



**KTH Electrical Engineering**

# **Fuel-Efficient Centralized Coordination of Truck Platooning**

SEBASTIAN VAN DE HOEF

Licentiate Thesis  
Stockholm, Sweden 2016



TRITA-EE 2016:074  
ISSN 1653-5146  
ISBN 978-91-7729-015-5

KTH Royal Institute of Technology  
School of Electrical Engineering  
Department of Automatic Control  
SE-100 44 Stockholm  
SWEDEN

Akademisk avhandling som med tillstånd av Kungliga Tekniska högskolan framlägges till offentlig granskning för avläggande av licentiatexamen i elektro- och systemteknik måndagen den 13 juni 2016 klockan 10.00 i E3, Kungliga Tekniska högskolan, Osquars backe 14, Stockholm, Sweden.

© Sebastian van de Hoef, June 2016. All rights reserved.

Tryck: Universitetservice US AB



## Abstract

The problem of how to coordinate a large fleet of trucks with given itinerary to enable fuel-efficient platooning is considered. Platooning is a promising technology that enables trucks to save significant amounts of fuel by driving close together and thus reducing air drag. A setting is considered in which a fleet of trucks is provided with transport assignments consisting of a start location, a destination, a departure time and an arrival deadline from a higher planning level. Fuel-efficient plans are computed by a centralized platoon coordinator. The plans consist of routes and speed profiles that allow trucks to reach their respective destinations by their arrival deadlines. Hereby, the trucks can meet on common parts of their routes and form platoons, resulting in a decreased fuel consumption.

First, routes are computed. Then, all pairs of trucks that can potentially platoon are identified. Potential platoon pairs are identified efficiently by extracting features from the routes and processing these features. In the next step, two types of plans are computed for each vehicle: default and adapted plans. An adapted plan is such that the vehicle can meet another vehicle en route and platoon. We formulate a combinatorial optimization problem that combines these plans in order to achieve low fuel consumption. An algorithm to compute optimal solutions to this problem is developed. The optimization problem is shown to be NP-hard, which motivates us to propose a heuristic algorithm that can handle realistically sized problem instances. The resulting plans are further optimized using convex optimization. The method is evaluated with Monte Carlo simulations in a realistic setting. We demonstrate that the proposed algorithm can compute plans for thousands of trucks and that significant fuel savings can be achieved.





# Acknowledgments

First of all, I would like to express my gratitude to my main supervisor Dimos Dimarogonas and to my co-supervisor Karl Henrik Johansson. Their continuous inspiration and feedback have made this thesis possible, and it has been an excellent experience to work with them.

I would like to thank the people that I have been working with in course of this thesis. In particular, I would like to mention my co-authors Panagiotis Tsiotras, Bart Besselink, Valerio Turri, Kuo-Yun Liang, Jonas Mårtensson, Assad Alam, and Håkan Terelius. Thanks to all participants in the COMPANION project for a very enjoyable collaboration. I am grateful to Matteo Vanin, Ari Hauksson, and David Spulak for their commitment in turning my ideas into Python code, and the smart mobility lab team for promoting transportation research. Thanks to the participants in the transport reading group and the formal methods reading group for great discussions and feedback.

I would like to express my gratitude to Pedro Lima, Antonio Adaldo, Pedro Pereira, Bart Besselink, Jana Tumova, and Miguel Galrinho for proofreading this thesis.

I am thankful for everybody working in the automatic control department for together creating such a enjoyable working environment. In particular, I would like to thank my past and present office neighbors for great company, Riccardo Risuleo for fun projects, Miguel Galrinho taking me to nice concerts, Martin Andreasson for exploring Chicago by bike, Demia Della Penda for making me “professor” ahead of time, Emma Tegling for organizing afterworks, Bart Besselink for making sure that I am not the only one with a sandwich for lunch, José Araújo for reflections on the life of a PhD student, Meng Guo for everlasting enthusiasm and practical advice, Dimitris Boskos for infinite patience explaining math, Kaveh Paridari for end-of-the-day chats, Pedro Lima for unlocking Gran Canaria’s cuisine, Jana Tumova for guidance in the world of computer science, Kuo-Yun Liang for inspecting every ancient stone in Athens, Pedro Pereira for never ending discussions, Marco Molinari for biking to work under any circumstances, Alexandros Nikou for happily using my assignment print script, Arda Aytakin for teaching about great software, and Antonio Adaldo for quality control of “Italian” pizza.



I would like to thank the administrators at the department for keeping this lab running.

The research in this thesis has been financed by the School of Electrical Engineering through the program of excellence, by the European Union thorough the COMPANION project, by the Swedish Research Council, and the Knut and Alice Wallenberg Foundation. I am grateful for the their support.

Last but not least I want to thank my family and friends for their support and company, in particular Anni for her dedication in proofreading my texts and both her and my parents for morally supporting me.

*Sebastian van de Hoef*  
Stockholm, June 2016



# Notation

Symbol	Meaning
$\mathcal{G}_r$	road network graph
$\mathcal{N}_r$	nodes of the road network graph
$\mathcal{E}_r$	edges of the road network graph
$L$	length associated with an edge in the road network graph
$e$	route of a vehicle
$N_e$	number of edges of a route
$t^S$	start time of an assignment
$t^D$	arrival time of an assignment
$t^A$	arrival time according to a vehicle plan
$t^M$	merge time
$t^{Sp}$	split time
$D$	length of a route
$v$	speed
$\mathbf{v}$	speed sequence of a vehicle plan
$N_v$	number of elements in a speed sequence
$\hat{\mathbf{t}}$	sequence of times instances the speed changes
$v_{\min}$	minimum speed
$v_{\max}$	maximum speed
$v_{cd}$	constant speed used in a default plan
$F_c$	fuel consumption of all plans combined
$F$	fuel consumption of a trajectory
$f$	fuel consumption per distance traveled
$f_0$	regular fuel consumption per distance traveled
$f_p$	fuel consumption per distance traveled as a platoon follower
$\mathcal{G}_c$	coordination graph
$\mathcal{N}_c$	set of assignments and nodes of the coordination graph
$\mathcal{E}_c$	edges of the coordination graph
$\Delta F$	edge weights of the coordination graph
$\mathcal{N}_l$	set of coordination leaders
$f_{ce}$	fuel savings as a function of the coordination leader set
$\mathcal{N}_n^i$	set of in-neighbors of a node $n$
$\mathcal{N}_n^o$	set of out-neighbors of a node $n$
$\Delta u$	change in $f_{ce}$ from adding/removing a coordination leader



# Contents

<b>Acknowledgments</b>	<b>v</b>
<b>Notation</b>	<b>vii</b>
<b>Contents</b>	<b>viii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Problem Formulation . . . . .	2
1.3 Thesis Outline and Contributions . . . . .	4
<b>2 Background</b>	<b>9</b>
2.1 Freight Transport . . . . .	9
2.2 Intelligent Transportation Systems . . . . .	10
2.3 Platooning . . . . .	11
2.4 Related Work . . . . .	12
2.5 Summary . . . . .	13
<b>3 Modeling</b>	<b>15</b>
3.1 Vehicle Plans for Coordination . . . . .	15
3.2 Platoon Coordinator . . . . .	19
3.3 Summary . . . . .	20
<b>4 Extracting Candidate Platoon Pairs</b>	<b>21</b>
4.1 Candidate Platoon Pairs . . . . .	22
4.2 Culling Candidate Platoon Pairs . . . . .	23
4.3 Features and Classifiers for Culling Platoon Pairs . . . . .	26
4.4 Simulations . . . . .	31
4.5 Summary . . . . .	35
<b>5 Pairwise Adapted Vehicle Plans</b>	<b>37</b>
5.1 The Optimal Rendezvous Speed . . . . .	37
5.2 Computing Adapted Vehicle Plans . . . . .	40





5.3	Summary . . . . .	44
<b>6</b>	<b>Computing Fuel-Efficient Vehicle Plans</b>	<b>45</b>
6.1	Combining Pairwise Plans to Save Fuel . . . . .	46
6.2	Exact Computation . . . . .	48
6.3	Heuristic Computation . . . . .	56
6.4	Joint Vehicle Plan Optimization . . . . .	57
6.5	Simulations . . . . .	60
6.6	Summary . . . . .	65
<b>7</b>	<b>Conclusions and Future Work</b>	<b>69</b>
7.1	Conclusions . . . . .	69
7.2	Future Work . . . . .	70
	<b>Bibliography</b>	<b>73</b>





# Chapter 1

## Introduction

THIS chapter introduces the thesis. In Section 1.1, we motivate why goods transportation is a topic worthwhile studying. Therein, we argue that road freight transport is in many aspects superior to other transport systems. However, road freight transport has a number of problems that need to be overcome in order to become sustainable. Truck platooning is a promising new technology that can help overcome some of these problems. In Section 1.2, we formulate the problem of forming truck platoons that this thesis attempts to solve. Section 1.3 gives an overview of this thesis. It summarizes, chapter by chapter, the contributions made. Furthermore, we indicate where presented material has been published.

### 1.1 Motivation

Goods transport is critical for economy, and transport volumes are tightly coupled with economic prosperity [68]. In the European Union, the entire transport sector accounts for 5% of the gross domestic product (GDP) [33]. Developments in transportation systems are key enablers for industrial development. Without improved ships and the invention of railways, the industrial revolution could not have taken place. More recently, increasing amounts of goods are moved by road freight transport [36]. This is due to a number of advantages road freight transport has compared to alternative means of transportation such as rail, water, or air. Trucks are very flexible. They can reach virtually every location that goods need to be transported to or from. The organizational overhead of trucks is low and many operators are small companies [35]. This enables quick adaptation to changing demands, and competition keeps prices low. Since a truckload is relatively small, it is often possible to transport goods directly from source to destination with little overhead for combining different transports in order to fill the vehicle [21]. Road freight transport has, on the other hand, a number of problems. Due to the decentralized nature of road infrastructure, traffic and thus transport times can vary and they are difficult to predict. Furthermore, every truck needs a driver, which



leads to high labor costs [81]. Driving a truck over long distances on highways can be at times a monotonous task. Nevertheless, the driver's full attention is required at all times, since even short moments during which a driver is not attentive can lead to fatal accidents [7]. Another problem is that the great majority of trucks is powered by fossil fuels, and despite various research efforts such as electric highways and alternative fuels, this is not likely to change soon, in particular in the domain of long haulage transport [34, 74]. Fuel accounts for roughly a third of a heavy truck's operation costs in long haulage transport [81]. The use of fossil fuels leads to problematic emissions, most prominently carbon-dioxide [74]. In 2014, the transport sector accounted for 20 % of greenhouse gas emissions in the European Union, of which 72 % were due to road transport.

Truck platooning is a technology that can help solve some of these problems. It refers to a group of vehicles forming a road train without any physical coupling between the vehicles. A short inter-vehicle distance is maintained by automatic control and vehicle-to-vehicle communication. Figure 1.1 shows two demonstrations of vehicle platooning. The small inter-vehicle spacing leads to an improved road throughput and the automatic control of the trailing vehicles improves safety. Similar to what racing cyclists exploit, the follower vehicles and, to a lesser degree, the lead vehicles experience a reduction in air drag, which translates into reduced fuel consumption [5, 6, 20, 57, 87, 99, 100]. Reduced fuel consumption, in turn, implies decreased operation costs and emissions. Advances in wireless communication, satellite based positioning, available computing power, and driver support systems in general have made the deployment of platooning systems feasible and platooning has attracted the attention of major truck manufacturers. It is likely that such systems will be commercially available in near future [52, 2].

## 1.2 Problem Formulation

Integrating platooning into the road freight transport system leads to a challenging coordination problem. While there have been promising demonstrations of intra-platoon control systems [43, 48, 56, 87, 89, 2], the question remains open where and when platoons should be formed. In some special cases, trucks have the entire or first part of their journey in common, for instance, when leaving from a distribution center. However, such special cases account only for a small fraction of road freight transport.

Consider Figure 1.2. Two trucks that travel between the same two regions but from different locations within the regions and at approximately at the same time. These trucks can adjust their speeds slightly at the beginning of their journeys, form a platoon at the start of the common part of their routes and thus save fuel during most of their journeys.

This example motivates the need for a coordination scheme that helps trucks form platoons (Figure 1.3). Such a system should retain the advantages trucks have over other transportation systems, such as flexibility and independence, while





Figure 1.1: Platooning demonstrations in the scope of the European truck platooning challenge 2016. (Image source: European Truck Platooning Challenge)

maximizing the gains from platooning, specifically reduced fuel consumption. In particular the reduction in fuel consumption can be jeopardized if vehicles drive at increased velocities in order to catch up to their assigned platoon partners [62]. This thesis investigates how vehicles with different start-destination-pairs can be coordinated to form platoons in a fuel efficient way.



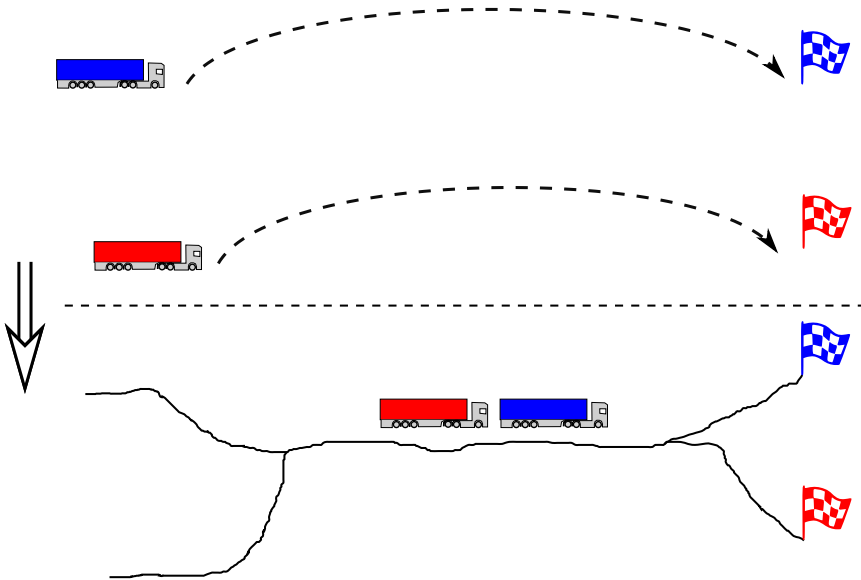


Figure 1.2: Two trucks with similar start-destination-pairs can meet en route and form a platoon.

### 1.3 Thesis Outline and Contributions

This section provides an overview of the thesis. It describes each chapter’s content and contribution. We also indicate publications in which material used in this thesis has been or is going to be published.

**Chapter 2: Background** Chapter 2 provides background of the thesis. Truck platooning is ultimately a measure to make road freight transport more efficient. A system for coordinating truck platooning will not be of practical use if the context in which it is used is not considered. Section 2.1 gives a brief overview over research on freight transport systems. The section illustrates the scale and development of today’s transport systems. We motivate why continuous improvement of transport systems is crucial for the economy.

Truck platooning is a relatively new technology that has been made possible by modern information and communications technology. Information and communications technology has had an impact on transportation systems in various ways. These developments are often described with the term “intelligent transportation systems”, of which an overview is given in Section 2.2.

The topic of this thesis is the coordination of truck platooning and it takes the ability of trucks to form platoons for granted. Section 2.3 summarizes work on the



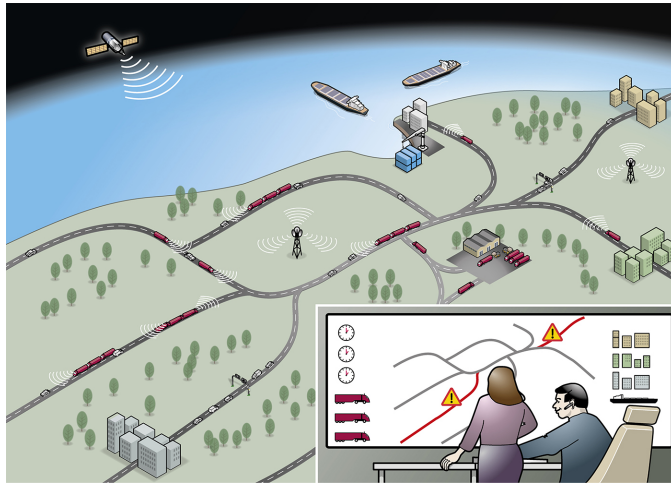


Figure 1.3: Illustration of an integrated platoon coordination system. Trucks communicate via vehicle to infrastructure communication with the centralized platoon coordinator that coordinates the dynamic formation of platoons and integrates with logistic operators.

control of truck platooning and its effect on fuel consumption.

Section 2.4 reviews related work on the coordination of truck platooning as well as work from other areas that play a role in some of the results presented in this thesis.

**Chapter 3: Modeling** In Chapter 3, we model the problem whose solution is investigated in the remainder of the thesis.

In Section 3.1, we introduce the notion of assignments and vehicle plans that fulfill these assignments. The vehicle plan, consisting of a route and speed profile, is the central data structure in this thesis. All remaining chapters deal with computing vehicle plans.

Section 3.2 introduces the centralized platoon coordinator. The centralized platoon coordinator communicates with the vehicles through vehicle-to-infrastructure communication. It receives assignment data from the vehicles and computes fuel-efficient vehicle plans, which are sent to the vehicles and executed.

The chapter is based on the publication:

- S. van de Hoef, K. H. Johansson, and D. V. Dimarogonas. Fuel-efficient en route formation of truck platoons. *IEEE Transactions on Intelligent Transportation Systems*, 2016. Submitted for publication



**Chapter 4: Extracting Candidate Platoon Pairs** Chapter 4 considers the computationally efficient identification of all vehicle pairs that can potentially platoon. To identify the platoon opportunities for the set of transport assignments is the first step in the computation of vehicle plans. In Chapters 5 and 6, we discuss how to compute and select pairwise vehicle plans.

In order to tell if two vehicles can platoon, the routes have to be compared. If the routes overlap, we compute whether the vehicles can meet on the common segment of their routes according to their time and speed constraints. Making these computations for each pair of vehicles is computationally expensive.

We propose a more efficient approach based on extracting low dimensional features from the vehicle assignments. These features can be used to efficiently dismiss a majority of the pairs that cannot platoon. The remaining pairs can then be processed using a computationally more expensive algorithm that compares the routes explicitly.

The chapter is based on the publication:

- S. van de Hoef, K. H. Johansson, and D. V. Dimarogonas. Computing feasible vehicle platooning opportunities for transport assignments. In *14-th IFAC Symposium on Control in Transportation Systems*, 2016. To be published

**Chapter 5: Pairwise Adapted Vehicle Plans** Chapter 5 considers the computation of a speed profile for one vehicle. Hereby, the speed at the beginning of each journey is adjusted in a way so that the vehicle meets another vehicle en route. The two vehicles platoon for some distance until they split up. The speed at the end of the journey is adjusted that the vehicle meets its arrival deadline. Fuel is saved during the platooning phase. We derive how to compute such a plan in a fuel-optimal way taking into account that fuel consumption depends on speed and whether the vehicle platoons or not.

The chapter is based on the publication:

- S. van de Hoef, K. H. Johansson, and D. V. Dimarogonas. Coordinating truck platooning by clustering pairwise fuel-optimal plans. In *18th IEEE International Conference on Intelligent Transportation Systems*, pages 408–415, 2015

**Chapter 6: Computing Fuel-Efficient Vehicle Plans** In chapter 6 we consider how to compute vehicle plans that are fuel-efficient for a fleet of vehicles as a whole. This is done by first computing a default plan and a number of adapted plans similar to the ones derived in Chapter 5 for each vehicle. Then, a subset of these plans is selected and combined in a fuel-efficient way.

In Section 6.1, we formulate the problem of selecting and combining vehicle plans in a fuel-efficient way as a combinatorial optimization problem. In Section 6.2, a branch-and-bound algorithm is developed that computes exact solutions to the combinatorial optimization problem. We also prove that the problem is NP-hard,





which means that it might take a lot of computational effort to compute an optimal solution.

The result that the optimization problem is NP-hard motivates the development of a heuristic algorithm in Section 6.3. The algorithm is similar to heuristic algorithms used in clustering and community detection. It improves the fuel-efficiency of the combined plans in every step until it reaches a local maximum.

Section 6.4 elaborates on how the selected default and adapted plans can be jointly improved using convex optimization. Hereby, the pairs of vehicles that platoon remain fixed but the speed profile that leads to this platooning is optimized considering all vehicles at once.

Section 6.5 provides a realistic simulation scenario. Using Monte Carlo simulations, we motivate that the developed methods can save significant amounts of fuel. It also shows that our method can handle realistic fleet sizes without running into computational problems.

The chapter is based on the following two publications:

- S. van de Hoef, K. H. Johansson, and D. V. Dimarogonas. Fuel-efficient en route formation of truck platoons. *IEEE Transactions on Intelligent Transportation Systems*, 2016. Submitted for publication
- S. van de Hoef, K. H. Johansson, and D. V. Dimarogonas. Fuel-optimal coordination of truck platooning based on shortest paths. In *American Control Conference*, pages 3740–3745, Chicago, IL, 2015

**Chapter 7: Conclusions and Future Work** In Chapter 7 we conclude the thesis and discuss future work. Section 7.1 summarizes and discusses the obtained results. Section 7.2 outlines possible ways to continue this work.

**Additional Publications** The following three related publications are not explicitly covered in this thesis:

- B. Besselink, V. Turri, S. van de Hoef, K.-Y. Liang, A. Alam, J. Mårtensson, and K. H. Johansson. Cyber-physical control of road freight transport. *Proceedings of the IEEE*, 104(5):1128–1141, 2016
- K.-Y. Liang, S. van de Hoef, H. Terelius, V. Turri, B. Besselink, J. Mårtensson, and K. H. Johansson. Networked control challenges in collaborative road freight transport. *European Journal of Control*, 2016. To be published
- S. van de Hoef, D. V. Dimarogonas, and P. Tsiotras. Spectral analysis of extended consensus algorithms for multiagent systems. In *53rd IEEE Conference on Decision and Control*, pages 2204–2209, Dec. 2014





## Chapter 2

# Background

THIS chapter provides background on the thesis. Since trucks are an integral part of the freight transport system, we begin by discussing work on planning and on related challenges in freight transportation in Section 2.1. Transportation systems have been heavily influenced by information and communications technology resulting in the emergence of the field of intelligent transportation systems. Platooning and its coordination is a technology enabled by modern information and communications technology. Section 2.2 gives a brief overview of developments in the field of intelligent transportation systems. Before there is a need to coordinate platooning, the technology to make platooning a reality has to be developed. A significant amount of work has been dedicated to that topic. Section 2.3 gives an overview of work on platooning. Finally, Section 2.4 lists some of the existing work on platoon coordination. Furthermore, it provides references to some of the methods not directly related to platooning that are used in this thesis.

### 2.1 Freight Transport

Driven by its vital role in the economy there is a large body of research on transportation systems. The work on logistics can be divided into three levels of planning: strategic level, tactical level, and operational level [29, 66]. On the strategic level, long term decisions are made. The construction of infrastructure such as depots and harbors fall in this category. In addition, the types of service offered by a transport operator are decided on this level. For example, the transportation of iron ore from a mine to a furnace poses different challenges than the just-in-time delivery of car components. On the tactical level decisions are made about how the transport should take place. Here, the transport mode is decided. On the operational level, the actual schedule of transports is decided. Assignment of drivers falls in this category, too, which is a challenging task due to the uncertainties in transport times and strict hours of service regulations [67, 37].

Different transport systems—mainly road, air, ship, and railway—have their



own domain specific challenges. For air traffic, airborne waiting times are very expensive and should be minimized while high safety standards have to be guaranteed [15]. In marine traffic, many of the challenges are related to the operations in ports [83, 17]. In railway systems, infrastructure disruptions are difficult to handle due to the infrastructure's high complexity [53]. Most of these domains have been significantly affected by information and communications technology. There is an increasing interest to integrate different transport systems and handle their mutual effects [21, 61, 66].

The total transport volumes are steadily increasing, the reasons being expanding trade, global economic development, globalization, and the ability to handle complicated supply chains. The costs for transportation are significant, and so is the environmental impact. If this growth should continue or even accelerate in a sustainable way, massive improvements in the efficiency of transport systems have to be implemented. This concerns handling of the sheer complexity of the future transport system, more efficient use of resources, transition to alternative energy sources, and increased automation [36, 74, 34].

## 2.2 Intelligent Transportation Systems

The development of information and communication technology has major impacts on transportation systems [11, 61, 15, 79]. A large body of work focuses on improving efficiency of road infrastructure by means of variable speed limits and traffic signals [76, 12]. On arterial networks it is possible to hold back traffic at certain points to keep the traffic flow at the point of maximum efficiency. In urban scenarios the timing of traffic lights can be adapted in an intelligent way to reduce journey times. In addition, quick response to incidents and adaptive routing of traffic are possible ways to improve the road transport system. While these schemes rely mostly on dedicated road-side infrastructure, the vehicles themselves can also play an increasingly active role. The widespread availability of global positioning system (GPS) receivers and mobile internet allow to collect data on the traffic situation directly from the vehicles without any dedicated sensors [47]. Live traffic feeds to navigation devices are a first step towards directly influencing vehicles without the need for variable signs, traffic lights, etc. Using vehicle-to-infrastructure (V2X) and vehicle-to-vehicle (V2V) communication, efficiency and safety of road transport is expected to improve further [46, 98].

On the level of individual vehicles, information and communications technology has been as influential as well. The computation of routes is widely spread technology and in combination with satellite based positioning helps avoid time and fuel consuming detours. The basis for the computation of short routes is Dijkstra's algorithm for computing shortest paths in graphs. The A-star algorithm includes additional heuristics in order to speed up the search. Since road networks are very large, one can further pre-process the close to static road-network data and reduce computation times for long routes by some orders of magnitude [3, 80, 13, 72].



Routes can be optimized for different criteria such as distance, journey time, or fuel consumption [22]. Furthermore, there are problem settings in which vehicles have to visit multiple locations in an efficient way [4]. A classical problem of this type is the traveling salesman problem where an optimal route connecting a given number of location is to be computed. This can be extended to the vehicle routing problem where several vehicles are considered [86]. Dynamic formulations consider that the locations being visited are not known a priori but that they are revealed as the system runs [24]. In addition, it is possible to consider additional stochastic elements in the problem [41, 40].

On vehicle level, huge progress has already been achieved in the development of advanced driver assistance systems and (semi-)autonomous vehicles [18, 75]. Technologies such as adaptive cruise control (ACC), cooperative adaptive cruise control (CACC) [71], and lane keeping assistance are direct enablers for platooning [42]. Despite impressive demonstrations of autonomous driving, it is likely that in near future commercially available systems will only allow the driver to hand over control in particular driving scenarios. Driving in a platoon on a highway is one such scenario [96].

## 2.3 Platooning

The term “platooning” is used to describe vehicles driving behind each other with the gap between adjacent vehicles being controlled. In its simplest form, this occurs naturally on busy roads. Modern sensor and wireless communication technology makes it possible to automatically control the inter-vehicle gaps. Such automatic control has a number of advantages over manual control by human drivers. Adaptive cruise control (ACC), which is currently being introduced to the automotive market [38], is both a convenience and a safety feature. It relieves the driver from the potentially boring task of controlling the distance to the vehicle in front. Furthermore, rear collision accidents due to insufficient gaps and inattentive drivers account for a significant number of accidents [7]. Adaptive cruise control can help to avoid such accidents. Automatic control of the inter-vehicle gaps, in particular cooperative adaptive cruise control (CACC) where vehicles communicate actively, makes it possible to reduce the inter-vehicle gaps compared to human controlled gaps without compromising safety [49, 88, 71]. This results in two other desirable effects of platooning. By reducing the inter-vehicle gaps, more vehicles can fit on the road, leading to more efficient use of the infrastructure. The small inter-vehicle gaps lead to a slipstream effect (Figure 2.1), which reduces the air drag experienced by the trailing vehicles [5, 6, 20, 57, 87, 99, 100]. Reduced air drag, in turn, leads to reduced fuel consumption. This effect is frequently exploited in bicycle races. Experiments motivate that the air drag of a heavy truck in a platoon can be lowered by 40 %, translating into an overall reduction in fuel consumption of over 10 %. The potential of platooning to reduce fuel consumption has recently been one of the main motivations to develop platooning systems for heavy trucks.



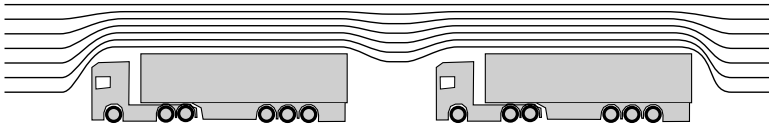


Figure 2.1: Platooning leads to a slipstream effect that can reduce the fuel consumption of the follower truck.

The automatic control of inter-vehicle gaps is a non-trivial problem that has attracted significant research interest [25, 8, 70, 43, 44, 49, 56, 60, 73, 77, 84, 85, 88, 89, 97]. Apart from the stringent requirements of safety, i.e., that vehicles do not collide under any circumstances, it is required that disturbances in one part of the platoon do not get amplified as they travel from vehicle to vehicle through the platoon. The phenomenon of spontaneously occurring traffic jams in heavy traffic is an example of a small disturbance being amplified as it travels upstream through the string of vehicles. To formalize this phenomenon, the notion of string stability has been introduced. Roughly speaking, a system is string stable when a disturbance on one subsystem is attenuated as it affects the next subsystem. When wireless communication is used to transmit control information between the vehicles, interference can cause information to be lost. The controller needs to handle such loss of information, for instance, by relying data from other sensors or increasing the gap between the vehicles. Surrounding traffic needs to be taken into account. For instance, other vehicles still have to be able to enter and exit the highway. Long platoons need to detect such vehicles and open gaps for the other vehicles when needed. When platooning is used as a measure to reduce fuel consumption, it is important that the control of the inter-vehicle gaps is performed in a way that the reduced air drag actually translates into reduced fuel consumption. If the vehicles brake and accelerate a lot in order to keep the gap at the desired value, they might consume more fuel compared to not platooning. In particular in hilly terrain a sophisticated fuel-efficient control strategy is crucial [90].

## 2.4 Related Work

Since platooning systems for reduced fuel consumption are not yet commercially available, the research on the formation of platoons is still in its infancy. Variations of platoon coordination have been considered in literature. In [59], the authors formulate a mixed integer linear programming problem without considering the speed dependency of fuel consumption, and prove that the problem is NP-hard. In [62] the authors consider a simple catch-up coordination scheme and evaluate it on real fleet data. In [58], local controllers for coordinating the formation of platoons are proposed. In [69], the authors use data-mining to identify economic platoons based on different criteria. Unlike this thesis, the method presented in [69] allows



that trucks wait for other trucks to form the platoon.

Various results from areas not directly related to platooning have been used in this thesis. The method to extract candidate platoon pairs discussed in Chapter 4 is inspired by a related problem in computer graphics, in particular finding the intersection of geometric objects. When the number of objects is large, instead of checking all possible pairs for intersection, it is more efficient to identify a smaller set of candidate pairs. The pairs in this set are then processed individually [14, 31, 54, 27, 64].

The proposed method to decide which trucks should platoon presented in Chapter 6.3 is inspired by clustering algorithms. Clustering is a widely used tool for analysis of large data sets. Data is structured into a finite number of sets. Elements within the set are in some way related. K-means clustering is a widely used technique in machine learning. An algorithm related to K-means clustering is called K-medoids clustering [50, 51, 55], to which the proposed algorithm in Chapter 6.3 is similar. Clustering of graphs has been investigated in the area of community detection. Community detection considers the problem of clustering a graph into densely connected groups of nodes [45, 39, 19].

This thesis explicitly considers that fuel consumption depends on vehicle speed and that platooning affects the fuel consumption. The development of an accurate fuel consumption model is non-trivial, as fuel consumption depends on a large number of factors such as road, weather, vehicle, driver, speed, load, traffic, etc. and various attempts have been made [30, 22]. In this work, we are mostly concerned with how speed and the role in a platoon affect fuel consumption. The effect of platooning on fuel consumption has been investigated both in simulations and in experiments with real vehicles. The reduction in fuel consumption is non-trivial to identify from measurements since it depends on the gap between the vehicles, the performance of the controller, the shape of the vehicles, the environmental conditions, etc. Nevertheless, research on the topic consistently shows a significant reduction of fuel consumption [5, 6, 20, 57, 87, 99, 100].

## 2.5 Summary

This chapter provides the background of the thesis. Freight transport is the backbone of industry and thus tightly coupled to economic prosperity. Therefore, transport has attracted the attention of many researchers and huge progress has been made on how to organize transport efficiently. Challenges arise in different modes of transport as well as on different levels. Increasing transport volumes and the environmental impact of transport poses new challenges.

The development of information and communications technology has had considerable impact on transport systems resulting in the research field of intelligent transportation systems. Available data enables smart decisions on how to route individual vehicles as well as entire transport streams, which leads to increased effi-



ciency of the transport system. Progress in sensors and automatic control improve safety and comfort on vehicle level.

One emerging technology on vehicle level is platooning, where vehicles form a road train by using automatic control of the inter-vehicle distances. Platooning has the potential to reduce fuel consumption, improve efficiency of road usage, improve safety, and help automate driving. While control of platoons is currently being developed as a commercial product for trucks, only few contributions on when, where, and how platoons should be formed are available in the literature. This thesis builds on these contributions and combines methods from computer graphics, clustering, and community detection to efficiently coordinate the en route formation of truck platoons.





# Chapter 3

## Modeling

IN this section, we model the problem of coordinating truck platooning in a fuel efficient way and we introduce notation that is used throughout this thesis. Section 3.1 introduces vehicle plans consisting of a route and a speed profile. Vehicle plans encode the result of the platoon coordination algorithms. We also relate fuel consumption and vehicle plans. The remainder of this thesis discusses how to compute fuel-efficient vehicle plans making use of platooning. Section 3.2 introduces a technical system that we call platoon coordinator. The platoon coordinator is a centralized entity that coordinates the dynamic en route formation of truck platoons. It receives assignment information and position data from the vehicles and computes vehicle plans that are sent back to them.

### 3.1 Vehicle Plans for Coordination

We have an index set  $\mathcal{N}_c$  of finitely many transport assignments, each tied to a specific truck. A transport assignment  $\mathcal{A} = (\mathcal{P}^S, \mathcal{P}^D, t^S, t^D)$  consists of a start position  $\mathcal{P}^S$ , a destination  $\mathcal{P}^D$ , a start time  $t^S$ , and an arrival deadline  $t^D$ . We model the road network as a directed graph  $\mathcal{G}_r = (\mathcal{N}_r, \mathcal{E}_r)$  with nodes  $\mathcal{N}_r$  and edges  $\mathcal{E}_r$ . Nodes correspond to intersections or endpoints in the road network and edges correspond to road segments connecting these intersections. The function  $L : \mathcal{E}_r \rightarrow \mathbb{R}^+$  maps each edge in  $\mathcal{E}_r$  to the length of the corresponding road segment. A vehicle position is a pair  $(e, x) \in \mathcal{E}_r \times [0, L(e)]$  where  $e$  indicates the current road segment and  $x$  how far the vehicle has traveled along that segment.

The goal is to compute fuel-efficient plans for the trucks that ensure arrival before each trucks' individual deadline. Each plan includes a route in the road network from start to destination and encodes a piecewise constant speed trajectory. The speed is constrained to a range of feasible speeds  $[v_{\min}, v_{\max}]$ , which is supposed to be the same for all vehicles and road segments.<sup>1</sup> For the sake of this high-level planning, it is reasonable to assume that trucks change their speed instantaneously.

---

<sup>1</sup>The approach developed in this thesis can be generalized in order to relax this assumption.



**Definition 1** (Vehicle Plan). A vehicle plan  $\mathcal{P} = (\mathbf{e}, \mathbf{v}, \hat{\mathbf{t}})$  consists of a route  $\mathbf{e}$ , a speed sequence  $\mathbf{v}$ , and a time sequence  $\hat{\mathbf{t}}$ . The route is a sequence of  $N_e$  edges in the road network  $\mathbf{e} = (\mathbf{e}[1], \dots, \mathbf{e}[N_e])$ ,  $\mathbf{e}[i] \in \mathcal{E}_r$ . The speed sequence is a sequence of  $N_v$  speeds  $\mathbf{v} = (\mathbf{v}[1], \dots, \mathbf{v}[N_v])$ , where speeds are within the feasible speed range  $0 < v_{\min} \leq \mathbf{v}[i] \leq v_{\max}$ . The time sequence  $\hat{\mathbf{t}} = (\hat{\mathbf{t}}[1], \dots, \hat{\mathbf{t}}[N_v + 1])$  defines when the speed changes. Speed  $\mathbf{v}[i]$  is selected from  $\hat{\mathbf{t}}[i]$  until  $\hat{\mathbf{t}}[i + 1]$ .

Note that  $N_e$  and  $N_v$  may be different for different vehicle plans.

We want to compute a vehicle plan for each truck. A valid vehicle plan brings the truck from its start position  $\mathcal{P}^S$ , where it is at time  $t^S$ , to its destination  $\mathcal{P}^D$  before its deadline  $t^D$ .

Vehicle plans are constrained by two conditions. The first condition requires the trip to start at the start time  $\hat{\mathbf{t}}[1] = t^S$  and ends before the deadline  $\hat{\mathbf{t}}[N_v + 1] = t^A \leq t^D$ . The second condition ensures that the truck arrives at its destination when the trip ends, i.e., the distance traveled is

$$D := \sum_{i=1}^{N_e-1} L(\mathbf{e}[i]) + x^D - x^S = \sum_{i=1}^{N_v} \mathbf{v}[i](\hat{\mathbf{t}}[i + 1] - \hat{\mathbf{t}}[i]).$$

A vehicle trajectory consists of an edge trajectory  $\epsilon$  and a linear position trajectory  $\xi$ . The edge trajectory for  $t \in [t^S, t^A]$  is given by  $\epsilon(t) = \mathbf{e}[j]$  where  $j$  depends on  $t$  and is the largest integer that satisfies

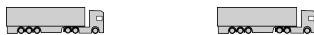
$$\sum_{i=1}^{j-1} L(\mathbf{e}[i]) - x^S < \int_{t^S}^t \phi(\tau) d\tau,$$

and where the speed trajectory  $\phi(t) = \mathbf{v}[i]$  for  $t \in [\hat{\mathbf{t}}[i], \hat{\mathbf{t}}[i + 1])$ ,  $i \in \{1, \dots, N_v\}$ . The linear position, i.e., the second element of the position, at time  $t$  is given by

$$\xi(t) = \int_{t^S}^t \phi(\tau) d\tau - \sum_{i=1}^{j-1} L(\mathbf{e}[i]) + x^S.$$

When trucks platoon, their positions coincide in our model. Each platoon consists of a platoon leader and a number of platoon followers. We introduce the platoon trajectory  $\pi_n : [t_n^S, t_n^A] \rightarrow \{0, 1\}$  for truck  $n \in \mathcal{N}_c$ . A platoon trajectory equals 1 when truck  $n$  is a platoon follower and 0 when it is a platoon leader or traveling alone. Thus,  $\pi_n(t) = 1$  implies that there is another truck  $m \in \mathcal{N}_c$  with  $m \neq n$  and  $(\epsilon_n(t), \xi_n(t)) = (\epsilon_m(t), \xi_m(t))$  and hence we neglect the physical dimension of the trucks.

Figure 3.1 illustrates the relation between vehicle plans and assignments. The route connects the assignment's start position  $\mathcal{P}^S$  and destination  $\mathcal{P}^D$ . The combination of the speed sequence  $\mathbf{v}$  and the time sequence  $\hat{\mathbf{t}}$  induces a position trajectory. When parts of these trajectories overlap, the vehicles can platoon.



We model the fuel consumption per distance traveled as a function of the speed and of whether the truck is a platoon follower or not. A platoon leader has the same fuel consumption as a truck that travels alone while a platoon follower has a reduced fuel consumption. We denote the fuel consumption per distance traveled as  $f : [v_{\min}, v_{\max}] \times \{0, 1\} \rightarrow \mathbb{R}^+$  where

$$f(v, p) = \begin{cases} f_0(v) & \text{if } p = 0 \\ f_p(v) & \text{if } p = 1. \end{cases} \quad (3.1)$$

The function  $f_0$  models the fuel consumption when the truck is a platoon leader or when it travels solo, and  $f_p$  the fuel consumption when the truck is a platoon follower. These functions can either be derived from an analytical model or fitted to data [30]. We purposely omit that fuel consumption depends on road and vehicle parameters in order to keep the presentation concise. All the presented results can be augmented to handle those additional parameters.

The problem that we want to solve is to find a vehicle plan for each vehicle, and we want to minimize the combined fuel consumption of these plans. The total fuel consumption  $F(\phi_n, \pi_n)$  associated to vehicle  $n$ 's plan is given by

$$F(\phi_n, \pi_n) = \int_{t_n^S}^{t_n^A} f(\phi_n(t), \pi_n(t)) \phi_n(t) dt, \quad (3.2)$$

where  $\phi_n$  is the speed trajectory,  $\pi_n$  the platoon trajectory,  $t_n^S$  the start time, and  $t_n^A$  the arrival time of truck  $n$ . The combined fuel consumption  $F_c$  is given by:

$$F_c = \sum_{n \in \mathcal{N}_c} F(\phi_n, \pi_n). \quad (3.3)$$

Our primary goal is to compute vehicle plans that minimize  $F_c$ .



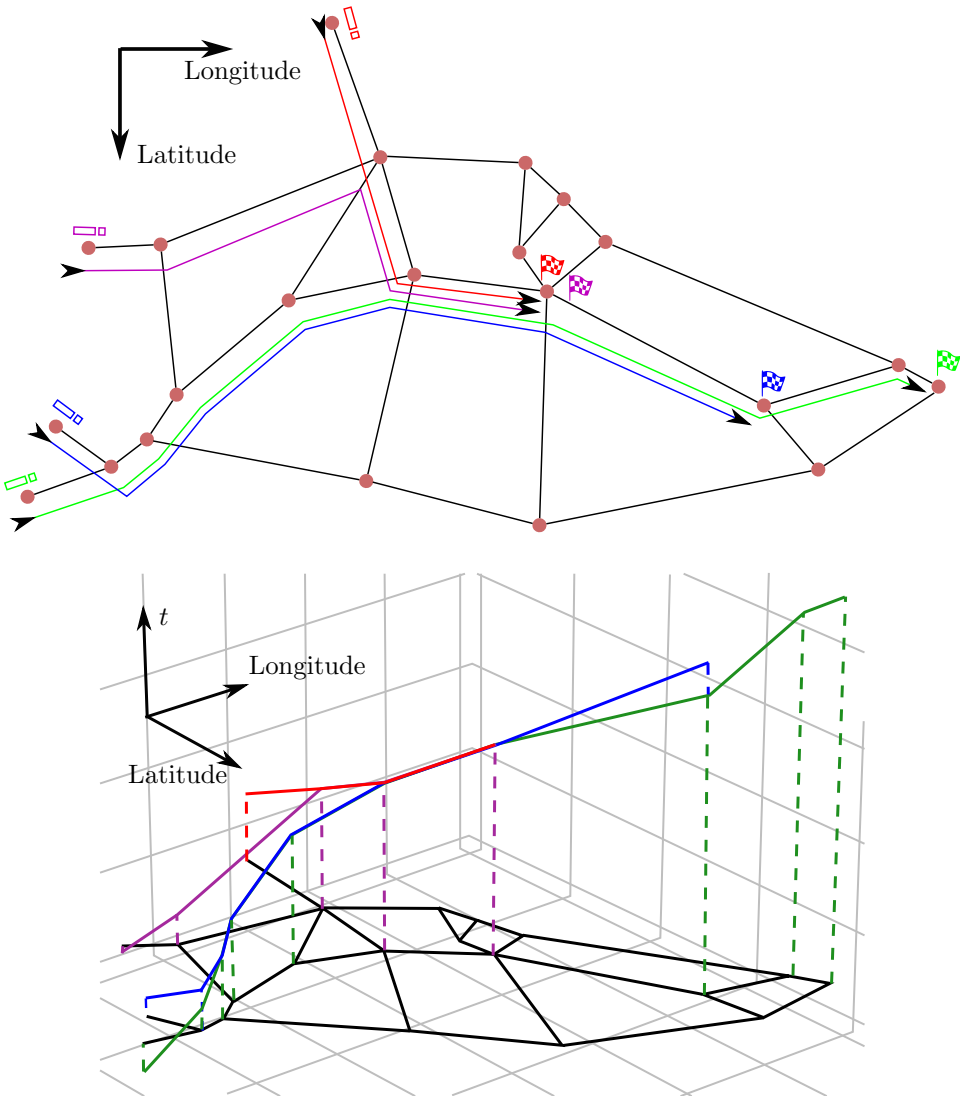


Figure 3.1: Each assignment consists of a start position and a destination in the network. Vehicle plans consist of a route, a speed sequence, and a time sequence. When the position trajectories of two vehicles partially coincide, these vehicles can form a platoon and save fuel.



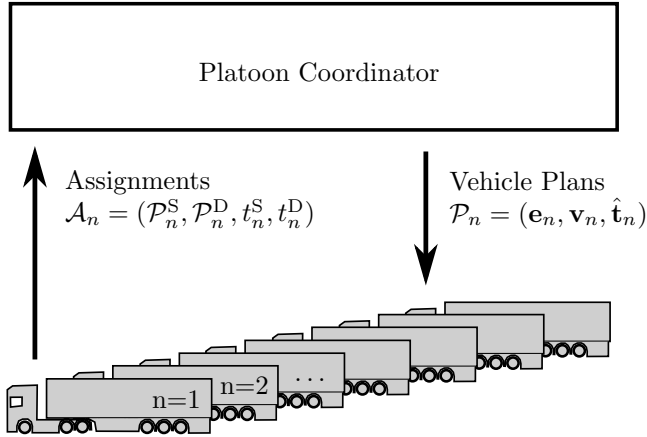


Figure 3.2: Schematic of the platoon coordinator. Trucks provide their assignment data and the platoon coordinator computes fuel efficient vehicle plans.

### 3.2 Platoon Coordinator

Consider the centralized platoon coordinator in Figure 3.2. Trucks connect to the coordinator via vehicle-to-infrastructure communication and share their assignment data. The coordinator then computes fuel-efficient vehicle plans for the trucks. These plans are sent to the trucks and executed. This process is repeated whenever there is updated information, such as deviations from the plans and new assignments. The current vehicle position is then the new start position of an assignment that is already being executed.

The computation of the vehicle plans happens in four stages:

1. Computation of the routes  $\mathbf{e}_n$ ,  $n \in \mathcal{N}_c$ : routes are calculated using an algorithm for route calculation in road networks.
2. Extraction of candidate platoon pairs: all pairs of vehicles that can platoon are identified.
3. Computation of pairwise vehicle plans: many plans involving two vehicles are computed. The fuel savings of these plans are recorded as the coordination graph  $\mathcal{G}_c$  introduced in Section 6.1.
4. Selection of pairwise plans: a consistent subset of the plans computed in the previous stage is combined by selecting a subset  $\mathcal{N}_1 \subset \mathcal{N}_c$ , so-called called coordination leaders.
5. Joint vehicle plan optimization: the selected pairwise plans are jointly optimized for low fuel consumption.



Stage 1. computes the routes  $\mathbf{e}_n$ ,  $n \in \mathcal{N}_c$  and stages 2.–4. compute the speed sequences  $\mathbf{v}_n$  and time sequences  $\hat{\mathbf{t}}_n$  for  $n \in \mathcal{N}_c$  making use of the ability of the trucks to form platoons in order to achieve lower fuel consumption. Algorithms for route calculation in road networks are well developed [80, 22] and not further discussed in this thesis. We discuss stages 2.–4. in the following chapters.

### 3.3 Summary

The problem considered in this thesis is to find and analyze a way of computing fuel-efficient vehicle plans for the assignments. An assignment consists of a start location, a goal location, a start time, and an arrival deadline. A valid vehicle plan connects start and goal location by a route in the road network and computes a speed profile that lets the vehicle reach its destination before the deadline. Using a fuel consumption model, the total fuel consumption of a vehicle plan can be computed. When two vehicle plans enable the trucks to platoon, it is possible to reduce the resulting fuel consumption.

The platoon coordinator is a system that receives the assignments of connected vehicles. In several stages it computes valid vehicle plans to fulfill the assignments. To do so, it computes vehicle plans in a way that lets vehicles form platoons during parts of their journey and save fuel. Deviations from the plans and new vehicles are accounted for by frequent replanning.

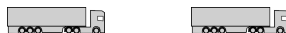


## Chapter 4

# Extracting Candidate Platoon Pairs

IN this chapter, we introduce a scalable way of computing all pairs of assignments that have an overlapping route. As introduced in Section 3.1, two trucks can only form a platoon if they have at least one edge of their routes in common. The topic of this thesis is to decide which platoons should be formed where and when. Thus, an obvious first step is to compute which pairs of transport assignments have at least one edge of their routes in common. The straightforward way of doing this is to compute for each pair of transport assignments individually the common edges in their routes. If there are common edges in the routes, we can determine if the trucks can form a platoon on any of those taking into account the start times, arrival deadlines, and speed constraints. This procedure involves computing a number of set intersections, and this number scales quadratically with the transport assignment count. Such computation becomes problematic for large vehicle fleets. Therefore, we introduce a computationally less expensive and scalable step to narrow down the set of candidate pairs.

Section 4.1 associates each vehicle plan with a sequence of time intervals and a sequence of two-dimensional positions. This information provides limits on the possible points in time a vehicle can be at a certain position as long as the vehicle travels according to a valid vehicle plan. If two vehicles can be at the same position at the same time, they are candidates for platooning. Section 4.2 introduces the concept of feature extraction and culling. Features are computed based sequences of positions and time intervals, and they are significantly less complex than these sequences. For some assignment pairs, it is possible to efficiently rule out the possibility of platooning based on these features. An algorithm for such computation is called a classifier. Section 4.3 develops appropriate features and classifiers. In Section 4.4, these classifiers are demonstrated in a simulation example.



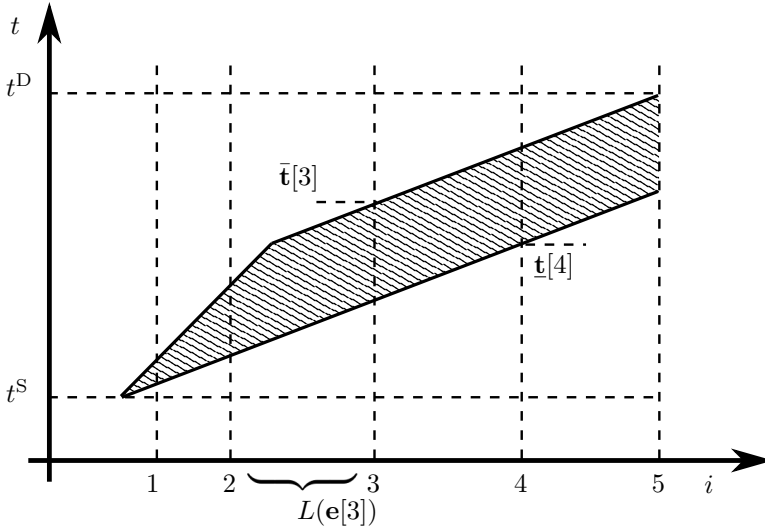


Figure 4.1: This plot illustrates the computation of the lower and upper bounds on the node arrival times. The truck start at the start time at the start position. The lower bound can be achieved with maximum speed  $v_{\max}$  from start to destination. The upper bound can be achieved by selecting the minimum speed  $v_{\min}$  up to the point where the truck arrives on deadline when it selects the maximum speed  $v_{\max}$  until the destination.

## 4.1 Candidate Platoon Pairs

We start by defining a function that indicates whether platooning between two transport assignments is possible or not. This is the case if there is at least one common edge in the routes of the transport assignments where the vehicles can intersect. To this end, we convert the routes to a sequence of nodes in  $\mathcal{G}_r$  and compute lower bounds  $\underline{t}$  and upper bounds  $\bar{t}$  on the points in time when these nodes can be reached. Overlapping time bounds on two consecutive nodes indicate that the two transport assignments can platoon<sup>1</sup>. We have for the sequence of nodes  $\mathbf{n} = \mathbf{n}[1], \dots, \mathbf{n}[N_e - 1]$  of a transport assignment with route  $\mathbf{e} = \mathbf{e}[1], \dots, \mathbf{e}[N_e]$  that  $\mathbf{n}[i] = n : (\cdot, n) = \mathbf{e}[i]$  for  $i = 1, \dots, N_e - 1$ . The possible arrival times at these

<sup>1</sup>This excludes the possibility of only platooning on the first or last link of a truck's route. However, these links are fairly small in realistic road networks (at most a few hundred meters) so that this simplification is of small practical relevance. A node for the start position and a node for the destination can be added to overcome this issue. We omit this for the sake of concise presentation.





nodes are computed according to

$$\underline{\mathbf{t}}[i] = \frac{\sum_{j=1}^i L(\mathbf{e}[j]) - x^S}{v_{\max}} + t^S \quad (4.1)$$

$$\bar{\mathbf{t}}[i] = \min \left( \frac{\sum_{j=1}^i L(\mathbf{e}[j]) - x^S}{v_{\min}} + t^S, t^D - \frac{\sum_{j=i+1}^{N_c-1} L(\mathbf{e}[j]) + x^D}{v_{\max}} \right) \quad (4.2)$$

Figure 4.1 illustrates the above definition of  $\underline{\mathbf{t}}$  and  $\bar{\mathbf{t}}$ . Recall that  $L$  associates edges in the road network with the length of the corresponding road segment.

Furthermore, each node can be associated with a two-dimensional position  $\mathbf{P} : \mathcal{E}_c \rightarrow \mathbb{R}^2$ . This can be, for instance, longitude and latitude of the node in the road network.

We introduce a function that indicates whether or not two transport assignments have the possibility to platoon.

**Definition 2** (Coordination Function). The coordination function  $C : \mathcal{N}_c \times \mathcal{N}_c \rightarrow \{0, 1\}$  has the following properties. Let  $\underline{\mathbf{t}}_i, \underline{\mathbf{t}}_j$  be the lower bounds and  $\bar{\mathbf{t}}_i, \bar{\mathbf{t}}_j$  the upper bounds on the node arrival times of transport assignments  $i$  and  $j$  according to (4.1), (4.2). Then it holds that  $C(i, j) = 1$ , if there are indices  $a, b$  such that

$$\begin{aligned} \mathbf{P}(\mathbf{n}_i[a]) &= \mathbf{P}(\mathbf{n}_j[b]) \\ \mathbf{P}(\mathbf{n}_i[a+1]) &= \mathbf{P}(\mathbf{n}_j[b+1]), \end{aligned}$$

and

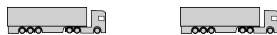
$$\begin{aligned} [\underline{\mathbf{t}}_i[a], \bar{\mathbf{t}}_i[a]] \cap [\underline{\mathbf{t}}_j[b], \bar{\mathbf{t}}_j[b]] &\neq \emptyset \\ [[\underline{\mathbf{t}}_i[a+1], \bar{\mathbf{t}}_i[a+1]] \cap [\underline{\mathbf{t}}_j[b+1], \bar{\mathbf{t}}_j[b+1]]] &\neq \emptyset \end{aligned}$$

Otherwise  $C(i, j) = 0$ .

Comparing the routes and the time bounds in order to evaluate  $C$ , is straightforward but computationally expensive. We refer to this as the *exact algorithm*. In the remainder of this chapter, we derive a scalable method for computing the set of all possible platoon pairs  $\mathcal{C} = \{(i, j) \in \mathcal{N}_c \times \mathcal{N}_c : C(i, j) = 1\}$ . Instead of iterating over all elements in  $\mathcal{N}_c \times \mathcal{N}_c$  and using the exact algorithm, we propose to first efficiently compute an over-approximation  $\hat{\mathcal{C}} \supset \mathcal{C}$  (see Figure 4.2) and then applying the exact algorithm.

## 4.2 Culling Candidate Platoon Pairs

The key idea of our approach is to extract features from the routes and time bounds  $(\mathbf{n}, \underline{\mathbf{t}}, \bar{\mathbf{t}})$  of the transport assignments, as illustrated in Figure 4.3, in order to compute  $\hat{\mathcal{C}}$ . These features can be more efficiently processed than  $(\mathbf{n}, \underline{\mathbf{t}}, \bar{\mathbf{t}})$ . The features



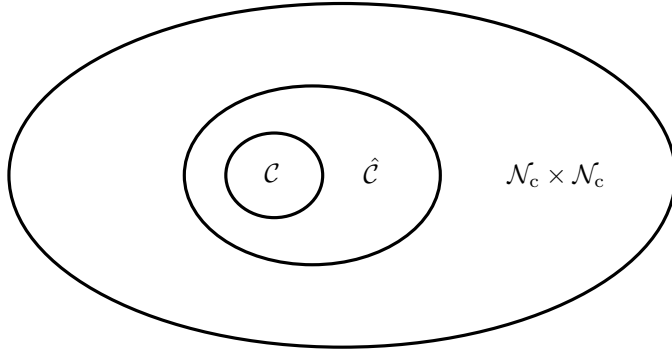


Figure 4.2: Instead of computing  $\mathcal{C}$  by directly iterating over all element in  $\mathcal{N}_c \times \mathcal{N}_c$ , we first compute an over-approximation of  $\mathcal{C}$  denoted  $\hat{\mathcal{C}}$  in an efficient way.

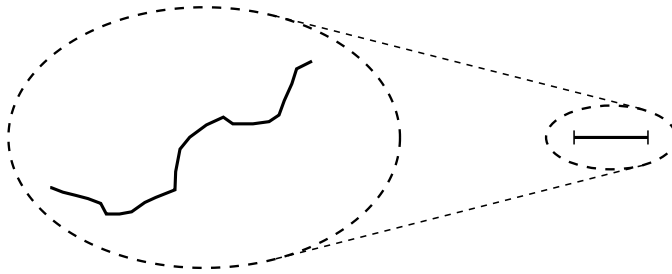


Figure 4.3: Each assignment's route and time bounds are used to compute features, such as an interval.

are designed in a way that no platooning opportunity in  $\mathcal{C}$  is excluded from  $\hat{\mathcal{C}}$ , so that  $\mathcal{C}$  can be computed from  $\hat{\mathcal{C}}$  using the exact algorithm. However, there might be some additional elements in  $\hat{\mathcal{C}}$  that do not actually correspond to platooning opportunities. We call these additional elements false-positives. The less false-positives there are in  $\hat{\mathcal{C}}$ , the faster the computation of  $\mathcal{C}$  from  $\hat{\mathcal{C}}$  is. This approach is inspired by algorithms for detecting collisions between a large number of geometric objects [14, 31]. Figure 4.2 illustrates the relation between  $\mathcal{N}_c \times \mathcal{N}_c$ ,  $\hat{\mathcal{C}}$ , and  $\mathcal{C}$ .

We consider two types of features. These are *interval features* and *binary features*. Interval features map each object to an interval. The corresponding classifier indicates an intersection between two objects if the intervals generated by the objects overlap. There are algorithms, such as [14, 31], that can compute this classifier for all object pairs more efficiently than checking each pair individually if the number of intersecting pairs is small. Binary features map each object to a boolean value. The corresponding classifier indicates an intersection between two objects



if the feature holds true for both objects. In Section 4.3, we derive appropriate features for the problem stated in Section 3.1.

The classifiers are aggregated using boolean connectives. We formalize this in the remainder of the section. Let  $\mathcal{N}$  be a set of objects. We define a classifier as a function  $c : \mathcal{N} \times \mathcal{N} \rightarrow \{0, 1\}$ . If  $c(i, j) = 0$ , we call the combination of  $c$  and  $(i, j)$  a negative, and if  $c(i, j) = 1$ , we call it a positive. Let  $g : \mathcal{N} \times \mathcal{N} \rightarrow \{0, 1\}$  be the ground truth, which can be computed by the exact algorithm. If, for a pair  $(i, j)$ , we have  $g(i, j) = 0$  and  $c(i, j) = 1$ , we call it a false-positive, and if  $g(i, j) = 1$  and  $c(i, j) = 0$ , we call it a false-negative. Our aim is to design classifiers that yield no false negatives for all elements of  $\mathcal{N} \times \mathcal{N}$  and few false-positives that have to be processed by the exact algorithm in addition to the true-positives.

We can identify two types of basic classifiers that are combined in a specific way in order to achieve the above objective. A classifier  $c$  is *required* if

$$\neg c(i, j) \Rightarrow \neg g(i, j)$$

for all  $i, j \in \mathcal{N} \times \mathcal{N}$ . In some cases, we have to take into account a set of classifiers to conclude that  $g$  does not hold. A set of classifiers  $\mathcal{S}_c$  is *required* if

$$\neg \bigvee_{c \in \mathcal{S}_c} c(i, j) \Rightarrow \neg g(i, j)$$

for all  $i, j \in \mathcal{N} \times \mathcal{N}$ . It is straightforward to construct a required classifier from a required set of classifiers.

**Proposition 1.** *If a set  $\mathcal{S}_c$  of classifiers is required, then  $\bigvee_{c \in \mathcal{S}_c} c$  is a required classifier.*

We can combine two required classifiers into one required classifier that performs no worse than any of the required classifiers it is combined of.

**Proposition 2.** *If  $c_1$  and  $c_2$  are required classifiers, then  $c_{12} := c_1 \wedge c_2$  is a required classifier. Let  $\bar{\mathcal{E}}_{12} = \{(i, j) \in \mathcal{N} \times \mathcal{N} : c_{12}(i, j) = 0\}$  be the set of negatives of  $c_{12}$  and let  $\bar{\mathcal{E}}_1, \bar{\mathcal{E}}_2$  be the set of negatives for  $c_1$  and  $c_2$  respectively. Then  $\bar{\mathcal{E}}_1 \subseteq \bar{\mathcal{E}}_{12}$  and  $\bar{\mathcal{E}}_2 \subseteq \bar{\mathcal{E}}_{12}$ .*

*Proof.* For  $c_{12}$  to be required, we need to show that  $\neg c_{12}(i, j) \Rightarrow \neg g(i, j)$  for all  $i, j \in \mathcal{N} \times \mathcal{N}$ . We have

$$\begin{aligned} (\neg c_1 \Rightarrow \neg g) \wedge (\neg c_2 \Rightarrow \neg g) &= (c_1 \vee \neg c_1 \wedge \neg g) \wedge (c_2 \vee \neg c_2 \wedge \neg g) \\ &= c_1 \wedge c_2 \vee \neg g \wedge (\neg c_1 \wedge \neg c_2 \vee \neg c_1 \wedge c_2 \vee c_1 \wedge \neg c_2) \\ &= c_1 \wedge c_2 \vee \neg g \wedge (\neg c_1 \vee \neg c_2) \\ &= c_1 \wedge c_2 \vee \neg g \wedge \neg(c_1 \wedge c_2) \\ &= \neg(c_1 \wedge c_2) \Rightarrow \neg g \\ &= \neg c_{12} \Rightarrow \neg g. \end{aligned}$$



Let  $(i, j) \in \bar{\mathcal{E}}_1$ . Then from the definition of  $\bar{\mathcal{E}}_1$  we have that  $c_1(i, j) = 0$ . We have that

$$c_{12}(i, j) = c_1(i, j) \wedge c_2(i, j) = 0 \wedge c_2(i, j) = 0.$$

It follows from the definition of  $\bar{\mathcal{E}}_{12}$  that  $(i, j) \in \bar{\mathcal{E}}_{12}$ . Similarly, we see that any element of  $\bar{\mathcal{E}}_1$  is an element of  $\bar{\mathcal{E}}_{12}$ .  $\square$

In this manner, we can combine as many required classifiers as we want and have at our disposal. With each classifier we add, we potentially decrease the set of remaining candidates that need to be checked by the exact algorithm. There is a trade-off between doing more work to evaluate more classifiers and having less instances that have to be processed by the exact algorithm [64].

### 4.3 Features and Classifiers for Culling Platoon Pairs

In order to apply the results from Section 4.2, we need to specify appropriate features and classifiers based on these features for the problem stated in Section 3.1. Once we know how to compute appropriate features that yield required classifiers or required sets of classifiers, we can use the results from Section 4.2 to execute the culling phase. The remaining candidate pairs are passed on to the exact algorithm to compute  $\mathcal{C}$ . Hence, we derive a selection of features and corresponding classifiers in this section. In Section 4.4, we demonstrate these classifiers and combinations of them in a simulation example.

The first feature projects the possible trajectories on a line, which yields an interval. Formally, we define this feature as follows.

**Definition 3.** Let  $\mathbf{p} \in \mathbb{R}^3$  be a three dimensional vector that defines the orientation of the line on which the trajectories are projected to. Then the associated interval feature is defined as

$$\mathcal{I} = [\min_{\mathbf{v} \in \mathcal{R}}(\mathbf{p}^T \mathbf{v}), \max_{\mathbf{v} \in \mathcal{R}}(\mathbf{p}^T \mathbf{v})] \quad (4.3)$$

with

$$\mathcal{R} = \left\{ \left[ \begin{array}{c} \mathbf{P}(\mathbf{n}[1]) \\ \mathbf{t}[1] \end{array} \right], \dots, \left[ \begin{array}{c} \mathbf{P}(\mathbf{n}[N_e - 1]) \\ \mathbf{t}[N_e - 1] \end{array} \right], \right. \\ \left. \left[ \begin{array}{c} \mathbf{P}(\mathbf{n}[1]) \\ \mathbf{t}[1] \end{array} \right], \dots, \left[ \begin{array}{c} \mathbf{P}(\mathbf{n}[N_e - 1]) \\ \mathbf{t}[N_e - 1] \end{array} \right] \right\}. \quad (4.4)$$

This feature is illustrated in Figure 4.4. The projection vector  $\mathbf{p}$  is a design choice. Proposition 2 allows us to combine arbitrarily many classifiers based on this kind of feature with different  $\mathbf{p}$ .

Next, we establish that if for a pair of transport assignments the intervals do not overlap, the coordination function is equal to zero. This allows us to define a required feature based on the overlap between these intervals.



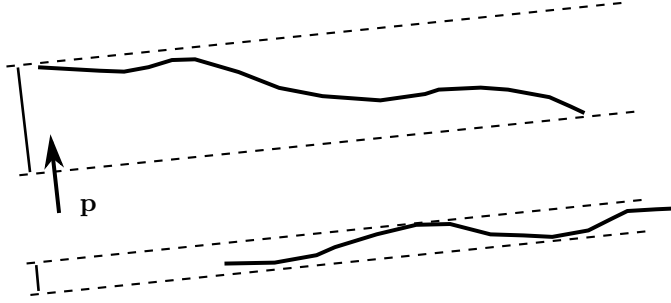


Figure 4.4: Illustration of the projection feature. It shows how the two routes (solid lines) are projected onto a line in the direction of the vector  $\mathbf{p}$ . The borders of the intervals are indicated with dashed lines. For illustration purposes the third dimension is omitted here. In this case the projection of the two routes does not overlap and we can conclude that these routes have no edges in common.

**Proposition 3.** *Let  $(i, j)$  refer to a pair of transport assignments. Let  $\mathcal{I}_i, \mathcal{I}_j$  be the interval features according to (4.3) for the two transport assignments. Then  $\mathcal{I}_i \cap \mathcal{I}_j = \emptyset \Rightarrow C(i, j) = 0$ .*

*Proof.* According to Definition 2,  $C(i, j) = 1$  implies that there must be indices  $a, b$  such that  $\mathbf{P}(\mathbf{n}_i[a]) = \mathbf{P}(\mathbf{n}_j[b])$  and  $[\underline{\mathbf{t}}_i[a], \bar{\mathbf{t}}_i[a]] \cap [\underline{\mathbf{t}}_j[b], \bar{\mathbf{t}}_j[b]] \neq \emptyset$ , where  $\mathbf{n}_i, \underline{\mathbf{t}}_i, \bar{\mathbf{t}}_i$  and  $\mathbf{n}_j, \underline{\mathbf{t}}_j, \bar{\mathbf{t}}_j$  are the node sequences and time bounds of transport assignment  $i, j$  respectively. We have

$$[\underline{\mathbf{t}}_i[a], \bar{\mathbf{t}}_i[a]] \cap [\underline{\mathbf{t}}_j[b], \bar{\mathbf{t}}_j[b]] \neq \emptyset \Leftrightarrow \underline{\mathbf{t}}_i[a] \leq \bar{\mathbf{t}}_j[b] \wedge \underline{\mathbf{t}}_j[b] \leq \bar{\mathbf{t}}_i[a].$$

Let

$$\begin{aligned} \mathbf{p} &= [\mathbf{p}[1], \mathbf{p}[2], \mathbf{p}[3]]^T, \\ \mathbf{P} &= \mathbf{P}(\mathbf{n}_i[a]) = \mathbf{P}(\mathbf{n}_j[b]), \\ P^0 &= [\mathbf{p}[1], \mathbf{p}[2]]\mathbf{P}. \end{aligned}$$

We have

$$\begin{aligned} & \underline{\mathbf{t}}_i[a] \leq \bar{\mathbf{t}}_j[b] \wedge \underline{\mathbf{t}}_j[b] \leq \bar{\mathbf{t}}_i[a] \\ \Rightarrow & \min(\mathbf{p}[3]\underline{\mathbf{t}}_i[a], \mathbf{p}[3]\bar{\mathbf{t}}_i[a]) \leq \max(\mathbf{p}[3]\underline{\mathbf{t}}_j[b], \mathbf{p}[3]\bar{\mathbf{t}}_j[b]) \\ \Rightarrow & \min(\mathbf{p}[3]\underline{\mathbf{t}}_i[a] + P^0, \mathbf{p}[3]\bar{\mathbf{t}}_i[a] + P^0) \leq \max(\mathbf{p}[3]\underline{\mathbf{t}}_j[b] + P^0, \mathbf{p}[3]\bar{\mathbf{t}}_j[b] + P^0) \\ \Rightarrow & \min\left(\mathbf{p}^T \begin{bmatrix} \mathbf{P} \\ \underline{\mathbf{t}}_i[a] \end{bmatrix}, \mathbf{p}^T \begin{bmatrix} \mathbf{P} \\ \bar{\mathbf{t}}_i[a] \end{bmatrix}\right) \leq \max\left(\mathbf{p}^T \begin{bmatrix} \mathbf{P} \\ \underline{\mathbf{t}}_j[b] \end{bmatrix}, \mathbf{p}^T \begin{bmatrix} \mathbf{P} \\ \bar{\mathbf{t}}_j[b] \end{bmatrix}\right) \\ \Rightarrow & \min_{\mathbf{v} \in \mathcal{R}_i}(\mathbf{p}^T \mathbf{v}) \leq \max_{\mathbf{v} \in \mathcal{R}_j}(\mathbf{p}^T \mathbf{v}), \end{aligned}$$



with  $\mathcal{R}_i, \mathcal{R}_j$  as in (4.4) for transport assignment  $i, j$ , respectively. Similarly, by swapping  $i$  and  $j$ , we can show that the conditions of the proposition imply that

$$\min_{\mathbf{v} \in \mathcal{R}_j} (\mathbf{p}^T \mathbf{v}) \leq \max_{\mathbf{v} \in \mathcal{R}_i} (\mathbf{p}^T \mathbf{v}).$$

The above two conditions combined imply that  $\mathcal{I}_i \cap \mathcal{I}_j \neq \emptyset$ . Thus

$$C = 1 \Rightarrow \mathcal{I}_i \cap \mathcal{I}_j \neq \emptyset,$$

or equivalently

$$\mathcal{I}_i \cap \mathcal{I}_j = \emptyset \Rightarrow C = 0.$$

□

Next, we introduce a binary feature that leads to a required classifier. This feature is based on the orientations of the individual links in a route. It is only useful if all segments in a route point approximately from start to goal location. Later on, we address the problem of outliers. Here, we derive a set of required classifiers each based on a binary feature from the orientation. The orientation  $\Theta(n_1, n_2) \in [0, 2\pi]$  of an edge  $(n_1, n_2) \in \mathcal{E}_r$  is the angle in polar coordinates of the vector  $\mathbf{P}(n_2) - \mathbf{P}(n_1)$ . We choose a partition of the interval  $[0, 2\pi]$ . Each element of the partition is related to one binary feature, which holds true if the orientation of at least one edge in the route falls in the range of that element. When two routes overlap there must be at least one edge that has the same orientation. Figure 4.5 illustrates the classifier.

**Proposition 4.** *Let  $(i, j)$  refer to the pair of transport assignments. Let  $\bar{\mathcal{P}}$  be a partition of  $[0, 2\pi]$ . If there is no element  $I \in \bar{\mathcal{P}}$  and edges in the routes of the transport assignments  $(\mathbf{n}_i[a], \mathbf{n}_i[a+1]), (\mathbf{n}_j[b], \mathbf{n}_j[b+1])$  such that  $\Theta(\mathbf{n}_i[a], \mathbf{n}_i[a+1]) \in I$  and  $\Theta(\mathbf{n}_j[b], \mathbf{n}_j[b+1]) \in I$ , then  $C(i, j) = 0$ .*

*Proof.* According to Definition 2,  $C(i, j) = 1$  implies that there must be indices  $a, b$  such that  $\mathbf{P}(\mathbf{n}_i[a]) = \mathbf{P}(\mathbf{n}_j[b])$  and  $\mathbf{P}(\mathbf{n}_i[a+1]) = \mathbf{P}(\mathbf{n}_j[b+1])$ , where  $n_i, n_j$  are the node sequences of transport assignment  $i, j$  respectively. For these it holds that  $\Theta(\mathbf{n}_i[a], \mathbf{n}_i[a+1]) = \Theta(\mathbf{n}_j[b], \mathbf{n}_j[b+1])$ . Since  $\bar{\mathcal{P}}$  is a partition of  $[0, 2\pi]$  and  $\Theta(\mathbf{n}_i[a], \mathbf{n}_i[a+1]) \in [0, 2\pi]$ , there must be  $I \in \bar{\mathcal{P}}$  with  $\Theta(\mathbf{n}_i[a], \mathbf{n}_i[a+1]) \in I$ . Since  $\Theta(\mathbf{n}_j[b], \mathbf{n}_j[b+1]) = \Theta(\mathbf{n}_i[a], \mathbf{n}_i[a+1])$ , it follows that also  $\Theta(\mathbf{n}_j[b], \mathbf{n}_j[b+1]) \in I$ . The proof follows from contradiction. □

Next, we discuss how we can make the orientation based classifier more efficient if we can disregard routes that overlap only over a short distance. Apart from the direct reduction in true positives, this approach will also reduce the false-positive rate of the classifiers, since some outlier route edges can be disregarded.

In order to cover the notion that there must be a minimum overlap in routes to be considered, we extend the definition of the coordination function (Definition 2).



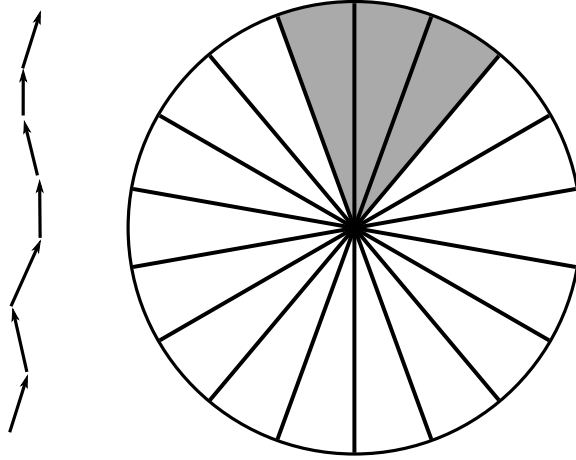


Figure 4.5: Illustration of the classifier based on the orientation. In this example the interval  $[0, 2\pi]$  is partitioned into  $20^\circ$  intervals. The arrows on the left symbolize edges of a route. The elements of the partition for which at least one edge in the route has the same orientation are filled with gray.

**Definition 4** (Minimum Distance Coordination Function).

A coordination function  $C : \mathcal{N}_c \times \mathcal{N}_c \rightarrow \{0, 1\}$  according to Definition 2 requires minimum distance  $l_{\min}$  if the following properties hold: if for a pair  $(i, j)$  we have  $C(i, j) = 1$ , there must be a set of pairs of indices  $\mathcal{A}$  such that for all  $(a, b) \in \mathcal{A}$  it holds that  $\mathbf{P}(\mathbf{n}_i[a]) = \mathbf{P}(\mathbf{n}_j[b])$  and  $\mathbf{P}(\mathbf{n}_i[a + 1]) = \mathbf{P}(\mathbf{n}_j[b + 1])$ , and  $[\underline{\mathbf{t}}_i[a], \bar{\mathbf{t}}_i[a]] \cap [\underline{\mathbf{t}}_j[b], \bar{\mathbf{t}}_j[b]] \neq \emptyset$  and  $[\underline{\mathbf{t}}_i[a + 1], \bar{\mathbf{t}}_i[a + 1]] \cap [\underline{\mathbf{t}}_j[b + 1], \bar{\mathbf{t}}_j[b + 1]] \neq \emptyset$ .

Furthermore, we require

$$\sum_{(a,b) \in \mathcal{A}} \|\mathbf{P}(\mathbf{n}_i[a]) - \mathbf{P}(\mathbf{n}_i[a + 1])\|_2 \geq l_{\min}.$$

We adapt the orientation-based classifier (Proposition 4) to exclude links of a total length less than  $l_{\min}$ . The approach is to calculate the fraction of route length that lies in each element of the partition. We can ignore the intersection with some elements of the partition as long as the lengths of the links whose orientation is contained in these elements sums up to a value less than  $l_{\min}/2$ . Figure 4.6 illustrates this approach.

**Proposition 5.** Let  $(i, j)$  refer to a pair of transport assignments. Let  $\bar{\mathcal{P}}$  be a partition of  $[0, 2\pi]$ . Let  $\mathcal{I}_i \subseteq \bar{\mathcal{P}}$  and let  $\bar{\mathcal{E}}_i \subseteq \mathcal{E}_i$ , where

$$\mathcal{E}_i = \{(\mathbf{n}_i[a], \mathbf{n}_i[a + 1]) : a \in \{1, \dots, N_{e,i} - 2\}\},$$



such that for all  $e \in \bar{\mathcal{E}}_i$ , it holds that there exists  $I \in \mathcal{I}_i$  with  $\Theta(e) \in I$  and we have

$$\sum_{(n_1, n_2) \in \mathcal{E}_i \setminus \bar{\mathcal{E}}_i} \|\mathbf{P}(n_1) - \mathbf{P}(n_2)\|_2 < l_{\min}/2.$$

Similarly, by replacing  $i$  by  $j$ , we define  $\mathcal{I}_j$  for transport assignment  $j$ . If  $\mathcal{I}_i \cap \mathcal{I}_j = \emptyset$ , then  $C(i, j) = 0$  with  $C$  according to Definition 4.

*Proof.* If  $C(i, j) = 1$ , then we have a set of pairs of indices  $\mathcal{A}$  such that for all  $(a, b) \in \mathcal{A}$  it holds that  $\mathbf{P}(\mathbf{n}_i[a]) = \mathbf{P}(\mathbf{n}_j[b])$  and  $\mathbf{P}(\mathbf{n}_i[a+1]) = \mathbf{P}(\mathbf{n}_j[b+1])$ . Thus, it also holds that  $\Theta(\mathbf{n}_i[a], \mathbf{n}_i[a+1]) = \Theta(\mathbf{n}_j[b], \mathbf{n}_j[b+1])$ . Since  $\bar{\mathcal{P}}$  is a partition of the image of  $\Theta(\cdot)$ , there is exactly one element  $I \in \bar{\mathcal{P}}$  with  $\Theta(\mathbf{n}_i[a], \mathbf{n}_i[a+1]) \in I$ , and since  $\Theta(\mathbf{n}_i[a], \mathbf{n}_i[a+1]) = \Theta(\mathbf{n}_j[b], \mathbf{n}_j[b+1])$ , we have

$$\Theta(\mathbf{n}_i[a], \mathbf{n}_i[a+1]) \in I \Leftrightarrow \Theta(\mathbf{n}_j[b], \mathbf{n}_j[b+1]) \in I.$$

Furthermore, we have from Definition 4 that

$$\sum_{(a,b) \in \mathcal{A}} \|\mathbf{P}(\mathbf{n}_i[a]) - \mathbf{P}(\mathbf{n}_i[a+1])\|_2 \geq l_{\min}.$$

Let  $\bar{\mathcal{A}}_i$  be a set of the indices of the head nodes of edges in  $(\mathcal{E}_i \cap \mathcal{E}_j) \setminus \bar{\mathcal{E}}_i$  paired with the corresponding indices in route  $j$ , with  $\mathcal{E}_i, \mathcal{E}_j, \bar{\mathcal{E}}_i$  as defined in the proposition. These are the pairs of indices of the edges in the common part of the route that are ignored in transport assignment  $i$ . Similarly, let  $\bar{\mathcal{A}}_j$  be the index pairs that are excluded due to transport assignment  $j$ . We need to show now that  $\mathcal{A}$  is not empty without the pairs in  $\bar{\mathcal{A}}_i$  and  $\bar{\mathcal{A}}_j$ , or in other words, that even if the features for either route ignore up to  $l_{\min}/2$  of the common part of the route, there are still edges left that let the set of classifiers indicate that the routes intersect. We have from the assumptions made in the proposition

$$\begin{aligned} \sum_{(a,b) \in \bar{\mathcal{A}}_i} \|\mathbf{P}(\mathbf{n}_i[a]) - \mathbf{P}(\mathbf{n}_i[a+1])\|_2 &< l_{\min}/2, \\ \sum_{(a,b) \in \bar{\mathcal{A}}_j} \|\mathbf{P}(\mathbf{n}_i[a]) - \mathbf{P}(\mathbf{n}_i[a+1])\|_2 &< l_{\min}/2, \end{aligned}$$

and from Definition 4 that

$$\sum_{(a,b) \in \mathcal{A}} \|\mathbf{P}(\mathbf{n}_i[a]) - \mathbf{P}(\mathbf{n}_i[a+1])\|_2 \geq l_{\min}.$$

Thus,

$$\sum_{(a,b) \in \mathcal{A} \setminus (\bar{\mathcal{A}}_i \cup \bar{\mathcal{A}}_j)} \|\mathbf{P}(\mathbf{n}_i[a]) - \mathbf{P}(\mathbf{n}_i[a+1])\|_2 > 0,$$





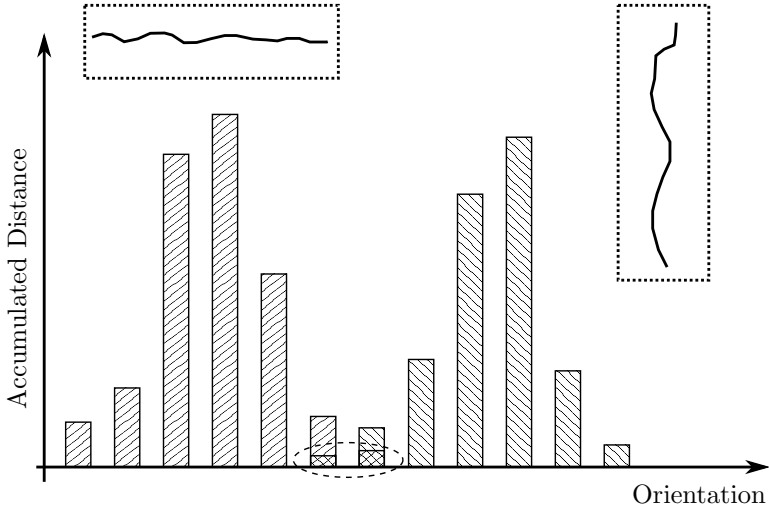


Figure 4.6: This figure illustrates how the performance of the orientation classifier can be improved when overlaps of length less than  $l_{\min}$  can be excluded. The figure shows the histogram of two routes. The routes are sketched on the top of the figure. There are two elements in the partition that contain orientations from both routes corresponding only to a small fraction of the total route length. The classifier according to Proposition 4 will indicate an intersection between these two route whereas the classifier according to Proposition 5 can exclude the few edges with similar orientation.

and since this is a sum over positive elements, we deduce that  $\mathcal{A} \setminus (\bar{\mathcal{A}}_i \cup \bar{\mathcal{A}}_j) \neq \emptyset$ . But then there is  $I \in \bar{\mathcal{P}}$  and  $(a, b) \in \mathcal{A} \setminus (\bar{\mathcal{A}}_i \cup \bar{\mathcal{A}}_j)$  such that

$$\Theta(\mathbf{n}_i[a], \mathbf{n}_i[a+1]) = \Theta(\mathbf{n}_j[b], \mathbf{n}_j[b+1]) \in I,$$

and thus  $\mathcal{I}_i \cap \mathcal{I}_j \neq \emptyset$ . By contraposition it follows that  $\mathcal{I}_i \cap \mathcal{I}_j = \emptyset \implies C(i, j) = 0$ .  $\square$

It is possible to combine various classifiers as defined in Propositions 3 and 5 in various ways according to Propositions 1 and 2 in Section 4.2.

## 4.4 Simulations

In this section, the method derived in this chapter is demonstrated in a realistic scenario. We show that the application of 6 classifiers can rule out 99% of the transport assignment pairs, leaving only 1% for the computationally expensive exact algorithm. The simulation setup is as follows. The start and goal locations are



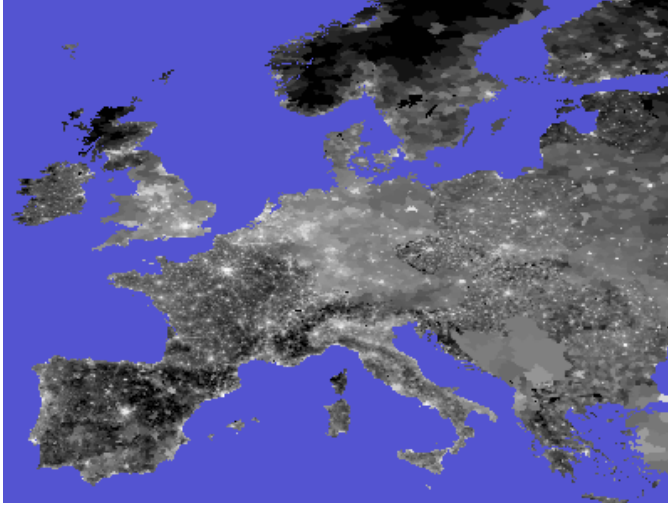


Figure 4.7: Population density map from which the start and goal locations are sampled. The brighter the pixel, the larger the population density in that area. Some areas outside Europe and areas without population are shown in blue.

sampled randomly with probability proportional to an estimate of the population density in the year 2000 [82]. We limit the area to a large part of Europe, which is shown in Figure 4.7.

We calculate shortest routes with the Open Source Routing Machine [65]. If the route is longer than 400 kilometers, a 400 kilometers long subsection of the route is randomly selected. The maximum speed is  $v_{\max} = 80$  km/h. We set the start times  $t^S$  of half the assignments to 0 and sample the start times of the remaining assignments uniformly in an interval of 0 to 24 h. The first half is to account for assignments that are currently on the road while the other half is to account for assignments that are scheduled to depart later. The deadlines  $t^D$  are set in such a way that the interval  $\bar{\mathbf{t}}[a] - \underline{\mathbf{t}}[a] = 0.5 h$  where  $a$  is any valid index. We consider the minimum length that two assignments have to overlap to be considered for platooning,  $l_{\min}$ , to be 20 km.

We implemented all features and corresponding classifiers that are described in Section 4.3, i.e., interval projection (Proposition 3) and minimum distance orientation partition (Proposition 5). Note that Proposition 4 is a special case of Proposition 3 with  $l_{\min} = 0$ . For interval projection we tested vectors of the form

$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} -\cos(\alpha) \\ \frac{-1}{\cos(50^\circ)} \sin(\alpha) \\ \frac{v_{\max} 180^\circ}{6371\pi} \end{bmatrix},$$



with  $\alpha = 0, \pi/4, \dots, 7\pi/4$ . The position  $\mathbf{P}$  is expressed here as latitude and longitude and measured in degrees. The vectors parametrized by  $\alpha$  are approximately orthogonal to a trajectory at maximum speed at the latitude of 50 degrees with heading angle  $\alpha$  and should work well for trajectory pairs that have similar orientation, that cover the same area, and that are only separated by a small time margin. We refer to the corresponding classifiers in the following discussion as  $c_{100}, c_{010}, c_{001}, c_{110}, c_{-110}, c_{\alpha 0}, \dots, c_{\alpha 7}$  respectively. For the orientation-based classifier, we use 100 equally sized cells to partition  $[0, 2\pi]$ . For each cell, the fraction of the route distance that falls in this cell is computed. Matches up to  $l_{\min}/2$  starting in ascending order of route distance contained in the cells are excluded. We refer to this classifier as  $c_o$ .

This simulation focuses on demonstrating that the culling phase is able to filter out a significant amount of assignments before they are passed on to the exact algorithm. Therefore, we do not focus on optimizing the implementation for speed and refrain from reporting running times of the simulations as they might be misleading and we know from related work [64] that these computations can be performed fast enough for the problem at hand if the false-positive rate of the classifiers is small.

We test 1000 transport assignments. All classifiers are evaluated in parallel. Next, the sequence of classifiers that filters the most assignments at every stage is computed. The number of positives for each classifier is listed in Table 4.1. Figure 4.9 shows the number of remaining pairs at each stage, the ground truth, and the sequence of classifiers for this sample. The optimization of the classifier order would typically be done when the system is designed and is to some extent specific for the exact transport setting. In a running platoon coordination system the order in which classifiers are applied would remain fixed.

We can see in Figure 4.9 that two classifiers,  $c_{110}$  and  $c_{\alpha 7}$ , combined are able to reduce the number of pairs by one order of magnitude. The first classifier,  $c_{110}$ , only takes into account longitude and latitude of the routes. The second one,  $c_{\alpha 7}$  is orthogonal to the first one,  $c_{110}$ , in the plane but also takes into account timing. The third classifier,  $c_{\alpha 3}$ , is also of the projection type, which is able to identify that a pair of assignments cannot platoon if they are geographically close but differ in timing, and it covers the opposite orientation compared to the previous classifier. The fourth classifier,  $c_{100}$ , covers a third direction in the plane. It is interesting to see that the fifth classifier,  $c_o$ , is the orientation-based classifier. Alone, it performs much worse than the other classifiers as can be seen in Table 4.1. Two transport assignments that take the same route in opposite directions and that “meet” on the way are impossible to identify as a negative with the projection based classifiers. The orientation-based classifier might be able to achieve that. The classifier that only takes into account start and arrival time,  $c_{001}$ , is selected last, since most of the cases it rules out are already covered by the classifiers  $c_{\alpha 0}, \dots, c_{\alpha 7}$ , and also because half the assignments start at the same time. We see that the benefit from adding more classifiers diminishes quickly as classifiers are added. All classifiers combined can reduce the number of pairs by two orders of magnitude and get within one order of magnitude from the ground truth. The false-positives are mostly very



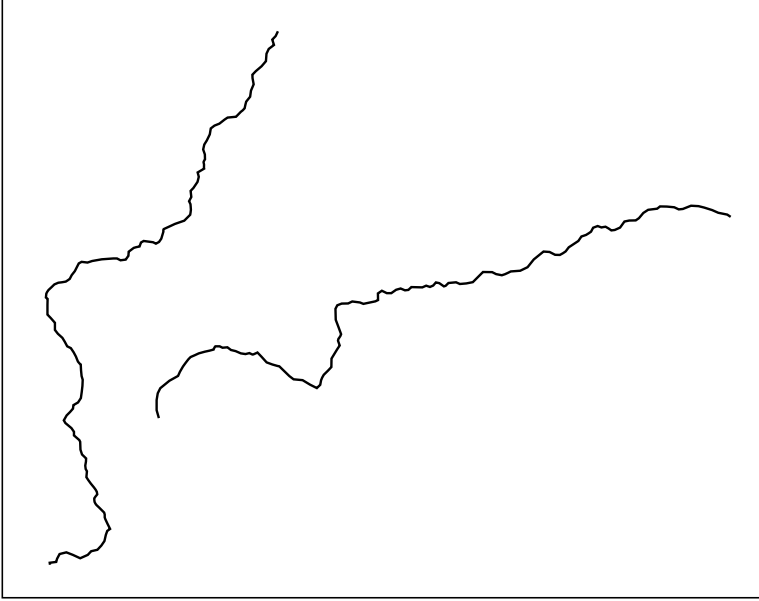


Figure 4.8: Example of a false-positive. The two routes do not overlap. However, the routes cannot be separated by a hyperplane, and since both routes are quite curvy the orientation based classifier cannot conclude that these route do not overlap.

None	499,500	$c_{-110}$	108,403	$c_{\alpha 4}$	134,019
$c_{100}$	104,380	$c_{\alpha 0}$	129,282	$c_{\alpha 5}$	107,287
$c_{010}$	101,542	$c_{\alpha 1}$	103,240	$c_{\alpha 6}$	105,883
$c_{001}$	208,896	$c_{\alpha 2}$	103,453	$c_{\alpha 7}$	109,934
$c_{110}$	98,343	$c_{\alpha 3}$	109,626	$c_o$	453,246

Table 4.1: Number of positives for different classifiers.

curvy routes that intersect geographically and are separated little in time in the area of the intersection. To be able to correctly identify such pairs as negatives is often not possible with the features presented in this chapter. Figure 4.8 shows an example of a false-positive. We get consistent results for different runs of the simulation.



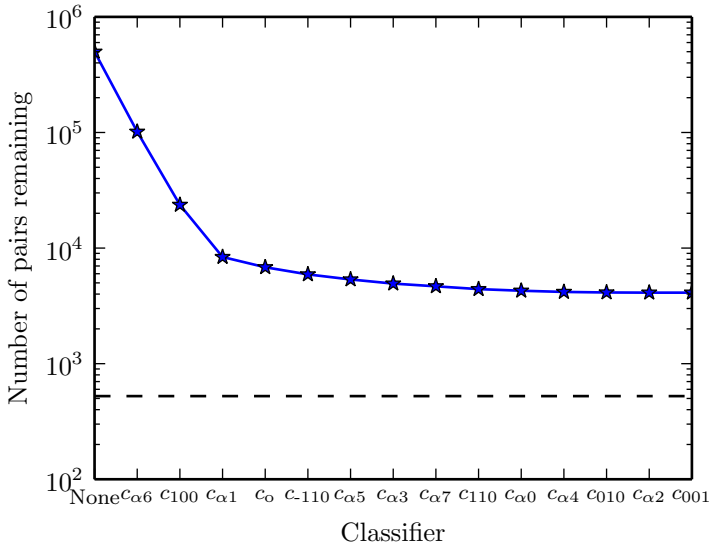


Figure 4.9: The number of remaining pairs when the classifiers are consecutively applied from left to right. The order the classifiers are chosen in a way that each stage removes as many pairs as possible. The classifier applied at each stage is indicated on the horizontal axis. The dashed line shows the ground truth from the exact algorithm.

## 4.5 Summary

Comparing the routes and the time bounds of a large number of assignments in order to find candidates for platooning is computationally expensive. A more efficient approach is to narrow down the set of candidates based on features. A feature is low dimensional data like a boolean truth value or an interval that can be efficiently processed in the form of classifiers. The smaller set of candidates can then be used as an input to computing fuel-efficient vehicle plans for all vehicles that are coordinated. Several classifiers can be combined to get even smaller sets of candidates, and, in some cases, classifiers have to be combined to be able to conclusively rule out that a pair of transport assignments is able to platoon. Two features and corresponding classifiers are derived. One is based on the projection of the route and time bounds onto a line. The other classifier is based on the intersection of route segment orientations with a partition of all possible orientations. The performance of this classifier can be improved by assuming that the common part of the routes of two assignments must have a minimum length to be relevant for platooning. Simulations indicate that the method developed in this chapter can significantly narrow down the set of candidate platoon pairs.





## Chapter 5

# Pairwise Adapted Vehicle Plans

**I**N this chapter, we consider a pair of assignments that offers the possibility for platoon formation on the overlapping part of the corresponding routes. We derive how one truck, the coordination follower, adapts its vehicle plan to another truck, the coordination leader, in a fuel-efficient way making use of the ability of the trucks to platoon.

Later on, in Chapter 6, we work with a wider definition of default plans and adapted plans. The derivations in this chapter can serve as a concrete example of how such plans can be computed. Realistic planning would have to take into account additional factors such as different speed limits along the route, traffic, rests of the driver, etc. The computation of vehicle plans under such additional constraints follows the lines of reasoning as presented in this chapter. However, these additional constraints add a lot of complexity in the notation, and are henceforth omitted.

In Section 5.1, we consider two vehicles with the same route. One vehicle selects a speed that allows the two vehicles to meet and form a platoon. We derive how to select this speed in a fuel-optimal way. In Section 5.2, we extend this result to the case in which the two vehicles have different but overlapping routes. One vehicle adapts its speed profile in a way that allows it to meet the other vehicle on the common section of the routes and the two vehicles form a platoon. Section 5.3 summarizes this chapter.

### 5.1 The Optimal Rendezvous Speed

Consider two vehicles on the same route as depicted in Figure 5.1. The vehicles are initially separated by a distance  $\Delta d$ . Vehicle 0 drives at a default speed, which is denoted  $v_0$ . Vehicle 1, which is behind vehicle 0, drives at a higher speed, denoted  $v_S^*$ . Since  $v_S^* > v_0$ , the distance between the vehicles decreases with time until the two vehicles meet and form a platoon. At this point, both vehicles continue driving in a platoon at default speed  $v_0$ .



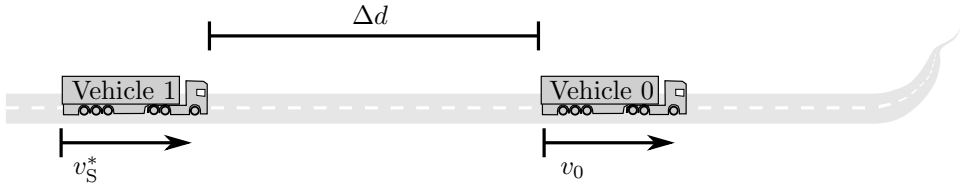


Figure 5.1: Two vehicles on the same route with distance  $\Delta d$ . Vehicle 0 has speed  $v_0$  and vehicle 1 has speed  $v_S^* > v_0$ . Since the speed of vehicle 1 is higher than the speed of vehicle 0, vehicle 1 will catch up with vehicle 0, and the two vehicles will form a platoon.

We want to select the rendezvous speed  $v_S^*$  in a fuel-optimal way, while  $v_0$  is not altered. To this end, we introduce a linear affine fuel-model<sup>1</sup>. The fuel consumption per distance traveled as platoon leader or alone is  $f_0(v) = F^0 + F^1 v$ , and the fuel consumption per distance traveled as platoon follower is  $f_p(v) = F_p^0 + F_p^1 v$ . We assume that the fuel consumption of a platoon follower is lower at default speed than if the vehicle was to travel alone, i.e.,  $F_p^0 + F_p^1 v_0 < F^0 + F^1 v_0$ . It is reasonable to assume this since without this assumption there is no reason to form platoons. We assume that  $v_0$  lies within the feasible speed-range as introduced in Section 3.1, Definition 1, i.e.,  $0 < v_{\min} \leq v_0 \leq v_{\max}$ . The optimal rendezvous speed  $v_S^*$  is also constrained to lie within the feasible speed range.

This problem setting is related to the optimal catch-up schemes derived in [62]. In fact, the catch-up schemes from [62] have been combined with the methods of Chapter 6 in a simulation study presented in [16].

A similar scenario to the one described above is setting where vehicle 1 is in front of vehicle 0. In that case, vehicle 1 selects a speed smaller than  $v_0$ . This means that vehicle 0 will catch up to vehicle 1 instead, and the two vehicles can form a platoon. In the remainder of this section, we consider both the case in which vehicle 1 is behind vehicle 0 and the case that vehicle 1 is in front of vehicle 0.

We model this scenario on a road network with one edge, denoted  $e$ . The length of the road segment corresponding to  $e$ , i.e.,  $L(e)$ , is assumed to be long enough to not impose any restrictions on where the two vehicles meet. The time, when the two vehicles start is denoted  $t^S$ , the time when they meet and start platooning is denoted  $t^M$ , and the time when they stop platooning is denoted  $t^{SP}$ .

The following proposition gives the optimal rendezvous speed  $v_S^*$  for vehicle 1.

**Proposition 6.** *Assume the following. The speed of vehicle 0 is constant  $v_0$  with  $v_0 \in \mathbb{R}$ ,  $v_0 > 0$ . The position of vehicle 0 at time  $t^S$  is  $(e, x_0(t^S))$ . The position of truck 1 at time  $t^S$  is  $(e, x_1(t^S))$ . Truck 1 platoons with truck 0 between time  $t^M$  and  $t^{SP}$  with  $t^{SP} > t^M$ . Truck 1 has constant speed  $v_S$  for time  $t^S$  to  $t^M$  and  $v_0$  from time  $t^M$  to  $t^{SP}$ . The rendezvous speed  $v_S$  is constrained to the interval  $[v_{\min}, v_{\max}]$ .*

<sup>1</sup>Extending the presented results to other fuel models is possible.





Then the rendezvous speed  $v_S^*$  that minimizes fuel consumption from time  $t^S$  to  $t^{Sp}$  is given by

$$v_S^* = \begin{cases} \max \left( v_0 \left( 1 - \sqrt{1 - \frac{F_p^1}{F^1} + \frac{\Delta F^0}{F^1 v_0}} \right), v_{\min} \right) & \text{if } \Delta d < 0 \\ \min \left( v_0 \left( 1 + \sqrt{1 - \frac{F_p^1}{F^1} + \frac{\Delta F^0}{F^1 v_0}} \right), v_{\max} \right) & \text{if } \Delta d > 0 \\ v_0 & \text{if } \Delta d = 0, \end{cases} \quad (5.1)$$

where  $\Delta d = x_0(t^S) - x_1(t^S)$  and  $\Delta F^0 = F^0 - F_p^0$ .

*Proof.* Let  $\Delta d^S = x_1(t^M) - x_1(t^S)$ . Let  $D_0 = x_1(t^{Sp}) - x_1(t^S)$ . We have the relation

$$\Delta d^S = \frac{v_S}{v_S - v_0} \Delta d. \quad (5.2)$$

At time  $t^M$  we have  $x_0(t^M) = x_1(t^M)$ . After the meeting point, both trucks platoon at speed  $v_0$ . Assume that 0 is the platoon leader. Hence, the total fuel consumption of 1 up to some distance from the current position  $D_0$ , which fulfills  $D_0 > \Delta d^S$ , becomes

$$f_0(v_S) \Delta d^S + f_p(v_0)(D_0 - \Delta d^S) = (f_0(v_S) - f_p(v_0)) \Delta d^S + f_p(v_0) D_0.$$

The fuel consumption of 0 is not affected by  $v_S$ . We see that the term  $f_p(v_0) D_0$  is not a function of  $v_S$ , so the optimal rendezvous speed does not depend on the total distance traveled. In order to find the optimal  $v_S$ , we can therefore consider the remaining terms denoted as  $f_r(v_S)$  and get with (5.2) and the definitions of  $f_0, f_p$

$$f_r(v_S) = (f_0(v_S) - f_p(v_0)) \Delta d^S = (F^1 v_S - F_p^1 v_0 + \Delta F^0) \frac{v_S}{v_S - v_0} \Delta d,$$

with  $\Delta F^0 = F^0 - F_p^0$ . We take the derivative of the above expression in order to find its extrema

$$\frac{\partial}{\partial v_S} f_r(v_S) = \frac{\Delta d}{(v_S - v_0)^2} (F^1 v_S^2 - 2F^1 v_0 v_S + F_p^1 v_0^2 - \Delta F^0 v_0).$$

In order to find the extrema  $\tilde{v}_S$ , we check where this expression is zero. We can assume that  $\Delta d \neq 0$ , otherwise  $\Delta d^S = 0$ , which means that the trucks can directly start platooning. Therefore,

$$0 = (F^1 (\tilde{v}_S)^2 - 2F^1 v_0 \tilde{v}_S + F_p^1 v_0^2 - \Delta F^0 v_0) \quad (5.3)$$

$$\tilde{v}_S = v_0 \left( 1 \pm \sqrt{1 - \frac{F_p^1}{F^1} + \frac{\Delta F^0}{F^1 v_0}} \right). \quad (5.4)$$

We have to differentiate between two cases. Either  $\Delta d > 0$ , which implies  $v_S > v_0$ , i.e., the coordination follower speeds up, or  $\Delta d < 0$ , which implies  $v_S < v_0$ , i.e.,



the coordination follower slows down. Otherwise  $\Delta d^S$  becomes negative. There are two solutions for  $\tilde{v}_S$ , one where  $\tilde{v}_S > v_0$ , and the other  $\tilde{v}_S < v_0$ . The appropriate one, depending on  $\Delta d$ , is  $\tilde{v}_S$ , the optimal unconstrained rendezvous speed.

We can verify that this is indeed a minimum by considering the asymptotic behavior of  $f_r(v_S)$  when  $f_r(v_S)$  approaches  $\pm\infty$  and when it approaches  $v_0$ . Assume  $\Delta d > 0$  so that  $\tilde{v}_S > v_0$ . We have

$$\begin{aligned}\lim_{v_S \rightarrow \infty} f_r(v_S) &= \infty, \\ \lim_{v_S \rightarrow v_0^+} f_r(v_S) &= \infty\end{aligned}$$

where we used that  $f_0(v_0) > f_p(v_0)$  so that the term  $f_0(v_0) - f_p(v_0)$  becomes positive, which is the prerequisite to save fuel by platooning. When we have  $\Delta d < 0$ , so that  $\tilde{v}_S < v_0$ , then

$$\begin{aligned}\lim_{v_S \rightarrow -\infty} f_r(v_S) &= \infty, \\ \lim_{v_S \rightarrow v_0^-} f_r(v_S) &= \infty.\end{aligned}$$

This shows that if  $\tilde{v}_S > v_{\max}$ , then  $v_S^* = v_{\max}$ , if  $\tilde{v}_S < v_{\min}$ , then  $v_S^* = v_{\min}$ , and  $v_S^* = \tilde{v}_S$  otherwise.

In order to have real solutions for (5.3), we need

$$\begin{aligned}1 - \frac{F_p^1}{F^1} + \frac{\Delta F^0}{F^1 v_0} > 0 &\Leftrightarrow F_p^1 v_0 + F_p^0 < F^1 v_0 + F^0 \\ &\Leftrightarrow f_p(v_0) < f_0(v_0),\end{aligned}$$

which is the condition that the coordination follower saves fuel when platooning. The larger the difference  $f_0(v_0) - f_p(v_0)$ , the larger the absolute difference between  $v_0$  and  $v_S^*$ , i.e., the longer the trucks platoon.  $\square$

## 5.2 Computing Adapted Vehicle Plans

In this section, we discuss how Proposition 6 can be used to compute an optimal speed profile when the two vehicles travel on different but intersecting routes. Similar to the previous section, we consider that one vehicle travels at the constant default speed. We call this vehicle the coordination leader. The other vehicle, referred to as the coordination follower, adapts its speed in order to meet the coordination leader on the common part of the route, and platoon for some distance. We derive how to make this adaptation happen in a fuel-optimal way.

The meaning of the terms “coordination leader” and “coordination follower” will become more apparent in Chapter 6 where adapted vehicle plans are systematically combined with the goal of minimizing the combined fuel consumption by forming platoons. Note that the notion of a platoon leader/follower is different from the



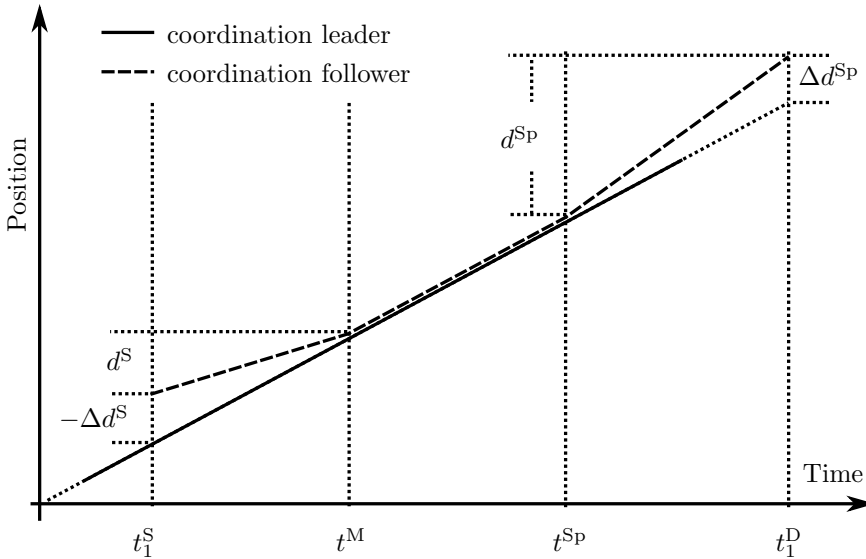


Figure 5.2: Speed profiles of the coordination leader and the coordination follower. The distance along the respective route with respect to a common reference point on the common part of the route is plotted over time. The coordination leader has a constant speed. In this example, the coordination follower drives slower at the beginning of its journey. Once it meets the coordination leader, the two vehicles platoon. At the end the coordination drives at an increased speed in order to make its deadline.

notion of a coordination leader/follower. A platoon leader is the lead truck in the platoon and a platoon follower is one of the trailing trucks. A coordination leader/follower, on the other hand, is a logical role in the composition of pairwise plans.

We consider plans of the following form. The coordination leader keeps a constant speed while the coordination follower selects a speed at the beginning of its journey that allows it to merge into a platoon with the coordination leader. Then the two platoon until they split up, followed by the coordination follower selecting a speed so that it arrives at its pre-specified deadline at its destination. For the sake of simplicity, we assume in this section the trucks to arrive at their destinations exactly on their respective deadlines. In Chapter 6, we also allow for arrival before the deadline. Figure 5.2 illustrates the three phases of the adapted speed profile.

In order to simplify notation, we define the distance  $d_e$  between two positions  $(e[i_1], x_1)$ ,  $(e[i_2], x_2)$  with respect to a route  $e$ .



**Definition 5** (Distance). Let  $i_1, i_2$  be such that  $N_e \geq i_2 \geq i_1$ . Then,

$$d_e((e[i_1], x_1), (e[i_2], x_2)) = \left| x_2 - x_1 + \sum_{i=i_1}^{i_2-1} L(e[i]) \right| \quad (5.5)$$

Consider a coordination leader with index 0 and a coordination follower with index 1. Two trucks can platoon only on the road segments corresponding to common edges of their routes. If their routes are shortest routes, it can be shown that the shared edges between two routes form a path as well (Lemma 1 in [93]), i.e., two routes meet and split up at most once. Trucks 0, 1 start at  $(e_0^S, x_0^S), (e_1^S, x_1^S)$  at time  $t_0^S, t_1^S$  and arrive at  $(e_0^D, x_0^D), (e_1^D, x_1^D)$  at time  $t_0^D, t_1^D$ , respectively. We denote the position at which the coordination leader and the coordination follower start platooning at time  $t^M$  as  $(e^M, x^M)$  and where they split at time  $t^{SP}$  as  $(e^{SP}, x^{SP})$ . These meeting points have to lie on the trajectory of the coordination leader with constant speed  $v_0$ :

$$\begin{aligned} d_{e_0}((e_0^S, x_0^S), (e^M, x^M)) &= v_0(t^M - t_0^S) \\ d_{e_0}((e_0^S, x_0^S), (e^{SP}, x^{SP})) &= v_0(t^{SP} - t_0^S). \end{aligned}$$

When platooning with the coordination leader the planned trajectory of the coordination follower consists of three phases: from start to the meeting point with speed  $v_S$ , from meeting point to the split point platooning as platoon follower of 0 with speed  $v_0$ , and from the split point to the destination with speed  $v_{SP}$ . We define  $d^S = d_{e_1}((e_1^S, x_1^S), (e^M, x^M))$  and  $d^{SP} = d_{e_1}((e^{SP}, x^{SP}), (e_1^D, x_1^D))$ . We have the relations

$$\begin{aligned} d^S &= v_S(t^M - t_1^S), \\ d^{SP} &= v_{SP}(t_1^D - t^{SP}). \end{aligned}$$

We define the virtual position difference at the start/end of the coordination follower's trajectory as

$$\begin{aligned} \Delta d^S &= d^S - (t^M - t_1^S)v_0 \\ \Delta d^{SP} &= d^{SP} - (t_1^D - t^{SP})v_0, \end{aligned} \quad (5.6)$$

which are equivalent to  $\Delta d$  in Proposition 6. If  $\Delta d^S > 0$  then  $v_S > v_0$ , if  $\Delta d^S < 0$  then  $v_S < v_0$ , if  $\Delta d^{SP} > 0$  then  $v_{SP} > v_0$ , and if  $\Delta d^{SP} < 0$  then  $v_{SP} < v_0$ . Then, we can compute according to (5.4) the appropriate, fuel-optimal speed  $v_S^*$  for the first and the last phase. Proposition 6 considers that the two vehicles are initially separated. The same lines of reasoning apply in order to determine the optimal speed of the coordination follower during the last phase.

This derivation has not taken into account so far that the first possible point to merge is when the coordination leader's and the coordination follower's routes meet. If  $v_S^*$  leads to a distance from  $(e_1^S, x_1^S)$  to the merge point that is too small,



then the coordination leader selects a speed that lets the coordination leader and coordination follower merge at the position where the two routes meet, denoted here  $(e^F, 0)$ . This speed is

$$v_S = \frac{d_{e_1}((e_1^S, x_1^S), (e^F, 0))}{t^M - t_1^S}.$$

The corresponding case might occur at split up, so that

$$v_{Sp} = \frac{d_{e_1}((e^L, L(e^L)), (e_1^D, x_1^D))}{t_1^D - t^{Sp}},$$

where  $(e^L, L(e^L))$  is the position where the coordination leader's and the coordination follower's routes split up.

The first test if platooning is possible and beneficial is, whether the calculated merge point lies before the split point or not, i.e., whether

$$d^S + d^{Sp} < d_{e_1}((e_1^S, x_1^S), (e_1^D, x_1^D)).$$

If this condition is fulfilled, we can calculate the fuel cost for the coordination follower with the speed profile that is adapted for platooning with the coordination leader as follows

$$F = d^S f_0(v_S) + d^{Sp} f_0(v_{Sp}) + (d_{e_1}((e_1^S, x_1^S), (e_1^D, x_1^D)) - d^S - d^{Sp}) f_p(v_0). \quad (5.7)$$

If  $F$  is smaller than the fuel consumption that results from traveling alone at a constant speed, it is beneficial that the vehicles platoon. The fuel savings that result from the adapted plan is the information that determines which vehicles should platoon. The algorithms developed in Chapter 6 use  $F$  for selecting from different possible adapted plans.

The results of this section can be summarized as follows. The optimal speed profile of a coordination follower with index 1 to a coordination leader with index 0 consists of three phases with constant speed:  $v_S$  from  $t_1^S$  to  $t^M$ , then  $v_0$  from  $t^M$  to  $t^{Sp}$ , and finally  $v_{Sp}$  from  $t^{Sp}$  to  $t_1^D$ , where the coordination follower is a platoon follower of the coordination leader from time  $t^M$  to  $t^{Sp}$ .

We see that the computation of such adapted plans involves sorting out some details, but this is not inherently difficult. Additional factors such as speed limits, traffic, rests, flexible start and arrival times, etc. can be added. An interesting question is how to go from two vehicles to a whole fleet. In the next chapter, we show how to combine the pairwise plans in a systematic way by using the property of the adapted plans that neither the speed profile nor the fuel consumption of the coordination leader changes.



### 5.3 Summary

This chapter considers that a truck, a coordination follower, follows an adapted vehicle plan so that it meets another truck, a coordination leader, during its journey and they platoon together. The speed profile of the coordination leader is given and not altered. The adapted plans consist of three phases. In the first phase the speed is set to a value so the trucks meet to form a platoon, which is the start of the second phase. During the second phase the trucks platoon. At the end of the second phase the trucks split up and the coordination follower selects a speed that lets the coordination follower arrive on its deadline at its destination. Based on an affine fuel model, an analytical expression of the fuel-optimal speed of the first and the last phase is derived. Taking into account that platooning can only happen on the common part of the routes, we arrive at the fuel-optimal adapted plan.

The computation of such plans can be tedious in more complicated settings such as varying speed limits but it does not pose fundamental challenges. The adapted plans derived here serve as an example and more involved settings will most likely follow the same lines of reasoning. When there are more than two trucks, adapted plans of multiple coordination follower can be combined following the methodology outlined in the following chapter.



## Chapter 6

# Computing Fuel-Efficient Vehicle Plans

THIS chapter presents a systematic way of combining the pairwise plans derived in the previous chapter assigning a plan to each vehicle. In Section 6.1, we introduce a more general notion of default plans and adapted plans compared to the one in Chapter 5. The problem of how to combine such plans into a fuel-efficient combined plan for all vehicles is expressed as a combinatorial optimization problem. Section 6.2 deals with the computation of exact solutions to this problem. The proposed method for exact computation is a branch and bound method. Branch and bound is a way to systematically explore all possible solutions of a problem. By comparing an upper bound of all solutions in a branch with the best solution found so far, it is possible to dismiss entire branches. Such an upper bound for our problem is derived. Two additional results on the structure of the optimal solution are established in order to reduce the search space. Finally, the problem is proven to be NP-hard, which is commonly believed to imply that the exact computation can take very long for some problem instances. This motivates the algorithm to compute a heuristic solution that is presented in Section 6.3. This algorithm can find a good solution efficiently but it is not guaranteed to converge to the best combination of pairwise plans. Similar approaches are often used when dealing with NP-hard problems. Once the pairwise plans are combined into a plan for all vehicles, it is possible to keep fixed which platoons should be formed and adjust the timing when these platoons should be formed and broken apart. Section 6.4 discusses how to do these adjustments in a way that minimizes fuel consumption. Section 6.5 evaluates the results from this chapter using Monte Carlo simulations in a realistic scenario.



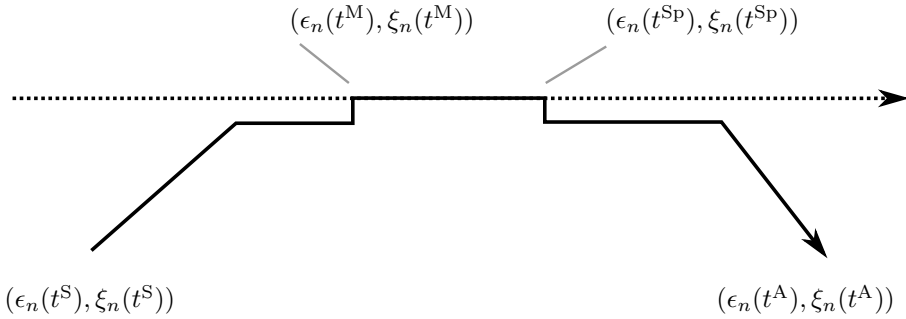


Figure 6.1: Overview of the relevant time instances of the adapted plan. The solid line illustrates the route of the adapted plan  $n$ , and the dashed line the one of the plan that it is adapted to and has index  $m$ . The parallel sections of the line indicate that the trucks share the route, and the section where the lines are on top of each other indicates that the trucks platoon there.

## 6.1 Combining Pairwise Plans to Save Fuel

To begin with, we need to be able to compute what we call a default plan. This is a valid vehicle plan according to Definition 1 with either the lowest possible or fuel optimal constant speed.

**Definition 6** (Default Plan). The default plan is a vehicle plan  $\mathcal{P} = (\mathbf{e}, \mathbf{v}, \hat{\mathbf{t}})$  with speed sequence  $\mathbf{v} = (v_{\text{cd}})$  and time sequence  $\hat{\mathbf{t}} = (t^S, D/v_{\text{cd}})$ . The most fuel optimal speed without platooning  $v_{\text{cd}}$  is computed as

$$v_{\text{cd}} = \underset{v \in (v_{\text{cm}}, v_{\text{max}}]}{\operatorname{argmin}} f_0(v),$$

where  $v_{\text{cm}}$  is the lowest constant speed to arrive before the deadline:

$$v_{\text{cm}} = \max \left( v_{\text{min}}, \frac{D}{t^D - t^S} \right).$$

An adapted plan, as introduced next, is such that the speed sequence  $\mathbf{v}_n$  and time sequence  $\hat{\mathbf{t}}_n$  of a follower truck  $n$  is adapted in a way that allows the follower to platoon during part of its journey with a leader  $m$ . The leader sticks to its default plan, which is important in order to be able to compose these plans. The plan is computed in a way that minimizes the fuel consumption of  $n$ .

**Definition 7** (Adapted Plan). An adapted plan is a vehicle plan  $\mathcal{P}_n = (\mathbf{e}_n, \mathbf{v}_n, \hat{\mathbf{t}}_n)$  adapted to vehicle plan  $\mathcal{P}_m = (\mathbf{e}_m, \mathbf{v}_m, \hat{\mathbf{t}}_m)$ , such that  $(\epsilon_n(t), \xi_n(t)) = (\epsilon_m(t), \xi_m(t))$  for  $t \in [\hat{\mathbf{t}}_n[2], \hat{\mathbf{t}}_n[N_v]]$ .





We denote the merge time as  $t^M = \hat{\mathbf{t}}_n[2]$  and the split time as  $t^{Sp} = \hat{\mathbf{t}}_n[N_v]$ . Truck  $n$  becomes the platoon follower of truck  $m$  at time  $t^M$ , stays platoon follower until  $t^{Sp}$ , when the two trucks separate. This sequence of events occurs only once. Figure 6.1 illustrates the adapted plan. We denote the speed trajectory  $\phi$  corresponding to the speed sequence  $\mathbf{v}$  and the time sequence  $\hat{\mathbf{t}}$  of the adapted vehicle plan of truck  $n$  adapted to truck  $m$  as  $\phi_{n,m}$ .

The fuel consumption of truck  $n$  with its plan adapted to truck  $m$  is modeled as in (3.2). We denote the platoon trajectory of the adapted plan  $\pi_{n,m}(t)$ . We have that  $\pi_{n,m}(t) = 1$  for  $t \in [t^M, t^{Sp}]$  and  $\pi_{n,m}(t) = 0$  for  $t \in [t^S, t^M) \cup [t^{Sp}, t^A)$ . The fuel consumption of  $m$  is not altered by the fact that  $n$  and  $m$  platoon, since  $m$ 's speed trajectory does not change and  $m$  takes the role of a platoon leader. The reduction in fuel consumption that results from  $n$  implementing the adapted plan and not its default plan is  $\Delta F(n, m) = F(\phi_n, \pi_n) - F(\phi_{n,m}, \pi_{n,m})$  where  $\pi_n \equiv 0$ , which is positive if  $n$  adapting to  $m$  saves fuel. If no plan that is adapted to  $m$  exists for  $n$ , we define  $\Delta F(n, m) = 0$ . There might exist no adapted plan because the routes do not overlap or because the constraint on the maximum speed in conjunction with the arrival deadline makes it impossible for the trucks to form a platoon.

We now compute  $\Delta F$  for all 2-permutations in  $\mathcal{N}_c$ . We are only interested in adapted plans that save fuel, i.e., for which  $\Delta F$  is positive. We can conveniently collect this information in a weighted graph that we call the coordination graph.

**Definition 8** (Coordination Graph). The coordination graph is a weighted directed graph  $\mathcal{G}_c = (\mathcal{N}_c, \mathcal{E}_c, \Delta F)$ . Recall that  $\mathcal{N}_c$  represents the trucks.  $\mathcal{E}_c \subseteq \mathcal{N}_c \times \mathcal{N}_c$  is a set of edges, and  $\Delta F : \mathcal{E}_c \rightarrow \mathbb{R}^+$  are edge weights, such that there is an edge  $(n, m) \in \mathcal{E}_c$ , if the adapted plan of  $n$  to  $m$  saves fuel compared to  $i$ 's default plan, i.e.,  $\mathcal{E}_c = \{(i, j) \in \mathcal{N}_c \times \mathcal{N}_c : \Delta F(i, j) > 0, i \neq j\}$ .

Furthermore, we introduce the set of in-neighbors of a node  $n \in \mathcal{N}_c$  as

$$\mathcal{N}_n^i = \{i \in \mathcal{N}_c : (i, n) \in \mathcal{E}_c\},$$

and the set of out-neighbors  $n$  as

$$\mathcal{N}_n^o = \{i \in \mathcal{N}_c : (n, i) \in \mathcal{E}_c\}.$$

We define the maximum over an empty set to be zero, i.e.,  $\max_{i \in \emptyset} (\cdot) = 0$ .

With these definitions, we are ready to formulate the problem of finding a fuel optimal set of coordination leaders  $\mathcal{N}_1$ .

**Problem 1.** Given as input a coordination graph  $\mathcal{G}_c = (\mathcal{N}_c, \mathcal{E}_c, \Delta F)$  find a subset  $\mathcal{N}_1 \subset \mathcal{N}_c$  of nodes that maximizes

$$f_{ce}(\mathcal{N}_1) = \sum_{i \in \mathcal{N}_c \setminus \mathcal{N}_1} \max_{j \in \mathcal{N}_i^o \cap \mathcal{N}_1} \Delta F(i, j). \quad (6.1)$$



The coordination leaders select their default plans. The remaining assignments, called coordination followers, select their plans adapted to the coordination leader that yields the largest fuel savings  $\Delta F(n, m)$ . Since the selection of adapted plans does not alter the speed trajectories of the coordination leaders, several coordination followers can select the same coordination leader without affecting the fuel savings that result from this adaptation, potentially resulting in platoons of more than two vehicles. The objective function  $f_{ce}(\mathcal{N}_1)$  equals the sum of all these fuel savings. If  $(n, m) \in \mathcal{E}_c$  with  $n \in \mathcal{N}_c \setminus \mathcal{N}_1$  and  $m = \arg \max_{m \in \mathcal{N}_c^o \cap \mathcal{N}_1} \Delta F(n, m)$ , we say that  $n$  is the coordination follower of  $m$  and  $m$  is the coordination leader of  $n$ . If  $m$  has no out-neighbor in  $\mathcal{N}_1$ , then  $\max_{m \in (\mathcal{N}_c^o \cap \mathcal{N}_1)} \Delta F(n, m) = \max_{m \in \emptyset} \Delta F(n, m) = 0$ .

At this point, we have a combinatorial problem, whose solution allows us to group transport assignments in a fuel-efficient way. All continuous optimization is contained in the adapted plans. Since an adapted plan only involves computing the speed profile for one vehicle, deriving such adapted plans is a task that is possible to handle, as demonstrated in Chapter 5. This simplification comes with a price on the fuel-savings that can be achieved. In Section 6.4, we address this problem to some extent by jointly optimizing the speed profile of each cluster. Furthermore, the envisioned system repeats the optimization frequently. A coordination follower that joins a platoon during the first part of its journey can, in a later optimization, become coordination follower of another truck and platoon for the remaining part of its journey.

One disadvantage of the approach presented in this section is that each truck can only join one platoon. This is however somewhat mitigated by the frequent re-planning envisioned for this system. At a later point in time, it might turn out more beneficial for a truck to leave its current platoon and join another one.

In the remainder of this chapter, we study Problem 1. We derive a branch and bound algorithm to compute optimal solutions and establish that it is NP-hard.

## 6.2 Exact Computation

Problem 1 is a combinatorial optimization problem. A common technique to solve such problems is the branch and bound technique [26]. Branch and bound is a systematic way to search for the optimal solution of a discrete optimization problem. It constructs a binary search tree whose leaves cover all possible values of the optimization variables. However, it can be possible to leave entire branches of the tree unexplored. A branch can be dismissed if it cannot contain any solution that is better than the best known solution so far.

Consider Algorithm 1, in which we tackle Problem 1 using the aforementioned branch and bound technique. Each node in the search tree encodes a subset of solutions, with the root encoding all solutions and a leaf exactly one. The solutions encoded by a node are characterized by two sets  $\bar{\mathcal{N}}_l$  and  $\bar{\mathcal{N}}_f$ . The set  $\bar{\mathcal{N}}_l$  contains all nodes that are assigned to be coordination leaders, i.e., elements of  $\mathcal{N}_1$ . The set  $\bar{\mathcal{N}}_f$  contains all nodes that are assigned *not* to be coordination leaders, i.e., that are



not elements of  $\mathcal{N}_1$ . For the remaining nodes in  $\mathcal{N}_c$  no decision has been made. At the root node, these sets are empty, and at every branching a node that is not of  $\mathcal{N}_1$  or  $\mathcal{N}_f$  is added to either  $\mathcal{N}_1$  or  $\mathcal{N}_f$ . A leaf is reached if all nodes are assigned to either  $\mathcal{N}_1$  or  $\mathcal{N}_f$ . Figure 6.2 shows an example of the search tree that can be traversed by Algorithm 1.

---

**Algorithm 1** The branch and bound algorithm to compute an optimal set of coordination leaders. The displayed version makes use of the result that at most half the assignments are coordination leaders. The remaining heuristics described in the chapter can be applied before adding a new node to  $\mathcal{Q}$ .

---

**Input:**  $\mathcal{G}_c$

**Output:**  $\mathcal{N}_1$

$\mathcal{N}_1 \leftarrow \emptyset$

$\mathcal{Q} \leftarrow \{(\emptyset, \emptyset)\}$

**while**  $\mathcal{Q} \neq \emptyset$  **do**

    Retrieve  $(\bar{\mathcal{N}}_1, \bar{\mathcal{N}}_f)$  from  $\mathcal{Q}$

**if**  $f_{ce}(\bar{\mathcal{N}}_1) > f_{ce}(\mathcal{N}_1)$  **then**

$\mathcal{N}_1 \leftarrow \bar{\mathcal{N}}_1$

**end if**

**if**  $\mathcal{N}_c \setminus (\bar{\mathcal{N}}_1 \cup \bar{\mathcal{N}}_f) \neq \emptyset$  **then**

        Select  $n \in \mathcal{N}_c \setminus (\bar{\mathcal{N}}_1 \cup \bar{\mathcal{N}}_f)$

**if**  $\bar{f}(\bar{\mathcal{N}}_1 \cup \{n\}, \bar{\mathcal{N}}_f) > f_{ce}(\mathcal{N}_1) \wedge |\bar{\mathcal{N}}_1| + 1 \leq \lfloor |\mathcal{N}_c|/2 \rfloor$  **then**

            Add  $(\bar{\mathcal{N}}_1 \cup \{n\}, \bar{\mathcal{N}}_f)$  to  $\mathcal{Q}$

**end if**

**if**  $\bar{f}(\bar{\mathcal{N}}_1, \bar{\mathcal{N}}_f \cup \{n\}) > f_{ce}(\mathcal{N}_1)$  **then**

            Add  $(\bar{\mathcal{N}}_1, \bar{\mathcal{N}}_f \cup \{n\})$  to  $\mathcal{Q}$

**end if**

**end if**

**end while**

---

In order to dismiss a branch, we keep track of the best solution  $\mathcal{N}_1$  found so far. We compare the best solution to an upper bound on the objective that can be achieved by the branch to be dismissed. If the branch contains no solution that is better than the best solution found so far, the branch can be dismissed.

The upper bound  $\bar{f}(\bar{\mathcal{N}}_1, \bar{\mathcal{N}}_f)$  is based on the intuition to assign every truck for which no decision has been made its best coordination leader from the certain coordination leaders  $\bar{\mathcal{N}}_1$  or the potential coordination leaders  $\mathcal{N}_c \setminus (\bar{\mathcal{N}}_1 \cup \bar{\mathcal{N}}_f)$ . Furthermore, the bound neglects, as far as the nodes  $\mathcal{N}_c \setminus (\bar{\mathcal{N}}_1 \cup \bar{\mathcal{N}}_f)$  are concerned, that coordination leaders do not contribute to the sum that defines  $f_{ce}$ .

**Proposition 7.** Let  $\mathcal{N}_1, \bar{\mathcal{N}}_1, \bar{\mathcal{N}}_f \subseteq \mathcal{N}_c$  be sets that fulfill  $\bar{\mathcal{N}}_1 \subseteq \mathcal{N}_1$ ,  $\bar{\mathcal{N}}_f \cap \mathcal{N}_1 = \emptyset$ ,  $\bar{\mathcal{N}}_u = \mathcal{N}_c \setminus (\bar{\mathcal{N}}_1 \cup \bar{\mathcal{N}}_f)$  and define

$$\bar{f}(\bar{\mathcal{N}}_1, \bar{\mathcal{N}}_f) = \sum_{i \in \bar{\mathcal{N}}_f \cup \bar{\mathcal{N}}_u} \max_{j \in \mathcal{N}_i^o \cap (\bar{\mathcal{N}}_1 \cup \bar{\mathcal{N}}_u)} \Delta F(i, j).$$



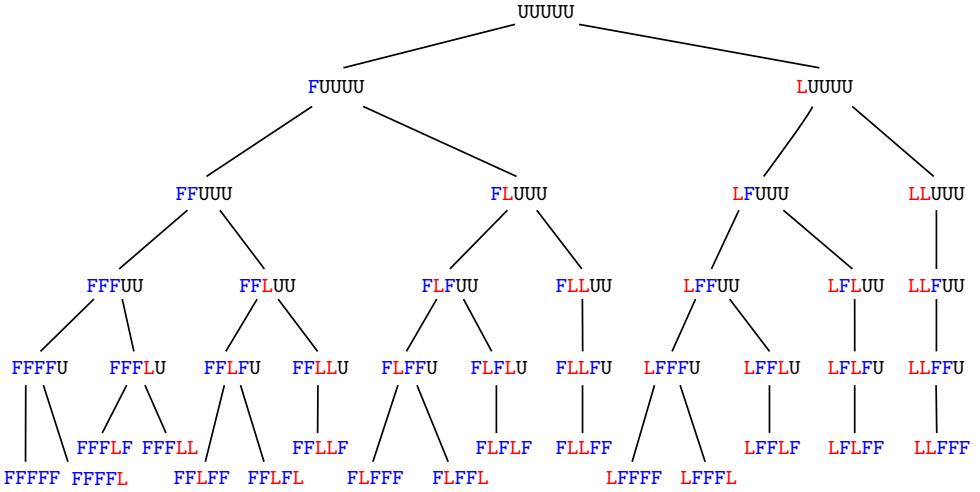


Figure 6.2: A search tree for five assignments. The membership of the assignments is indicated with letters. The letter **U** indicates that the assignment belongs to  $\mathcal{N}_c \setminus (\bar{\mathcal{N}}_1 \cup \bar{\mathcal{N}}_f)$ , the letter **F** indicates that the assignment belongs to  $\bar{\mathcal{N}}_f$ , and the letter **L** indicates that the assignment belongs to  $\bar{\mathcal{N}}_1$ . In this example, no branches are dismissed due to the upper bound  $\bar{f}$ . Solutions with more than two coordination leaders are not explored since, according to Proposition 8, there must be an optimal solution with at most two coordination leaders.

The value of  $f_{ce}(\mathcal{N}_1)$  as defined in (6.1) is upper bounded by

$$f_{ce}(\mathcal{N}_1) \leq \bar{f}(\bar{\mathcal{N}}_1, \bar{\mathcal{N}}_f).$$

*Proof.* We have that

$$\begin{aligned} \bar{\mathcal{N}}_f \cup \bar{\mathcal{N}}_u &= \bar{\mathcal{N}}_f \cup (\mathcal{N}_c \setminus (\bar{\mathcal{N}}_1 \cup \bar{\mathcal{N}}_f)) \\ &= \bar{\mathcal{N}}_f \cup (\mathcal{N}_c \setminus \bar{\mathcal{N}}_1) \\ &\supseteq \mathcal{N}_c \setminus \bar{\mathcal{N}}_1 \\ &\supseteq \mathcal{N}_c \setminus \mathcal{N}_1 \end{aligned}$$

and

$$\begin{aligned} \bar{\mathcal{N}}_1 \cup \bar{\mathcal{N}}_u &= \bar{\mathcal{N}}_1 \cup (\mathcal{N}_c \setminus (\bar{\mathcal{N}}_1 \cup \bar{\mathcal{N}}_f)) \\ &= \bar{\mathcal{N}}_1 \cup (\mathcal{N}_c \setminus \bar{\mathcal{N}}_f) \\ &\supseteq \mathcal{N}_c \setminus \bar{\mathcal{N}}_f \\ &\supseteq \mathcal{N}_1 \end{aligned}$$



and hence

$$\begin{aligned}
f_{ce}(\mathcal{N}_1) &= \sum_{i \in \mathcal{N}_c \setminus \mathcal{N}_1} \max_{j \in \mathcal{N}_i^o \cap \mathcal{N}_1} \Delta F(i, j) \\
&\leq \sum_{i \in \bar{\mathcal{N}}_f \cup \bar{\mathcal{N}}_u} \max_{j \in \mathcal{N}_i^o \cap \mathcal{N}_1} \Delta F(i, j) \\
&\leq \sum_{i \in \bar{\mathcal{N}}_f \cup \bar{\mathcal{N}}_u} \max_{j \in \mathcal{N}_i^o \cap (\mathcal{N}_1 \cup \bar{\mathcal{N}}_u)} \Delta F(i, j),
\end{aligned}$$

where we used that  $\Delta F(i, j) > 0$  for all  $(i, j) \in \mathcal{E}_c$ .  $\square$

We can improve the performance of the algorithm by establishing results on the structure of the optimal solution. Every time a branch does not contain at least one solution that matches this structure, the branch can be dismissed. The first result on the structure of the optimal solution is an upper bound on the maximum number of coordination leaders, i.e., on the cardinality of the optimal  $\mathcal{N}_1$ . It states that there is an optimal solution with at most  $\lfloor |\mathcal{N}_c|/2 \rfloor$  coordination leaders.

**Proposition 8.** *There exists an optimal solution  $\mathcal{N}_1$  to Problem 1 with  $|\mathcal{N}_1| \leq \lfloor |\mathcal{N}_c|/2 \rfloor$ .*

*Proof.* First of all, we note that Problem 1 is an unconstrained optimization problem and the optimization argument belongs to a finite set. Therefore, a solution always exists.

The existence of an optimal solution  $\mathcal{N}_1$  with  $|\mathcal{N}_1| \leq \lfloor |\mathcal{N}_c|/2 \rfloor$  is proven by contradiction. Assume that every optimal solution  $\mathcal{N}_1$  to Problem 1 fulfills  $|\mathcal{N}_1| > \lfloor |\mathcal{N}_c|/2 \rfloor$ . Then  $|\mathcal{N}_c \setminus \mathcal{N}_1| < |\mathcal{N}_1|$ . Hence, there is least at one  $n \in \mathcal{N}_1$  for which there is no  $i \in \mathcal{N}_c \setminus \mathcal{N}_1$  for which  $n = \arg \max_{j \in \mathcal{N}_i^o \cap \mathcal{N}_1} \Delta F(i, j)$ . Thus,  $n$  can be removed from  $\mathcal{N}_1$  without decreasing  $f_{ce}(\mathcal{N}_1)$ . This reasoning can be repeatedly applied until  $|\mathcal{N}_1| \leq \lfloor |\mathcal{N}_c|/2 \rfloor$  with  $f_{ce}(\mathcal{N}_1)$  no smaller than the optimal  $\mathcal{N}_1$ . Thus, the smaller  $\mathcal{N}_1$  is as well an optimal solution to Problem 1. This, however, contradicts the assumption.  $\square$

This proposition helps when computing an optimal solution since coordination leaders sets with cardinality larger than  $\lfloor |\mathcal{N}_c|/2 \rfloor$  do not have to be considered.

The next result that helps prune the search tree is that a node is either a coordination leader itself or at least one node in its two-hop out-neighbor set is a coordination leader. To this end, we define the set of two-hop out-neighbors of a node  $n \in \mathcal{N}_c$  as

$$\mathcal{N}_n^{2o} = \mathcal{N}_n^o \cup \bigcup_{i \in \mathcal{N}_n^o} \mathcal{N}_i^o.$$

Figure 6.3 shows an example of the set  $\mathcal{N}_n^{2o} \cup \{n\}$ .

**Proposition 9.** *Let  $\mathcal{N}_1$  be an optimal solution to Problem 1. For each  $n \in \mathcal{N}_c$  with  $\mathcal{N}_n^{2o} \neq \emptyset$ , we have that  $\mathcal{N}_1 \cap (\mathcal{N}_n^{2o} \cup \{n\}) \neq \emptyset$ .*



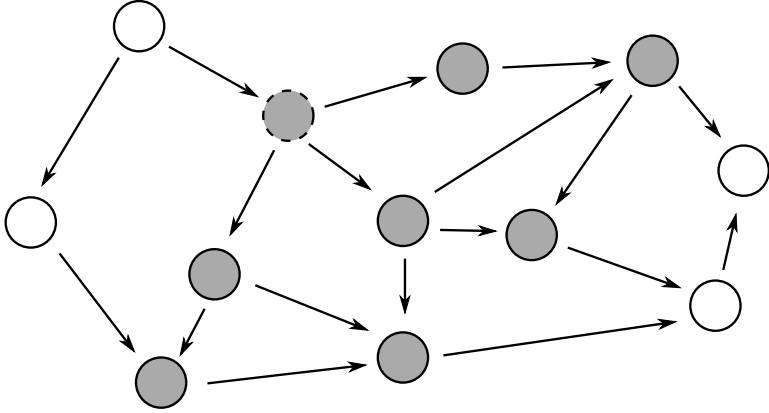


Figure 6.3: Example of a node's two-hop out-neighbor set. The gray circles that have a solid line represent the two-hop out-neighbor set of the node drawn as a circle with a dashed line. An optimal solution  $\mathcal{N}_1$  to Problem 1 contains at least one of the gray-filled nodes.

*Proof.* Assume  $\mathcal{N}_n^{2o} \neq \emptyset$ . If  $n \in \mathcal{N}_1$ , then clearly  $\mathcal{N}_1 \cap (\mathcal{N}_n^{2o} \cup \{n\}) \neq \emptyset$ . If  $n \notin \mathcal{N}_1$  and  $\mathcal{N}_1 \cap \mathcal{N}_n^{2o} = \emptyset$ , then we can add any node in  $\mathcal{N}_n^{2o}$  to  $\mathcal{N}_1$  and increase  $f_{ce}$ , which contradicts the assumption that  $\mathcal{N}_1$  is a solution to Problem 1. This is because for any  $i \in \mathcal{N}_n^{2o}$  it holds that  $\max_{j \in \mathcal{N}_i^o \cap \mathcal{N}_1} \Delta F(i, j) = \max_{j \in \emptyset} \Delta F(i, j) = 0$ , but  $\max_{j \in \mathcal{N}_n^o} \Delta F(n, j) > 0$ .  $\square$

At every node in the search, we can compute if any solution in the corresponding branch can be an optimal solution using Proposition 9. This is the case when a node and its two-hop neighbor set is fully contained in  $\tilde{\mathcal{N}}_f$ . If that is the case, there is no need to further explore the branch in question.

Proposition 9 can be used to compute a lower bound on the number of coordination leaders in an optimal solution. It can be easier to test whether a branch contains solutions with enough coordination leaders, i.e., whether  $|\tilde{\mathcal{N}}_f| + |\tilde{\mathcal{N}}_u|$  is greater than or equal to the lower bound, compared to using Proposition 9 directly as outlined above.

Proposition 9 tells us that each union of a node and its two-hop out-neighbors contains at least one coordination leader, unless that node's two-hop out-neighbor set is empty. However, in most cases these sets overlap and one coordination leader is contained in the two-hop out-neighbor sets of several nodes. We can, nevertheless, select some of these sets so that the selected sets mutually do not intersect. A coordination leader cannot be contained in two of these sets.

**Proposition 10.** *Let  $\mathcal{N}_1$  be an optimal solution to Problem 1 and let the set of sets  $\underline{\mathcal{D}} \subset \{\{n\} \cup \mathcal{N}_n^{2o} : n \in \mathcal{N}_c, \mathcal{N}_n^{2o} \neq \emptyset\}$  be defined such that any two elements of*



$\mathcal{D}_1, \mathcal{D}_2 \in \underline{\mathcal{D}}$  have zero intersection  $\mathcal{D}_1 \cap \mathcal{D}_2 = \emptyset$ . Then it holds that  $|\mathcal{N}_1| \geq |\underline{\mathcal{D}}|$ , and for every  $d \in \underline{\mathcal{D}}$ , it holds that  $d \cap \mathcal{N}_1 \neq \emptyset$ .

*Proof.* From Proposition 9 it follows that  $\mathcal{D}_1 \cap \mathcal{N}_1 \neq \emptyset$  and  $\mathcal{D}_2 \cap \mathcal{N}_1 \neq \emptyset$ . Since  $\mathcal{D}_1 \cap \mathcal{D}_2 = \emptyset$  it holds also that  $(\mathcal{D}_1 \cap \mathcal{N}_1) \cap (\mathcal{D}_2 \cap \mathcal{N}_1) = \emptyset$ . This holds for any two elements  $\mathcal{D}_1, \mathcal{D}_2$  in  $\underline{\mathcal{D}}$ . Thus, every element of  $\underline{\mathcal{D}}$  contains at least one element of  $\mathcal{N}_1$ .

Since this holds for any two elements in  $\underline{\mathcal{D}}$ , there is at least one unique element in  $\mathcal{N}_1$  for every element in  $\underline{\mathcal{D}}$ , i.e., there are at least as many elements in  $\mathcal{N}_1$  as in  $\underline{\mathcal{D}}$ .  $\square$

The set  $\underline{\mathcal{D}}$  is an independent subset of the set  $\{\{n\} \cup \mathcal{N}_n^{2o} : n \in \mathcal{N}_c, \mathcal{N}_n^{2o} \neq \emptyset\}$ . Maximal independent sets, i.e., sets  $\underline{\mathcal{D}}$  where no element from  $\{\{n\} \cup \mathcal{N}_n^{2o} : n \in \mathcal{N}_c, \mathcal{N}_n^{2o} \neq \emptyset\}$  can be added without violating that any two subsets have non-zero intersection, can be computed with a greedy algorithm. The problem of finding the maximum independent set—this is, the independent set with largest cardinality—is however NP-hard [78], so finding the largest value for the bound might not always be feasible.

The quality of this bound depends on the graph. Consider Figure 6.4. The optimal solution with the graph shown on the left side will have 4 or 5 coordination leaders, namely the middle layer of nodes. Adding the top node to the set of coordination leaders does not change the objective. All sets  $(\mathcal{N}_n^{2o} \cup \{n\})$  include the node on the top of the graph, and therefore  $|\underline{\mathcal{D}}| = 1$  for any choice of  $\underline{\mathcal{D}}$ . When the weights of the edges from the middle layer to the top node are changed in a way so that they are larger than the edges from the bottom layer to the middle layer, then the top node becomes the only coordination leader and the bound is tight. On the other hand, the graph shown on the right-hand side of the figure will admit a tight bound regardless of the weights. For every pair of nodes that is connected by an edge, the top node becomes coordination leader. These pairs of nodes are the sets  $(\mathcal{N}_n^{2o} \cup \{n\})$ , which are all independent.

An obvious property to investigate when having developed an algorithm to solve a combinatorial optimization problem is the algorithm's worst case complexity. Like many combinatorial optimization problems, Problem 1 can be shown to be NP-hard. This means it is unlikely, even though not yet proven, that there can be an algorithm that solves any instance of the problem efficiently, meaning that the number of computation steps needed to compute the result cannot be upper bounded by a polynomial evaluated on the size of the input. The size of the input is measured in terms of number of edges and nodes in the coordination graph.

**Proposition 11.** *Problem 1 is NP-hard.*

*Proof.* We show the result by reduction of the optimization version of the set covering problem to Problem 1. The optimization version of the set problem covering is well known to be NP-hard. Reduction to a known hard problem is a common



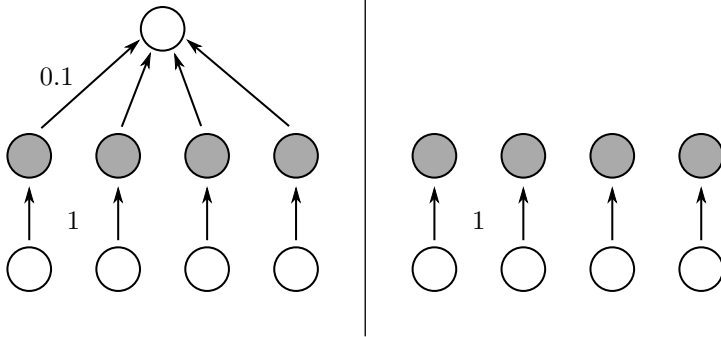


Figure 6.4: Two different coordination graphs that illustrate how the usefulness of Proposition 10 depends on the coordination graph. An optimal solution of Problem 1 on both graphs has at least 4 coordination leaders. Proposition 10 shows that an optimal solution on the left graph has at least one coordination leader whereas an optimal solution on the right graph has at least 4 coordination leaders, which are drawn as gray-filled circles.

proof technique for this kind of result [28]. We do this by constructing a coordination graph  $\mathcal{G}_c$  for which there is a one-to-one correspondence between coordination leaders and selected sets for the cover. Then we show that the minimum number of leaders that corresponds to a set cover gives the maximum value for  $f_{ce}$ .

Consider the following set covering problem. We have a finite set  $\mathcal{U}$ . Furthermore, let  $\mathcal{S}_u$  be a family of subsets of  $\mathcal{U}$  with  $\bigcup_{\mathcal{S} \in \mathcal{S}_u} \mathcal{S} = \mathcal{U}$ . The problem is to find the smallest number of subsets in  $\mathcal{S}_u$  whose union is  $\mathcal{U}$ .

We construct the coordination graph as the one shown in Figure 6.5. We introduce a node for each element in  $\mathcal{U}$ . We denote the set of these nodes with  $\mathcal{N}_3$  and let  $\mu_3 : \mathcal{U} \rightarrow \mathcal{N}_3$  be a bijective mapping from the elements in  $\mathcal{U}$  to the nodes in  $\mathcal{N}_3$ . We introduce a node for each element in  $\mathcal{S}_u$ . We denote the set of these nodes with  $\mathcal{N}_2$  and let  $\mu_2 : \mathcal{S}_u \rightarrow \mathcal{N}_2$  be a bijective mapping from elements in  $\mathcal{S}_u$  to nodes in  $\mathcal{N}_2$ . Consider a node  $n_2 \in \mathcal{N}_2$  that corresponds to the element  $\mathcal{S} \in \mathcal{S}_u$ . The in-neighbors of  $n_2$  are  $\mathcal{N}_{n_2}^i = \{\mu_3(\mathcal{S}) : \mathcal{S} \in \mu_2^{-1}(n_2)\}$ . The weight of the corresponding edges is 1. We introduce an additional node  $N_1$ . There is an edge from each node in  $\mathcal{N}_2$  to  $N_1$  with weight 0.5. Clearly, this reduction is linear in the size of the input  $\mathcal{U}, \mathcal{S}_u$ .

Since  $N_1$  has no out-neighbors, its membership in  $\mathcal{N}_1$  can only increase  $f_{ce}(\mathcal{N}_1)$ . Since all nodes in  $\mathcal{N}_3$  have no in-neighbors, adding a node in  $\mathcal{N}_3$  to  $\mathcal{N}_1$  can only decrease  $f_{ce}(\mathcal{N}_1)$ . Thus, the problem of finding the optimal  $\mathcal{N}_1$  reduces to finding which nodes in  $\mathcal{N}_2$  belong to  $\mathcal{N}_1$ . In the optimal solution, each node in  $\mathcal{N}_3$  has at least one out-neighbor in  $\mathcal{N}_1$ . Otherwise we could add any out-neighbor of that node to  $\mathcal{N}_1$  and increase  $f_{ce}(\mathcal{N}_1)$  by at least 0.5. Therefore,  $\{\mu_2^{-1}(n) : n \in \mathcal{N}_1 \cap \mathcal{N}_2\}$





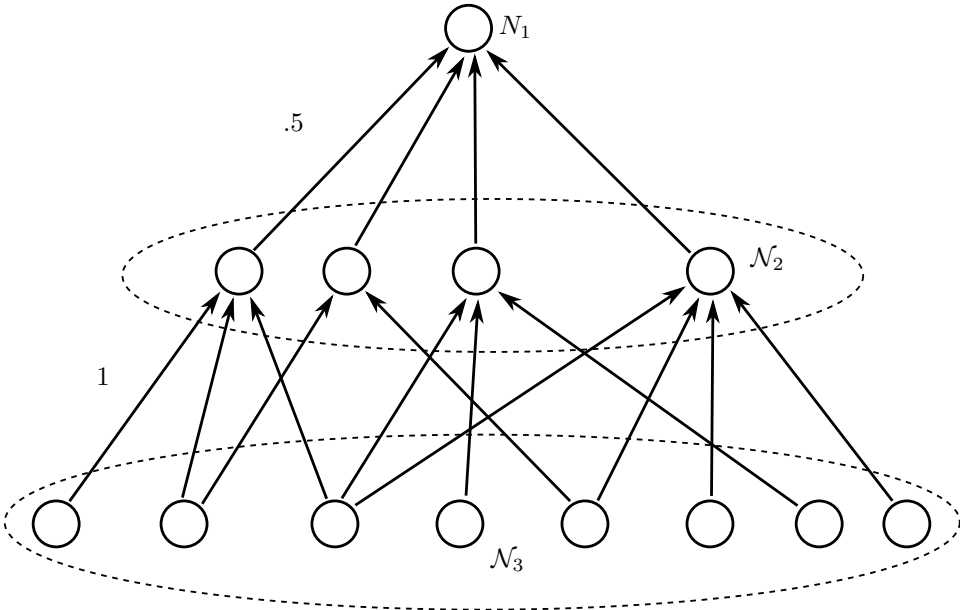


Figure 6.5: Illustration of the graph used to prove that Problem 1 is NP-hard.

is a set cover of  $\mathcal{U}$ . Otherwise there would be  $u \in \mathcal{U}$  such that there is no  $\mathcal{S} \in \{\mu_2^{-1}(n) : n \in \mathcal{N}_1 \cap \mathcal{N}_2\}$  with  $u \in \mathcal{S}$ . If such a  $u$  existed,  $\mu_3(u)$  would be a node with no out-neighbor in  $\mathcal{N}_1 \cap \mathcal{N}_2$ . Furthermore, let  $\bar{\mathcal{S}}_u \subseteq \mathcal{S}_u$  be a set cover of  $\mathcal{U}$ . Then  $\{\mu_2(\mathcal{S}) : \mathcal{S} \in \bar{\mathcal{S}}_u\}$  has the property that  $\bigcup_{n \in \{\mu_2(\mathcal{S}) : \mathcal{S} \in \bar{\mathcal{S}}_u\}} \mathcal{N}_n^i = \mathcal{N}_3$ , so any set cover has the property that all nodes in  $\mathcal{N}_3$  have at least one out-neighbor in  $\mathcal{N}_1$ . Each node in  $\mathcal{N}_2$  contributes with 0.5 to the objective if it is not in  $\mathcal{N}_1$ . Therefore, the optimal  $\mathcal{N}_1$  contains a minimum number of nodes from  $\mathcal{N}_2$  such that every node in  $\mathcal{N}_3$  has at least one out-neighbor in  $\mathcal{N}_1 \cap \mathcal{N}_2$ . Since any  $\mathcal{N}_1 \cap \mathcal{N}_2$  that fulfills this property maps to a set cover  $\bar{\mathcal{S}}_u$  and vice versa, and since  $|\mathcal{N}_1 \cap \mathcal{N}_2| = |\bar{\mathcal{S}}_u|$ , we have that  $\bar{\mathcal{S}}_u$  is the solution to the set covering problem. Thus, the NP-hard set-covering problem can be reduced to Problem 1, which shows that Problem 1 is NP-hard.  $\square$

Exact solutions to NP-hard problems can be hard to compute, which is why heuristic and approximate solutions are often used. These algorithms compute good solutions that are not necessarily optimal in a computationally efficient way. A heuristic algorithm for Problem 1 is developed in the next section.



### 6.3 Heuristic Computation

In this section we present an algorithm that computes heuristic solutions to Problem 1. Motivated by the result that Problem 1 is NP-hard, we apply an iterative strategy that converges to a local maximum.

Consider Algorithm 2. The input is a coordination graph  $\mathcal{G}_c$  and the output is a set of coordination leaders  $\mathcal{N}_1$ . Initially  $\mathcal{N}_1$  is an empty set. In each iteration, a node  $n \in \mathcal{N}_c$  is selected for which the objective function  $f_{ce}$  is increased if it is added to  $\mathcal{N}_1$  or removed from  $\mathcal{N}_1$ , and  $\mathcal{N}_1$  is updated accordingly. The difference in  $f_{ce}$  when adding or removing a node in  $\mathcal{N}_c$  to or from the set of coordination leaders  $\mathcal{N}_1$  is given by a function  $\Delta u$ . The algorithm iterates until no further increase of  $f_{ce}$  is possible.

The function  $\Delta u$  that measures how much is gained from switching whether  $n$  belongs to  $\mathcal{N}_1$  is defined as follows:

$$\Delta u(n, \mathcal{N}_1) = \begin{cases} f_{ce}(\mathcal{N}_1 \setminus \{n\}) - f_{ce}(\mathcal{N}_1) & \text{if } n \in \mathcal{N}_1 \\ f_{ce}(\mathcal{N}_1 \cup \{n\}) - f_{ce}(\mathcal{N}_1) & \text{otherwise.} \end{cases} \quad (6.2)$$

If  $n \notin \mathcal{N}_1$ , we get

$$f_{ce}(\mathcal{N}_1 \cup \{n\}) - f_{ce}(\mathcal{N}_1) = \sum_{i \in \mathcal{N}_n^i \setminus \mathcal{N}_1} \left( \max_{j \in \mathcal{N}_i^o \cap (\mathcal{N}_1 \cup \{n\})} \Delta F(i, j) - \max_{j \in \mathcal{N}_i^o \cap \mathcal{N}_1} \Delta F(i, j) \right) - \max_{i \in \mathcal{N}_n^o \cap \mathcal{N}_1} \Delta F(n, i).$$

The sum over  $i$  covers nodes that can select  $n$  as their new coordination leader. The last summand accounts for  $n$  possibly not being a coordination follower any longer.

If  $n \in \mathcal{N}_1$ , we get

$$f_{ce}(\mathcal{N}_1 \setminus \{n\}) - f_{ce}(\mathcal{N}_1) = \sum_{i \in \mathcal{N}_n^i \setminus \mathcal{N}_1} \left( \max_{j \in \mathcal{N}_i^o \cap (\mathcal{N}_1 \setminus \{n\})} \Delta F(i, j) - \max_{j \in \mathcal{N}_i^o \cap \mathcal{N}_1} \Delta F(i, j) \right) + \max_{i \in \mathcal{N}_n^o \cap (\mathcal{N}_1 \setminus \{n\})} \Delta F(n, i).$$

The sum over  $i$  covers nodes that can have  $n$  as their coordination leader before the change. The last summand accounts for  $n$  possibly becoming a coordination follower.

In this paper, we consider two methods to select  $n$  from the set  $\{\bar{n} \in \mathcal{N}_c : \Delta u(\bar{n}, \mathcal{N}_1) > 0\}$ . The first method is to select  $n$  in a greedy manner according to  $n = \arg \max_{\bar{n} \in \mathcal{N}_c} \Delta u(\bar{n}, \mathcal{N}_1)$ . The second method is to choose  $n$  randomly with equal probability from the set  $\{\bar{n} \in \mathcal{N}_c : \Delta u(\bar{n}, \mathcal{N}_1) > 0\}$ .

Algorithm 2 is guaranteed to converge in finite time. This is due to the number of possible subsets of  $\mathcal{N}_c$  being finite and thus the number possible assignments of  $\mathcal{N}_1$  is finite. In every iteration  $f_{ce}(\mathcal{N}_1)$  strictly increases, which means that  $\mathcal{N}_1$



---

**Algorithm 2** Iterative Algorithm to compute the set of coordination leaders  $\mathcal{N}_1$ .

---

**Input:**  $\mathcal{G}_c$

**Output:**  $\mathcal{N}_1$

$\mathcal{N}_1 \leftarrow \emptyset$

**while**  $\{\bar{n} \in \mathcal{N}_c : \Delta u(\bar{n}, \mathcal{N}_1) > 0\} \neq \emptyset$  **do**

Select  $n \in \{\bar{n} \in \mathcal{N}_c : \Delta u(\bar{n}, \mathcal{N}_1) > 0\}$

**if**  $n \in \mathcal{N}_1$  **then**

$\mathcal{N}_1 \leftarrow \mathcal{N}_1 \setminus \{n\}$

**else**

$\mathcal{N}_1 \leftarrow \mathcal{N}_1 \cup \{n\}$

**end if**

**end while**

---

changes in every iteration and the same assignment for  $\mathcal{N}_1$  never reoccurs. So in the worst case Algorithm 2 iterates over all subsets of  $\mathcal{N}_c$  before termination. It is also possible to interrupt the algorithm before termination and use the value of  $\mathcal{N}_1$  at this point in the execution. It is easy to see that a coordination leader set  $\mathcal{N}_1$  computed by Algorithm 2 fulfills the condition on the optimal solution stated in Proposition 9, i.e., that every union of a node and its two-hop out-neighbors contains at least one coordination leader.

Algorithm 2 can be efficient. Note for instance that the function  $\Delta u$  can be computed based on the sub-graph induced by the one- and two-hop neighbors of  $n$  only. This means that the average complexity of computing  $\Delta u$  is a function of the average node degree but not of the number of nodes in the coordination graph. Furthermore, if a node is added to or removed from  $\mathcal{N}_1$ , then only the  $\Delta u$  for the two-hop neighbors needs to be recomputed.

Simulations suggest that selecting  $n$  in a greedy or a random manner makes little difference for the quality of the computed solution. However, greedy node selection tends to lead to less iterations of the algorithm and is thus better suited for a serial implementation. Random node selection might be preferable for a parallel implementation due to the reduced need for synchronization.

Having computed the set of coordination leaders, there is immediately a vehicle plan for each truck. These plans are jointly optimized as discussed in the following section.

## 6.4 Joint Vehicle Plan Optimization

In this section we derive how to jointly optimize the vehicle plans that are selected by Algorithm 2. We do this by formulating a convex optimization problem with linear constraints for a group consisting of a coordination leader and its coordination followers. Hereby, the timing when platoons are assembled and broken apart is adjusted while the locations where this happens is not changed. Trucks that are



not matched to any coordination leader or are not coordination leaders themselves just follow their default plans and are not considered in this section.

Consider a coordination leader  $n_1 \in \mathcal{N}_1$  and its followers

$$\mathcal{N}_{\text{fl},n_1} = \{n \in \mathcal{N}_c \setminus \mathcal{N}_1 : n_1 = \arg \max_{i \in \mathcal{N}_1 \cap \mathcal{N}_n^o} \Delta F(n, i)\}.$$

We construct an ordered set of time instances  $\mathbf{t} = (\mathbf{t}[1], \mathbf{t}[2], \dots)$ . This set contains the start time and the arrival deadline of the coordination leader, and the merge times and the split times of its followers. We divide the distance traveled by the leader from start to destination at these time instances and get the distances

$$\mathbf{W}_{n_1}[i] = \frac{\mathbf{t}[i+1] - \mathbf{t}[i]}{v_{\text{cd}}},$$

between these points, where  $v_{\text{cd}}$  is the speed of the leader according to its default plan. These are the distances between the points where coordination followers join or leave the platoon. Similarly, for a coordination follower  $n \in \mathcal{N}_{\text{fl},n_1}$ , we have

$$\mathbf{W}_n = (\mathbf{v}_n[1](t_n^{\text{M}} - t_n^{\text{S}}), \mathbf{W}_{n_1}[i_n^{\text{M}}], \dots, \mathbf{W}_{n_1}[i_n^{\text{SP}}], \mathbf{v}_n[N_v](t_n^{\text{A}} - t_n^{\text{SP}}).$$

The variables  $t_n^{\text{S}}, t_n^{\text{M}}, t_n^{\text{SP}}, t_n^{\text{A}}$  denote the start time, merge time, split time, and arrival time of follower  $n$  according to its adapted plan. The first element of  $\mathbf{W}_n$  is the distance along the route from start to the merge point. For the part of the route the follower platoons with the coordination leader, the entries are the same as for the coordination leader. The indices  $i_n^{\text{M}}, i_n^{\text{SP}}$  are defined accordingly. The last element of  $\mathbf{W}_n$  is the distance from the split point to the destination of the follower. Figure 6.6 illustrates the definition of  $\mathbf{W}_n$ . We introduce sequences  $\mathbf{p}_n = (\mathbf{p}_n[1], \dots, \mathbf{p}_n[|\mathbf{W}_n|])$  that indicate on which segments of the journey the coordination follower is a platoon follower. If truck  $n$  is a platoon follower on the segment that corresponds to  $\mathbf{W}_n[i]$  for some  $i$ , then  $\mathbf{p}_n[i] = 1$ . Otherwise we have  $\mathbf{p}_n[i] = 0$ . For the coordination leader  $n_1$ , we have  $\mathbf{p}_{n_1} = (0, \dots, 0)$  and for a coordination follower  $n \in \mathcal{N}_{\text{fl},n_1}$ , we have that  $\mathbf{p}_n = (0, 1, 1, \dots, 1, 0)$ .

We express the speed and time sequence of truck  $n \in \{n_1\} \cup \mathcal{N}_{\text{fl},n_1}$  as traversal times  $\mathbf{T}_n = (\mathbf{T}_n[1], \dots, \mathbf{T}_n[|\mathbf{W}_n|])$  of the segments  $\mathbf{W}_n$ . The speed on each such segment remains constant and can be computed as

$$\mathbf{v}_n[i] = \frac{\mathbf{W}_n[i]}{\mathbf{T}_n[i]}.$$

The traversal times of the segments in all trucks' routes are the optimization variables. Working with traversal times rather than the sequence of speeds  $\mathbf{v}$  allows us to state the optimization problem with linear constraints. The times when the speed changes  $\hat{\mathbf{t}}_n$ , are computed as  $\hat{\mathbf{t}}_n[i] = t_n^{\text{S}} + \sum_{j=1}^{i-1} \mathbf{T}_n[j]$  for  $i = 1, \dots, N_{v,n} + 1$ .



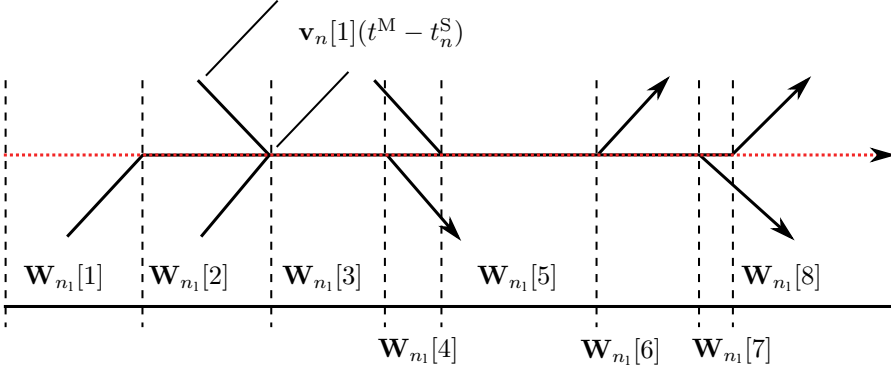


Figure 6.6: Illustration of how the sequences  $\mathbf{W}_n$  are defined. The red, dotted line represents the route of the coordination leader and the black, solid lines with arrows represent the routes of the coordination followers. The thin lines indicate the distances that the elements of  $\mathbf{W}_{n_1}$  correspond to.

With these definitions, we are ready to state the following problem:

**Problem 2.**

$$\min_{\{\mathbf{T}_n : n \in \{n_1\} \cup \mathcal{N}_{\text{fl}, n_1}\}} \sum_{i=1}^{N_{v,n}} f\left(\frac{\mathbf{W}_n[i]}{\mathbf{T}_n[i]}, \mathbf{p}_n[i]\right) \mathbf{W}_n[i] \quad (6.3a)$$

s.t.

for  $n \in \{n_1\} \cup \mathcal{N}_{\text{fl}, n_1}$  :

$$\frac{\mathbf{W}_n[i]}{v_{\max}} \leq \mathbf{T}_n[i], \quad i \in \{1, \dots, N_{v,n}\} \quad (6.3b)$$

$$\frac{\mathbf{W}_n[i]}{v_{\min}} \geq \mathbf{T}_n[i], \quad i \in \{1, \dots, N_{v,n}\} \quad (6.3c)$$

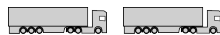
$$t_n^S + \sum_{i=1}^{N_{v,n}} \mathbf{T}_n[i] \leq t_n^D \quad (6.3d)$$

and for  $n \in \mathcal{N}_{\text{fl}, n_1}$  :

$$t_n^S + \mathbf{T}_n[1] = t_{n_1}^S + \sum_{i=1}^{i_n^M - 1} \mathbf{T}_{n_1}[i] \quad (6.3e)$$

$$\mathbf{T}_n[1+i] = \mathbf{T}_{n_1}[i_n^M + i - 1], \quad i \in \{1, \dots, i_n^{\text{Sp}} - i_n^M + 1\}. \quad (6.3f)$$

Notice that the objective function (6.3a) equals the combined fuel consumption  $\sum_{n \in \{n_1\} \cup \mathcal{N}_{\text{fl}, n_1}} F(\phi_n, \pi_n)$  for the assignments  $\{n_1\} \cup \mathcal{N}_{\text{fl}, n_1}$ , which is part of the sum



that defines the combined fuel consumption of all assignment  $F_c$  defined in (3.3). It is composed of the fuel consumption of the coordination leader and the coordination followers. The coordination leader is considered to travel alone or take the role as the platoon leader throughout its journey. The coordination followers travel alone on the first and the last segment of their journey. They become platoon followers in-between these segments.

There are two sets of constraints. The first set applies to all trucks and ensures that the sequences  $\mathbf{T}_n$  correspond to valid vehicle plans. In particular, the constraints (6.3b) and (6.3c) express that the trajectories stay within the allowed range of speed. The constraints (6.3d) express that all trucks arrive before their deadline. The second set of constraints ensures that platooning happens as specified in the original pairwise plans. The constraints (6.3e) ensure that the coordination leader and each of its followers arrive at the same time at their respective merge point. The constraints (6.3f) ensure that the speed of the leader and the speed of the follower are the same when they are supposed to platoon.

When  $f_0, f_p$  are such that  $f_0(T^{-1})$  and  $f_p(T^{-1})$  are convex in  $T$  for  $T > 0$ , then the objective (6.3a) is a sum of convex functions and hence convex. For instance, polynomials with arbitrary constant part and non-negative coefficients fulfill this requirement. Furthermore, all constraints are linear. Thus, Problem 2 is a convex optimization problem for which well developed numerical solvers are readily available [23, 9]. The optimization is initialized with the pairwise plans.

## 6.5 Simulations

In this section, we evaluate the coordination method outlined in the previous sections with Monte Carlo simulations. We show that the coordination of truck platooning can lead to significant reductions in fuel consumption compared to the current situation where trucks do not platoon, as well as compared to spontaneous platooning where trucks only form platoons if they happen to be in the vicinity of another.

We generate transport assignments randomly. The start and goal locations are sampled within mainland Sweden. The probability of an assignment starting or ending at a particular location is proportional to the population density [82], see Figure 6.7. The resolution is 0.1 degrees in longitude and latitude and the road network node that is closest to the sampled coordinate is chosen. We calculate the routes with the Open Source Routing Machine [65]. Assignments for which no route can be found are disregarded. If the route is longer than 400 kilometers, a 400 kilometers long subsection of the route is randomly selected. This is to take into account that merge points too far from the current position should not be considered for coordination since the uncertainty becomes too large due to traffic, new assignments, and rest periods of the driver. Start locations along the route are considered since we believe that platoon coordination systems will frequently re-plan for assignments that are already en route and suspended for the driver to take a rest.



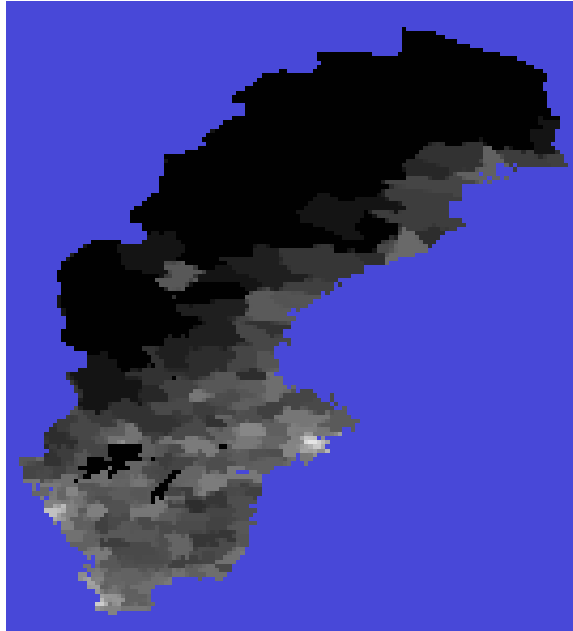


Figure 6.7: Population density map from which the start and goal locations are sampled. The brighter the pixel, the larger the population density in that area. Areas not belonging to mainland Sweden are shown in blue.

The fuel model is an affine approximation around 80 km/h of the analytical fuel model in [16]. We have for the fuel per distance traveled in kilograms diesel per meter

$$\begin{aligned} f_0(v) &= 8.4159 \cdot 10^{-6} v + 4.8021 \cdot 10^{-5} \\ f_p(v) &= 5.0495 \cdot 10^{-6} v + 8.5426 \cdot 10^{-5}. \end{aligned}$$

According to this model, the relative reduction in fuel consumption of a platoon follower is 15.9 percent at a speed of 80 km/h.

We consider a default speed of 80 km/h and we assume that the speed can be freely chosen between  $v_{\min} = 70$  km/h and  $v_{\max} = 90$  km/h throughout the entire journey. We sample the start time of the assignments uniformly in an interval of 2 hours and compute the arrival deadlines according to the default speed.

The pairwise plans are such that trucks platoon as long as possible. Once a coordination follower splits up from the coordination leader, it drives fast enough to arrive in time at its destination and at least at default speed. The split points are such that arriving in time is feasible. Thus, trucks are guaranteed to meet their deadlines and the initial value for the joint vehicle plan optimization fulfills the





Figure 6.8: The routes of a platoon coordinator with four coordination followers. The route of the coordination leader is shown in black, the routes of the coordination followers are dashed. The beginning of a route is marked with a star. The merge point of a follower is indicated with an upwards-facing triangle and the split point with a downwards-facing triangle.

constraints. Figure 6.8 shows an example of the routes of a coordination leader and its coordination followers and where the coordination followers join and leave the platoon.

We compare our proposed platoon coordinator to fuel savings that arise from spontaneous platooning, i.e., that trucks happen to get into each others vicinity and then spontaneously form platoons. To this end, we collect all the link arrival times according to the default plans for each link in the scenario. We sort these times and collect them in ascending order in groups of at most one minute difference in their edge arrival time. We assume that each of these groups forms a platoon driving at default speed and that the default trajectory is not altered by the platooning. This is a generous estimate since it neglects any kind of coordination effort, which would be present for time gaps up to one minute.





In order to assess the quality of the solution computed by Algorithm 2, we establish an upper bound on the solution of Problem 1. This upper bound is based on the intuition to assign every truck its best coordination leader and ignore that coordination leaders do not contribute to the objective. We have that

$$\begin{aligned}
 f_{ce}(\mathcal{N}_1) &= \sum_{i \in \mathcal{N}_c \setminus \mathcal{N}_1} \max_{j \in \mathcal{N}_i^o \cap \mathcal{N}_1} \Delta F(i, j) \\
 &\leq \sum_{i \in \mathcal{N}_c \setminus \mathcal{N}_1} \max_{j \in \mathcal{N}_i^o} \Delta F(i, j) \\
 &\leq \sum_{i \in \mathcal{N}_c} \max_{j \in \mathcal{N}_i^o} \Delta F(i, j),
 \end{aligned} \tag{6.4}$$

where the second inequality holds since  $\Delta F(i, j) > 0$  for all  $(i, j) \in \mathcal{E}$ . This bound can also be derived from the upper bound used in the branch and bound algorithm (Proposition 7) by setting  $\bar{\mathcal{N}}_1 = \bar{\mathcal{N}}_f = \emptyset$ .

This bound can only be tight when there is an optimal solution where no coordination leader has an out-neighbor. Otherwise the coordination leaders cannot contribute to the sum. Nevertheless, the bound helps us assess how far a heuristic solution can be away from the optimum.

We implemented platoon coordination in Python and used CVXOPT [9] for convex optimization. The execution of Algorithm 2 takes less than a second for 2000 transport assignments. Even faster computation times could be achieved by optimizing the implementation.

Each simulation consists of the following steps:

1. Random generation of transport assignments
2. Computation of routes and default plans
3. Computation of the coordination graph
4. Computation of coordination leaders according to Section 6.3
5. Joint vehicle plan optimization according to Section 6.4

We evaluate how different numbers of assignments affect the amount of platooning and the fuel savings relative to the default plans. For comparison we compute the fuel savings of spontaneous platooning. We run Algorithm 2 with greedy and random node selection and compute the upper bound of the objective function  $f_{ce}$ . The results are averaged over 150 simulation runs.

Figure 6.9 visualizes an example coordination graph. In addition it shows which assignments are selected in step 4). We can see that only a small fraction of assignment pairs can save fuel by forming a platoon. As the number of assignments grows, more opportunities are available for each assignment which can translate into larger fuel savings [58].



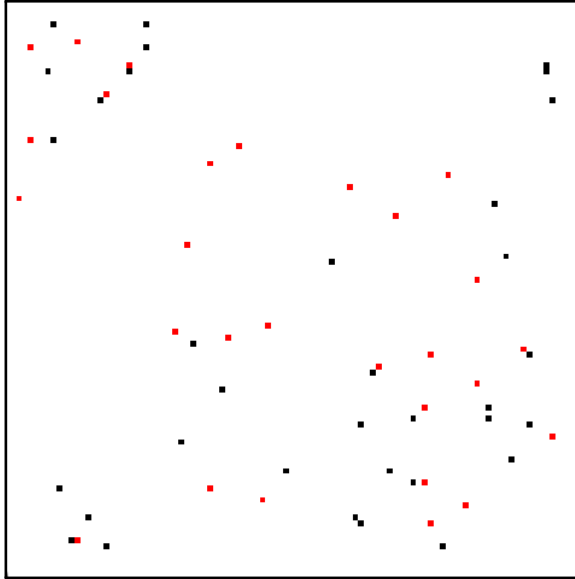


Figure 6.9: This plots visualizes the adjacency matrix of a coordination graphs with 100 assignments. Nonzero entries are indicated with a black or a red dot, each corresponding to an edge in the coordination graph. Edges whose corresponding plans are selected by the Algorithm 2 correspond to the red dots.

Figure 6.10 shows the effect on the fuel savings when the numbers of transport assignments that are coordinated is varied. It is possible to make a number of observations based on these data. First of all, the fuel savings increase rapidly with the number of transport assignments when the absolute number of assignments is small. As more and more assignments are added, this trend stagnates and the relative fuel savings increase only slowly. Ideally this should approach asymptotically the maximum fuel savings of 15.9% as the number of transport assignments goes to infinity, since then virtually every truck is a platoon follower for its entire journey. There is only a small difference between greedy and random node selection, however, with the greedy node selection outperforming the random node selection consistently. For a parallel or even a distributed implementation of Algorithm 2, random node selection would be preferable due to the reduced need for synchronization whereas greedy node selection is faster in a centralized setting. Furthermore, the results after selecting the coordination leaders and before the joint convex optimization are less than the upper bound but only about 30% worse. Since the upper bound is not tight, this indicates that Algorithm 2 performs well. We can see a clear improvement in the fuel savings by the joint optimization of the vehicle plans. Spontaneous platooning gives fuel savings that are less than half of what can



be achieved by coordination. Also bear in mind that this is a generous estimate of fuel savings by spontaneous platooning so that the real difference would probably be even larger.

We conclude that coordinated platooning can yield significant fuel savings and that coordination is crucial in leveraging these savings. For 2000 transport assignments starting over the course of two hours, we get 7.6% reduction in fuel consumption. A number of 2000 trucks starting in that time interval on an area like Sweden is a realistic number. The total distance traveled in the simulated scenario is in the same order of magnitude as the total distance traveled by domestic road freight transport in Sweden within two hours, assuming that traffic volume is equally spread over the year [1]. The density of the road freight traffic that was simulated is only a fraction of the total road freight traffic in countries with high population density, .

Figure 6.11 shows how the distribution of platoon sizes changes with the number of transport assignments. We can see that the larger the number of transport assignments, the more distance is traveled in large platoons. For 2000 assignments, over half the distance traveled is in a platoon. Most of the distance is traveled in platoons with ten or less vehicles. This is promising since large platoons might be difficult to control and thus the platoon coordinator would have to prevent planning for larger platoons. Since these large platoons only account for a small fraction of the distance traveled, this would not have too large an impact on the total fuel savings. The largest platoon formed has 28 vehicles. A noticeable effect occurs at a number of 200 transport assignments when more distance is traveled in relatively large platoons compared to the distribution with a number of 300 transport assignments. It seems that some kind of phase transition occurs at these points, where enough assignments are in the system to go from one coordination leader with many followers to having several coordination leaders that are better suited for their followers. To understand this phenomenon is subject of future work.

The simulations show that computing plans for a large number of vehicles to form platoons is feasible with the methods outlined in this paper. It motivates that real-time platoon coordination enables significant reductions in fuel consumption and might be the key to leveraging the full potential of truck platooning.

## 6.6 Summary

This chapter describes how to compute vehicle plans for platooning. The key element making the presented approach to coordinating truck platooning feasible for large numbers of vehicles is the systematic combination of default plans and adapted plans. Some vehicles, the coordination leaders, get their default plans assigned. The remaining vehicles use the most fuel efficient plan that is adapted to one of the coordination leaders. The fuel efficiency of the adapted plans is based on becoming a platoon follower of a coordination leader during a part of the route. By selecting coordination leaders in a smart way, the fuel savings that result from the adapted plans is maximized.



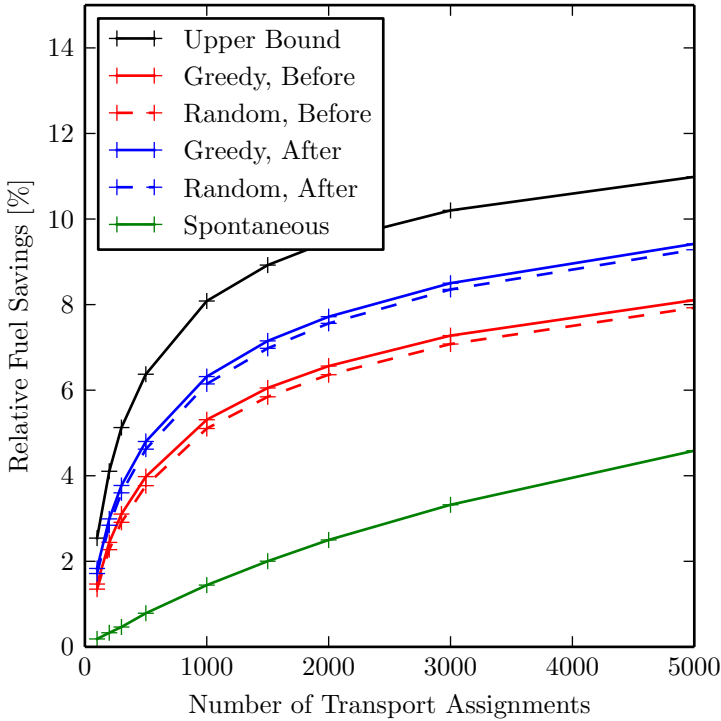


Figure 6.10: The relative fuel savings due to platooning compared to the default plans with varying numbers of assignments. “Greedy” indicates that greedy node selection was used in the clustering algorithm, whereas “Random” indicates random node selection. The keywords “Before”/“After” refer to the relative fuel savings before/after the joint optimization of the vehicle plans. “Spontaneous” are the relative fuel savings based on the estimate of fuel savings due to spontaneous platooning. “Upper Bound” refers to the upper bound the  $f_{ce}$  as stated in (6.4).

The proper selection of coordination leaders can be done by a branch and bound algorithm. This algorithm can explore all possible allocations of coordination leaders. However, it can skip over entire sets of possible allocations when no optimal allocation can be element of these sets. This is done by comparing the best solution found so far in the execution of the algorithm to an upper bound of the solutions contained in the set. The result that not more than half the vehicles should be coordination leaders and a method to compute that subsets of vehicles have to contain at least one coordination leader, can help to further dismiss sets of suboptimal solutions. Unfortunately, the problem of selecting coordination leaders is proven to



be NP-hard, which means that any algorithm that computes exact solutions might have very long running times.

This motivates the design of algorithms that compute a good selection of coordination leaders efficiently but not necessarily the best one. One possible choice is an algorithm that starts from an empty set of coordination leaders and iteratively adds and removes leaders from that set, increasing the fuel savings in each iteration. Such an algorithm can compute a good result efficiently for large numbers of vehicles.

The combination of default and adapted plans can be further improved in order to reduce fuel consumption. By committing to which vehicles platoon on which parts of the routes, it is possible to adjust the timing that leads to such platoons. Since this affects the speed profiles, it changes the fuel consumption. Using convex optimization, the timing can be adjusted in a fuel-optimal way.

Simulations show that this method has the potential to coordinate a realistic fleet of vehicles and achieve significant fuel savings compared to the current situation in which no platooning is used. The method also improves over so-called spontaneous platooning where vehicles happen to get into each others vicinity and form platoons.



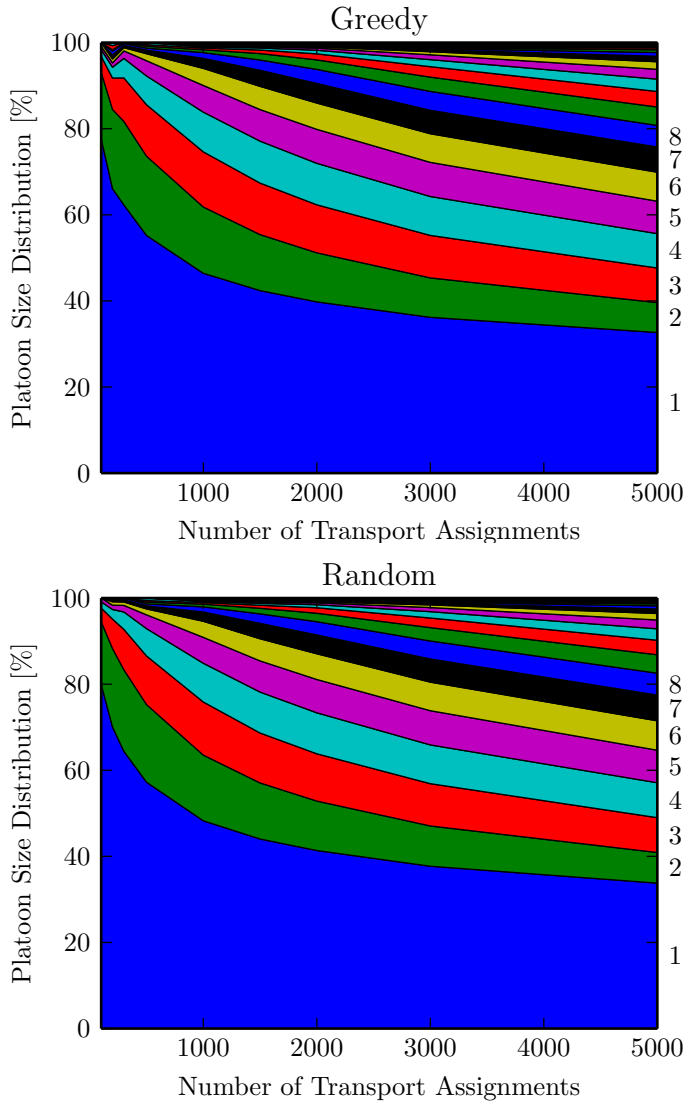


Figure 6.11: This figure shows the distribution of platoon sizes per distance traveled over the number of assignments in percent. The upper plot shows the results of greedy node selection whereas the lower plot shows those of random node selection in the clustering algorithm. To the right, the size of platoon is indicated for a platoon size up to eight. So, when the distance between the first and the second boundary from below is for instance at 20%, it means that 20% of the distance was traveled as member of a platoon of size 2.



# Chapter 7

## Conclusions and Future Work

THIS chapter concludes the thesis. Section 7.1 recaps and discusses the presented results. Section 7.2 provides some possible directions for future work on the topic.

### 7.1 Conclusions

This thesis considers the problem of coordinating the formation of truck platoons in a fuel-efficient way. A centralized coordination system for truck platooning was proposed. A possibly large number of vehicles would connect to this system over vehicle-to-infrastructure communication. Each vehicle provides start and goal position as well as time of its next transport assignments. The system returns vehicle plans consisting of routes and speed profiles that lead to globally reduced fuel consumption based on platooning. As time evolves, this process is repeated to account for deviations and new assignments.

The computation of the vehicle plans is a complex problem. One element that makes the problem difficult is the potentially large number of vehicles. This was successfully approached by dividing the computation into tractable stages. First, the routes are determined. Algorithms for route computation in road networks are readily available and this computation was not further investigated. Then, a default speed plans and a selection of adapted plans is computed. Since the computation of such plans is limited to the calculation of the speed profile for one vehicle, it is possible to derive fuel-optimal speed profiles. Each adapted plan involves platooning for a certain distance as platoon follower of the vehicle the plan is adapted to. This can lead to a lower fuel consumption. Such plans were derived for an affine fuel model. Similar plans can be derived for other fuel models, changing speed ranges along the route and other additional factors. At the next stage these plans are combined systematically in order to maximize the global fuel savings. The problem of combining the plans was formulated as a problem of selecting of a subset of vehicles that get their default plans assigned and to which



the remaining vehicles can adapt in order to form platoons. A branch and bound method for solving the combinatorial optimization problem of selecting coordination leaders was proposed. The method uses several heuristics to prune the search space. However, it was shown that the problem is NP-hard, which is a strong indicator that exact solution might be very expensive to compute. Therefore, an iterative algorithm was proposed that can quickly compute good but not necessarily optimal solutions.

The effectiveness of the method was demonstrated in a realistic simulation study. Significant amounts of fuel can be saved by platooning. The coordination of platooning was shown to be crucial in fully exploiting the potential of platooning to reduce fuel consumption. This thesis demonstrates that the dynamic coordination of forming platoons is feasible. The proposed methods are implemented and tested in the scope of the COMPANION project [32] in a demonstrator that includes real and simulated vehicles. Even though the development is still ongoing at the time this thesis is written, preliminary results are promising and indicate that a platoon coordinator, as proposed here, can function in practice.

## 7.2 Future Work

Despite the promising results, there are various open questions. One such question is to understand how the transport assignments and the road network relate to fuel savings achieved by this method. Clearly, the spatial and temporal distribution of the assignments has an influence on how much fuel can be saved from platooning. One extreme case would be that all assignments have the same data and the trucks can form a platoon from start to destination. Another extreme would be that no two assignments are executed during the same time. In these two cases it is easy to predict how much fuel can be saved through platooning. It should be possible to explicitly estimate the fuel savings based on the assignment data in the general case or to find bounds.

A related question is to analyze how well the heuristic algorithm for the selection of default and adapted plans performs. This can be done analytically as well as in simulations. For the latter, we need to implement the branch and bound algorithm efficiently using all the heuristics, so that we can compare the results of the heuristic algorithm to exact solutions.

We also want to study what happens if the platoon coordinator runs as a model predictive controller under the presence of disturbances as opposed to computing a plan for a set of assignments only once. Realistically, trucks will not be able to execute the plans exactly, but there will be disturbances due to traffic, weather, the driver taking a break, etc. Furthermore, new assignments could be added on the fly with a receding time horizon. In control theory, feedback is used in order to attenuate disturbances. For the truck platoon coordination problem this means repeating the calculation of the plans based on updated information, similar to a model predictive controller. Additionally, there significant communication delays





that need to be accounted for. We intend to study how plans and the achieved fuel consumption change when such repeated replanning is employed and derive analytical results related to the stability and convergence of the overall system.

Apart from addressing uncertainty by using feedback, it might be beneficial to take possible disturbances into account explicitly. Using historic or live traffic data, we can get an idea of how well a plan can be followed by the vehicle. For instance, to plan for merge points after a region with heavy traffic might be too uncertain, and a plan that merges the platoon before the region of heavy traffic should be preferred. To this end the framework needs to be extended so that stochastic effects can be modeled and accounted for. Scenario based model predictive control, as well as stochastic and robust optimization can be appropriate sources of inspiration.

One of the assumptions made is that routes are not altered to facilitate platooning. This was motivated by the high complexity routing adds to the problem and the intuition that the road system is built in a way that for long distances there are typically not many alternatives of comparable length. Future work is to analyze how much could be gained from adapting the routes as well, and investigate if there are efficient ways of adapting the route for platooning for instance by considering few route alternatives such as the ones derived in [10].

Finally, it will be interesting to investigate how a platoon coordination system can work in practice. The way such a system is perceived by fleet owners, professional drivers, and the general public will play an important role in the success of the system. While some experiments with real vehicles have been made, more tests have to be performed in order to make large-scale deployment of platoon coordination systems a reality.





# Bibliography

- [1] *Lastbilstrafik 2014 Swedish national and international road goods transport 2014*. Technical report, Trafikanalys, 2015.
- [2] European truck platooning challenge 2016. Dutch Ministry of Infrastructure and the Environment, the Directorate General Rijkswaterstaat, the Netherlands Vehicle Authority (RDW), Conference of European Directors of Roads (CEDR) and European Automobile Manufacturers' Association (ACEA), 2016.
- [3] I. Abraham, A. Fiat, A. V. Goldberg, and R. F. Werneck. Highway dimension, shortest paths, and provably efficient algorithms. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 782–793. Society for Industrial and Applied Mathematics, 2010.
- [4] N. Agatz, A. Erera, M. Savelsbergh, and X. Wang. Optimization for dynamic ride-sharing: A review. *European Journal of Operational Research*, 223(2):295–303, 2012.
- [5] A. Alam, B. Besselink, V. Turri, J. Mårtensson, and K. H. Johansson. Heavy-duty vehicle platooning towards sustainable freight transportation: A cooperative method to enhance safety and efficiency. *IEEE Control Systems Magazine*, 35(6):34–56, 2015.
- [6] A. Alam, A. Gattami, and K. H. Johansson. An experimental study on the fuel reduction potential of heavy duty vehicle platooning. In *Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems*, pages 306–311, Madeira, Portugal, 2010.
- [7] C. J. Almqvist and K. Heinig. *European Accident Research and Safety Report 2013*. Technical report, Volvo Trucks, 2013.
- [8] L. Alvarez and R. Horowitz. *Safe Platooning in Automated Highway Systems*. Research report, University of California, Berkeley, CA, USA, 1997.
- [9] M. S. Andersen, J. Dahl, and L. Vandenberghe. CVXOPT: A Python package for convex optimization, 2013.



- [10] R. Bader, J. Dees, R. Geisberger, and P. Sanders. Alternative route graphs in road networks. In A. Marchetti-Spaccamela and M. Segal, editors, *Theory and Practice of Algorithms in (Computer) Systems: First International ICST Conference*, pages 21–32, Rome, Italy, Apr. 2011. Springer Berlin Heidelberg.
- [11] D. Banister and D. Stead. Impact of information and communications technology on transport. *Transport Reviews*, 24(5):611–632, 2004.
- [12] L. D. Baskar, B. De Schutter, J. Hellendoorn, and Z. Papp. Traffic control and intelligent vehicle highway systems: a survey. *Intelligent Transport Systems, IET*, 5(1):38–52, Mar. 2011.
- [13] H. Bast, S. Funke, D. Matijevic, P. Sanders, and D. Schultes. In transit to constant time shortest-path queries in road networks. In *9th Workshop on Algorithm Engineering and Experiments*, pages 46–59, 2007.
- [14] J. Bentley and T. Ottmann. Algorithms for reporting and counting geometric intersections. *IEEE Transactions on Computers*, C-28(9):643–647, Sept. 1979.
- [15] D. Bertsimas and S. S. Patterson. The air traffic flow management problem with enroute capacities. *Operations Research*, 46(3):406–422, 1998.
- [16] B. Besselink, V. Turri, S. van de Hoef, K.-Y. Liang, A. Alam, J. Mårtensson, and K. H. Johansson. Cyber-physical control of road freight transport. *Proceedings of the IEEE*, 104(5):1128–1141, 2016.
- [17] C. Bierwirth and F. Meisel. A survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research*, 202(3):615–627, 2010.
- [18] K. Bimbraw. Autonomous cars: Past, present and future a review of the developments in the last century, the present scenario and the expected future of autonomous vehicle technology. In *12th International Conference on Informatics in Control, Automation and Robotics*, volume 01, pages 191–198, July 2015.
- [19] V. Blondel, J. Guillaume, R. Lambiotte, and E. Mech. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics*, page P10008, 2008.
- [20] C. Bonnet and H. Fritz. Fuel consumption reduction in a platoon: Experimental results with two electronically coupled trucks at close spacing. *SAE Technical Paper 2000-01-3056*, 2000.
- [21] Y. Bontekoning and H. Priemus. Breakthrough innovations in intermodal freight transport. *Transportation Planning and Technology*, 27(5):335–345, 2004.



- [22] K. Boriboonsomsin, M. J. Barth, W. Zhu, and A. Vu. Eco-routing navigation system based on multisource historical and real-time traffic information. *IEEE Transactions on Intelligent Transportation Systems*, 13(4):1694–1704, Dec. 2012.
- [23] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.
- [24] F. Bullo, E. Frazzoli, M. Pavone, K. Savla, and S. L. Smith. Dynamic vehicle routing for robotic systems. *Proceedings of the IEEE*, 99(9):1482–1504, 2011.
- [25] R. Chandler, R. Herman, and E. Montroll. Traffic dynamics: Studies in car following. *Operations Research*, 6(2):165–184, 1958.
- [26] J. Clausen. *Branch and Bound Algorithms - Principles and Examples*, 1999.
- [27] J. D. Cohen, M. C. Lin, D. Manocha, and M. Ponamgi. I-COLLIDE: An interactive and exact collision detection system for large-scale environments. In *Proceedings of the ACM Interactive 3D Graphics Conference*, pages 189–196. ACM, 1995.
- [28] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson. *Introduction to Algorithms*. MIT Press, 2nd edition, 2009.
- [29] T. G. Crainic and G. Laporte. Planning models for freight transportation. *European Journal of Operational Research*, 97(3):409–438, 1997.
- [30] E. Demir, T. Bektaş, and G. Laporte. A review of recent research on green road freight transportation. *European Journal of Operational Research*, 237(3):775–793, 2014.
- [31] H. Edelsbrunner and H. A. Maurer. On the intersection of orthogonal objects. *Information Processing Letters*, 13(4/5):177–181, 1981.
- [32] S. Eilers, J. Mårtensson, H. Pettersson, M. Pillado, D. Gallegos, M. Tobar, K. H. Johansson, X. Ma, T. Friedrichs, S. S. Borojeni, and M. Adolfson. COMPANION – Towards co-operative platoon management of heavy-duty vehicles. In *18th IEEE International Conference on Intelligent Transportation Systems*, pages 1267–1273, 2015.
- [33] European Commission. *Roadmap to a Single European Transport Area — Towards a competitive and resource efficient transport system*. Publications Office of the European Union, Luxembourg, 2011.
- [34] European Commission. *EU Energy, Transport and GHG Emissions Trends to 2050*. Publications Office of the European Union, Luxembourg, 2013.



- [35] European Commission. *Collection and Analysis of Data on the Structure of the Road Haulage Sector in the European Union*. Publications Office of the European Union, Luxembourg, 2014.
- [36] European Commission. *EU transport in figures – Statistical pocketbook 2015*. Publications Office of the European Union, Luxembourg, 2015.
- [37] European Parliament and Council. *Regulation (EC) No 561/2006*, 2006.
- [38] European Parliament and Council. *Regulation (EC) No 661/2009*, 2009.
- [39] S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3–5):75–174, 2010.
- [40] S. Gao and I. Chabini. Optimal routing policy problems in stochastic time-dependent networks. *Transportation Research Part B: Methodological*, 40(2):93–122, 2006.
- [41] M. Gendreau, G. Laporte, and R. Séguin. Stochastic vehicle routing. *European Journal of Operational Research*, 88(1):3–12, 1996.
- [42] H. Gharavi, K. V. Prasad, and P. A. Ioannou. Special issue on advanced automobile technologies. *Proceedings of the IEEE*, 95(2), 2007.
- [43] S. Hallé, B. Chaib-draa, and J. Laumonier. Car platoons simulated as a multi-agent system. In *Proceedings of the 4th Workshop on Agent-Based Simulation*, pages 57–63, 2003.
- [44] H. Hao and P. Barooah. Stability and robustness of large platoons of vehicles with double-integrator models and nearest neighbor interaction. *International Journal of Robust and Nonlinear Control*, 23(18):2097–2122, 2013.
- [45] S. Harenberg, G. Bello, L. Gjeltema, S. Ranshous, J. Harlalka, R. Seay, K. Padmanabhan, and N. Samatova. Community detection in large-scale networks: a survey and empirical evaluation. *Wiley Interdisciplinary Reviews: Computational Statistics*, 6(6):426–439, 2014.
- [46] H. Hartenstein and K. Laberteaux. A tutorial survey on vehicular ad hoc networks. *IEEE Communications Magazine*, 46(6):164–171, 2008.
- [47] J. Herrera, D. Work, R. Herring, X. Ban, Q. Jacobson, and A. Bayen. Evaluation of traffic data obtained via GPS-enabled mobile phones: The Mobile Century field experiment. *Transportation Research Part C: Emerging Technologies*, 18(4):568–583, 2010.
- [48] R. Horowitz and P. Varaiya. Control design of an automated highway system. *Proceedings of the IEEE*, 88(7):913–925, July 2000.



- [49] P. Ioannou and C. Chien. Autonomous intelligent cruise control. *IEEE Transactions on Vehicular Technology*, 42(4):657–672, 1993.
- [50] A. K. Jain. Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31(8):651–666, 2010.
- [51] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.
- [52] R. Janssen, H. Zwijnenberg, I. Blankers, and J. de Kruijff. *Truck Platooning Driving the Future of Transportation*. TNO Mobility and Logistics, 2015.
- [53] J. Jespersen-Groth, D. Potthoff, J. Clausen, D. Huisman, L. Kroon, G. Maróti, and M. N. Nielsen. *Robust and Online Large-Scale Optimization: Models and Techniques for Transportation Systems*, chapter Disruption Management in Passenger Railway Transportation, pages 399–421. Lecture Notes in Computer Science. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
- [54] P. Jiménez, F. Thomas, and C. Torras. 3D collision detection: A survey. *Computers and Graphics*, 25:269–285, 2000.
- [55] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An introduction to Cluster Analysis*. John Wiley & Sons, Inc., 2008.
- [56] R. Kunze, R. Ramakers, K. Henning, and S. Jeschke. *Intelligent Robotics and Applications*, volume 5928 of *Lecture Notes in Computer Science*, chapter Organization and Operation of Electronically Coupled Truck Platoons on German Motorways, pages 135–146. Springer Berlin Heidelberg, 2009.
- [57] M. Lammert, A. Duran, J. Diez, K. Burton, and A. Nicholson. Effect of platooning on fuel consumption of class 8 vehicles over a range of speeds, following distances, and mass. *SAE International Journal of Commercial Vehicles*, 7(2):626–639, 2014.
- [58] J. Larson, K.-Y. Liang, and K. H. Johansson. A distributed framework for coordinated heavy-duty vehicle platooning. *IEEE Transactions on Intelligent Transportation Systems*, 16(1):419–429, Feb. 2015.
- [59] E. Larsson, G. Sennton, and J. Larson. The vehicle platooning problem: Computational complexity and heuristics. *Transportation Research Part C: Emerging Technologies*, 60:258–277, 2015.
- [60] W. Levine and M. Athans. On the optimal error regulation of a string of moving vehicles. *IEEE Transactions on Automatic Control*, AC-11(3):355–361, 1966.



- [61] L. Li, R. R. Negenborn, and B. D. Schutter. Intermodal freight transport planning — A receding horizon control approach. *Transportation Research Part C: Emerging Technologies*, 60:77–95, 2015.
- [62] K.-Y. Liang, J. Mårtensson, and K. H. Johansson. When is it fuel efficient for a heavy duty vehicle to catch up with a platoon? In *7th IFAC Symposium on Advances in Automotive Control*, 2013.
- [63] K.-Y. Liang, S. van de Hoef, H. Terelius, V. Turri, B. Besselink, J. Mårtensson, and K. H. Johansson. Networked control challenges in collaborative road freight transport. *European Journal of Control*, 2016. To be published.
- [64] F. Liu, T. Harada, Y. Lee, and Y. J. Kim. Real-time collision culling of a million bodies on graphics processing units. In *Proceedings of the ACM SIGGRAPH Asia*, pages 154:1–154:8. ACM, 2010.
- [65] D. Luxen and C. Vetter. Real-time routing with openstreetmap data. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 513–516. ACM, 2011.
- [66] C. Macharis and Y. Bontekoning. Opportunities for OR in intermodal freight transport research: A review. *European Journal of Operational Research*, 153(2):400–416, 2004. Management of the Future MCDA: Dynamic and Ethical Contributions.
- [67] G. Mbiydzennyuy. Arrival times with hours of service regulations for truck drivers-tracks and gaps from current research. In *18th IEEE International Conference on Intelligent Transportation Systems*, pages 2631–2636, Sept. 2015.
- [68] A. McKinnon. Life without trucks: The impact of a temporary disruption of road freight transport on a national economy. *Journal of Business Logistics*, 27(2):227–250, 2006.
- [69] P. Meisen, T. Seidl, and K. Henning. A data-mining technique for the planning and organization of truck platoons. In *International Conference on Heavy Vehicles, Heavy Vehicle Transport Technology*, pages 389–402, 2008.
- [70] S. Melzer and B. Kuo. A closed-form solution for the optimal error regulation of a string of moving vehicles. *IEEE Transactions on Automatic Control*, 16(1):50–52, 1971.
- [71] V. Milanés, S. E. Shladover, J. Spring, C. Nowakowski, H. Kawazoe, and M. Nakamura. Cooperative adaptive cruise control in real traffic situations. *IEEE Transactions on Intelligent Transportation Systems*, 15(1):296–305, Feb. 2014.





- [72] R. H. Möhring, H. Schilling, B. Schütz, D. Wagner, and T. Willhalm. Partitioning graphs to speedup Dijkstra’s algorithm. *ACM Journal of Experimental Algorithmics*, 11(2.8):1–29, Feb. 2007.
- [73] G. Naus, R. Vugts, J. Ploeg, M. Van de Molengraft, and M. Steinbuch. String-stable CACC design and experimental validation: a frequency-domain approach. *IEEE Transactions on Vehicular Technology*, 59(9):4268–4279, 2010.
- [74] OECD/International Transport Forum. *International transport outlook 2015*. OECD Publishing/ITF, 2015.
- [75] U. Ozguner, C. Stiller, and K. Redmill. Systems for safety and autonomous behavior in cars: The DARPA grand challenge experience. *Proceedings of the IEEE*, 95(2):397–412, Feb. 2007.
- [76] M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang. Review of road traffic control strategies. *Proceedings of the IEEE*, 91(12):2043–2067, Dec. 2003.
- [77] H. Raza and P. Ioannou. Vehicle following control design for automated highway systems. *IEEE Control Systems Magazine*, 16(6):43–60, 1996.
- [78] J. Robson. Algorithms for maximum independent sets. *Journal of Algorithms*, 7(3):425–440, 1986.
- [79] K. Roy and C. Tomlin. Enroute airspace control and controller workload analysis using a novel slot-based sector model. In *American Control Conference*, page 6. IEEE, June 2006.
- [80] P. Sanders and D. Schultes. Engineering fast route planning algorithms. In C. Demetrescu, editor, *Experimental Algorithms: 6th International Workshop*, pages 23–36, Rome, Italy, June 2007. Springer Berlin Heidelberg.
- [81] Scania AB. *Annual report*, 2014.
- [82] Socioeconomic Data and Application Center. *Population Density Grid, v3, 2000*, 2015.
- [83] D. Steenken, S. Voß, and R. Stahlbock. Container terminal operation and operations research - a classification and literature review. *OR Spectrum*, 26(1):3–49.
- [84] D. Swaroop, J. Hedrick, C. Chien, and P. Ioannou. A comparison of spacing and headway control laws for automatically controlled vehicles. *Vehicle System Dynamics*, 23(1):597–625, 1994.
- [85] D. Swaroop and J. K. Hedrick. String stability of interconnected systems. *IEEE Transactions on Automatic Control*, 41(3):349–357, Mar. 1996.



- [86] P. Toth and D. Vigo. *The Vehicle Routing Problem*. Society for Industrial and Applied Mathematics, 2002.
- [87] S. Tsugawa. An overview on an automated truck platoon within the Energy ITS Project. *Advances in Automotive Control*, 7:41–46, 2013.
- [88] S. Tsugawa, S. Kato, T. Matsui, H. Naganawa, and H. Fujii. An architecture for cooperative driving of automated vehicles. In *Proceedings of the Intelligent Transportation Systems Conference*, pages 422–427, Dearborn, USA, 2000.
- [89] S. Tsugawa, S. Kato, K. Tokuda, T. Matsui, and H. Fujii. A cooperative driving system with automated vehicles and inter-vehicle communications in demo 2000. In *IEEE Intelligent Transportation Systems Proceedings*, pages 918–923, 2001.
- [90] V. Turri, B. Besselink, J. Mårtensson, and K. H. Johansson. Fuel-efficient heavy-duty vehicle platooning by look-ahead control. In *53rd IEEE Conference on Decision and Control*, pages 654–660, Dec. 2014.
- [91] S. van de Hoef, D. V. Dimarogonas, and P. Tsiotras. Spectral analysis of extended consensus algorithms for multiagent systems. In *53rd IEEE Conference on Decision and Control*, pages 2204–2209, Dec. 2014.
- [92] S. van de Hoef, K. H. Johansson, and D. V. Dimarogonas. Coordinating truck platooning by clustering pairwise fuel-optimal plans. In *18th IEEE International Conference on Intelligent Transportation Systems*, pages 408–415, 2015.
- [93] S. van de Hoef, K. H. Johansson, and D. V. Dimarogonas. Fuel-optimal coordination of truck platooning based on shortest paths. In *American Control Conference*, pages 3740–3745, Chicago, IL, 2015.
- [94] S. van de Hoef, K. H. Johansson, and D. V. Dimarogonas. Computing feasible vehicle platooning opportunities for transport assignments. In *14-th IFAC Symposium on Control in Transportation Systems*, 2016. To be published.
- [95] S. van de Hoef, K. H. Johansson, and D. V. Dimarogonas. Fuel-efficient en route formation of truck platoons. *IEEE Transactions on Intelligent Transportation Systems*, 2016. Submitted for publication.
- [96] M. van Schijndel-de Nooij, B. Krosse, T. van den Broek, S. Maas, E. van Nunen, H. Zwijnenberg, A. Schieben, H. Mosebach, N. Ford, M. McDonald, D. Jeffery, J. Piao, and J. Sanchez. *Definition of necessary vehicle and infrastructure systems for Automated Driving*. Technical report, European Commission, 2011.
- [97] P. Varaiya. Smart cars on smart roads: problems of control. *IEEE Transactions on Automatic Control*, 38(2):195–207, 1993.



- [98] M. Whaiduzzaman, M. Sookhak, A. Gani, and R. Buyya. A survey on vehicular cloud computing. *Journal of Network and Computer Applications*, 40:325–344, 2014.
- [99] H. Wolf-Heinrich and S. R. Ahmed. *Aerodynamics of Road Vehicles*. Society of Automotive Engineers, 1998.
- [100] M. Zabat, N. Stabile, S. Farascaroli, and F. Browand. *The Aerodynamic Performance Of Platoons: A Final Report*. Technical report, Institute of Transportation Studies, University of California Berkeley, Jan. 1995.

