2008] for mitigating shockwaves. It has been extended in later papers (see Section 2.2.1, in the state of the art). This algorithm is based on shockwave theory developed by Lighthill-Wihthman-Richards. We use the notation ρ for the density (vehicles per unit of length), q for the flow (vehicles per unit of time) and v is the average speed of the section. The main result of shockwave the theory used in this paper is the speed of a front between two traffic states, 1 and 2 (c_{12}):

$$c_{12} = \frac{q_1 - q_2}{\rho_1 - \rho_2} \tag{3.10}$$

It also uses the basic relation of flow theory:

$$q = \rho \cdot v \tag{3.11}$$

This algorithm makes use of fixed sensors and probe vehicle data, though its first version only used fixed sensors. The objective of the algorithm is to create a region upstream of the shockwave with a lower speed and flow that makes the tail of the shockwave propagate at a shorter speed than the head. Thus, eventually the head will meet the tail and the shockwave will be resolved. After this, the shockwave is resolved, and although a region with a higher density and lower speed remains, this region is assumed to have a higher flow than the shockwave, so the traffic is improved.

To achieve this, the algorithm detects the shockwave using the first detection algorithm explained (ASM) and then, mitigates it using variable speed limits (VSL) on an upstream region of the shockwave. Specifically, we can draw the diagram shown in Figure 3.1 of the situation of a shockwaves and the application of speed limits.

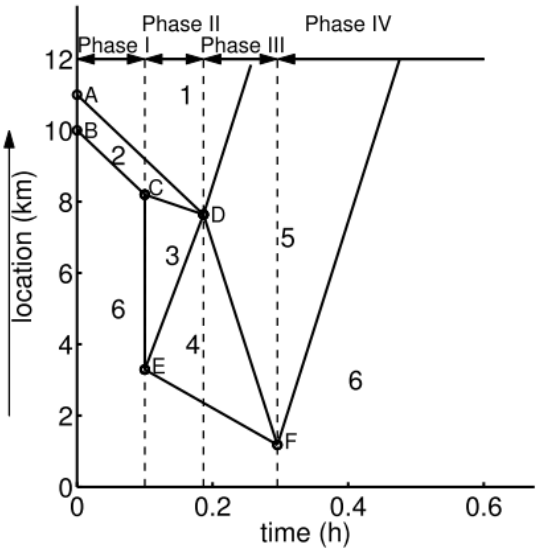


Figure 3.1: Regions defined in the application of SPECIALIST. Extracted from: Hegyi et al. [2013].

This diagram contains six different states (we also call them regions):

- **state 1**: this is the state downstream of the shockwave.
- **state 2**: this is the state of the shockwave.
- **state 3**: this is the state that is created after applying the VSL.
- **state 4**: this is the state that is created in the region upstream of the region where the speed limits are applied.
- **state 5**: this is the state that remains after the shockwave is resolved.
- **state 6**: this is the state upstream the shockwave, before applying any variable speed.

Some of the states of these regions are known (or can be known) and others parameters of the algorithm. Specifically, the states 1, 2, 3 and 6 can be estimated with the available measures and knowing that $\rho_3 = \rho_6$ and the speed of state 3 is the fixed controlled speed. The speed of state 4 is also the controlled speed, but its density (ρ_4) cannot be known. This density, together with the density and flow of state 5 (ρ_5 and q_5) and the applied variable speed limit are the tuning parameters of the algorithm.

In Figure 3.2 we can see how we compute the length of the region where the VSL are applied. This triangle is formed by the points C, D and the interception of line BC with the line of time 0.2. of Figure 3.1. In the diagram we see that the angles that we have are:

$$\alpha = \arctan |c_{23}| \tag{3.12}$$

$$\beta = \arctan |c_{12}| \tag{3.13}$$

$$\gamma = \frac{\pi}{2} - \arctan |c_{34}| \tag{3.14}$$

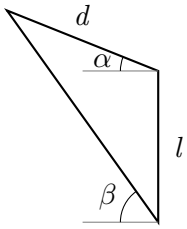


Figure 3.2: Zoom in the triangle where the head and tail of the shockwave meet.

We observe that we know all the angles except the one that is opposite to the side l. We can obtain it taking into account that the sum of all angles of the triangle is π. With this, we obtain that this angle is $\alpha - \beta$. With this angle, we can apply the sinus theorem and we obtain:

$$\frac{l}{\sin(\alpha - \beta)} = \frac{d}{\sin(\pi/2 - \beta)} \tag{3.15}$$

And we can isolate the distance d:

$$d = l \cdot \frac{\sin(\pi/2 - \beta)}{\sin(\alpha - \beta)} \tag{3.16}$$

Once we know this value, the length of the VSL region can be computed as the sum of the two distances L_1 and L_2 defined in the diagram.

$$L_1 = d \cdot \sin \alpha \tag{3.17}$$

$$L_2 = d \cdot \frac{\cos \alpha}{\tan \gamma} \tag{3.18}$$

Combining these expressions with equation 3.16 we can obtain the length of the controlled region:

$$L = l \cdot \frac{\sin(\pi/2 - \beta)}{\sin(\alpha - \beta)} \cdot \left[\sin \alpha + \frac{\cos \alpha}{\tan \gamma}\right] \tag{3.19}$$

We can see that the region where we have to apply the speed limits depends linearly on the length of the shockwave. We can also vary this length depending on the speeds of borders between regions, which we can modify depending on the applied speed limit and the tunning parameters of the algorithm that we explained before.

The procedure to apply this algorithm is the following:

1. The shockwave is detected and located with some method (it can be any algorithm).
2. All the traffic states are determined, by direct measure and also using the parameters of the algorithm.
3. Check if the shockwave is solvable. Some times, the traffic states are in a configuration where the shockwave is not solvable. This is explained in more detail in Section 4.6.1. If the shockwave is not solvable, the algorithm ends here.
4. If it is solvable, the VSL region of application is computed with the previous formula and the VMS are activated.
5. Once the shockwave is no longer detected (it has been mitigated) the VSL are released progressively.

3.4 Implementation

All the algorithms and the posterior analysis of the results obtained is done with Python. Specifically, the simulator API functions must use Python 2.7.13. To implement these algorithms we have developed the following main classes:

- **section**: this class has the parameters that define a section such as length, number of lanes and id. It also stores the data obtained in that section and contains the methods to apply the ASM algorithm on the data.
- **freeway**: it is responsible of storing the ordered sequence of the sections studied. It also allows to retrieve one section given its id or its position and obtains global positions (with respect to the whole freeway) from local measures in a section.
- **detector**: it contains the methods used to detect the shockwave. This is the only one that changes when the detection method is modified.

- **simulation manager**: it is the main class that is executed at every simulation step. It calls the updates of the parameters and then calls the detection method to detect shockwaves. It also handles that if a shockwave is detected, the SPECIALIST algorithm is called (if we are mitigating).
- **data manager**: this class updates all the data available, fixed sensors and PVD. It does it with the PVD manager and several sensor managers that are associated with each sensor.
- **PVD manager**: it updates all the PVD data. It controls that it is updated with the fixed frequency, determined by the sending time.
- **sensor manager**: similarly that the PVD manager, for its corresponding sensor it updates its measures at the appropriates time steps.
- **SPECIALIST**: this class computes the SPECIALIST algorithm. It estimates all the traffic states that must be estimated and performs a solvability assessment. If the shockwave is solvable, it computes the length of application of the VSL and activates them.
- **VMS manager**: this is the class that stores the VMS configuration available that SPECIALIST activates. It stores which VMS are activated and the speed that they are setting. It must also be called at each time step to apply these speed limitations.

The main libraries used are *numpy* to handling the arrays of measures, *sklearn* for the linear models of the clusterwise linear regression and *matplotlib* for generating the figures.

We have used Git[2] for version control. Moreover, we have used Jenkins [3] to automatize the execution of the implemented tests on the programmed classes.

[2] `https://git-scm.com`, last accessed: 26/06/2019
[3] `https://jenkins.io/`, last accessed: 26/06/2019

Chapter 4

Computational experiments

In this chapter, we explain the experiments that we performed and their results in order to determine the performance of the algorithms implemented and the effect of all their parameters. We tested the ASM detection algorithm, the Izadpanah's detection algorithm and the mitigation algorithm SPECIALIST.

We examined these algorithms in four different networks, that are also described in this section. First, we explain the simulation framework in which all simulations were performed. Then, we comment how to measure the performance of each algorithm. After, we explain the design of experiments and the results for each algorithm, and we make a final comment on each of them. SPECIALIST was executed with the best detector method and its best parameters obtained for each scenario. To conclude, we used the best configuration of SPECIALIST to run it on the test site, the AP7.

We consider that our methods are designed to be calibrated and then deployed on a network. Hence, in this work the calibration of the parameters only depends on the topology of the network, not in the specifics of the traffic situation, such as the flow. Thus, our design of experiments focused on obtaining the best parameters for each algorithm that show a better overall performance. Our analysis focused also on how the different algorithms behave depending on the penetration rate of connected vehicles.

4.1 Simulation framework

To evaluate the algorithms implemented we simulate networks and apply them. However, there are mainly three types of traffic simulation approaches (Barceló [2014]):

- macroscopic: only the evolution of the macroscopic measures of the flow (density, speed and flow) is simulated. Thus, these simulations are faster but they provide less details than the other types, since only the macroscopic measures are computed.

- microscopic: in this type of simulation the behaviour of every vehicle is emulated. So, the positions and speeds of all of them are known at each simulation step. The fact of having to simulate all vehicles makes this type of simulations slower than macroscopic simulations, but they provide much more detailed information, since we know the trajectory for each individual vehicle.

- mesoscopic: this type of simulations are an intermediate point between the two previous types. There are many different types of simulations that are included in this category.

Usually they are microscopic simulations with simplifications, such as not knowing the exact position of each vehicle in a link, just knowing in which link each vehicle is. They provide less detailed information than microscopic simulations but they are executed faster.

Since we wanted to simulate connected vehicles, we needed to simulate each vehicle individually. Thus, we performed a traffic microsimulation.

To perform the experiments we used the traffic simulation software Aimsun[1], specifically the version Aimsun Next 8.3. Aimsun Next allows to perform the microsimulation we needed (it can also perform meso-simulations).

We have chosen this simulator because Aimsun is also a partner of the C-ROADS project and they have created the model of the AP-7 as part of their contribution. Also, the group at the inLab FIB where this master thesis was developed already had experience working with Aimsun. An example of a screenshot of an Aimsun Next simulation is shown in Figure 4.1.

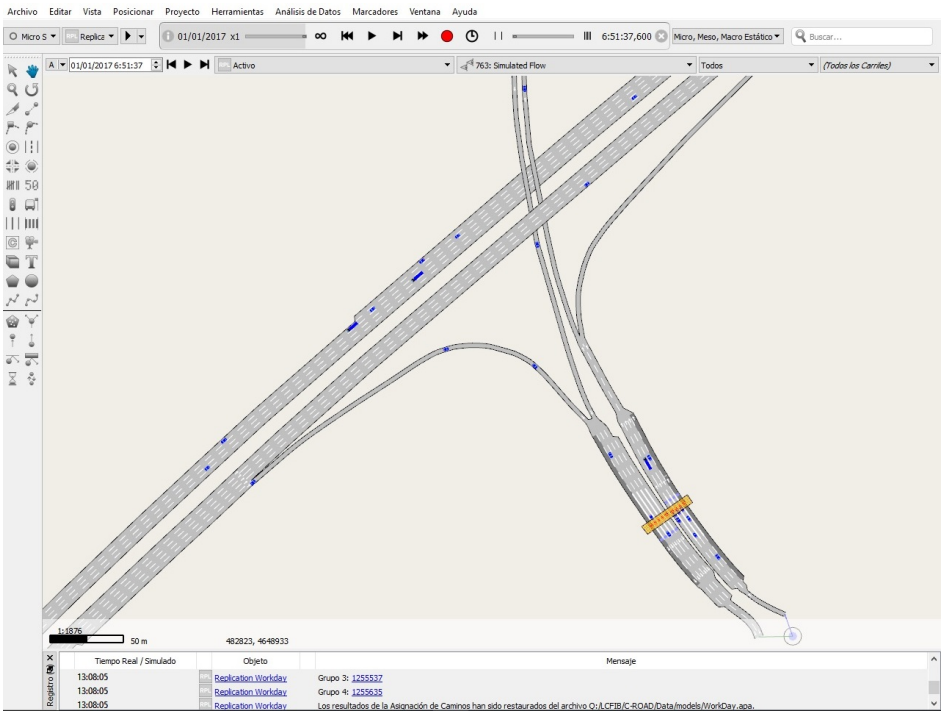


Figure 4.1: Screenshot of an Aimsun simulation.

In addition to the graphical program, Aimsun also provides a console program for executing simulations without the GUI (the graphical interface and representation of the simulation). Simulations executed with the console are faster, so to perform several simulations automatically we used it. These simulations are controlled using Aimsun scripting, that allows to control the launch of simulations and configure them in the Aimsun console program.

The implementation of the algorithms, which interact with the simulation (by extracting data or directly modifying the behaviour of the simulation), is done with the Aimsun API, which is different from Aimsun scripting. This API allows to interact with the simulation in real time while it is being executed. For instance, at each timestep, it provides access to the

[1] `https://www.aimsun.com`, last accessed: 21/06/2019.

information of every vehicle and allows to modify it, or it provides access to the fixed sensors data.

All the experiments have been executed in a computer running Windows 10 (64 bits) with the following specifications: a CPU Intel Core i7-600K 4.0 GHz, a hard disk Samsung SSD 850 EVO M.2 500 GB and 4 RAMs of 16GB (total of 64 GB).

4.2 Scenarios

The algorithms proposed are designed to work with one lane. However, we also want to test how they work in a freeway with several lanes, where each lane is treated individually. To this purpose, we used four networks to test our methods. Three dummy networks which consists only of a straight freeway of 10 km with 1, 2 or 3 lanes. Then we have the AP7 model, which is described later in Section 4.7.

Each network was simulated with different penetration rates, giving several scenarios. Which penetration rates are executed for each algorithm is explained in the design of experiments of each one. We are particularly interested in PVD, so we did not consider scenarios without PVD.

In order to generate a shockwave, we created a high input flow in the networks. Specifically, we put enough traffic to saturate the freeway with as much traffic as possible. Then, we stop a car enough time to create the desired shockwave. In the dummy networks, there is enough to stop a car for 5 seconds, but in the AP7 to generate a shockwave we need to stop it for 20 seconds. In these scenarios of the dummy networks, the simulations executed last 30 min and the shockwave is simulated at the 15 min at the last 100 m of the freeway.

To mitigate the shockwave, we applied VSL, as explained in Section 3.3. In all scenarios, we used a configuration of variable message signals spaced 1 km through all the freeway.

In addition, we set fixed sensors, which are magnetometers, each 500 m and they update their data each 15 seconds. Both parameters are fixed for all scenarios.

4.3 Performance metrics

Regarding the shockwave detection, we did not have a database of shockwave measures to calibrate our algorithms. Thus, we had to generate our shockwaves, as explained before. The only things that we can control about a shockwave are its generation time and its starting position. Thus, what we can measure is if the shockwave is well detected at its beginning. Once it is detected, since we do not know how it has propagated[2], we cannot measure if the detection algorithm is determining with a good precision the head and the tail of the shockwave or not. Thus, the only thing that we measure of our detection algorithms is the delay in the detection, which is how many seconds are between the generation time of the shockwave and the detection time of the algorithm.

With respect to the shockwave mitigation, we can measure if the SPECIALIST algorithm has actually mitigated the shockwave or not (if it is still being detected) and also we can measure how much improvement it produces on the network. The metric for this is the total time spent (TTS), which measures the total time that cars have spent inside the network

[2]There is a traffic macroscopic theory, Lighthill-Whitham-Richards (LWR) that can compute the speed of a shockwave, but we would need to know the density inside the shockwave. To compute this, we would have to know where is the shockwave, which we do not know. In addition, this would give the ideal behaviour of a shockwave, which may differ from the real one, specially in networks with more than 1 lane.

(the sum of all travel times). In addition, we analyze the last instantaneous TTS. This is the average total time spent of the vehicles that arrive at the end of the network in the last minute of the simulation. The TTS allows us to see how the networks is affected in the short term by the mitigation algorithm, because we run short simulation of 30 min. The last instantaneous TTS allows us to see how the state of the network is affected more in the long term, because we see which TTS the vehicles tend to have after applying the algorithm. It is a measure that helps to determine in which state the network is left, while the TTS represents in which state the network has been. Both metrics are represented divided by a reference value which is the metric obtained with the scenario where the shockwave is not mitigated (for each network).

4.4 Experiments of ASM detection

In this section we explain the experiments performed to analyze the effect of the parameters of the ASM detection algorithm and its performance. First we present the design of experiments, then the results obtained and finally we make a conclusion and show the best parameters obtained for each scenario.

4.4.1 Design of experiments

This method has parameters related to the ASM (the algorithm to merge PVD and fixed sensors data) and also to the threshold that we use to detect congestion regions. These parameters are shown in Table 4.1. There are many parameters, so it is not feasible to perform a full factorial design or to analyze them all. In addition, some of them are more relevant and interpretable and others are more difficult to understand and do not seem so relevant.

parameter	**description**	**fixed**
V_c	First parameter of $w(x,t)$ (see Section 3.3).	80 km/h
ΔV	Second parameter of $w(x,t)$.	10 km/h
σ^c	Congested parameter (associated with position) of the ASM weight function.	-
σ^f	Free flow parameter (associated with position) of the ASM weight function.	-
τ^c	Congested parameter (associated with time) of the ASM weight function.	-
τ^f	Free flow parameter (associated with time) of the ASM weight function.	-
ΔL_max	Maximum distance considered in a neighbourhood in ASM.	100 m
ΔT_max	Maximum time difference considered in a neighbourhood in ASM.	30 s
c^c	Congested parameter that shifts the time of the ASM weight function.	-25 km/h
c^f	Free flow parameter that shifts the time of the ASM weight function.	80 km/h
θ^{pvd}	First parameter of the weight of the PVD source.	1
θ^{fixed}	First parameter of the weight of the fixed detector source.	4
μ^{pvd}	Second parameter of the weight of the PVD source.	3
μ^{fixed}	Second parameter of the weight of the fixed detectors source.	2
speed threshold	Threshold for defining a region as congested.	-
length threshold	Minimum distance for considering that we have a congested region.	-
sending time	This variable indicates how often cars send PVD.	-
dx	Size of each cell in the discretization.	-

Table 4.1: Parameters of ASM detection.

In Table 4.1, aside from the meaning of each parameter, we also present some fixed values extracted from recommended values in the literature. One exception are the variables

ΔL_max and ΔT_max, that do not have any recommended values, but since they are not very relevant variables we fixed them to reasonable values. The parameters that we had not fixed are the ones more relevant and the ones that we tested.

The parameters σ^c and τ^c are paired because they define the size of the congested kernel for the ASM method. The same happens with parameters σ^f and τ^f that define the size of the free flow kernel. Thus, we joined them to define only the kernels sizes, whose several recommended sizes are given in the SPECIALIST paper. Taking this into consideration, the values tested were the following ones:

- (σ^c, τ^c): (600 m, 60 s), (200 m, 20 s) and (40 m, 10 s) .
- (σ^f, τ^f): (600 m, 60 s), (200 m, 20 s) and (40 m, 10 s).
- speed threshold: 20, 30, 40 and 50 (km/h).
- length threshold: 100, 200, 300, 500 and 700 (m).
- dx: 100, 150 and 200 (m).
- sending time: 1, 5 and 10 (s).

The kernel sizes are called large, medium and small to avoid writing the tuple of parameters in the figures.

As we just said, we did not perform a full factorial design because there would be too many experiments. What we did was to define a reference configuration of the parameters and then to analyze, for instance, the speed threshold we keep all the reference values fixed and only change the speed threshold. We did this for all the studied parameters.

In addition to these parameters, we also tested the algorithms with different penetration rates. The penetration rates tested were 5%, 10%, 20%, 50%, 70% and 100%. We tried all the combinations explained before with each penetration rate. Thus, with this different penetration rates we had 96 experiments for each network, that are doubled because each network is tested with and without fixed sensors.

The reference configuration of the parameters used is the following one:

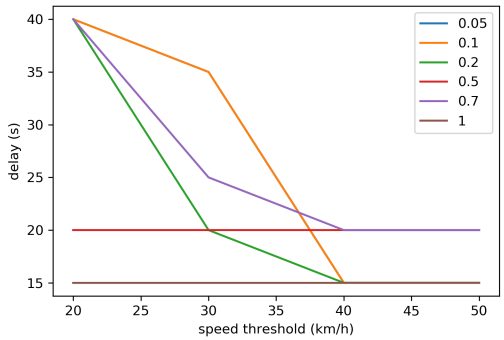
- (σ^c, τ^c) = (200 m, 20 s).
- (σ^f, τ^f) = (200 m, 20 s).
- speed threshold = 30 km/h.
- length threshold = 100 m.
- dx = 100 m.
- sending time = 5 s.

4.4.2 Results

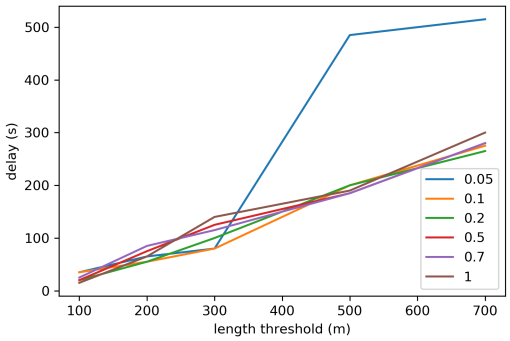
Here we present the results obtained with this detection algorithm. First we comment the results of the algorithms without using the fixed sensors (only with probe vehicle data) and then we will explain the results obtained taking into account also the fixed sensors.

1 lane without sensors

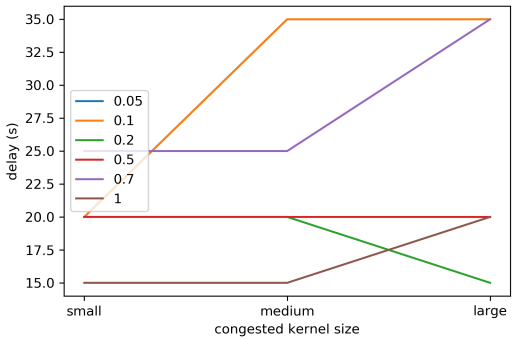
In this section we analyze the results obtained with the network of 1 lane without the fixed sensors. Since these are the first results, we explain here which was the expected effect of each parameter.



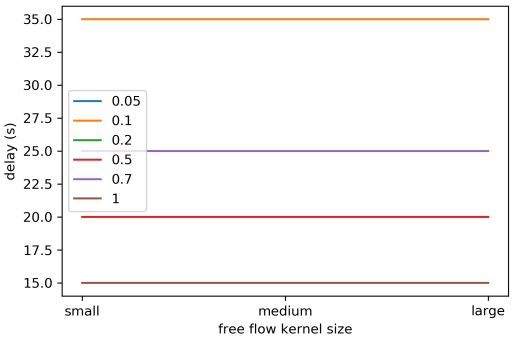
(a) Effect of the speed threshold on detection delay.



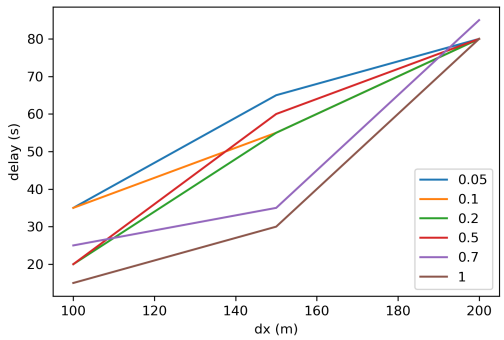
(b) Effect of the length threshold on detection delay.



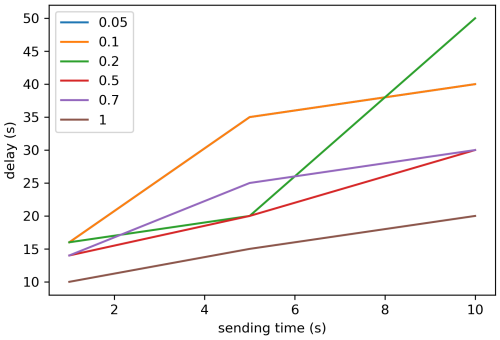
(c) Effect of the congested kernel size on detection delay.



(d) Effect of the free flow kernel size on detection delay.



(e) Effect of the discretization on detection delay.



(f) Effect of the sending time of the PVD on detection delay.

Figure 4.2: Effects of the parameters of ASM detection on delay for the 1 lane network without fixed sensors.

In Figure 4.2a we observe that the detection time decreases when the speed threshold increases, as we could expect. This happens because if we define a shockwave as a region with a speed lower than this threshold, a higher threshold will be less selective so more vehicles will be added to the shockwave. Thus, it will be larger sooner and will be detected before. However, we observe that the behaviour depends on the penetration rate. For a penetration rate of 100%, the detection time does not change at all, and neither does for a penetration rate of 50%.

In Figure 4.2b the trend observed also corresponds to the expected one. However, in this case the relation is even more clear and very similar for all penetration rates except 5%, that has the detection time highly increased for length thresholds greater than 400 m. If we increase the length threshold, we require the shockwave to be much larger to be detected, thus this time to grow is more time that the detection is delayed. It is also remarkable that the increase in the delay due to the length threshold is much larger than the one that is observed with other parameters. Here the delay may range from 10 to 300 s for the majority of penetration rates, where in the other parameters the range is much smaller, between 10 to 80 s.

We can see the effects of the kernel size in figures 4.2c and 4.2d. The free flow kernel does not affect at all the delay for any penetration rate. In contrast, the congested kernel size has very different effects on each penetration rate. This makes sense because near the shockwave, the congested kernel has almost all the weight, so it makes sense that the free flow kernel does not have any effect at all. It is difficult to explain the effect of the congested kernel size. In general, it increases the detection time. But for penetration rate of 20% it decreases it and the 50% penetration rate is not affected at all, so there is no clear effect.

Figure 4.2e shows the effect of the discretization. As expected, a higher discretization (minor dx) leads to faster detections. This behaviour is consistent for all penetration rates. This happens because the shockwave is better located and sooner some of the discretization points have a lower speed. With a large cell size, there are many vehicles that contribute to the speed of that cell, so more of them need to be in the shockwave to have a significant effect on the speed computed on the cell. If the cell is smaller, we need less vehicles in the shockave to change significantly the speed computed at that cell.

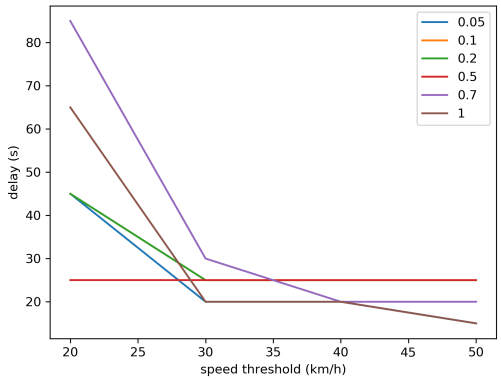
In Figure 4.2f the effect of sending time is represented. Again, the expected values are the ones observed, higher sending times lead to a greater delay in the detection This happens because the PVD data is updated more often, so the new vehicles affected by the shockwave are detected earlier.

Overall, we observe that with 50% of penetration rate the algorithm shows a better performance than with 70%. Further, we had expected the different lines corresponding to the penetration rates to be more clearly defined in the different figures, however we see them crossing in almost all of them. This means that the penetration rate does not have a clear effect of improving the performance if the penetration rate is higher, as it seems reasonable and we assumed before running these experiments. It is also remarkable that the delay that can be achieved is in the range 10-20 seconds with appropriate values for the algorithm.

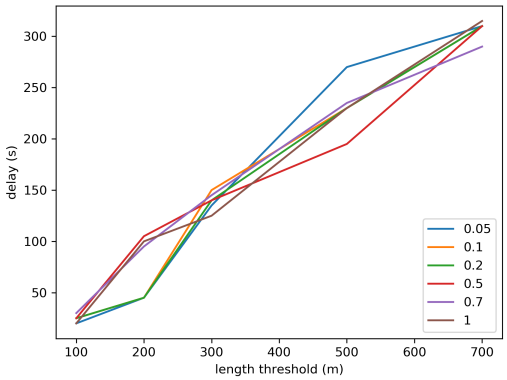
2 lane without sensors

Here we present the results obtained with the second network, where there are 2 lanes.

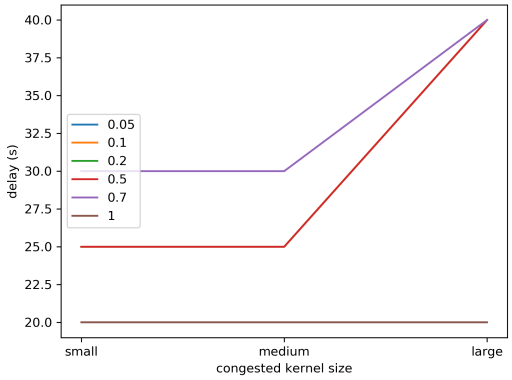
In Figures 4.3a and 4.3b we observe the same effects as in the previous network. The main difference is that, regarding the length threshold, we observe that the 5% penetration



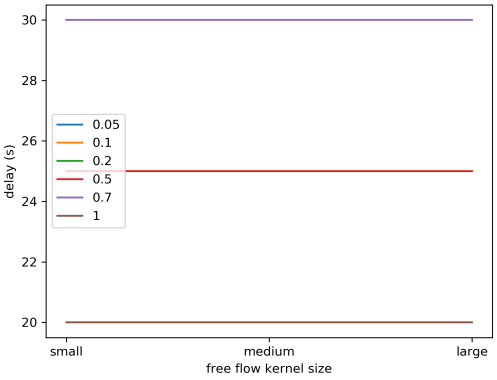
(a) Effect of the speed threshold on detection delay.



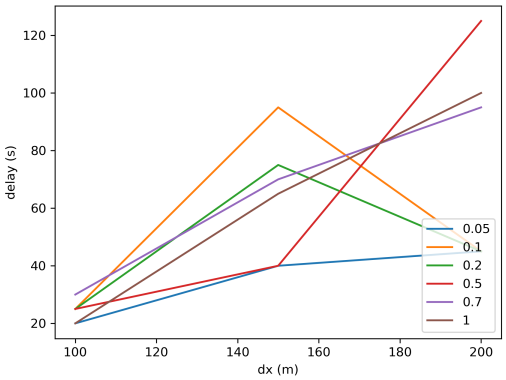
(b) Effect of the length threshold on detection delay.



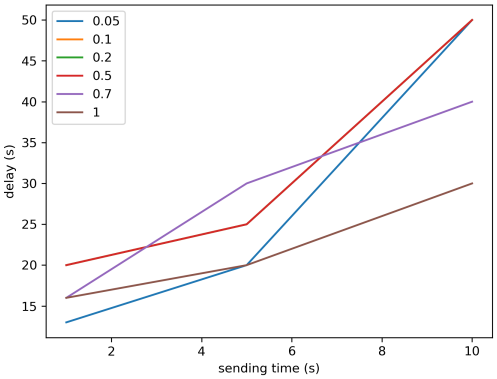
(c) Effect of the congested kernel size on detection delay.



(d) Effect of the free flow kernel size on detection delay.



(e) Effect of the discretization on detection delay.



(f) Effect of the sending time of the PVD on detection delay.

Figure 4.3: Effects of the parameters of ASM detection on delay for the 2 lanes network without fixed sensors.

rate does not show a high increase of the delay compared to the one obtained with other penetration rates.

The effect of the congested kernel size shown in Figure 4.3c in this case is clearer, it increases the delay, while in the previous scenario this effect was not clear. The other kernel (Figure 4.3d) does not have any effect again, like in previous experiments.

The curves in Figure 4.3e are surprising. For two penetration rates, a discretization of 200m presents a better result than with 150m. This was unexpected and was not observed in the previous scenario. This is not a clear effect, because some penetration rates show the expected trend, but it is remarkable.

Figure 4.3f presents also the same shape than the same figure of the previous scenario.

To sum up, we observe some differences between this scenario and the previous one. In particular, in the effects of the congested kernel size and in discretization. However, the strange fact that in many cases a 50% penetration rate works better than a 70% also happens here. In this scenario, also the best detection delays that can be achieved are between 10 and 20 seconds.

It is also noticeable that in these experiments the different penetration rate do not show again a consistent behaviour where higher penetration rates lead to better results.

3 lane without sensors

This section presents the results obtained with the scenarios of the 3 lanes dummy network.

Again, Figure 4.4a exhibits a similar shape to the one observed in the same figure for network with 1 lane. We see a trend that the greater the speed the lower the delay, but not for all penetration rates.

Figure 4.4b also presents a similar behaviour than the previous ones. Like in the second scenario, there is no relevant increase in the delay for a 5% penetration rate. Again, the relation is almost linear for all penetration rates and gets a very high detection delay of 300 s.

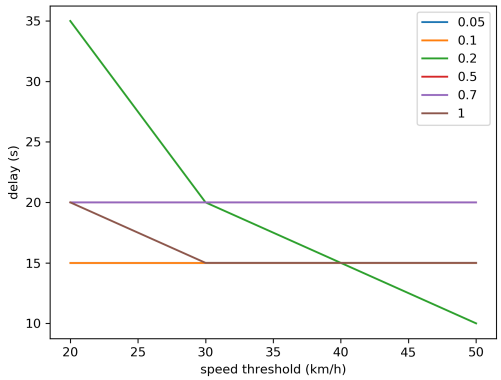
Where we can observe a significant difference is in Figure 4.4c. In this case, there is no clear pattern. For some penetration rates it is better to use a small congested kernel, for others it is better the large. Also, for 100% penetration rate increasing to a large congested kernel increases the delay, while for a 20% of penetration rate, it improves it. In contrast to this kernel, the free flow kernel does not affect at all again, as we appreciate in Figure 4.4d.

In Figure 4.4e we observe a shape similar to the one in the first scenario, though there are some penetration rates for which increasing the size of the discretization (this means, less points) does not mean a decrease in the performance. For penetration rate of 10%, we even observe a slight increase. This is similar to the effect observed for the scenario 2, but no so clear.

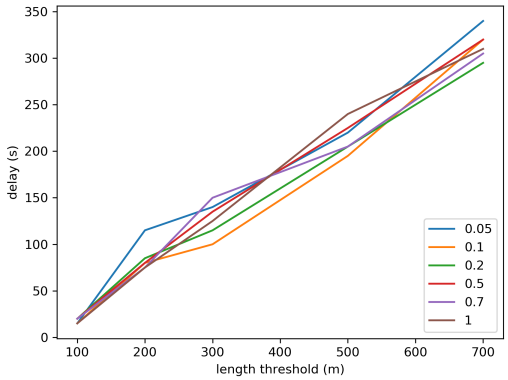
Finally, in Figure 4.4f the behaviour is the expected one again, the delay decreases with the sending time.

In general, the effect that we had observed before than with a 50% of penetration rate similar performances, even better ones than the ones obtained with 70% of connected cars is also appreciated in this scenario, particularly in Figures 4.4c and 4.4d.

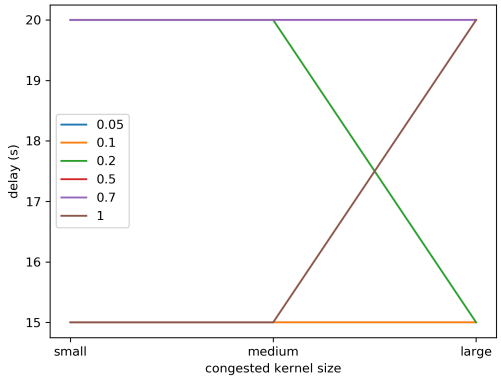
The fact that higher penetration rates do not lead to better results is observed in this network too. With the best configurations, we achieve a delay of 10-20 s.



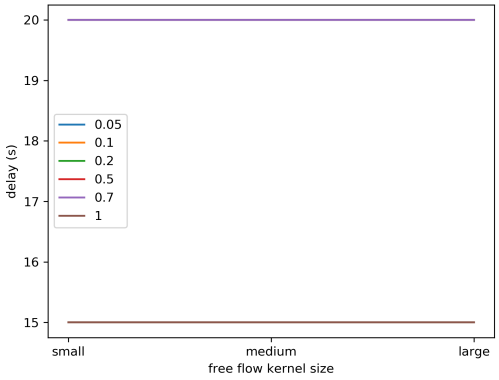
(a) Effect of the speed threshold on detection delay.



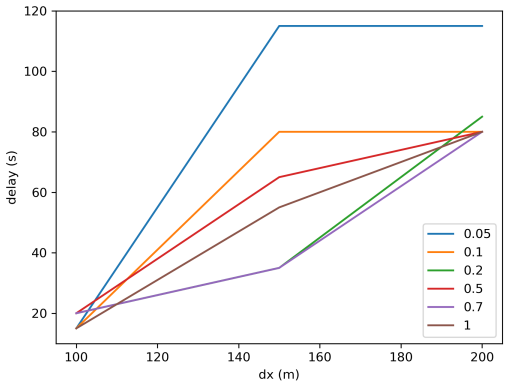
(b) Effect of the length threshold on detection delay.



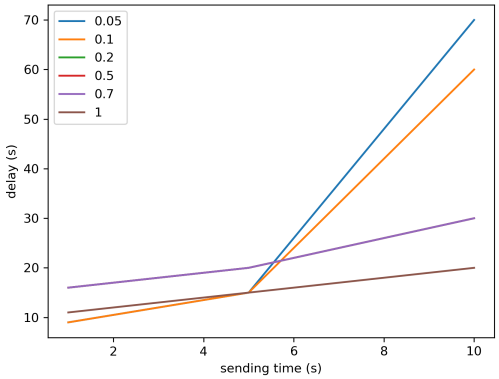
(c) Effect of the congested kernel size on detection delay.



(d) Effect of the free flow kernel size on detection delay.



(e) Effect of the discretization on detection delay.



(f) Effect of the sending time of the PVD on detection delay.

Figure 4.4: Effects of the parameters of ASM detection on delay for the 3 lanes network without fixed sensors.

1 lane with sensors

We comment in this section the results obtained with the 1 lane network with fixed sensors, that are places one each 500 m.

In Figure 4.5a we can appreciate that the trend is the same presented by the scenarios without sensors. However, the delay range of 10-70 s is larger than the one obtained without sensors.

Figure 4.5b presents also the same shape that in the scenarios without sensors. However, on network 1 without sensors we saw a high increase of the delay for a 5% of penetration rate that does not appear here. This is probably due to the presence of the fixed sensors. The other penetration rates show very similar results than without sensors.

Again, the effect of the congested kernel size (Figure 4.5c) is not clear and depends on the penetration rate. In contrast, and also as in the previous scenarios, the free flow kernel size does not have any effect.

In this case, the effect of the discretization is the expected one (Figure 4.5e). In fact, the figure is very similar to the one without sensors.

The last Figure 4.5f presents the expected shape. It is remarkable thought that a 10% penetration rate appears to work better than the 20%.

We can observe that the patterns observed are similar than the ones observed without the magnetometers. So, it seems that having fixed sensors does not help to detect shockwaves. We should also notice that the presence of detectors neither does deteriorate the detection.

2 lane with sensors

Here we explain the results that the detection in the 2 lane network showed with fixed sensors.

The effect of the speed threshold shown in figure 4.6a is the same one observed in all previous figures that represent the same effect. We notice also from this figure that the penetration rate that shows an overall lower delay is the lowest one, with 5% of penetration rate. In addition, we observe that the figure is almost identical to its analogous without fixed sensors.

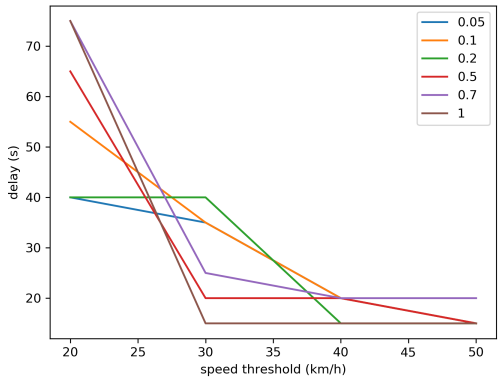
Figure 4.6b presents a shape very similar to all the previous ones too. It is remarkable, though, that in this case the penetration rates of 5% and 20% have a considerably lower delay for a threshold of 300 m.

In Figure 4.6c we observe a trend that a greater congested kernel size gives a larger delay. In this figure this is quite clear, while in the previous ones the effect of this variable was not clear at all. It is also noticeable that the lower delay is achieved with the lower penetration rate, while the 100% penetration rate has the worst performance. This same surprising result appears in Figure 4.6d.

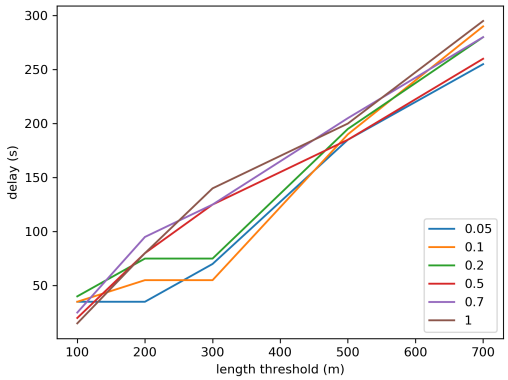
Figure 4.6e shows the same strange shape that Figure 4.3e (the same figure in scenario 2 without sensors). Some penetration rates show a better performance with a less fine discretization. As we commented in the other figure, this was unexpected.

The last Figure 4.6f does show the expected behaviour. However, we see again that the lower penetration rate has the best delay and the 100% penetration rate shows the worst one.

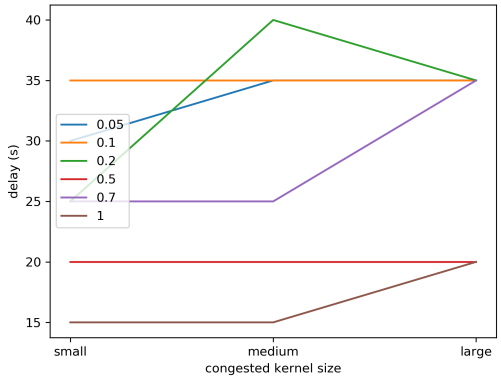
To sum up, these figures show again that it seems that the use of fixed sensors does not improve the detection. In fact, there is not a significant difference between these results and the ones without sensors, so the same conclusion explained in Section 4.4.2 applies here.



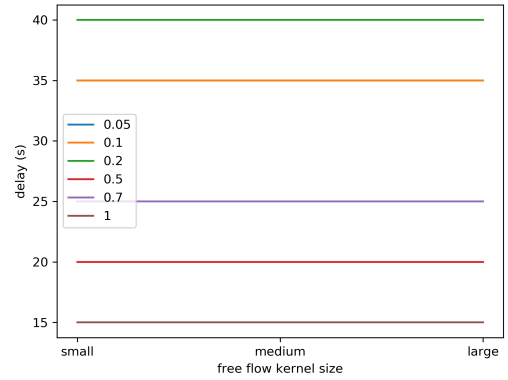
(a) Effect of the speed threshold on detection delay.



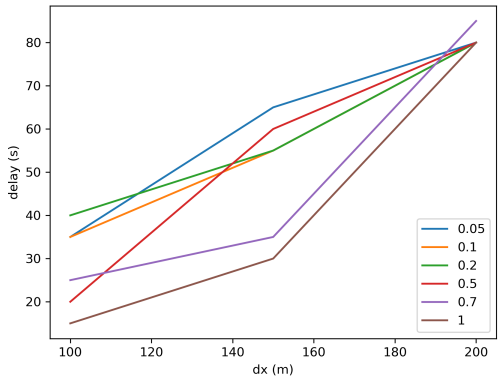
(b) Effect of the length threshold on detection delay.



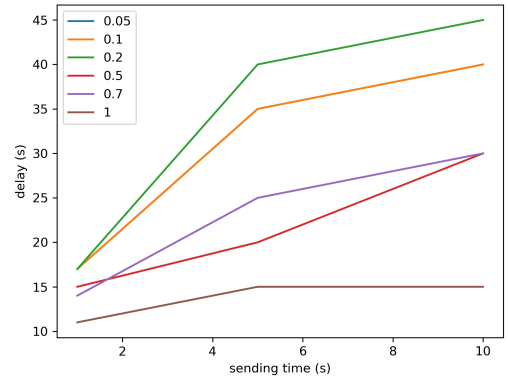
(c) Effect of the congested kernel size on detection delay.



(d) Effect of the free flow kernel size on detection delay.

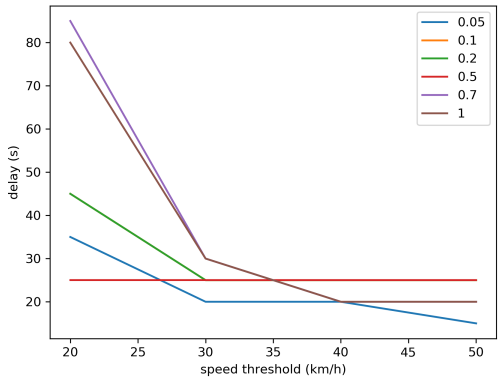


(e) Effect of the discretization on detection delay.

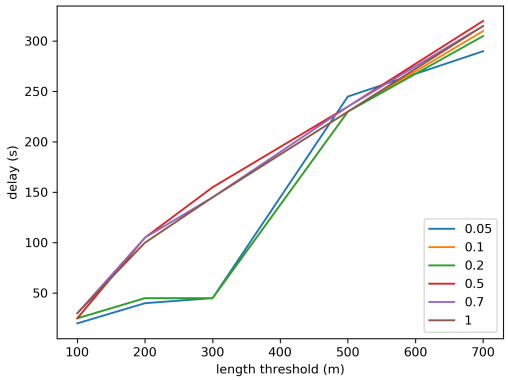


(f) Effect of the sending time of the PVD on detection delay.

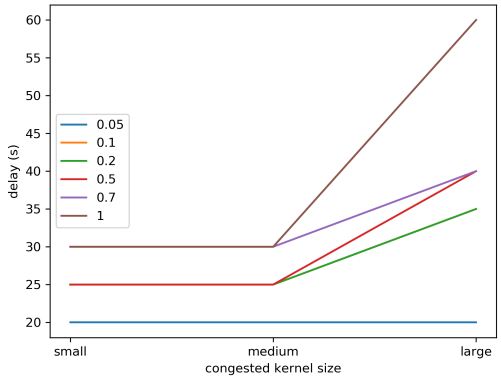
Figure 4.5: Effects of the parameters of ASM detection on delay for the 1 lane network with fixed sensors.



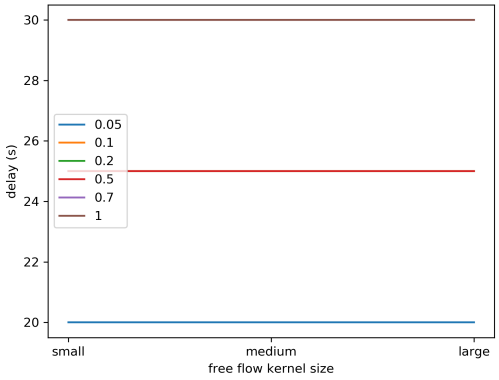
(a) Effect of the speed threshold on detection delay.



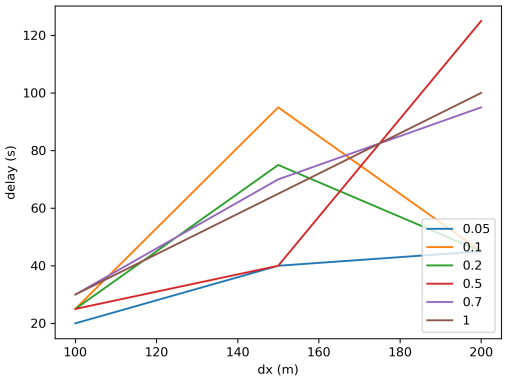
(b) Effect of the length threshold on detection delay.



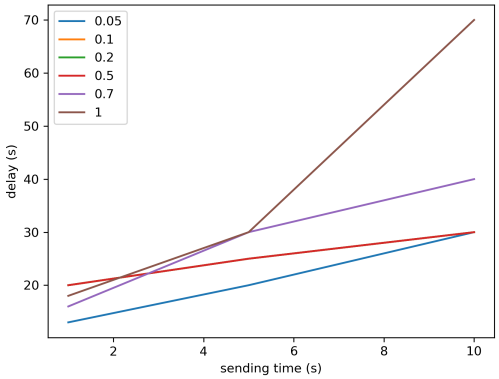
(c) Effect of the congested kernel size on detection delay.



(d) Effect of the free flow kernel size on detection delay.



(e) Effect of the discretization on detection delay.



(f) Effect of the sending time of the PVD on detection delay.

Figure 4.6: Effects of the parameters of ASM detection on delay for the 2 lanes network with fixed sensors.

3 lane with sensors

The results of the ASM detection algorithm for the 3 lane network with sensors are presented here.

We can see that Figure 4.7a presents the same shape that all others, a greater threshold leads to a faster detection (lower delay). Nevertheless, in this case the greater values of delay achieved are considerably larger than the maximum values obtained on the same scenarios without sensors.

Figure 4.7b shows the same trend observed in the other scenarios. Although, we see that with 5% and 10% of penetration rate a better performance is achieved.

Surprisingly, in this scenario it is the congested kernel size the parameter that has no effect (Figure 4.7c) while the free flow kernel size has some effect (Figure 4.7d). It is also surprising that the lowest penetration rates have a better performance than the 100% penetration rate.

The discretization (Figure 4.7e) in this case behaves as expected. We observe that here the lower penetration rates do perform worst, as expected, but the 100% penetration rate is not the best one. This happens again with the sending time (Figure 4.7f), where the an increasing sending time decreases the performance but again the worst performance is achieved with 100% penetration rate.

From all these figures, we observe again that the effect of the penetration rate is unexpected because the 100% penetration rate is far from achieving the best performance. We also observe that fixed sensors do not improve the detection significantly. There is one remarkable exception, which is that for a sending time of 10 s, the delay is considerable reduced for penetrations rates of 70% and 100%.

4.4.3 Conclusions

Overall, it is remarkable that the algorithm did not perform as expected regarding the penetration rate. We expected the algorithm to perform better with higher penetration rates, but we do no observe this. There is not any clear pattern with respect of how the penetration rate affects. It is worth mentioning that this happens in both cases, with and without fixed sensors. In contrast, the other parameters, behave quite similarly to what we had expected.

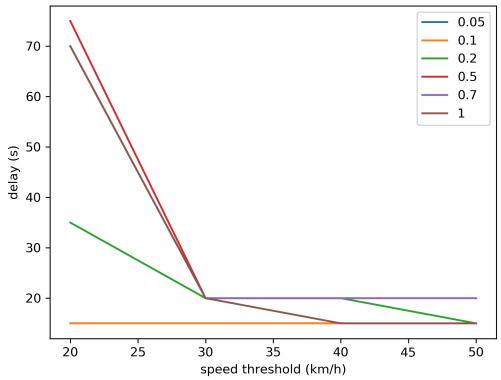
Another relevant conclusion is that there is no significant difference in detection delay due to the use of fixed sensors. The best (and worse) delays achieved in both types of scenarios are very similar. This is remarkable because we could expect that with low penetration rates the detectors would help, but this is not observed.

The average times of execution of each simulation are shown in Table 4.2. As we can see, the execution times are much higher with the sensors. This is because with the sensors we have to perform many more computations: we have to update the speeds computed with the sensors and we also have to perform the fusion algorithm, that is not needed if we only have PVD data.

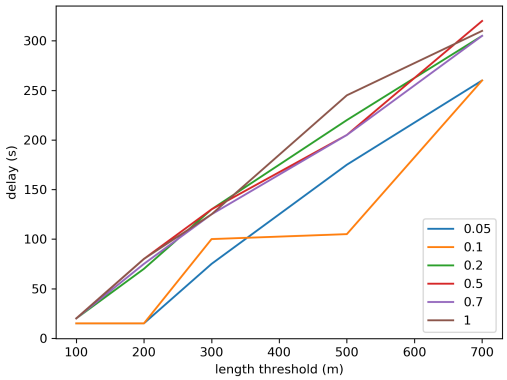
scenario	with sensors	without sensors
1 lane	30 s	14 s
2 lane	80 s	37 s
3 lane	99 s	58 s

Table 4.2: Average simulation time for ASM detection scenarios.

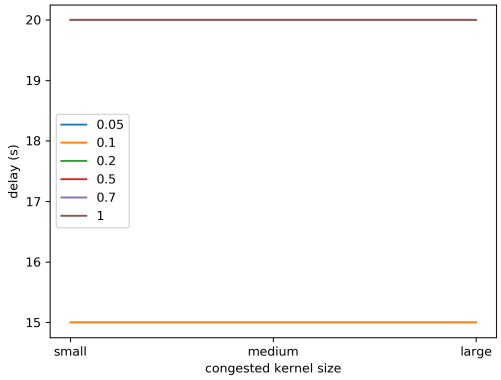
For each scenario, the best parameters are the ones shown in Table 4.3. These parameters



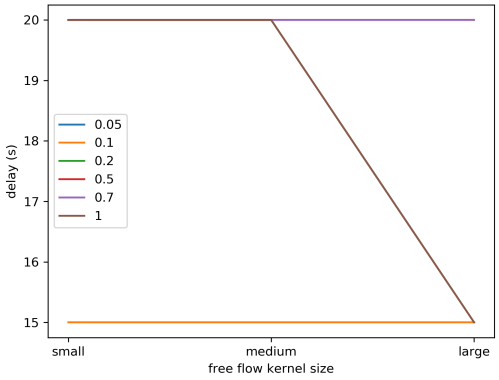
(a) Effect of the speed threshold on detection delay.



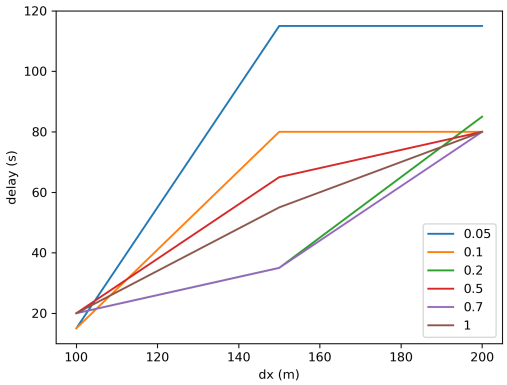
(b) Effect of the length threshold on detection delay.



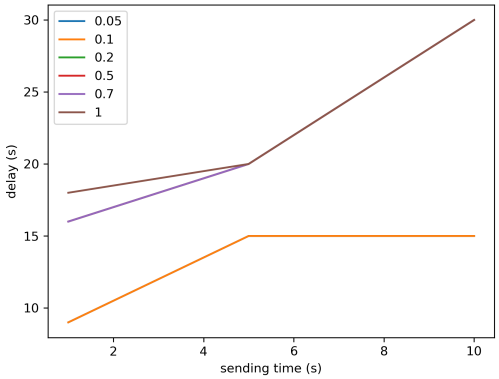
(c) Effect of the congested kernel size on detection delay.



(d) Effect of the free flow kernel size on detection delay.



(e) Effect of the discretization on detection delay.



(f) Effect of the sending time of the PVD on detection delay.

Figure 4.7: Effects of the parameters of ASM detection on delay for the 3 lanes network with fixed sensors.

are the ones used in the mitigation experiments to detect the shockwave. The delay achieved with these best configurations depends on each one, but it is around 10 s. The results presented in Hegyi et al. [2013] also achieve a detection delay lower than a minute (no precise results are presented in that paper, just figures). So, we achieve a similar detection delay.

network	pr	σ^f (m)	τ^f (s)	σ^c (m)	τ^c (s)	speed threshold (km/h)	length threshold (m)	sending time (s)	dx (m)
1 lane	0.05	600	60	600	60	50	100	1	100
1 lane	0.1	600	60	600	60	50	100	1	100
1 lane	0.2	600	60	600	60	40	100	1	100
1 lane	0.5	600	60	600	60	50	100	1	100
1 lane	0.7	600	60	600	60	40	100	1	100
1 lane	1	600	60	600	60	30	100	1	100
2 lane	0.05	600	60	600	60	50	100	1	100
2 lane	0.1	600	60	600	60	30	100	1	100
2 lane	0.2	600	60	600	60	30	100	1	100
2 lane	0.5	600	60	600	60	20	100	1	100
2 lane	0.7	600	60	600	60	40	100	1	100
3 lane	1	600	60	600	60	40	100	1	100
3 lane	0.05	600	60	600	60	20	100	1	100
3 lane	0.1	600	60	600	60	20	100	1	100
3 lane	0.2	40	10	600	60	50	100	1	100
3 lane	0.5	600	60	600	60	30	100	1	100
3 lane	0.7	600	60	600	60	30	100	1	100
3 lane	1	40	10	600	60	40	100	1	100

Table 4.3: Best parameters for ASM detection algorithm for each scenario.

4.5 Experiments of Izadpanah's detection

This section explains the experiments done to analyze the effect of the parameters of Izadapanah's detection algorithm. We present the design of the experiments, the results obtained and finally some conclusions.

4.5.1 Design of experiments

This algorithm has the parameters shown in Table 4.4:

parameter	**description**	**fixed**
speed threshold	It defines the threshold change of speed in an inflection point.	-
sending time	This variable indicates how often cars send PVD.	-
resampling	This is the number of resampling in the clusterwise linear regression.	1
γ	This is the weight of the k-means component in the clusterwise linear regression.	0.1

Table 4.4: Parameters of Izadpanah's detection.

The values related with the clusterwise linear regression, the resampling and the γ, are fixed because we only have one shockwave in the scenarios that we have tested. The clustering parameters are relevant when there are many shockwaves, which is not our case, so calibrating

these parameters is not important in our case. We would require an scenario with more than one shockwave to perform a good analysis and this is out of the scope of this master thesis.

Thus, the only two parameters analyzed are the threshold and the sending time. The different values tested are the following ones:

- threshold: 20, 30, 40 (km/h)
- sending time: 1, 5, 10 (s), like in the previous algorithm

Despite having only two parameters to analyze, we did not perform a factorial design. We could not because this algorithm is much slower than the previous one (the time to simulate in commented in Section 4.5.3). We used the same design than with the first experiments, being the reference parameters:

- threshold = 20 km/h
- sending time = 5 s

Again, we run all experiments for many penetration rates. We used the following penetration rates: 5%, 10%, 20% and 50%. Again, we could not simulate larger penetration rates (70% and 100%) because this algorithm is too slow and the simulation time increases with the penetration rate. Since this algorithm does not make use of the fixed sensors, we only have the scenarios without them.

4.5.2 Results

In this section the results obtained are presented and commented.

1 lane

In Figure 4.8a we can see that the threshold does not have any effect in the delay. What we can observe that the detection time is consistently lower with a higher penetration rate.

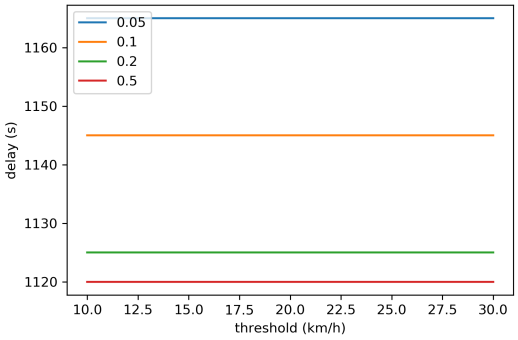
Figure 4.8b shows that the lower the sending time, the better is the detection. We also observe here that the penetration rate improves the detection.

2 lanes

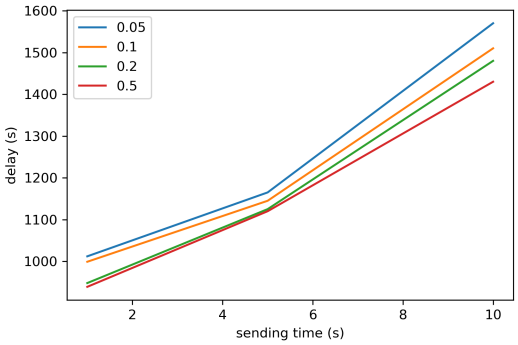
We can see that in Figure 4.9a there is a huge difference, that leads to a faster detection if the threshold is lower. We also see that there is a clear reduction of the delay with the higher penetration rate. The 50% penetration rate achieves a 200 s delay, which is very low compared the results of the previous networks with the same algorithm.

Figure 4.9b is quite similar to the analogous one of the scenario with 1 lane. The main difference is that in this case for a sending time of 5 s and with a penetration rate of 50% the detection is surprisingly fast, more than with a sending time of 1 s.

In this network the best configuration achieves a detection time much faster than the best one of the previous network. Again, we see consistent results with the penetration rate. Here, higher penetration rates have a lower delay.

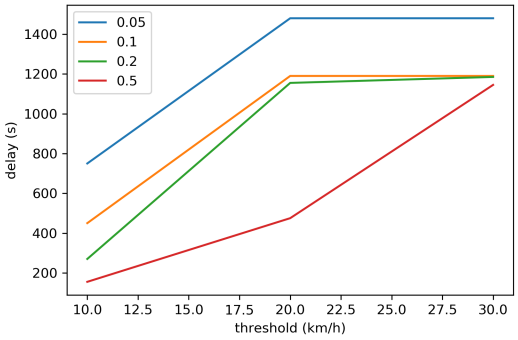


(a) Effect of the speed threshold on detection delay.

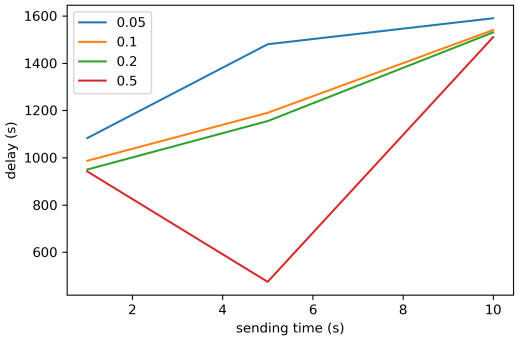


(b) Effect of the sending time on detection delay.

Figure 4.8: Effects of the parameters of Izadpanah's detection on delay for the 1 lane network.



(a) Effect of the speed threshold on detection delay.



(b) Effect of the sending time on detection delay.

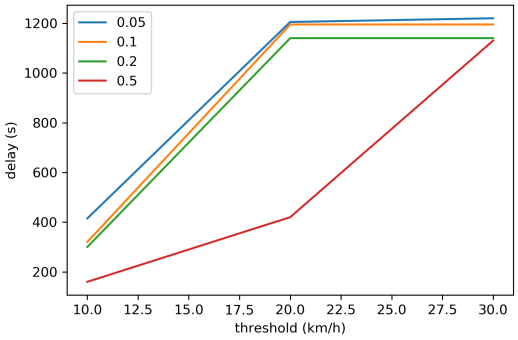
Figure 4.9: Effects of the parameters of Izadpanah's detection on delay for the 2 lane network.

3 lanes

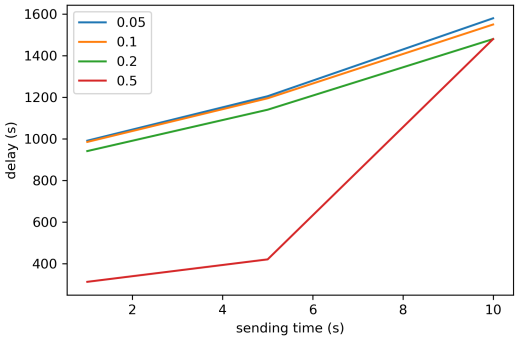
Again, Figure 4.10a is very similar to the one observed in the second scenario. Just that in this case the difference of delay between 5% penetration rate and 10% is very small. With a 50% penetration rate we achieve a quite faster detection, just under 200 s of delay too. In this case, the other penetration rates achieve a delay under 800 s.

In Figure 4.10b we observe that the differences between the lower penetration rates are very small too. In this case, the 50% penetration rate shows the same trend than the others, the lower is the sending time the lower is the delay, like we observed in the 1 lane scenario but not in the previous one.

The summary is that in this scenario the results are better, but still far from the ones obtained with ASM detection. We also observe here that the effect of the penetration rate is the expected one.



(a) Effect of the speed threshold on detection delay.



(b) Effect of the sending time on detection delay.

Figure 4.10: Effects of the parameters of Izadpanah's detection on delay for the 3 lane network.

4.5.3 Conclusions

In conclusion, we observe that the delay obtained with this algorithm is much higher than the one achieved with the previous one. The lower delay achieved in between 150-200 seconds. In addition, the time to execute the experiments is also much higher, the averages for scenarios with 1, 2 and 3 lanes are, respectively: 84 min, 117 min and 172 min. So, this algorithm results in being much slower and it shows a greater delay than the previous one, around 20 times larger. Since in all aspects it is worst than the previous one, the mitigation experiments with the SPECIALIST algorithm.

One remarkable aspect is that with this algorithm we observe that the penetration rate affects the results as expected. Here, a greater penetration rate decreases the detection delay. This contrasts with the results obtained with the previous algorithms, where the penetration rate did not improve the delay.

The best parameters for each scenario are shown in Table 4.5.

network	pr	threshold (km /h)	sending time (s)
1 lane	0.05	20	1
1 lane	0.1	20	1
1 lane	0.2	20	1
1 lane	0.5	20	1
2 lane	0.05	10	1
2 lane	0.1	10	1
2 lane	0.2	10	1
2 lane	0.5	10	5
3 lane	0.05	10	1
3 lane	0.1	10	1
3 lane	0.2	10	1
3 lane	0.5	10	1

Table 4.5: Best parameters for Izadpanah's detection algorithm for each scenario.

4.6 Experiments of SPECIALIST

In this section we explain the mitigation experiments performed with the SPECIALIST algorithm. First we describe the design of experiments and then the results obtained. Since only the first detection algorithm showed a good performance, we only apply the SPECIALIST with this algorithm. In addition, in order to define the length of the VSL region, we need to obtaine measures of the flow, so only scenarios with fixed sensors are considered. This is because we only can measure the flow with fixed sensors, not with PVD.

4.6.1 Design of experiments

SPECIALIST has the following parameters, ρ_4, ρ_5, q_5 and vsl, as explained in Section 3.3. These parameters are, respectively, density of region 4, density of region 5, flow of region 5 and the variable speed limits. The parameters used for ASM detection are the ones that had a better performance in their experiments (shown in Table 4.3). Regarding the SPECIALIST parameters, we must take into account that they are not independent. Not all possible combination of them are feasible. There are the following restrictions based on the resolvability conditions explained in Hegyi et al. [2008]. The variables c_{54} and c_{46} are the propagation speed of the border between these two sections.

- $c_{54} < 0 \Rightarrow q_5 > q_4$ and $\rho_4 > \rho_5$
- $c_{46} < 0 \Rightarrow q_6 > q_4$ and $\rho_4 > \rho_6$
- $c_{54} < c_{46} \Rightarrow \frac{q_5 - q_4}{\rho_5 - \rho_4} < \frac{q_4 - q_6}{\rho_4 - \rho_6}$
- conditions on state 5 $\Rightarrow q_5 > q_1$ and $d_5 > d_1$

In the above constraints, we have to take into account that $q_4 = vsl \cdot \rho_4$, so it is defined by the parameters.

Thus, only those combinations of parameters that fulfil these conditions can be used. We must take into account that the values ρ_6 and q_6 are not known, they are computed by the algorithm. However, we computed these states before and then we used them to select the feasible parameters, because region 6 is the free flow state.

We can use these constraints to avoid executing simulations that we know that will not trigger the mitigation. For this, we estimated the values of q_6 and ρ_6 for the three scenarios. They are different, so this is why we have chosen different parameters for each scenario. For instance, none of the combinations that works for the scenario with 3 lanes activates the algorithm in the scenario with 1 lane, because their flow properties are very different. The parameters used for each scenario are shown in Table 4.6.

parameters	**scenario 1**	**scenario 2**	**scenario 3**
ρ_4 (veh/km/lane)	50, 54, 58	24, 28, 32	18, 22, 26
ρ_5 (veh/km/lane)	46, 50, 54	20, 24, 28	14, 18, 22
q5 (veh/h)	4000, 4500, 5000	2000, 2500, 3000	1500, 2000, 2500
vsl (km/h)	40, 50, 60	40, 50, 60	40, 50 , 60

Table 4.6: Parameters tested for each scenario.

We performed a full factorial design, because after applying the conditions there are not so many configurations. Running a simulation for each configuration with all penetration

rates (the same ones used in the experiments of ASM detection) we have to perform: 162, 126 and 180 experiments for the 1 lane, 2 lanes and 3 lanes scenarios, respectively.

4.6.2 Results

Here we present the results obtained with SPECIALIST for all scenarios and all penetration rates. We must take into account that some of the previously defined values of the parameters may be missing. This happens because, in addition to the explained constraints, there are more constraints that are checked when all the traffic states are estimated and it may happen that a particular values do not activate the variable speed limits or they are activated but they do not mitigate the shockwave. For each scenario, we show the number of configurations that can set the VSL and how many of them actually mitigate the shockwave.

1 lane

In this section we present the results obtained with the SPECIALIST. In this case, of the 162 experiments, the SPECIALIST is activated in 144 cases but only in 51 of them actually mitigated the shockwave. So, just in 35% of the cases where the VSL are activated the shockwave is solved.

Figures 4.11a, 4.11b, 4.11c and 4.11d look the same. We see that the parameters barely have any effect. We also can see that the algorithm does not seem to improve the network at all. The TTS is the same that we had without activating any algorithm. We observe that some of the values that we tried did not work, because no configuration with them activated the speed limits and mitigated the shockwave.

The figures of the last instantaneous TTS, 4.12a, 4.12b, 4.12c and 4.12d, show the same. There is no improvement at all, and all penetration rates behave the same. Seeing these results, we can say that the algorithm did not work with this scenario.

2 lanes

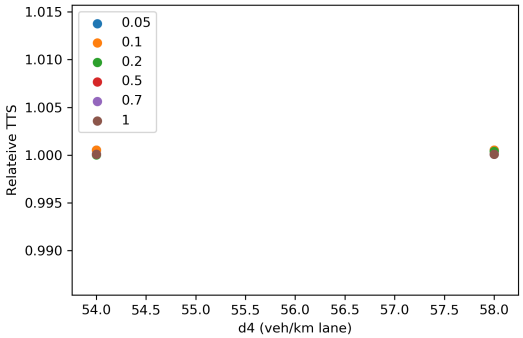
The results obtained with this scenario are much better than the previous ones. In this case, the algorithm was activated in 126 cases and mitigated the shockwave in 90, so in a 71% of the cases it solved the shockwave.

Nevertheless, the results obtained show that the TTS does not improve. In Figure 4.13a we observe that one of the values tested does not work with any combination, that is why it does not appear in the figure. We observe that the TTS is always greater than the one obtained in the reference scenario (the shockwave without mitigation) but not much, just between 0.5 - 2%. The lower value of the density ρ_4 shows better results than the higher. Again, it does not seem to be a pattern regarding the penetration rate, we do not observe an improvement with the higher penetration rate.

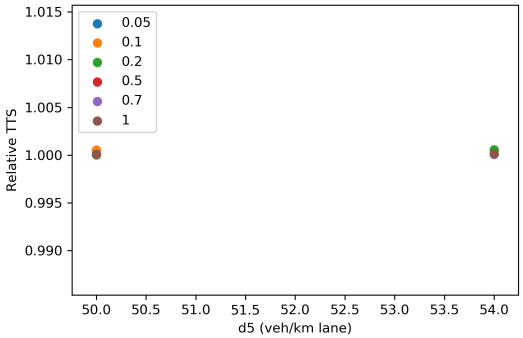
Figure 4.13b shows the effect of ρ_5. One of the proposed values did not work. The results obtained are very similar to the ones of ρ_4. The lower value works better, and the range of the TTS is again between 0.5-2% higher than the reference value. There is also no clear effect of the penetration rate.

In Figure 4.13c we see that the effect of q_5 is irrelevant, it behaves the same for all given values. The TTS is between 0.5-1.5% higher.

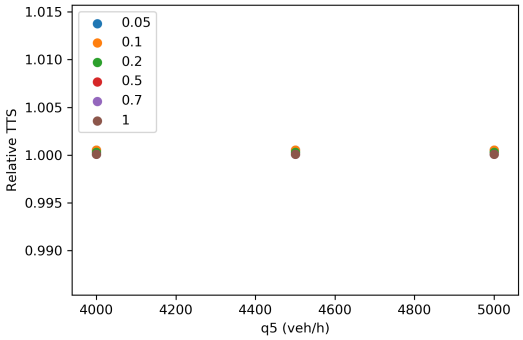
The effect of the VSL is represented in Figure 4.13d. In this case some of the values actually improve the one obtained with the reference scenario. We see a trend which is



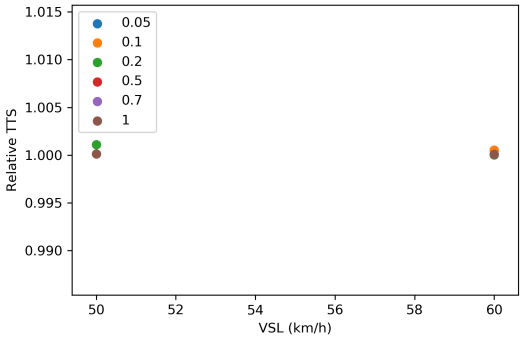
(a) Effect of the density of state 4 on the total time spent.



(b) Effect of the density of state 5 on relative total time spent.



(c) Effect of the flow 5 on the total time spent.



(d) Effect of the variable speed limits on the total time spent.

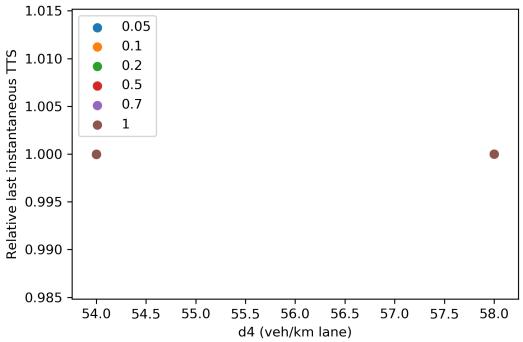
Figure 4.11: Effects of the SPECIALIST parameters on the total time spent for the 1 lane network.

that the higher the VSL, the lower the TTS. This makes sense, because if the VSL are higher, vehicles travel at a higher speed. The only problem would be if it is too high and the shockwave is not mitigated, but this does not happen with these parameters. The penetration rates do not seem to have any clear behaviour like in the previous figures.

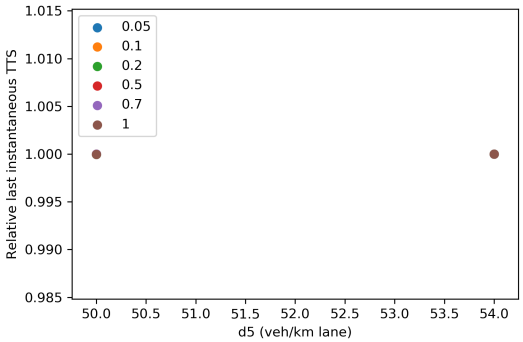
The last instantaneous TTS actually shows some improvement. In Figure 4.14a we observe that with all penetration rates we achieve a measure that is better than the reference one. The reduction is between a 10-15%. This same reduction is also observed in Figure 4.13b. It is also remarkable that with the higher density, 28 veh/km/lane, all penetration rates show an improvement in the delay.

We observe in Figure 4.14c that the flow of state 5 does not have any effect at all. What we observe is that all penetration rates show some improvement. This improvement in quite spread and ranges from 10% to 15%.

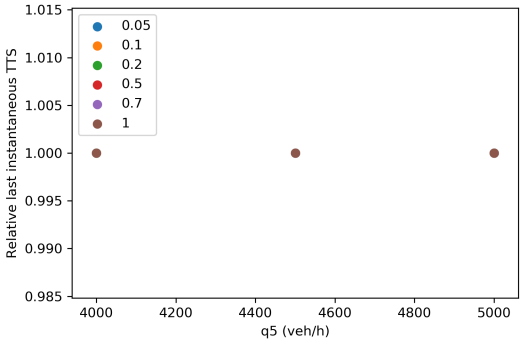
The last figure of this scenario, Figure 4.14d shows the effect of the VSL. With 60 km/h we have spread results, some penetration rates improve a 10% and other do not show any improvement. With 50 km/h, all penetration rates have a significant improvement in last instantaneous TTS of 12-15%. For a VSL of 40 km/h, the improvement depends on the penetration rate, but only a 50% penetration rate shows a considerable reduction of the



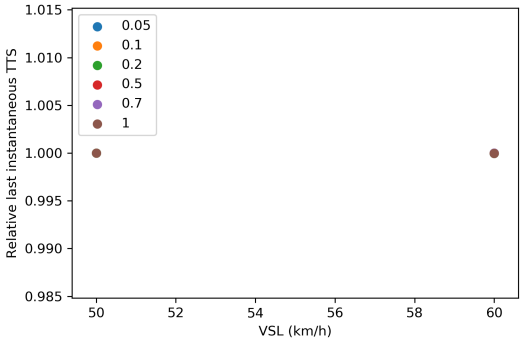
(a) Effect of the density of state 4 on the total time spent.



(b) Effect of the density of state 5 on the total time spent.



(c) Effect of the flow 5 in the total time spent.



(d) Effect of the variable speed limits on the total time spent.

Figure 4.12: Effects of the SPECIALIST parameters on last instantaneous TTS for the 1 lane network.

measure in a 12%.

In theese scenarios, we observe that the algorithm works. The TTS does not change, but it does the last instantaneous TTS. This indicates that the state of the network in which it is left is better than the reference state where the shockwave is not solved.

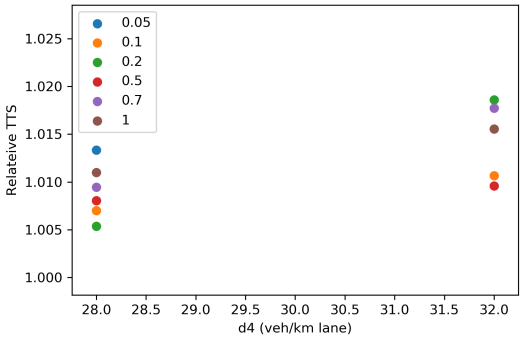
3 lanes

For this scenario, the SPECIALIST is activated with 171 of the configuration and in 126 of them the shockwave is mitigated. So, in a 74% of the cases where it is activate it mitigates the shockwave.

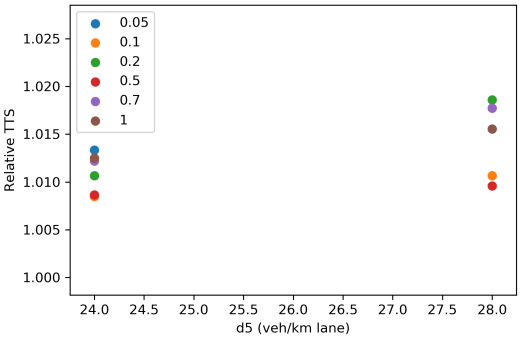
In Figure 4.15a we observe that in all cases we obtain a value that is larger than the one obtained in the reference scenario. However, the increase is quite small, it only achieves a 3% at most. We also observe that this increase grows with the value of ρ_4 and also that different penetration rates tend to give closer results.

In Figure 4.15b we observe a similar pattern than in the previous one. We see a trend where the greater the density, the bigger is the TTS. This increase is similar than the previous one, at most 4%.

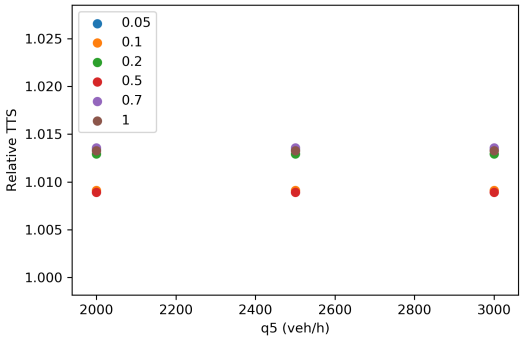
We can see that, once more, the flow of state 5 does not have any effect. The average



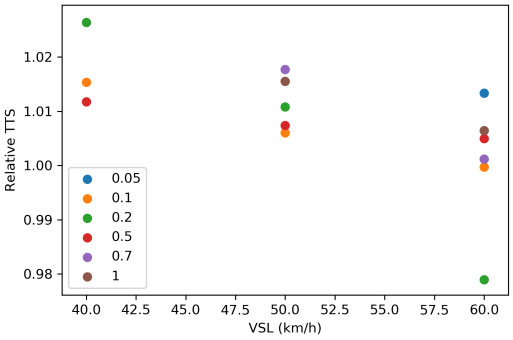
(a) Effect of the density of state 4 on the total time spent.



(b) Effect of the density of state 5 on the total time spent.



(c) Effect of the flow 5 on the total time spent.



(d) Effect of the variable speed limits on the total time spent.

Figure 4.13: Effects of the SPECIALIST parameters on results for the 2 lane network.

values achieved are between a 1.5-3% larger than the reference ones.

The last figure of the TTS, Figure 4.15d shows that all the values are greater than the reference value too. For all of the values of the VSL, the results are quite spread.

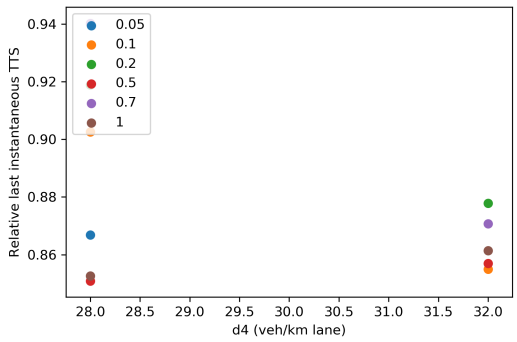
In spite of the these results, the figures of the instantaneous TTS are quite different. In many of them we achieve a result better than the reference one. Figure 4.16a shows a trend were the greater the density more less disperse are the measures and best is the average.

In Figure 4.16b we observe that for a density of 18 and 22 veh/km/lane and all the penetration rates the results obtained are better than the ones in the reference scenario.

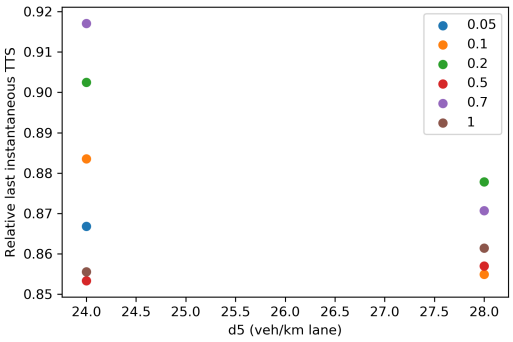
Not surprisingly after seeing the figures of the other networks, we observe in figure 4.16c that the flow of state 5 is irrelevant. The remarkable result is that a 20% penetration rates reduces the last instantaneous TTS more than a 10%.

The final figure shows that, for 40 and 50 km/h the results are worse than the reference scenario, up to a 5%. however, for 60 km/h almost all penetration rates show a reduction of more than 10%, except for a 50% penetration rate, which is surprising.

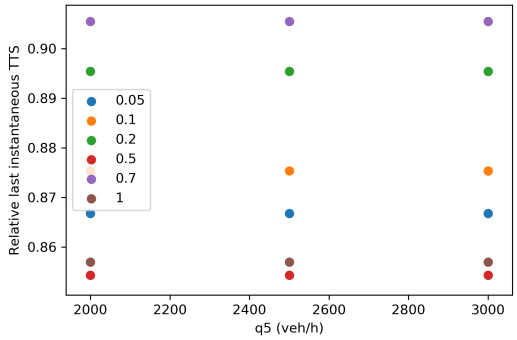
We observe that the algorithm works in these cases. The TTS does not change for most combinatios, but it does the last instantaneous, like in the 2 lanes network.



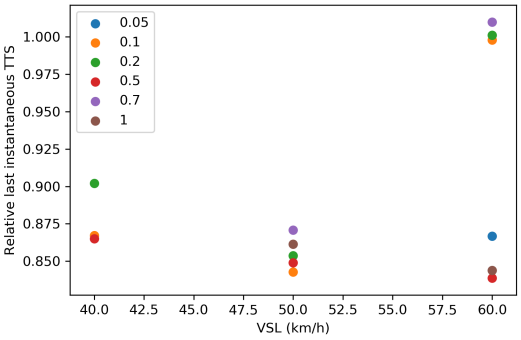
(a) Effect of the density of state 4 in the last instantaneous TTS.



(b) Effect of the density of state 5 in the last instantaneous TTS.



(c) Effect of the flow 5 in the last instantaneous TTS.



(d) Effect of the variable speed limits in the last instantaneous TTS.

Figure 4.14: Effects of the SPECIALIST parameters on the last instantaneous TTS for the 2 lane network.

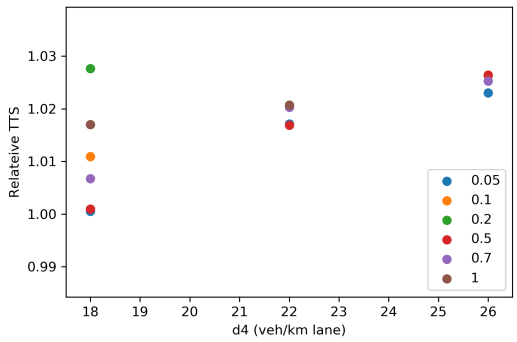
4.6.3 Conclusions

Overall, we observed that the mitigation algorithms did not work as well as expected. In particular, for the first scenario, although the shockwave is mitigated, the metrics that we measured are not affected. This indicates that the network did not improve at all. In contrast, with 2 or 3 lanes, we observe that there is a small improvement for some configurations in the TTS, and another in the last instantaneous TTS. These improvements indicate that we did actually improve the traffic in these scenarios. The reason why we think that the algorithm does not work in the 1 lane network is explained at the final conclusions in Chapter 5.

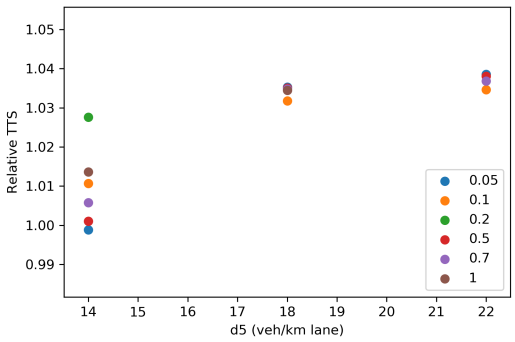
We observed that the parameters q_5 is absolutely irrelevant. Regarding the other parameters, we cannot extract a global conclusion because not clear pattern appears from the results that we have.

4.7 Testing in AP7 freeway

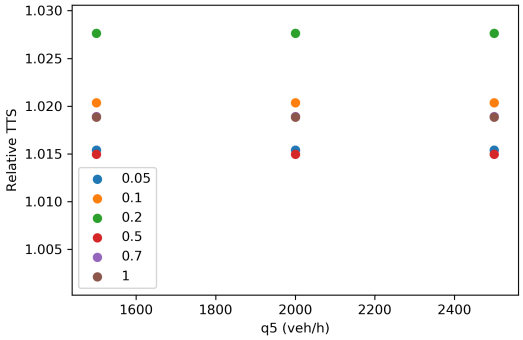
To conclude our experiments, we tested the mitigation with the best parameters obtained in the 3 lane scenario in a real section of the AP7 freeway. We only focused on the zone around the "ronda de Girona", which is the site of the catalan sub-pilot of C-ROADS Spain.



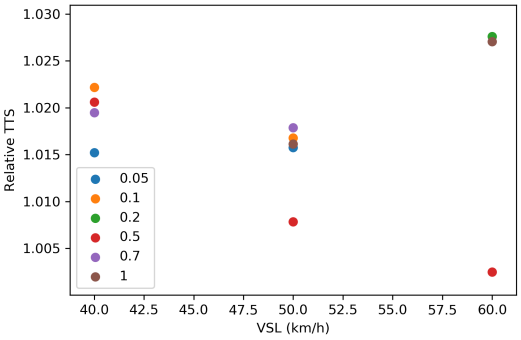
(a) Effect of the density of state 4 on the total time spent.



(b) Effect of the density of state 5 on the total time spent.



(c) Effect of the flow 5 on the total time spent.



(d) Effect of the variable speed limits in the total time spent.

Figure 4.15: Effects of the SPECIALIST parameters on the relative total time spent for the 3 lane network.

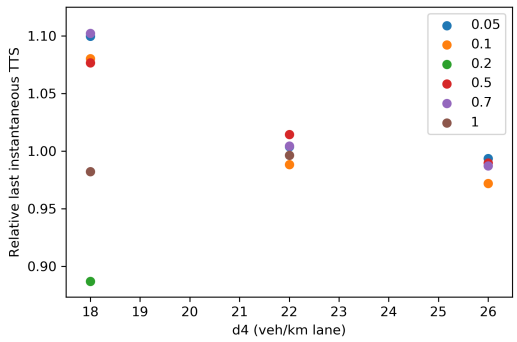
In Figure 4.17 we see area of the pilot. Specifically, we run the mitigation algorithm in the section between the entrances/exits 8 and 9, in the direction from La Roca to La Jonquera (from South to North in the figure). This region has an approximate length of 12 km and 3 lanes in all its extension.

We run the SPECIALIST algorithm for the same penetration rates that we tested it. For each one, we used the best parameters obtained in the previous section. Each simulation lasted 2 hours and the shockwave is generated at the minute 60.

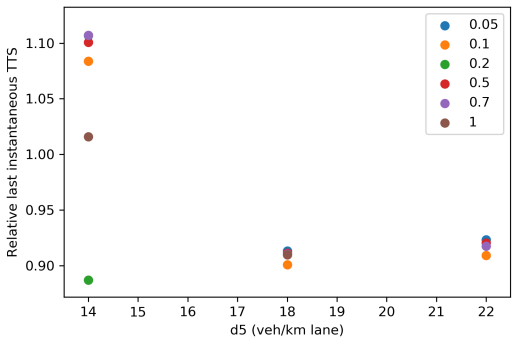
In this case, the shockwave is generated at the end of the studied region, just a 300 meters before the exit lane (exit 8), but stopping the car 20 seconds instead of 5 s as we did with on dummy models. To generate a shockwave we also had to saturate the network. A screenshot of the shockwave (when it is not mitigated) is shown in Figure 4.18.

The results obtained, regarding the TTS are shown in Figure 4.19. Figure 4.20 shows the results of the last instantaneous TTS. We observe that all penetration rates achieve a very similar performance. The TTS is reduced a 18%. The second figure shows that the last instantaneous TTS is reduced a 16.5%.

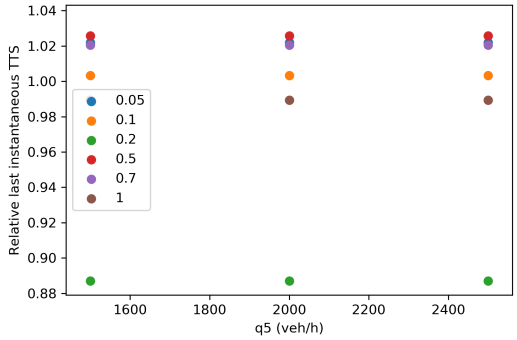
These are way better results than the ones observed in the previous section, with the dummy network with 3 lanes. One possible reason is that these simulations were too short



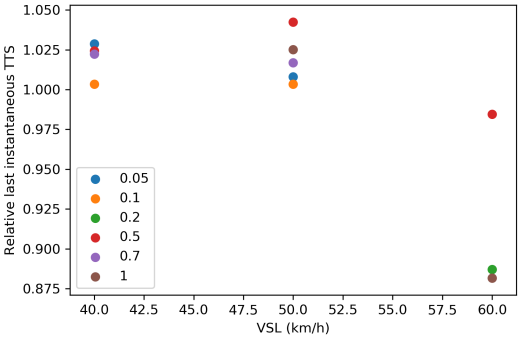
(a) Effect of the density of state 4 on the last instantaneous TTS.



(b) Effect of the density of state 5 on the last instantaneous TTS.



(c) Effect of the flow 5 on the total time spent.



(d) Effect of the variable speed limits on the last instantaneous TTS.

Figure 4.16: Effects of the SPECIALIST parameters on the last instantaneous TTS for the 3 lane network.

on that scenario to capture the full effect of the shockwave and so the benefits of mitigating it did not seem so good. In this scenario, the simulations are much longer because we simulate one hour after the shockwave. Thus, the reference scenario is very affected by the shockwave and this is why here we observe a great improvement on the metrics, particularly in the TTS. On the dummy scenario shockwave were also mitigated, so this is the most plausible explanation for this difference observed in the measures.

The fact that the reduction is approximately the same for all penetration rates is a consequence of the results shown in Section 4.4. There we could see that the different penetration rates allow to detect the shockwave with a very similar delay if the best parameters are chosen. Since the detection its almost the same, the mitigation is activated at almost the same time and thus the effect of the mitigation is very similar.

Overall, we observe that in this network we are able to mitigate the shockwave with all the tested penetration rates. In addition, this mitigation has a direct effect on the measures of the network, improving its performance with a very significant reduction of the total time spent. Thus, we can say that the algorithm has been simulated successfully in a real freeway.

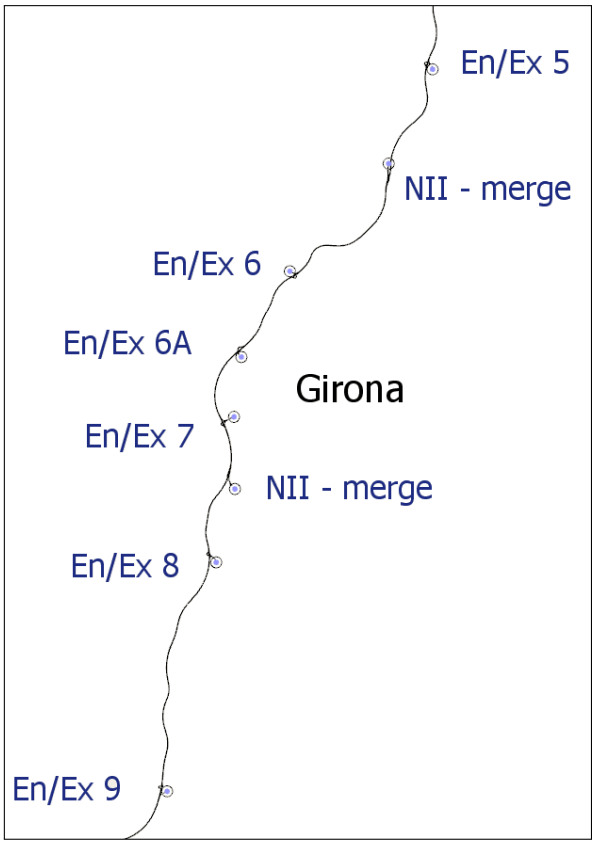


Figure 4.17: Model of the AP7 near to Girona.

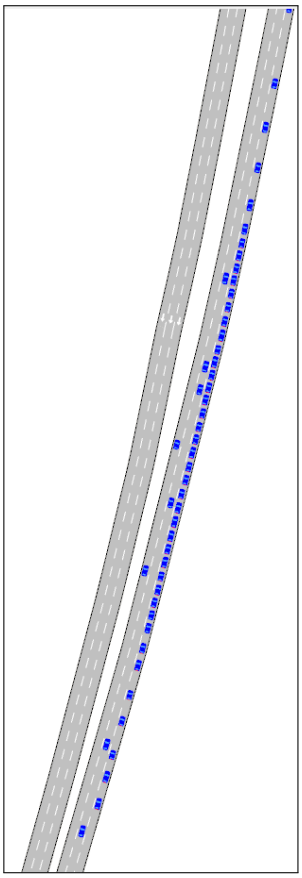


Figure 4.18: Screenshot of the shockwave generated on the AP7 in Aimsun.

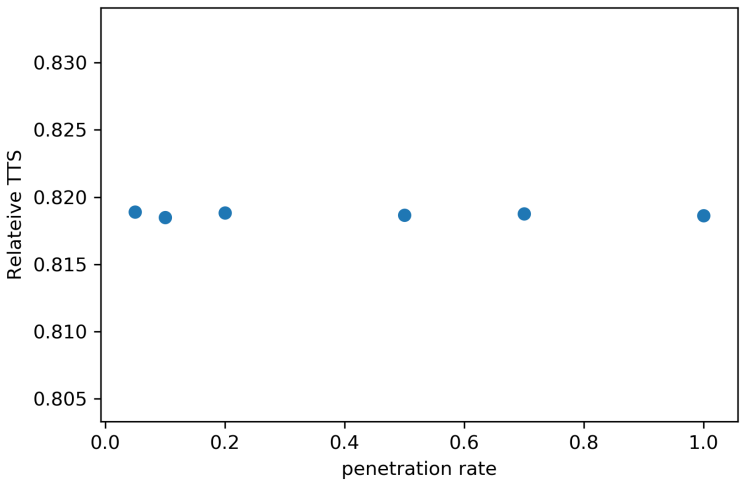


Figure 4.19: Relative TTS obtained in the AP7.

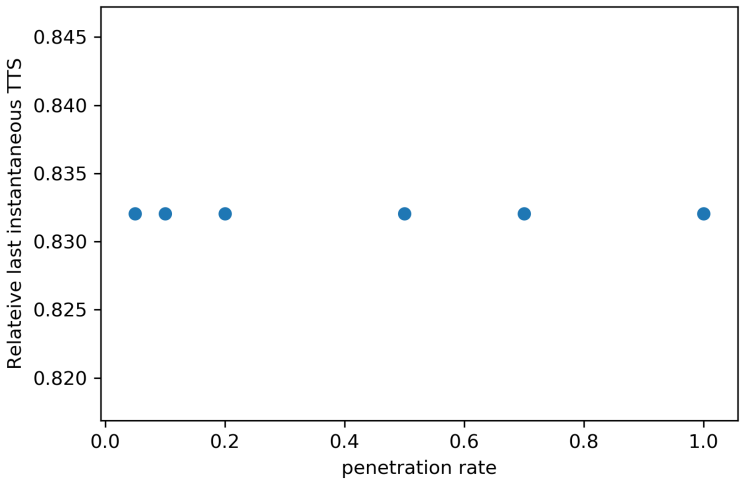


Figure 4.20: Relative last instantaneous TTS obtained in the AP7.

Chapter 5

Conclusions

In this chapter we present the main conclusions that we have obtained. We also explain the main contributions that we have done. Then, we propose some future work that could be done to continue the work developed in this master thesis.

The initial objectives of this master thesis were exposed in Section 1.3. We implemented two shockwave detection methods, ASM detection and Izadpanah's detection. We tested these methods in the dummy networks with different parameters to evaluate its effect. Then, the best method, which is ASM detection, was used to mitigate shockwaves with the SPECIALIST algorithm on the dummy networks. Finally, the best parameters of the SPECIALIST were used to mitigate shockwaves in the model of the AP7.

We have observed that ASM detection works very well, the detection takes approximately 10 s. These are similar results to the ones obtained by Hegyi et al. [2013]. This fast detection is very good and key to mitigate the shockwave.

On the other hand, Izadpanah's detection did not work that well. We were able to detect shockwaves, but the delay in the detection was much higher. Moreover, the time to compute this algorithm is much higher than the one that ASM detection requires.

With the first algorithm, we observed that the effect of the penetration rate is not clear at all. We expected that higher penetration rates would improve the detection, but this does not happen. The effect of the penetration rate is difficult to explain for this algorithm, because it works by computing the speeds at some discretizated cells. We could think that a higher penetration rate is better. However, maybe having less connected cars allows the values of the cells to detect the shockwaves faster. This can occur because if there are less connected vehicles very few need to be in the shockwave for it to be detected. If a huge portion of vehicles are connected cars, then we would require more vehicles in the shockwave to see a change in the speed of the cell.

Where we can clearly observe that a higher penetration rate improves the delay is with Izadpanah's detection. In this case, however, it is much easier to explain the effect of the penetration rate. If the penetration rate is higher, we have more trajectories of the vehicles, so we can detect more inflection points. Thus, in this case it is easier to explain the effect of the penetration rate and why it matches the observed behaviour.

Regarding the mitigation algorithm, we did not observe a considerable improvement on the dummy networks, particularly regarding the TTS. There was a greater improvement on the last instantaneous TTS, but the significant thing is that we observe that the shockwaves in the real freeway are mitigated. This indicates that probably we run too short simulation to see the full effect of the shockwave on the dummy networks. The improvement of the TTS

in the real network is a 18%, so this is a very good improvement. We also have to take into account that the travel time is not only the travel time of the studied zone, it is the total time since the start of the freeway. This makes this increase even more significant.

It is also remarkable that we did not observe any mitigation on the 1 lane network. Probably, this occurs because if there are several lanes, once the input flow on the shockwave is reduced, there are less cars on the other lanes and vehicles can escape the shockwave changing to the other lanes. This does not happen on the 1 lane network because the shockwave must be mitigated only by the effect of a reduced input flow. Since we are not resolving the shockwave in most cases in this scenario, this may indicate that the length of VSL computed is actually too short and should be increased to reduce more the flow on the shockwave.

5.1 Contributions

The first contribution of this master thesis is a state of the art on shockwave damping, where we explain the main methods used in the literature.

In this master thesis we have implemented one algorithm proposed in the literature (Izadpanah) and a simplified version of another one (ASM detection). We have combined them with the SPECIALIST mitigation algorithm. We have been able to implement an algorithm that works successfully in networks with several lanes and test it in a real network successfully.

We programmed a code that may be useful in the future to simulation applications of VSL and to collect and process PVD and fixed sensors data. This code has been developed for Aimsun but we tried to make it as generalizable to other simulators as possible.

All the work done is also part of the contribution of the inLab FIB to the C-Roads project.

5.2 Future work

There are many ways to extend this work. For instance, we could study these algorithms with longer simulations to analyze better in the long term how the networks are affected by the presence and posterior mitigation of a shockwave. It could also be interesting to measure how the mitigation algorithm affects the flow of the affected regions. In theory, if we reduce the TTS we sould observe an increase on the flow, but this would be worth studying.

A major metric that we could not analyze well is the false positives proportion. We only could check that our algorithms did not detect shockwave where we had not generated them. However, in a real scenario, vehicle may generate situations that do not appear in a simulation. These situations could trigger our algorithm even if they are not shockwaves. An analysis of these situations and their effects on the detection would be very interesting and necessary to deploy these algorithms in a real freeway. However, this experiment would require working with real data and not simulated one.

Another extension would be to consider different types of shockwaves and flows. We have worked with a fixed type of shockwave and a fixed flow for each network, but this could be extended to creating shockwaves with different sizes in scenarios with various flows.

Moreover, there are some relevant factors that we have considered fixed but that could actually modify the detection and the mitigation. These are the distance between fixed sensors, that fixed to 500 m, this could be a parameter of study. Another one is the distance between the variable message signs. Also related with the VMS, we could study the performance of the mitigation in an scenario where not all vehicles comply with the speed limit.

Also regarding how drivers receive the speed limit we could also study the effect of receiving in-car information instead of using VMS. Even it could be studied how these algorithms would perform if we had autonomous cars and we could set the speed of each car individually.

It is also interesting to study how the use of these algorithms could reduce the emissions of the network, that could be a new metric to consider.

Further research on why the penetration rate does not seem to behave as expected in ASM detection would be worth investigating.

In addition, the second detection algorithm, Izadpanah's detection, could also be speed up using parallel programming. This could not be done in this work due to limitations of the simulator used, but it could speed considerable this algorithm and it may get feasible computational times. This, with a new algorithm to detect inflection points may improve the idea of Izadpanah and the other authors and achieve better results.

Bibliography

J. Barceló. *Fundamentals of Traffic Simulation*. Springer, 2014. ISBN 978-1-4419-6142-6. doi: 10.1007/978-1-4419-6142-6.

M. Behl and R. Mangharam. Pacer Cars: Real-Time Traffic Shockwave Suppression (Work in Progress Session). *Real–Time Systems Symposium(RTSS)*, (June), 2010. URL `http://cse.unl.edu/{~}rtss2008/archive/rtss2010/WIP2010/11.pdf`.

A. Csikós, I. Varga, and K. M. Hangos. Freeway shockwave control using ramp metering and variable speed limits. *2013 21st Mediterranean Conference on Control and Automation, MED 2013 - Conference Proceedings*, pages 1569–1574, 2013. doi: 10.1109/MED.2013.6608931.

I. Gitman, J. Chen, E. Lei, and A. Dubrawski. Novel prediction techniques based on cluster-wise linear regression. *CoRR*, abs/1804.10742, 2018. URL `http://arxiv.org/abs/1804.10742`.

E.F. Grumert and A. Tapani. Traffic State Estimation Using Connected Vehicles and Stationary Detectors. *Journal of Advanced Transportation*, 2018, 2018. ISSN 20423195. doi: 10.1155/2018/4106086.

Y. Han, A. Hegyi, Y. Yuan, S. Hoogendoorn, M. Papageorgiou, and C.s Roncoli. Resolving freeway jam waves by discrete first-order model-based predictive control of variable speed limits. *Transportation Research Part C: Emerging Technologies*, 77:405–420, 2017. ISSN 0968090X. doi: 10.1016/j.trc.2017.02.009. URL `http://dx.doi.org/10.1016/j.trc.2017.02.009`.

A. Hegyi and S. P. Hoogendoorn. Dynamic speed limit control to resolve shock waves on freeways - Field test results of the SPECIALIST algorithm. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pages 519–524, 2010. ISSN 2153-0009. doi: 10.1109/ITSC.2010.5624974.

A. Hegyi, B. De Schutter, and J. Hellendoorn. Optimal coordination of variable speed limits to suppress shock waves. *IEEE Transactions on Intelligent Transportation Systems*, 6(1): 102–112, 2005. ISSN 15249050. doi: 10.1109/TITS.2004.842408.

A. Hegyi, M. Burger, B. De Schutter, J. Hellendoorn, and T.J.J. Van Den Boom. Towards a practical application of model predictive control to suppress shock waves on freeways. *2007 European Control Conference, ECC 2007*, 19, 2007.

A. Hegyi, S.P. Hoogendoorn, M. Schreuder, H. Stoelhorst, and F. Viti. SPECIALIST: A dynamic speed limit control algorithm based on shock wave theory. *2008 11th International*

IEEE Conference on Intelligent Transportation Systems, pages 827–832, 2008. doi: 10.1109/ITSC.2008.4732611.

A. Hegyi, B. D. Netten, M. Wang, W. Schakel, T. Schreiter, Y. Yuan, B. Van Arem, and T. Alkim. A cooperative system based variable speed limit control algorithm against jam waves - An extension of the SPECIALIST algorithm. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 2(Itsc):973–978, 2013. doi: 10.1109/ITSC.2013.6728358.

J. C. Herrera and A. M. Bayen. Incorporation of Lagrangian measurements in freeway traffic state estimation. *Transportation Research Part B: Methodological*, 44(4):460–481, 2010. ISSN 01912615. doi: 10.1016/j.trb.2009.10.005. URL `http://dx.doi.org/10.1016/j.trb.2009.10.005`.

B. K. P. Horn. Suppressing traffic flow instabilities. *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, (Itsc):13–20, 2013. doi: 10.1109/ITSC.2013.6728204.

P. Izadpanah, B. Hellinga, and L. Fu. Automatic Traffic Shockwave Identification Using Vehicles' Trajectories. *Transportation Research Board*, 2000(January 2009), 2009.

J. Li, C. Wang, S. He, and T. Z. Qiu. Dynamic traffic shockwave speed estimation in connected vehicle environment. *2017 4th International Conference on Transportation Information and Safety, ICTIS 2017 - Proceedings*, pages 54–59, 2017. doi: 10.1109/ICTIS.2017.8047742.

N. Motamedidehkordi, M. Margreiter, and T. Benz. Shockwave Suppression by Vehicle-to-Vehicle Communication. *Transportation Research Procedia*, 15:471–482, 2016. ISSN 23521465. doi: 10.1016/j.trpro.2016.06.040. URL `http://dx.doi.org/10.1016/j.trpro.2016.06.040`.

Observatorio del Transporte y la Logística en España. Informe anual 2017, 2017. URL `http://observatoriotransporte.fomento.es/NR/rdonlyres/EE4D9E3E-74A9-4C1F-A5FC-\284D30BBAFFA/148831/INFORMEOTLE2017.pdf`.

F. Rempe, L. Kessler, and K. Bogenberger. Fusing probe speed and flow data for robust short-term congestion front forecasts. *5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems, MT-ITS 2017 - Proceedings*, (1):31–36, 2017. doi: 10.1109/MTITS.2017.8005695.

I. Schelling, A. Hegyi, and S.P. Hoogendoorn. SPECIALIST-RM - Integrated variable speed limit control and ramp metering based on shock wave theory. *14th International IEEE Conference on Intellignet Transportation Systems*, 2011.

T. Seo, T. Kusakabe, and Y. Asakura. Estimation of flow and density using probe vehicles with spacing measurement equipment. *Transportation Research Part C: Emerging Technologies*, 53:134–150, 2015. ISSN 0968090X. doi: 10.1016/j.trc.2015.01.033. URL `http://dx.doi.org/10.1016/j.trc.2015.01.033`.

Spookfiles. First dutch cooperative vehicle-roadside system in place, 2016. URL `http://www.spookfiles.nl/sites/www.spookfiles.nl/files/documenten/shockwave_traffic_jams_a58_-_background_information.pdf`.

M. Treiber and D. Helbing. Reconstructing the Spatio-Temporal Traffic Dynamics from Stationary Detector Data. *Cooper@tive Tr@nsport@tion Dyn@mics*, 1(3):1–24, 2002.

M. Treiber and A. Kesting. *Traffic Flow Dynamics*. Springer, 2013. ISBN 978-3-642-44796-9.

J. W. C. Van Lint and S. P. Hoogendoorn. A Robust and Efficient Method for Fusing Heterogeneous Data from Traffic Sensors on Freeways. *Computer-Aided Civil and Infrastructure Engineering*, 25(8):596–612, 2010. ISSN 10939687. doi: 10.1111/j.1467-8667.2009.00617.x.

L. J. J. Wismans, L. C. W. Suijs, L. Krol, and E. C. van Berkum. In-Car Advice to Reduce Negative Effects of Phantom Traffic Jams. *Transportation Research Record: Journal of the Transportation Research Board*, 2489:1–10, 2015. ISSN 0361-1981. doi: 10.3141/2489-01. URL `http://trrjournalonline.trb.org/doi/10.3141/2489-01`.