MANAGING ALGORITHMIC DRIVERS IN A BLOCKED-LANE SCENARIO

Dmitrii Tikhonenko

TESI DOCTORAL UPF / Year 2020

THESIS SUPERVISORS
Mihalis Markakis
Kalyan Talluri
Department of Economics, Finance and Business

To Diana.

Acknowledgements

My greatest gratitudes are towards my advisers, Mihalis Markakis and Kalyan Talluri, for their constant guidance throughout these years. This work would not be possible without their bright ideas, support, and patience. I learned so many different skills from them, such as how to conduct research, how to present it, and how to write it down. How not to lose enthusiasm over failures and stalled progress. And of course, our countless discussions and brainstorms are what forged the shape of this thesis.

I want to express my deep appreciation to Gregory Fridman. I owe him a very idea to start this journey and the initial momentum on this path. His energetic attitude has always been highly inspiring to me.

Many thanks to Piotr Zwiernik; working on a math course with him was a pleasure. My rigorous analysis capabilities that affected this thesis directly have been improved manifold during seminars of this course. Thanks to all students for their keen questions, too.

I much appreciate the valuable comments of Victor Martínez-de-Albéniz. I thank participants of EUROYoung, TSL Workshop, seminars at Cornell, INSEAD, and Durham Universities, who gave helpful feedback.

I wish to thank Marta Araque and Laura Agusti for their highest-class administrative support during all these years. I admire their eagerness to help in any situation. During my last year, the internship at IESE Business School helped work on my thesis a lot, and I highly appreciate the opportunity. On that side, I thank Irene Sancho for her administrative help.

I give my final gratitude to my parents, who always support me in any life situation.

The thesis is supported by the ECO2013-41131-R and ECO2016-75905-R (AEI/FEDER, UE) grants from the Spanish Ministry of Economy, Industry and Competitiveness, which I gratefully acknowledge.

Abstract

Due to the emergence of new technologies, algorithm-assisted drivers are close to becoming a reality. In this thesis, different aspects of managing such drivers in a blocked-lane scenario are discussed. The first chapter presents an algorithm for the optimal merging of self-interested drivers. The optimal policy can include undesirable velocity oscillations. We propose measures for a central planner to eradicate them, and we test the efficiency of our algorithm versus popular heuristic policies. In the second chapter, a mechanism for positional bidding of the drivers is developed. It allows trading of highway positions of the drivers with heterogeneous time valuations, resulting in a socially beneficial outcome. The final chapter presents a deep learning policy for centralized clearing of the bottleneck in the shortest time. Its use is fast enough to allow future operational applications, and a training set consists of globally optimal merging policies.

Resum

En aquesta tesi, es discuteixen diferents aspectes de la gestió els conductors assistits per algoritmes en un escenari de carril bloquejat. El primer capítol presenta un algorisme de la gestió òptima dels conductors egoistes. La política òptima pot incloure oscil·lacions de velocitat no desitjades. Proposem mesures per a un planificador central per erradicar-les i comprovem l'eficiència del nostre algoritme enfront de les polítiques heurístiques populars. En el segon capítol, es desenvolupa un mecanisme per a la licitació posicional dels conductors. Permet negociar posicions per carretera dels conductors amb valoracions de temps heterogènies, donant lloc a un resultat socialment beneficiós. El capítol final presenta una política d'aprenentatge profund per a l'aclariment centralitzat del coll d'ampolla en el menor temps possible. El seu ús és prou ràpid per permetre futures aplicacions operatives, i un conjunt de formació consisteix en polítiques de fusió òptimes a nivell mundial.

Prologue

Smooth vehicular traffic flow can reduce environmental pollution, improve quality-of-life for commuters, and save billions of dollars in lost productivity. The nature of road traffic, however, poses significant modeling and technical challenges. It is highly dynamic and stochastic, with a large number of autonomous, self-interested agents, of varying skills and attitudes, with the decisions of one agent potentially affecting the actions of many others. Moreover, traffic is generally controlled only by coarse static rules and guidelines, which by necessity have to be applicable to a broad spectrum of driving conditions.

Despite many decades of sustained research by various communities, a number of research questions and challenges in optimal traffic control remain, as evident from the prevalence of traffic jams and delays to this day. Some of these research questions have taken on renewed urgency with the advent of sensors, vehicle-to-vehicle communication (V2V), vehicle-to-infrastructure communication (V2I), 5G, and driverless car technologies, which enable more information and more control, and allows algorithmic decision-making that can be based on clear objectives.

In this new transportation paradigm where driving is assisted by algorithms, either fully or partially, the real challenge, as should be apparent upon some reflection, is the *management* of algorithm-assisted vehicular traffic. For instance, what are fair, safe, and efficient real-time protocols for overtaking and lane-changing? How should one model the objective functions of different drivers? How can we take these different objective functions into account for the benefit of everyone? How can a central planner best control the flow? While the technology behind driverless cars is advancing at a breath-taking pace, much is still unknown about such issues of optimization, incentives, and collective decision-making by heterogeneous agents, and the resulting outcomes.

This thesis addresses these questions in application to one particular but common in practice scenario: mandatory lane changing. Due to an accident or construction work, one of the two lanes of a road segment is blocked, as illustrated in Figure 1. All drivers that happen to be on the blocked lane need to merge to the free-flowing lane to continue with their routes. Throughout the thesis, we consistently call one lane B (Blocked) and the other F (Free). This scenario is a backbone of the thesis, and in the different chapters, we model and analyze the problem from different angles. Nevertheless, all chapters share some similarities in the underlying assumptions. First, we assume that the drivers are "algorithmic"—that is, they are self-interested and rational, which is expected from algorithm-assisted drivers in the nearest future. Second, we cut smaller details of traffic dynamics and come to a stylized model. We focus on proving theoretical results and getting valuable insights.

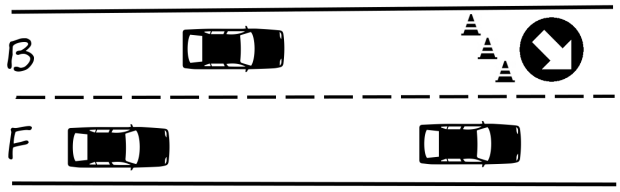


Figure 1: Illustration of a mandatory lane-changing scenario

In the first chapter, we develop a Dynamic Programming (DP) formulation of a single merging car, and we characterize the optimal policy. As we surprisingly found out, the optimal solution can oscillate between high and low velocities during merging attempts. Traffic oscillations are not a new phenomenon, as is discussed in detail in the literature review of Chapter 1. However, our results show that optimizing nature of algorithmic drivers can induce a new endogenous source of oscillations. We take a more in-depth look into possible causes via our theoretical analysis; we also propose a central planner strategy on how to diminish these oscillations. We compare our policy to popular heuristics policies such as "merge early" and "merge late". Cellular Automata simulations show a significant advantage of using our merging policy, reducing the merging driver's total travel time.

In the second chapter, we turn our attention to a different aspect of merging. We assume that all drivers on the road have different "urgency" of their trips or time valuation. This valuation is inherently a private value, and neither the central planner nor other drivers can freely observe it. The drivers are ready to give up some of their travel time to the merging driver in exchange for monetary payment. Can we organize an auction format that would allow finding equilibrium and possess desirable properties? Auctions are a rich topic, but many existing formats such as first-price auction, position auction, or VCG mechanism are not applicable in this rather complex environment. On the other hand, formats that can be found in traffic engineering literature typically lack equilibrium analysis. We propose two ad-hoc bidding schemes called Tail-To-Head (T2H) and Head-to-Tail (H2T). We show that equilibrium for both mechanisms can be computed via Dynamic Program, and provide analytical solutions. Furthermore, we show both analytically and numerically that Tail-To-Head is efficient in terms of social welfare and guarantees zero externalities, while Head-to-Tail excels at delivering high utility of the merging driver.

In the third chapter, we consider the merging problem from the central planner's position who can control drivers, either directly or through instructions or recommendations. There are several drivers on both lanes, and the central planner needs to find a way to clear the bottleneck as fast as possible. We formulate this

problem using Mixed-Integer Programming (MIP). For very few cars on the road, it can be solved fast. However, when there are more than a dozen vehicles, solving time is prohibitive for real-time recommendations. We propose Deep Neural Network architecture, which we train on a large massive of optimal MIP solutions. The output of this neural network is merging time and merging order. To obtain full recommended trajectories, we post-process these outputs using Linear Programming formulation. The resulting policy is fast to compute and is within 1.5% optimality gap from the exact MIP solutions.

Contents

Chapter 1

MANAGING LANE-CHANGING OF ALGORITHM-ASSISTED DRIVERS

1.1 Introduction

The development of new communication technologies opens up numerous possibilities for car traffic. It is conceivable that central planners, in a not-too-distant future, will be able to coordinate road traffic, or at least provide real-time recommendations to drivers. These developments raise the need for better understanding, via models and analysis, of optimal policies that take drivers' objectives and self-interested nature into consideration. Governments, car manufacturers, and start-up companies are already testing such scenarios—for instance, the EU-funded 5GCAR project (Sequeira et al. (2019))—and developing new standards and protocols to turn this potential into reality; see New York Times (2016), DOT (2018).

Meanwhile, issues of driver lane-changes and traffic oscillations remain to be not fully understood even for the present-day traffic conditions. For instance, highway design choices, such as short additional overtaking lanes on two-lane highways, are common in some countries, e.g., Spain. Are they beneficial, or in fact detrimental because of the forced merges they induce? Researchers have also linked accidents, stop-and-go traffic, and traveling shock-waves to lane changes. However, the fundamental reason as to how a single lane change can induce such traffic phenomena is still not fully understood.

Many macroscopic as well as microscopic models of traffic introduce exogenous stochastic terms to model human behavior such as over or under-reacting to traffic with random braking, or acceleration or reacting with time delay; for

instance, see Section G of the survey paper Helbing (2001). These stochastic errors then propagate upstream and make the traffic flow "unstable", occasionally even causing it to come to a complete halt for seemingly no reason. Hence, by implication, the "causal" factor for these puzzling stop-and-go waves and traffic oscillations is the suboptimal and/or unpredictable nature of human driving. This naturally raises the question that we address in this chapter: *would traffic oscillations go away if vehicles were controlled or assisted by algorithms, removing human driving "behavior" from the equation?*

In this chapter, we develop and analyze this question in the setting of a stylized model of mandatory lane changing. Due to an accident or construction work, one of the two lanes of a road segment is blocked. All drivers that happen to be on the blocked lane need to merge to the free lane in order to continue with their routes. Drivers on the blocked lane know that they have to merge, and each driver has the following decisions to make: the velocity at which she moves; whether, and when to attempt to merge to the free lane.

In our model, the blocked-lane drivers have only a *stochastic* view of the state of the free lane: although they know the exact locations of merging opportunities on the free lane at the time of their planning decisions, the gaps may not be there, or may be of insufficient size when they actually get there, due to uncertainty in the driving behavior of free-lane drivers.

What makes the problem interesting is that moving at higher velocity on the blocked lane reduces travel time in the short run but requires a larger gap to merge at that velocity, reducing the chance of merging and increasing the risk of a long wait at the blockage point. Consequently, blocked-lane drivers are trading off faster travel time at the current stage with an increased risk of not being able to merge before reaching the end of the road segment.

The contributions of this work are as follows:

1. We provide a Dynamic Programming (DP) formulation of the problem for a single merging car, and we characterize the optimal policy. We show that the latter exhibits a surprising structure: in the presence of merging uncertainty, it may be optimal for a driver, in certain parameter regimes, to oscillate between high and low velocities while attempting to merge; i.e., rational self-interested driving behavior, based on global stochastic optimization, may lead to oscillatory traffic patterns. So on the *descriptive* side, our main finding is that traffic oscillations can arise endogenously due to drivers' global optimization decisions. This adds another potential source of disturbance, in addition to suboptimality or unpredictability in their driving behavior, which has been the focus of existing literature. So algorithm-assisted driving, with each driver minimizing their individual travel times, may eliminate some sources of velocity oscillations, but adds a new one;

2. On the *normative* side, we show how a traffic regulator can set velocity limits so that oscillations do not arise even in the presence of algorithm-assisted drivers, optimizing their total expected travel time. This would lead to safer, smoother, and more efficient traffic flow. We use Monte Carlo discrete-event simulations of more realistic traffic environments to show that merging policies arising from our DP approach can both improve throughput and reduce delays compared to various alternative policies proposed in the literature.

Our research highlights the additional disturbances that algorithmic drivers can bring to traffic. The fundamental reason is that while manual drivers can do *local/heuristic* optimization based on immediate surrounding vehicles and conditions, algorithm-assisted drivers can make *globally* optimal decisions because of their computational power and ability to process significant amounts of information on-the-fly. This research highlights the fact that, if left to rational selfish objectives, uncertainty on other agents' future actions and locations can lead to undesirable phenomena. Hence, altruistic or cooperative behavior, e.g., courtesy yielding, has to be incentivized or imposed on such algorithms for the greater good.

Our approach is fundamentally different from traffic modeling via differential equations that link macroscopic measures such as density and flow; such equations may fit observed data well, but do not provide causal or normative insights. The very simplicity of our model also differentiates it from complex microscopic simulations that are not amenable to analysis; such simulation set-ups capture many features of human driving behavior, but often have too many confounding factors to pinpoint fundamental causes.

The chapter is organized as follows. To motivate our work, we analyze in Section 1.2 a large highway dataset and show that traffic oscillations are not uncommon around mandatory merging points, even with manual driving. In Section 1.3 we survey the related literature, focusing primarily on articles that research lane changing. In Section 1.4, we present a stylized discrete-stage and velocity model of lane changing in forced-merge scenarios, and show that velocity oscillations that can occur at optimality. In Section 1.5, we show how a central planner can set velocity limits to eliminate these kinds of oscillations. In Section 1.6 we relax some of the simplifying assumptions and test the robustness of our model. In Section 1.7 we introduce details of our numerical set-up and investigate the throughput-delay tradeoffs of the optimal policy in more realistic environments, through Monte Carlo discrete-event simulations. In Section 1.8 we conclude.

1.2 Data and evidence

In this section, we analyze merging instances from the on-ramp in the US Highway 101 Dataset, which is part of the Next Generation Simulation (NGSIM) project of the US DOT[1]. The particular road segment has 5 main lanes and an auxiliary lane, with both an on-ramp and an off-ramp and total length of 2100 feet. Video data over three 15-minute intervals, recorded between 7.50am and 8.35am, has been transcribed into high-precision snapshots of x and y coordinates and velocities at 100 millisecond intervals. The data is invaluable for calibration of simulations, and understanding of human driving behavior. Our analysis focuses on velocity oscillations in one of the two forced-merging situations in the data: vehicles coming from the on-ramp (lane 6) have to merge to the rightmost main lane (lane 5) within a specified distance, mimicking our blocked-lane scenario. In total we analyze 280 such cases. To the best of our knowledge, the NGSIM dataset has not been analyzed before with the focus on driver-level oscillations while attempting to merge.

The main factor in identifying a merge attempt was a temporal traversal movement towards lane 5, which we identified by checking difference over a period on the local x-coordinate. The x-axis is almost perpendicular (but not perfectly) to the direction of the lanes. As a general rule, a merge is identified if a vehicle is moving towards the target lane at the velocity of at least 5 feet per second, and if a period of such movement could be observed for at least 10 consecutive frames (i.e. one second).

To identify velocity oscillations, we use the following criteria:

- Cases where a vehicle left the auxiliary lane within 4 seconds were discarded, as there was no time for several merging opportunities.
- We compare the velocity of the merging vehicle to the velocity of the vehicle ahead on the same lane, checking for correlations. This is to rule out the possibility that the merging car simply follows the velocity pattern of the preceding car.
- We track the velocity of the car behind on the target lane as additional confirmation of merge attempt: in many cases, the car on the target lane reacts by decelerating, and sometimes, by accelerating.
- Our main focus is on the cases when the driver oscillates while merging with the same driver on the target lane or skips one or more target lane drivers after a failed attempt.

[1] https://data.transportation.gov/Automobiles/Next-Generation-Simulation-NGSIM-Vehicle-Trajector/8ect-6jqj

In total, we found 27 cases (9.65% of all merges) where velocity oscillations of merging cars satisfy the above specification. If we only consider the cases where it took more than 4 seconds for the driver to merge, then 19.7% merges show merging-related oscillations. Usually, oscillations are not significant and appear as small (3-7 feet/s) and short (1-2 seconds) velocity drop. However, there are also a few cases where the driver shows a consistent and significant velocity drop. In Fig. 1.1 we demonstrate four such examples of velocity oscillations of the merging car. One particularly interesting example is vehicle 2122, which changes velocity from low to high and then back to low while continuing merging attempts. Car 1789 shows local temporal drops of velocity which coincide with merging attempts. Cars 344 and 1098 demonstrate examples of merging attempts at a lower velocity if compared to earlier or later ones. We emphasize that many of those oscillations are not small variations in velocity which are usually part of the execution of the merging maneuver, but rather more strategic change of velocity. This is especially clear in case of the third merging attempt of car 1098 in Fig. 1.1.

In addition, we found that *courtesy yielding*, where a target-lane driver slows down, presumably to create a large enough gap for the merging driver, does occur occasionally but not always. There are also occasional cases of *blocked merging*, where a target-lane driver not only does not courtesy-yield but instead closes the gap, presumably to prevent merging.

It is, of course, impossible to attribute precise causal factors to the above phenomena from a purely observational dataset, such as the NGSIM one. In Section 1.4.2, we list several reasons due to which oscillations may arise in manual driving.

Since algorithmic driving is based on precise objective functions, our theoretical findings highlight the need for "fair" protocols to be built into algorithm-assisted driving: if vehicles on the target lane have no incentives to show "courtesy" to merging drivers, then some form of altruistic/cooperative behavior may have to be imposed by a central planner; otherwise, velocity oscillations, detrimental to the overall traffic flow, may arise.

We complement our analysis with further evidence on velocity oscillations, this time from the literature. In Table 1.1, we review papers that perform empirical analysis on merging, and show either oscillations or shock-waves originating from lane changes. In Section 1.3, we further review papers on lane changing that are based on simulations and theoretical models.

1.3 Literature on traffic merging and lane-changing

Traffic modeling follows either a macroscopic approach, studying traffic at the population/fluid level, or a microscopic approach, studying the behavior of indi-

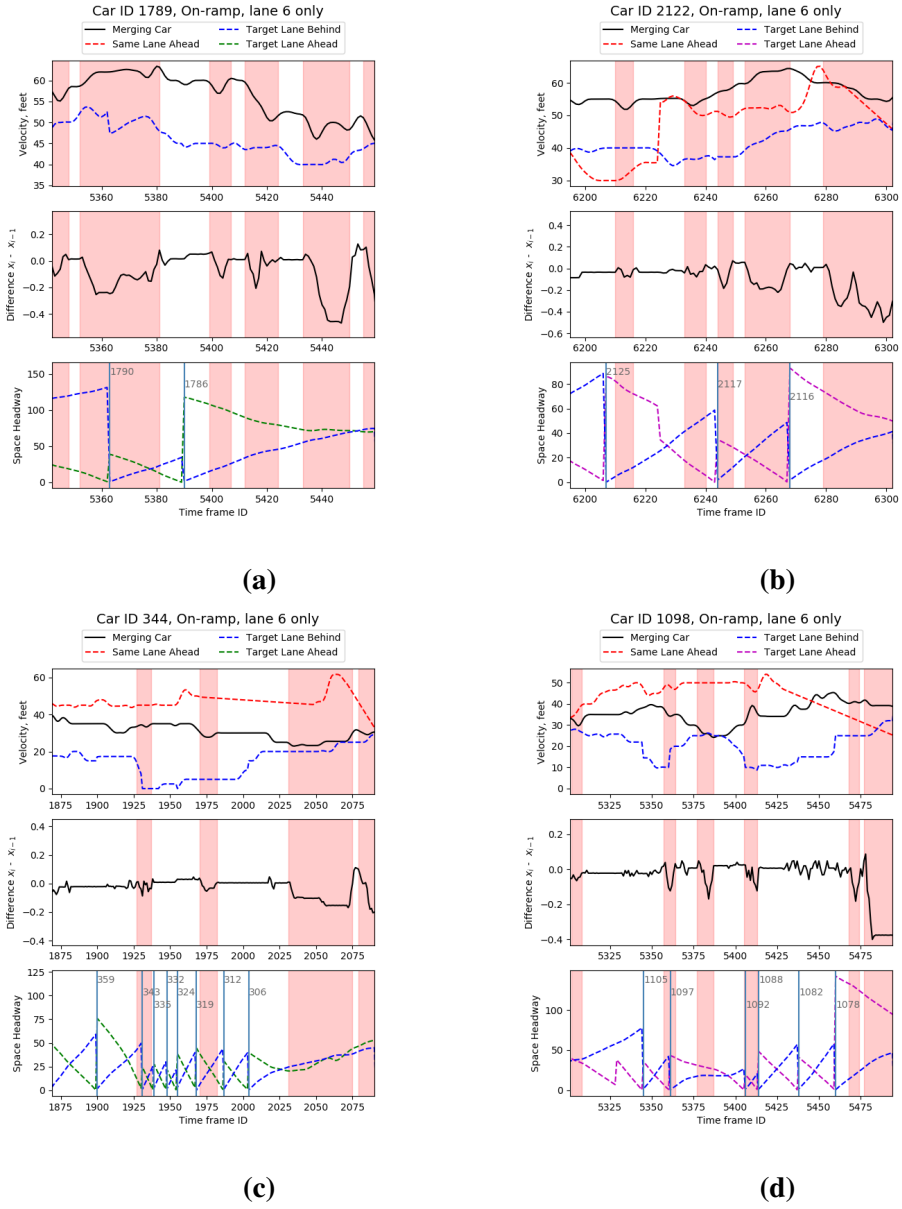


Figure 1.1: Velocity oscillations of merging drivers. Each of 4 cases displays 3 subplots. *Top:* velocity of the merging driver, the velocity of the driver ahead on the same lane, and the driver behind on the target lane. *Middle:* traversal velocity of the merging car. *Bottom:* distance to cars behind and ahead (on the target lane), vertical bars show when the driver behind changes, along with a new car ID number. Identified merging attempts (which often comes with velocity drop) are shown in red strips. Time measured in frames, one frame is 0.1 seconds.

Table 1.1: Empirical, observed data-based papers studying lane changes or forced-merging at on and off ramps, focusing mostly on describing the effects of merging. Data is transcribed from video recordings of sections ranging from 150m to 1 km during periods of 45min to 8 hours. Further empirical studies, not necessarily focused on oscillations but demonstrating their presence or effects in traffic are Hidas (2005), Kerner et al. (2006), Schönhof and Helbing (2007), Ahn and Cassidy (2007), Zheng et al. (2010), Zheng et al. (2013), Zheng (2014), Ahn et al. (2010), Knoop et al. (2018), Sheu (2013) and Wei et al. (2000).

Reference	Data & Methodology	Focus	Main modeling elements	Conclusions
Daamen et al. (2010), Loot (2009)	Data at on-ramps from helicopter-mounted video in Netherlands	Gap acceptance theory for merge behavior	Data Analysis, verification of gap-acceptance, car-following and relaxation theories	During free-flow, most vehicles merge at the first half of the acceleration lane; under congested traffic conditions, relatively more merges happen at the end of the lane—supporting a hypothesis of travel-time minimization objective
Mauch and Cassidy (2002)	Observed data on a 10-km stretch	Modeling oscillations and testing the Kinematic Wave Theory	Descriptive; Estimation of parametric equations	Formation of new oscillations strongly correlated with vehicle lane-changing and less to endogenous car-following effects in moderately dense queues. "*...marked improvements in traffic flow theories will likely come by incorporating lane-changing effects*"
Lee et al. (2016)	Stochastic model of lane changes based on the NGSIM data	Data Analysis; Develop a probability model for a discretionary lane change	Parametric equation estimation by regression	Explains discretionary lane changes under relative velocity (both positive as well as negative), and relative lead gap

vidual drivers[2]. Both strands of literature are too vast, and the models too many, to even attempt to summarize here. Instead we refer to a relatively recent survey due to Helbing (2001) from a macroscopic point of view, and to Chowdhury et al. (2000) from the microscopic Cellular-Automata perspective.

[2]A smaller body of work is classified as *mesoscopic* that attempts to describe microscopic dynamics as a function of macroscopic parameters.

In the rest of this section, we concentrate only on research relevant to lane-changing. Over the last two decades, lane-changing has gained prominence in the traffic literature. It is increasingly recognized as one of the principal reasons for creating disturbances in traffic flows, as well as oscillations leading to stop-and-go waves in traffic patterns.

The merging problem in particular is the topic of early influential papers in Operations Research, such as Evans et al. (1964) and McNeil and Smith (1969). These papers build detailed models of merging, and compute quantities of interest such as the merging delay from on-ramp to highway. However, while traffic flow is modeled as a stochastic process, the modeling of individual behavior and incentives of drivers are absent from these early works. After a rather long hiatus, the Operations Research community is picking up this important topic, as in Jain and Smith (1997), Heidemann (2001), based on queueing theory, as well as Gregoire et al. (2015), Le et al. (2015) and Como et al. (2016), from a control and optimization perspective.

Merging is essentially an optimal stopping problem with similarities to the classical "parking problem" (see McQueen and Miller Jr. (1960)). The main difference in our setting is the existence of an additional decision variable, the velocity, which affects in a non-trivial way both the travel time and the probability of finding space to merge.

Many empirical works have focused on how drivers behave at merging points, e.g., Chang and Kao (1991), Hounsell et al. (1992), Hidas (2005), Liu and Hyman (2012), and Knoop et al. (2012). Many models have been proposed on explaining the data with parametric forms for the merge probabilities and driver behavior, as in Gipps (1986), Kita (1999), Daganzo (2002a), Daganzo (2002b), Hidas (2002), Jin (2010), Laval and Daganzo (2006), Kondyli and Elefteriadou (2011), Zhang et al. (2012), Keyvan-Ekbatani et al. (2016) and Ngoduy et al. (2019). Choudhury et al. (2007), Choudhury and Ben-Akiva (2013) and Kesting et al. (2007) propose sophisticated merging decision structures and regimes, including potential cooperation with cars in the target lane.

A notable difference with our model, though, is the lack of a velocity decision, with the exception of Sun and Elefteriadou (2010) which does have a velocity element. We refer the reader to Rahman et al. (2013), and Zheng (2014) for a broader review of lane-changing models. The thesis of Ahmed (1999) proposes a logistic-regression model to determine the probability of a driver choosing to merge, while Toledo et al. (2003) propose an integrated lane-changing model that works for mandatory and discretionary considerations based on a gap-acceptance model.

Note that while the vast majority of the literature concentrates on local explanations of driver choices, that is, based only on immediate surroundings, Nilsson et al. (2016) and Rios-Torres and Malikopoulos (2017) propose global driver op-

timization models based on optimal control theory.

Tarko et al. (1999) and McCoy and Pesti (2001) have documented experiments with "zipper merging" and "late merging" strategies, and have reported on improvements over the uncontrolled random merging benchmark. A similar approach is also taken in Grillo et al. (2008). These works advance our understanding of typical traffic scenarios, but lack an explicit control/optimization dimension, as well as any theoretical support of their findings. Baykal-Gürsoy et al. (2009) and Duret et al. (2010) can be viewed as a step in that direction: the authors introduce theoretical models for predicting the effect of lane-changing using kinematic-wave theory, with a future plan to use these results for better decisions in traffic management.

Given the recent interest in driverless cars, a number of researchers have turned their attention to connected automated vehicles and how they interact with manually driven vehicles. In this stream, some notable papers are the following: Buisson et al. (2018), Lee et al. (2019), Zheng et al. (2020). Systems such as adaptive cruise control for automated vehicles are proposed, either at the individual or cooperative level (Van Arem et al. (2006), Kesting et al. (2008)). They can control car-following and lane-changing decisions of automated vehicles, leading to better flow characteristics, including diminishing stop-and-go waves (Wang et al. (2015), Wang et al. (2016)).

Finally, a related strand of literature lies in the intersection of Traffic Engineering and Computer Science, where lane-changing is studied through simulations at a microscopic level (for instance, merging rules studied in MITSIM and Yang and Koutsopoulos (1996)). From this literature, let us highlight Ebersbach and Schneider (2004) and Han and Ko (2012), which focus on merging into a highway with a blocked lane and mandatory on-ramp merging, respectively.

1.4 Modeling and analysis of merging

In this section we study a discrete-time and discrete-velocity model of merging. By using a stylized model, we hope to provide insights into the dynamics of the merging process. Later, we show that the same insights hold in continuous-time and/or continuous-velocity models.

1.4.1 Mandatory lane-changing under uncertainty

Consider a two-lane road, with traffic moving in the same direction on both lanes. One lane is blocked unexpectedly due to accident, construction, or maintenance work, while traffic on the other lane continues to flow freely. We refer to the lanes as B (Blocked) and F (Free). Every car that happens to be on lane B when

the blockage occurs, needs to merge to lane F at some place before the blockage point, i.e., we have a scenario of *mandatory lane-changing*.

Table 1.2: Notation used in the chapter

Notation	To represent
B	Blocked lane
F	Free lane
d_k	Distance between merging positions k and $k+1$
v_k^s, v_k^e	velocity at the start and end of stage k respectively
v_H, v_L	Upper (superscript H) and lower (superscript L) bounds on velocity respectively
u_k	0-1 decision variable of the driver to merge
V_k^F	Velocity on the F lane in stage k modeled as i.i.d random variables
Y_k	0-1 random variable; 1 means there is a sufficient gap to allow merging at end of stage $k-1$
$q(\cdot)$	Probability of the Bernoulli r.v Y_k as a function of velocities and distances between merging points
$c(\cdot)$	Impact of the relaxation on the merging car in terms of travel time
P	Penalty in expeted time units if the car reaches the blockage without merging
$T_k^B(\cdot)$	Value function representing optimal travel time to the blockage point for the car in stage k as a function of the velocity choice

We focus on the behavior of a single car on the blocked lane, whose goal is to merge to the free lane and get through the bottleneck in the shortest possible time. The timing of decision making is fundamentally discrete, in the sense that the driver of a vehicle on the B-lane can merge only at N specific gaps/merging opportunities present on the free lane. Let the starting point of the car be at merging position 0, and the bottleneck be at merging position $N+1$. We denote the distance between positions k and $k+1$ by $d_k, k = 0, \ldots, N$. We assume that the driver knows the location of the gaps, e.g., via inspection or V2I information. This discretizes the road segment into cells that we call *stages*, and we define stage k as the road segment between positions k and $k+1$. We use the superscripts B and F to denote stages on the B and F lane, respectively.

The vehicle reaches stages 1 to N sequentially. The driver on lane B has two decisions to make at the start of stage $k \in \{0, \ldots, N-1\}$: (a) whether to merge to lane F at the end of stage k, if there is a sufficiently large gap to do so; and (b) whether to accelerate, decelerate, or keep the same speed during stage k, effectively determining the velocity of the car at the end of stage k. We use a binary decision variable, u_k, whose value is 1 if the driver's decision is to try merge, and the value 0 otherwise. Let v_k^s and v_k^e denote the velocity of the vehicle at the start and at the end of stage k, respectively. In our benchmark model, we assume that these velocities can only take two values, a high and a low one, v_H and v_L, respectively. Later, we show that our insights hold even when the velocity decision is continuous.

Three important assumptions underlie our model:

1. The driver on lane B cannot predict with certainty the state of lane F at the end of a given stage, i.e. before she actually reaches that point. In other

words, while the *ex ante* start/end of the stages are synonymous to merging opportunities, they may or may not materialize when the B-lane driver actually reaches there. This is because the size of the gap itself depends on the driving behavior of vehicles on the free lane, specifically the vehicles on either side of the gap, which cannot be predicted accurately, since there is no communication/coordination between vehicles.

2. Decisions and actions are not executed instantaneously. Instead, drivers (both human, as well as algorithmic) plan a bit ahead to execute an action. In our case, the driver decides to merge at the start of the stage and the decision — if it is to merge — is attempted at the end of the stage. Such a lag, in both information and action, is certainly realistic for humans, and will be reduced but unlikely to go away completely even in fully automated driving scenarios.

3. For safety, convenience, and fuel efficiency, a constant acceleration or deceleration rate is applied whenever a change in velocity is mandated. In particular, if varying acceleration/deceleration patterns were allowed, the fastest way to reach the end of a stage with a target velocity could very well be to accelerate and subsequently decelerate within the stage. These trivial, in some sense, oscillations are *not* the focus of our work. In contrast, what we uncover is that velocity oscillations can arise across stages, even if drivers maintain a constant acceleration or deceleration within each stage.

We assume that the velocity of the F lane at stage k is a random variable, V_k^F, with known distribution. To be more precise, this is the velocity of a free-lane vehicle that happens to be between the k^{th} and the $(k+1)^{st}$ merging position. We also assume that once the vehicle merges to lane F, it never merges back to the lane B. Indeed, when drivers are aware of a blockage, constant lane-changing is rarely observed in practice. As a result, conditional on a successful merging, the expected remaining travel time is a simple function of these velocities, and of the point where the merging occurs. In particular, the (expected) time to traverse stage k on lane F is equal to $\mathbb{E}\big[d_k \big/ V_k^F\big]$. While on lane B, the time that it takes to traverse stage k is deterministic and equal to $2d_k \big/ \big(v_k^s + v_k^e\big)$, i.e., reciprocal to the vehicle's average velocity, as we assume constant acceleration or deceleration.

Let Y_{k+1} be an indicator random variable that takes the value $y_{k+1} = 1$ if a sufficiently large gap (to allow merging) is realized at the end of stage k on the free lane, at the point in time when the blocked lane driver is attempting to merge; and $y_{k+1} = 0$, otherwise. Following the literature, we model the random variable of finding a gap to merge $Y_{k+1} = q\big(v_k^e, V_k^F, d_k\big)$, with a (stationary) function $q(\cdot)$ of the velocity of the blocked-lane car v_k^e, velocity on the free lane V_k^F at the end of stage k, and the distance d_k between the k^{th} and the $(k+1)^{st}$ merging position.

It should be intuitively clear why the first two arguments of q affect the ability to merge; the last one, d_k, can also play a role because conditional on an unsuccessful merging attempt in the previous stage, the length of the current stage may affect the driver's uncertainty on merging in the near future.

A merging car creates a local disturbance in the flow of the target lane; in our case, lane F. This disturbance affects one or more cars on the target lane, as well as the merging car itself. This phenomenon, termed *relaxation* in the literature, is well studied and understood empirically; see Zheng et al. (2013) for a recent review. We do not attempt to model the details of the relaxation process as our focus is on the merging car and its objective of minimizing its total travel time. Hence, for simplicity and tractability, we capture the aggregate impact of relaxation on the merging car by a *merging penalty* which, following the literature, we assume to be a (stationary) function $c\big(v_k^e, V_k^F, d_k\big)$ of the velocity of the B-lane car and of the F lane at the end of stage k, as well as of the distance between the k^{th} and the $(k+1)^{st}$ merging position.

If the driver reaches the bottleneck stage without having merged to lane F, she incurs an expected late-merging penalty of P time units, and then exits the bottleneck. (In this late-merging penalty we also include the traveling time from the last merging opportunity, at position N, to the bottleneck, at position $N+1$.) This penalty models the fact that the car needs to come to a complete stop, and then wait for a large enough gap on lane F in order to bypass the blockage. The regime of interest is when P is reasonably large, as otherwise there may be no incentive to merge and the solution to the driver's problem becomes trivial.

The B-lane driver's problem is then a *Dynamic Program* (DP) with the following primitives.

State: The state of the system at the start of stage k consists of the lane on which the car is located, F or B, and its current velocity v_k^s, in case the car is still on the B lane.

Decision: The decision at stage k, assuming that the car is still on lane B, consists of two components. The first component represents the driver's intention to merge at the end of stage k, $u_k \in \{0,1\}$. The second component is the velocity to achieve at the end of stage k, $v_k^e \in \{v_L, v_H\}$. If the car has already merged to lane F at stage k, then there is no decision to be made.

Uncertainty: The uncertainty is on the existence of a sufficient gap on lane F at the end of stage k (equivalently, at merging position $k+1$), which is captured by the indicator random variable Y_{k+1}.

Dynamics: If the car is on lane B at the end of stage k, and $u_k = y_{k+1} = 1$, then the car merges to lane F at the start of stage $k+1$. Otherwise, the car remains on lane B and moves at velocity $v_{k+1}^s = v_k^e$.

Objective Function: The objective for the B-lane vehicle is to take decisions to minimize expected remaining travel time.

We denote by $T_k^B\left(v_k^s\right)$ the optimal expected remaining travel time, if at the start of stage k the car is on lane B and moves at velocity v_k^s. Similarly, T_k^F represents the cost-to-go if the car is on lane F. We are interested in calculating $T_0^B(v_0^s)$ and, along the way, in obtaining the optimal merging and velocity control policy. This can be done via the following DP recursion:

$$T_k^B\left(v_k^s\right) = \min_{v_k^e \in \{v_L, v_H\}} \left\{ \frac{2d_k}{v_k^s + v_k^e} + \min \left\{ \underbrace{T_{k+1}^B\big(v_k^e\big)}_{u_k=0}, \right. \right. \tag{1.1}$$

$$\left. \left. \underbrace{\mathbb{E}\Big[q\big(v_k^e, V_k^F, d_k\big)\Big(T_{k+1}^F + c\big(v_k^e, V_k^F, d_k\big)\Big) + \Big(1 - q\big(v_k^e, V_k^F, d_k\big)\Big)T_{k+1}^B\big(v_k^e\big)\Big]}_{u_k=1} \right\} \right\},$$

for $k = 0, \ldots, N-1$, with boundary condition:

$$T_N^B\left(v_N^s\right) = P,$$

where

$$T_k^F = \mathbb{E}\left[\sum_{j=k}^{N} \frac{d_j}{V_j^F}\right], \qquad k = 0, \ldots, N.$$

Note that in Eq. (1.1) decisions are made at the start of stage k, whereas the realization of the random variables occurs when they reach the gap. Hence, the blocked-lane driver minimizes their expected travel time with respect to the distribution of V_k^F.

Structure of the optimal policy

To simplify the notation and analysis, we assume that $d_0 = d_1 = \ldots = d_N = 1$, and that the random variables $\left\{V_k^F\right\}_{k=0}^N$ are independent and identically distributed with a probability law V_F on $(0, \infty)$, and we denote $\mathbb{E}\big[V_F\big] = \overline{v}^F$. From an analytical standpoint, the former assumption is basically a normalization, made without much loss of generality. The latter assumption, however, is a stronger one: while assuming that the velocities of the free lane at different stages are identically

distributed seems reasonable for a lane that is free flowing, assuming that they are also mutually independent could be restrictive. In fact, one would expect at least a weak dependence between the velocities of consecutive stages. Assuming independence makes the analysis tractable. To validate our insights, in the simulation set-up presented later, we consider the model presented above in its full generality.

Note that the independence assumption for the sequence $\{V_k^F\}_{k=0}^N$ implies that $T_{k+1}^B(v_k^e)$ and T_{k+1}^F are independent of $q(v_k^e, V_k^F, d_k)$. Moreover, in order to ease the notation, we define

$$q(v_k^e) \equiv \mathbb{E}\left[q(v_k^e, V_F, 1)\right],$$

and

$$c(v_k^e) \equiv \frac{\mathbb{E}\left[q(v_k^e, V_F, 1)c(v_k^e, V_F, 1)\right]}{q(v_k^e)},$$

where the expectations are taken with respect to the distribution of V_F.

We assume that $\overline{v}^F \leq v_L$, i.e., we have a relatively congested free lane, which is typical in mandatory lane-changing scenarios.

We let $q(v_L) = q_L$ and $q(v_H) = q_H$, and we assume that $q_L > q_H > 0$. This is motivated by the fact that, all else being equal, the driver requires less space to merge if the difference between the velocities in the two lanes is small; see Lee (2006), Choudhury et al. (2010) and Toledo et al. (2003) for exact estimates of this effect; and in our case v_L is closer to $\overline{v}^F$, as our main interest is in scenarios where the free lane is relatively dense. We discuss the empirical background and modeling aspects of merging such as gap-acceptance in more detail in Section 1.7.1.

Similarly, we let $c(v_L) = c_L$ and $c(v_H) = c_H$, and we assume that $c_H < c_L$. The reasoning is that, on average, merging requires a larger gap if $v_k^e = v_H$, but conditional on a successful merge event, the relaxation process is smoother overall. In-depth studies of relaxation are limited, but statistical support for the aforementioned claim can be found in van Beinum et al. (2018). A more detailed discussion of relaxation and merging penalties can be found in Section 1.7.1.

For technical reasons, we assume that the late merging penalty satisfies $P \gg 1 \,/\, \min\{v_L, \overline{v}^F\} + c_L$, which is consistent with the intuition that penalty needs to be "large enough."

With these assumptions and notation, the DP formulation reduces to:

$$T_k^B(v_k^s) = \min_{v_k^e \in \{v_L, v_H\}} \left\{ \frac{2}{v_k^s + v_k^e} + \min \left\{ T_{k+1}^B(v_k^e), q(v_k^e)\left(T_{k+1}^F + c(v_k^e)\right) + \left(1 - q(v_k^e)\right) T_{k+1}^B(v_k^e) \right\} \right\}, \quad (1.2)$$

for $k = 0, \ldots, N-1$, with boundary condition:

$$T_N^B\left(v_N^s\right) = P, \tag{1.3}$$

and

$$T_k^F = \frac{N-k+1}{\overline{v}^F}, \qquad k = 0, \ldots, N. \tag{1.4}$$

Let us denote by (u_k^*, v_k^{*e}) the element of an *optimal* policy at the k^{th} stage. Our initial finding is quite intuitive, and follows directly from the DP recursion in Eq. (1.2).

Proposition 1.1. *Consider the mandatory lane-changing problem in Eq.* (1.2)-(1.4)*. If it is optimal not to merge at stage k, i.e., $u_k^* = 0$, then it is optimal to have high velocity during stage k, i.e., $v_k^{*e} = v_H$.*

In other words, the decision pair $(0, v_L)$ can never be part of an optimal merging and velocity control policy. In order to characterize further the structure of optimal policies, we need to introduce following notation:

$$B_k\left(v_k^s\right) \equiv \frac{2}{v_k^s + v_L} - \frac{2}{v_k^s + v_H} + q_L\left(T_{k+1}^F + c_L\right) - q_H\left(T_{k+1}^F + c_H\right),$$

and

$$\Delta T_{k+1} \equiv (1-q_H) T_{k+1}^B(v_H) - (1-q_L) T_{k+1}^B(v_L).$$

In our formulation, merging can be viewed as an optimal stopping problem, where the driver incurs a lump-sum cost $T_{k+1}^F + c(v_k^e)$ if she merges successfully at the end of stage k, and after which the whole process terminates. In that light, $B_k\left(v_k^s\right)$ can be interpreted as the expected loss of choosing v_L over v_H at stage k, while ΔT_{k+1} as the expected continuation benefit of that same decision.

We now prove two technical lemmas, which will facilitate proofs of our main theoretical results.

Lemma 1.1. $T_k^B(v_H) \leq T_k^B(v_L)$*, for all* $k \in \{1, \ldots, N-1\}$.

Proof. Fix an arbitrary $k \in \{1, \ldots, N-1\}$. The result follows directly from Eq. (1.2), by noting that the objective function in the optimization problem defining $T_k^B(v_H)$ takes lower values than the objective function in the optimization problem defining $T_k^B(v_L)$, for every feasible solution. ■

Lemma 1.2. *If it is optimal to merge at stage k, i.e., $u_k^* = 1$, then*

$$\frac{(v_L - v_H)}{v_L(v_L + v_H)} \leq T_k^B(v_H) - T_k^B(v_L) \leq \frac{(v_L - v_H)}{v_H(v_L + v_H)}.$$

Proof. Let us assume that it is optimal to merge, and that $v_k^{*e} = \tilde{v}$ is the optimal velocity at the end of stage k, if the velocity at the start of the stage is v_H. Eq. (1.2) implies that

$$T_k^B(v_H) = \frac{2}{v_H + \tilde{v}} + q(\tilde{v})\left(T_{k+1}^F + C(\tilde{v})\right) + \big(1 - q(\tilde{v})\big)T_{k+1}^B(\tilde{v}).$$

Note that,

$$T_k^B(v_L) \leq \frac{2}{v_L + \tilde{v}} + q(\tilde{v})\left(T_{k+1}^F + C(\tilde{v})\right) + \big(1 - q(\tilde{v})\big)T_{k+1}^B(\tilde{v}),$$

since $\tilde{v}$ is a feasible, but not necessarily optimal solution to the optimization problem defining $T_k^B(v_L)$. Together, the two equations imply that

$$\begin{aligned} T_k^B(v_H) - T_k^B(v_L) &\geq \frac{2}{v_H + \tilde{v}} - \frac{2}{v_L + \tilde{v}} = \frac{2(v_L - v_H)}{(v_L + \tilde{v})(v_H + \tilde{v})} \\ &\geq \frac{2(v_L - v_H)}{(v_L + v_L)(v_L + v_H)} = \frac{(v_L - v_H)}{v_L(v_L + v_H)}, \end{aligned}$$

where the right-most inequality uses the fact that the numerator is negative, so that the fraction decreases by substituting v_L for $\tilde{v}$.

This proves the lower bound on $T_k^B(v_H) - T_k^B(v_L)$. The upper bound can be proved by considering that $v_k^{*e} = \hat{v}$ is the optimal velocity at the end of stage k, if the velocity at the start of the stage is v_L, and working similarly. ■

Proposition 1.2. *Consider the mandatory lane-changing problem in Eq. (1.2)-(1.4). An optimal policy has the following multi-threshold structure:*

1. $(u_k^*, v_k^{*e}) = (0, v_H)$ *if and only if* $T_{k+1}^B(v_H) \leq T_{k+1}^F + c_H$*;*
2. $(u_k^*, v_k^{*e}) = (1, v_H)$ *if and only if* $T_{k+1}^B(v_H) \geq T_{k+1}^F + c_H$ *and* $\Delta T_{k+1} \leq B_k(v_k^s)$*;*
3. $(u_k^*, v_k^{*e}) = (1, v_L)$ *if and only if* $T_{k+1}^B(v_H) \geq T_{k+1}^F + c_H$ *and* $\Delta T_{k+1} \geq B_k(v_k^s)$.

Combining the last two cases, we obtain that it is optimal to merge if and only if $T_{k+1}^B(v_H) \geq T_{k+1}^F + c_H$.

Proof. **Part (i)**: Assume that $(u_k^*, v_k^{*e}) = (0, v_H)$ is the optimal solution. This implies that

$$\begin{aligned}
&\frac{2}{v_k^s + v_H} + T_{k+1}^B(v_H) \\
&\quad \leq \min_{v_k^e \in \{v_L, v_H\}} \left\{ \frac{2}{v_k^s + v_k^e} + q(v_k^e)(T_{k+1}^F + C(v_k^e)) + (1 - q(v_k^e))T_{k+1}^B(v_k^e) \right\} \\
&\quad = \frac{2}{v_k^s + v_H} + q_H(T_{k+1}^F + c_H) + (1 - q_H)T_{k+1}^B(v_H),
\end{aligned}$$

where the inequality follows from Eq. (1.2), for $u_k^* = 0$; and the equality from Proposition 1.1, which disqualifies $(0, v_L)$ as an optimal solution. Rearranging terms, we have that $T_{k+1}^B(v_H) \leq T_{k+1}^F + c_H$, since $q_H > 0$.

Conversely, assume that $T_{k+1}^B(v_H) \leq T_{k+1}^F + c_H$. We note that

$$\begin{aligned}
&\min_{v_k^e \in \{v_L, v_H\}} \left\{ \frac{2}{v_k^s + v_k^e} + q(v_k^e)(T_{k+1}^F + C(v_k^e)) + (1 - q(v_k^e))T_{k+1}^B(v_k^e) \right\} \\
&\quad \geq \frac{2}{v_k^s + v_H} + \min_{v_k^e \in \{v_L, v_H\}} \left\{ q(v_k^e)T_{k+1}^B(v_H) + (1 - q(v_k^e))T_{k+1}^B(v_H) \right\} \\
&\quad = \frac{2}{v_k^s + v_H} + T_{k+1}^B(v_H).
\end{aligned}$$

While the equality is derived through simple algebra, the inequality is based on the following facts: (a) $T_{k+1}^B(v_H) \leq T_{k+1}^F + c_H \leq T_{k+1}^F + c_L$, by assumption and the ordering of merging penalties; (b) Lemma 1.1, which implies that $T_{k+1}^B(v_H) \leq T_{k+1}^B(v_L)$. Combined with Proposition 1.1, which precludes $(0, v_L)$ as an optimal solution, the above inequality implies that $(u_k^*, v_k^{*e}) = (0, v_H)$.

Putting the two parts together, we establish the first part of the proposition:

$$(u_k^*, v_k^{*e}) = (0, v_H) \iff T_{k+1}^B(v_H) \leq T_{k+1}^F + c_H.$$

Parts (ii)-(iii): In part (i), we have established that the condition $T_{k+1}^B(v_H) \geq T_{k+1}^F + c_H$ implies that $u_k^* = 1$. Moreover, the condition $\Delta T_{k+1} \leq B_k(v_k^b)$ is equivalent to

$$\begin{aligned}
&\frac{2}{v_k^s + v_H} + q_H(T_{k+1}^F + c_H) + \big(1 - q_H\big)T_{k+1}^B\big(v_H\big) \\
&\qquad \leq \frac{2}{v_k^s + v_L} + q_L(T_{k+1}^F + c_L) + \big(1 - q_L\big)T_{k+1}^B\big(v_L\big),
\end{aligned}$$

if we use the definitions of ΔT_{k+1} and $B_k(v_k^s)$, and rearrange terms. This implies that the high velocity, v_H, is optimal, conditional on being optimal to merge. Combining the two arguments, we have that

$$T_{k+1}^B(v_H) \geq T_{k+1}^F + c_H \text{ and } \Delta T_{k+1} \leq B_k(v_k^s) \iff (u_k^*, v_k^{*e}) = (1, v_H),$$

and

$$T_{k+1}^B(v_H) \geq T_{k+1}^F + c_H \text{ and } \Delta T_{k+1} \geq B_k(v_k^s) \iff (u_k^*, v_k^{*e}) = (1, v_L),$$

which prove the second and third part of the result, respectively. ■

We emphasize that Proposition 1.2 does not imply the uniqueness of an optimal solution, i.e., in boundary cases such as $T_{k+1}^B(v_H) = T_{k+1}^F + c_H$ or $\Delta T_{k+1} = B_k(v_k^s)$, there may be more than one optimal solutions. This is the reason that we do not use strict inequalities in the statement or the proof of the result. Note that optimal policies have an intuitive structure: if merging is not optimal, something that is associated with moving at high velocity, then the expected remaining travel time on lane F must be larger than that on lane B, even if successful merging was guaranteed, and vice versa. Moreover, if merging is optimal, then the driver should choose v_L over v_H only if the expected continuation benefit of that decision outweighs the expected loss from the merge.

Proposition 1.3. *Consider the mandatory lane-changing problem in Eq. (1.2)-(1.4). If it is optimal to merge at stage k, i.e., $u_k^* = 1$, then it is optimal to merge at stage $k+1$, i.e., $u_{k+1}^* = 1$.*

Proof. Assume that it is optimal to merge at stage k, i.e., $u_k^* = 1$. The velocity in the free lane is fixed, $\overline{v}^F$, therefore

$$\begin{aligned} T_{k+1}^F = \frac{1}{\overline{v}^F} + T_{k+2}^F \implies T_{k+1}^F + c_H &= \frac{1}{\overline{v}^F} + T_{k+2}^F + c_H \\ \implies T_{k+1}^B(v_H) &\geq \frac{1}{\overline{v}^F} + T_{k+2}^F + c_H, \end{aligned} \tag{1.5}$$

since Proposition 1.2 implies that $T_{k+1}^B(v_H) \geq T_{k+1}^F + c_H$. Now, Eq (1.2), with $u_k^* = 1$, implies that

$$\begin{aligned} &T_{k+1}^B(v_H) = \\ &= \min_{v_{k+1}^e \in \{v_L, v_H\}} \left\{ \frac{2}{v_H + v_{k+1}^e} + \left\{ q(v_{k+1}^e)(T_{k+2}^F + C(v_k^e)) + (1 - q(v_{k+1}^e))T_{k+2}^B(v_{k+1}^e) \right\} \right\} \\ &\leq \frac{1}{v_H} + q_H \left(T_{k+2}^F + c_H\right) + \left(1 - q_H\right) T_{k+2}^B(v_H). \end{aligned}$$

Combined with Eq. (1.5), we have that

$$\begin{aligned} & \frac{1}{\overline{v}^F} + T^F_{k+2} + c_H \leq \frac{1}{v_H} + q_H \left(T^F_{k+2} + c_H\right) + \left(1 - q_H\right) T^B_{k+2}(v_H) \\ \iff \quad & \left(1 - q_H\right) \left(T^F_{k+2} + c_H\right) \leq \frac{1}{v_H} - \frac{1}{\overline{v}^F} + \left(1 - q_H\right) T^B_{k+2}(v_H) \\ \iff \quad & T^F_{k+2} + c_H \leq T^B_{k+2}(v_H), \end{aligned}$$

since $\overline{v}^F \leq v_H$. This is equivalent to $u^*_{k+1} = 1$, according to Proposition 1.2. ■

This result can be viewed as the direct extension of the classical result regarding the "parking problem". While such results in optimal stopping problems are usually established by invoking the One-Step Lookahead rule (the "parking problem" is one example), we provide a proof from first principles.

1.4.2 Traffic oscillations

We now focus on the final region where it is optimal to merge, and also where optimizing driver behavior can lead to velocity oscillations, under certain conditions.

Final Merging Zone: This is the region defined as

$$T^B_k(v_L) \geq T^B_k(v_H) \geq T^F_{k+1} + c_H = \frac{N - k + 1}{\overline{v}^F} + c_H. \tag{1.6}$$

Proposition 1.3 establishes that there is typically a zone of consecutive stages, reaching the blockage point, where it is optimal to merge. On the other hand, Proposition 1.2 allows for merging to happen at either velocities, depending on the state of the system and the parameters of the problem. For convenience, let us adopt the shorthand notation

$$\mathbb{E}_{k+1}[L] \equiv q_L \left(T^F_{k+1} + c_L\right) + (1 - q_L) T^B_{k+1}(v_L), \tag{1.7}$$

which denotes the expected remaining travel time at the *end* of stage k, but right before the merging opportunity is revealed (i.e., before the random variable Y_{k+1} is realized), assuming that $v^e_k = v_L$; $\mathbb{E}_{k+1}[H]$ is defined similarly.

The DP recursion can be expressed in the following form:

$$T^B_k\left(v^s_k\right) = \min_{v^e_k \in \{v_L, v_H\}} \left\{ \frac{2}{v^s_k + v^e_k} + \min\left\{\mathbb{E}_{k+1}\left[v^e_k\right], T^B_{k+1}\left(v^e_k\right)\right\} \right\}. \tag{1.8}$$

The metric of interest here is $\mathbb{E}_{k+1}[L] - \mathbb{E}_{k+1}[H]$, i.e., the future loss/benefit from choosing v_L over v_H at stage k. Depending on the exact value of this quantity, the optimal velocity decisions are made as follows:

1. If $\mathbb{E}_{k+1}[L] - \mathbb{E}_{k+1}[H] \leq \frac{2}{v_L+v_H} - \frac{1}{v_L}$ and $v_k^s = v_L$, then $v_k^{*e} = v_L$;
2. If $\mathbb{E}_{k+1}[L] - \mathbb{E}_{k+1}[H] \geq \frac{2}{v_L+v_H} - \frac{1}{v_L}$ and $v_k^s = v_L$, then $v_k^{*e} = v_H$;
3. If $\mathbb{E}_{k+1}[L] - \mathbb{E}_{k+1}[H] \leq \frac{1}{v_H} - \frac{2}{v_L+v_H}$ and $v_k^s = v_H$, then $v_k^{*e} = v_L$;
4. If $\mathbb{E}_{k+1}[L] - \mathbb{E}_{k+1}[H] \geq \frac{1}{v_H} - \frac{2}{v_L+v_H}$ and $v_k^s = v_H$, then $v_k^{*e} = v_H$.

To see this, note that if $v_k^s = v_L$ and merging is optimal, Eq. (1.2) implies that $v_k^{*e} = v_L$, as long as

$$\frac{2}{v_L + v_L} + q_L \left(T_{k+1}^F + c_L\right) + (1 - q_L) T_{k+1}^B(v_L) \leq \frac{2}{v_L + v_H} + q_H \left(T_{k+1}^F + c_H\right) + (1 - q_H) T_{k+1}^B(v_H).$$

This is precisely the first case above. The other cases can be proved similarly.

It can be verified that

$$\frac{2}{v_L + v_H} - \frac{1}{v_L} \leq \frac{1}{v_H} - \frac{2}{v_L + v_H} \leq 0,$$

since $v_L \leq v_H$. Hence, in the more interesting part of the state space, the merging zone, one can view the optimal policy as having three regions:

Region ***L***: If $\mathbb{E}_{k+1}[L] - \mathbb{E}_{k+1}[H] < \frac{2}{v_L+v_H} - \frac{1}{v_L}$, then $v_k^{*e} = v_L$. This is the case when conditions 1 and 3 above are satisfied simultaneously.

Region ***H***: If $\mathbb{E}_{k+1}[L] - \mathbb{E}_{k+1}[H] > \frac{1}{v_H} - \frac{2}{v_L+v_H}$, then $v_k^{*e} = v_H$. This is the case when conditions 2 and 4 above are satisfied simultaneously.

Region ***X***: If $\frac{2}{v_L+v_H} - \frac{1}{v_L} \leq \mathbb{E}_{k+1}[L] - \mathbb{E}_{k+1}[H] \leq \frac{1}{v_H} - \frac{2}{v_L+v_H}$, then $v_k^{*e} \neq v_k^{*s}$.

Choosing v_L over v_H always comes at a short-term disadvantage, since it takes longer to traverse the current stage. Thus, the velocity decision depends critically on the future benefit or loss from choosing one over the other. If choosing v_L over v_H is harmful or largely indifferent, then the optimal velocity decision is v_H; this is region H. In contrast, if choosing v_L over v_H is quite beneficial, then the optimal velocity decision is v_L; this is region L.

Proposition 1.4. *Consider the mandatory lane-changing problem in Eq.* (1.2)-(1.4)*. If the driver enters region L in the merging zone, she stays in region L (i.e., she keeps trying to merge at v_L) until she exits the bottleneck.*

Proof. This result is established as part of the proof of Proposition 1.5. ■

Together, Propositions 1.1 and 1.4 imply that, in the general case, the optimal policy starts with a region where it is optimal not to merge and move at high velocity, and ends with a region where it is optimal to merge and move at low velocity.

Now, to understand why velocity oscillations may be optimal for an optimizing driver, consider the car on lane B entering stage k with velocity $v_k^s = v_H$. (An identical argument can be made if the car enters stage k at low velocity.) During the next two stages, the driver can act in one of the following three ways (while intending to merge each time, but unable to find sufficient gap to merge):

(a) $H \to H \to H$;

(b) $H \to L \to L$;

(c) $H \to L \to H$.

Sequence (c) is preferable to (a) for the driver if

$$\frac{2}{v_H + v_L} + \mathbb{E}_{k+1}[L] \leq \frac{1}{v_H} + \mathbb{E}_{k+1}[H] \iff \mathbb{E}_{k+1}[L] - \mathbb{E}_{k+1}[H] \leq \frac{1}{v_H} - \frac{2}{v_L + v_H}.$$

Similarly, sequence (c) may be preferable to (b) in the second step because

$$\frac{2}{v_L + v_H} + \mathbb{E}_{k+2}[H] \leq \frac{1}{v_L} + \mathbb{E}_{k+2}[L] \iff \frac{2}{v_L + v_H} - \frac{1}{v_L} \leq \mathbb{E}_{k+2}[L] - \mathbb{E}_{k+2}[H].$$

So (c) is preferable to *both* (a) and (b) when condition X is satisfied. Hence, as long as the quantity $\mathbb{E}_k[L] - \mathbb{E}_k[H]$ stays in region X for consecutive stages, it is optimal for the driver to oscillate between high and low velocity over all the stages in region X: sticking to a low velocity has too high immediate cost, whereas sticking to a high velocity has too high long-term cost, so a rational decision maker attempts to "interpolate" the two extremes, as we explain further below.

The nature of oscillations

Velocity oscillations can arise for different reasons. One obvious reason is the driver fixes her attempted merging points at the (known, estimated, or calculated) gaps on the free lane, targeting optimal velocities at the merging points, and between these merge points minimizes her travel time on the blocked lane. This results in a straightforward bang-bang type behavior, where the driver accelerates and then decelerates to the required velocity between the merging points. These are caused by rational optimizing behavior, but they are not the focus of our work as the phenomenon is not that surprising.

Even if we allow only constant acceleration/deceleration between merging points though, velocity oscillations may arise for two reasons:

1. The driver may unexpectedly see a gap between planned merging points, in which case she may attempt to decelerate and merge on the spot. Again, this phenomenon is not particularly surprising and it is not the focus of our work;

2. Even with constant acceleration/decelertion between decision points, purely for optimization reasons, tthe optimal velocities at the planned merging points themselves vary, so that we can have patterns such as $v_0^* > v_2^* > v_1^*$ or $v_1^* > v_2^* > v_0^*$, which requires the driver to decelerate and then accelerate, or vice versa, at consecutive stages.

This last phenomenon is counterintuitive and has not been identified in the literature so far.

Convexity and oscillations

At the heart of the existence of region X is the convexity of function $\frac{1}{u}$, representing travel time as a function of velocity, itself a law of physics. Intuitively speaking, the travel time at the average velocity is less than the average of the travel times at high and low velocities:

$$\frac{1}{(v_L + v_H)/2} \leq \frac{1}{2}\left(\frac{1}{v_L} + \frac{1}{v_H}\right), \tag{1.9}$$

which can be equivalently written as

$$\frac{2}{v_L + v_H} - \frac{1}{v_L} \leq \frac{1}{v_H} - \frac{2}{v_L + v_H}. \tag{1.10}$$

This "convexity gap," i.e., the difference between the left and right-hand side, is precisely the range of region X.

Now, $\mathbb{E}_{k+1}[L] - \mathbb{E}_{k+1}[H]$ is the difference in the expected travel time of arriving at the end of the stage at either velocities, L or H, and can be positive or negative depending on the primitives of the problem. It may also happen to fall (strictly) in the range defined by the region X inequalities. In that case, if the driver happens to be at velocity L at the end of the stage and is unable to merge, then she is better off moving to H and trying to merge at the next stage, as the lower travel time from average speed dominates:

$$\frac{2}{v_L + v_H} - \frac{1}{v_L} < \mathbb{E}_{k+1}[L] - \mathbb{E}_{k+1}[H] \iff \frac{2}{v_L + v_H} + \mathbb{E}_{k+1}[H] < \frac{1}{v_L} + \mathbb{E}_{k+1}[L]; \tag{1.11}$$

while if she happens to be at H just before the end of stage k, she is better off slowing down to L and try to merge at the end of the next stage:

$$\mathbb{E}_{k+1}[L]-\mathbb{E}_{k+1}[H] < \frac{1}{v_H}-\frac{2}{v_L+v_H} \iff \frac{2}{v_L+v_H}+\mathbb{E}_{k+1}[L] < \frac{1}{v_H}+\mathbb{E}_{k+1}[H]. \tag{1.12}$$

This can happen repeatedly across consecutive stages, as long as the difference of the expected values stays in region X.

1.4.3 Insight from numerical experiments

To obtain additional insight on whether velocity oscillations occur frequently enough to be of concern, we derive the optimal policy numerically for a broad range of parameter values. Figure 1.2 summarizes our findings, illustrating a variety of scenarios that one may encounter. Despite the differences between these

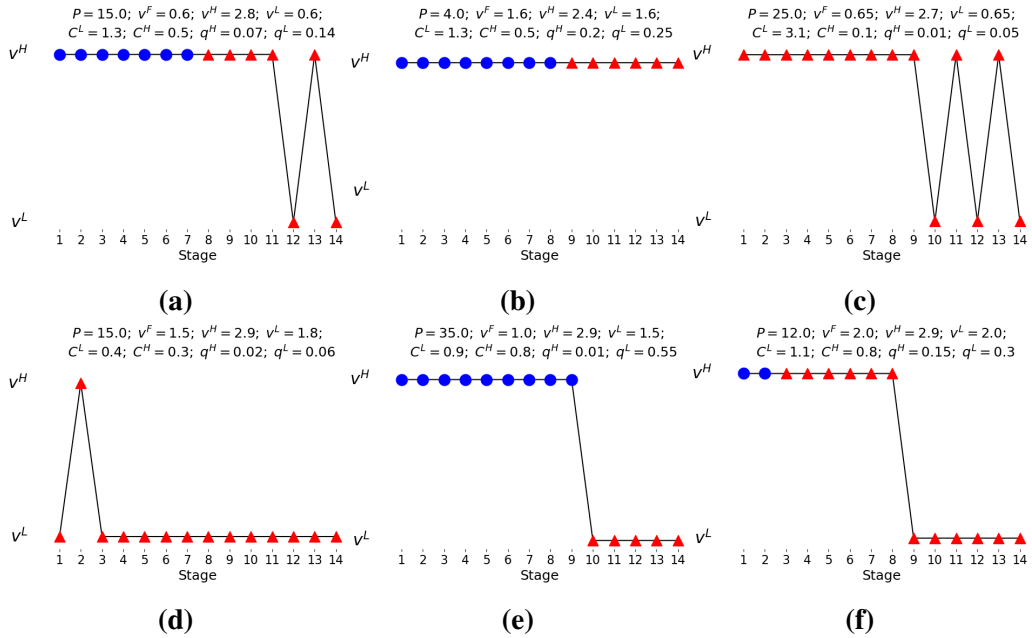
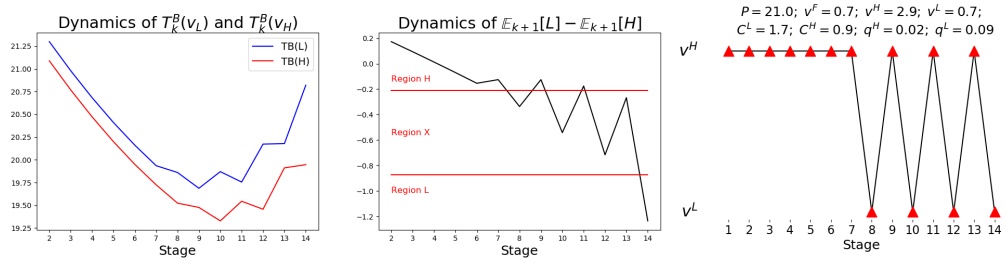


Figure 1.2: Examples of the optimal merging and velocity control policy for different choice of parameters P, v_H, v_L, $\overline{v}^F$, q_H, q_L, c_L and c_H, assuming $N = 15$. Optimal merging decisions are represented with the color and the shape of points (blue circle for "not merge", red triangles for "merge"), while the optimal velocities are shown along the y-axis. The corresponding stages are along the x-axis.

cases, a common theme emerges. The optimal policy, in general, seems to have the following three phases, in succession:

$$\big(\text{no merge, high velocity}\big) \longrightarrow \big(\text{merge, high velocity}\big) \longrightarrow \big(\text{merge, low velocity}\big),$$

In Figure 1.3 we dive deeper into one of the examples that exhibit the aforementioned velocity oscillations: the left-most figure depicts the expected travel time, at either velocities, for the different stages. Note that after stage 10 the expected travel time increases, due to the fact that incurring the late merging penalty P becomes more and more likely. The center figure shows the evolution, in time, of the quantity $\mathbb{E}_{k+1}[L] - \mathbb{E}_{k+1}[H]$, which alternates between regions H and X from stage 7 and onwards. Consistent with our analysis, the optimal policy, which is presented in the right-most figure, exhibits velocity oscillations from stage 7 and onwards.



(a) **(b)** **(c)**

Figure 1.3: The dynamics of $\mathbb{E}_{k+1}[L] - \mathbb{E}_{k+1}[H]$ lead to an optimal solution that alternates between regions H and X.

1.5 Managing lane-changing of algorithm-assisted drivers

Velocity oscillations while attempting to merge increase the risk of accidents. In this section we show how the traffic manager can avoid global optimization-induced oscillations, by setting appropriate speed limits in the Final Merging Zone.

One way to limit the velocity oscillations is to require that $\mathbb{E}_k[L] - \mathbb{E}_k[H]$ is monotonically non-increasing in k. Then, the dynamics in the Final Merging Zone will be as follows: $H \dots HX \dots XL \dots L$, i.e., with a single region of oscillations. Under stronger conditions, the region of oscillations can be completely avoided.

Proposition 1.5. *Consider the mandatory lane-changing problem in Eq.* (1.2)-(1.4).

1. *Assume that*

$$\frac{q_L - q_H}{\overline{v}^F} + \frac{2(1-q_L)}{v_L + v_H} - \frac{1-q_H}{v_H} + \frac{(1-q_L)(v_L - v_H)}{v_L(v_L + v_H)} \geq 0,$$

and

$$\frac{q_L - q_H}{\overline{v}^F} - \frac{2(q_L - q_H)}{v_L + v_H} + q_H q_L (c_L - c_H) + \frac{2}{v_L + v_H} - \frac{1}{v_H} + \frac{(v_L - v_H)(1 - q_L)(1 - q_H)}{v_L (v_L + v_H)} \geq 0.$$

Then, $\mathbb{E}_k[L] - \mathbb{E}_k[H]$ is monotonically non-increasing in k, so there can be at most one region of velocity oscillations.

2. *A stronger condition guarantees oscillations never occur. Assume that*

$$\frac{q_L - q_H}{\overline{v}^F} + \frac{2(1 - q_L)}{v_L + v_H} - \frac{1 - q_H}{v_H} + \frac{(1 - q_L)(v_L - v_H)}{v_L (v_L + v_H)} \geq 0,$$

and

$$\frac{q_L - q_H}{\overline{v}^F} - \frac{2(q_L - q_H)}{v_L + v_H} + q_H q_L (c_L - c_H) + \frac{2}{v_L + v_H} - \frac{1}{v_H} + \frac{(v_L - v_H)(1 - q_L)(1 - q_H)}{v_L (v_L + v_H)} \geq \frac{(v_L - v_H)^2}{v_L v_H (v_L + v_H)}.$$

Then, the optimal solution lies in region X for, at most, one stage. Hence, the optimal policy has an intuitive "monotonic" structure:

$$(\textit{no merge, high velocity}) \longrightarrow (\textit{merge, high velocity}) \longrightarrow (\textit{merge, low velocity}),$$

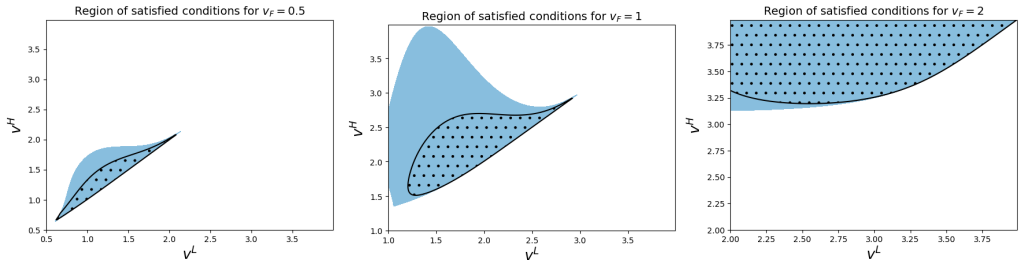
which excludes oscillations.

Proof. See Appendix 1.9. ■

Interpreting the sufficient conditions on speed limits in Proposition 1.5 is difficult. However, as there are just two dimensions in our case, v_L and v_H, we can plot the respective regions to (a) check if the conditions have any bite; (b) visualize and determine useful ranges for a policy maker.

In Figure 1.4 we provide illustrative examples of the velocity regions that satisfy the aforementioned conditions. First, we set the values for q_L, q_H, c_L, c_H in accordance to functions of $\overline{v}^F, v_H$ and v_L that have appeared in the literature; see Section 1.7.1 for more details. Then, for different values of $\overline{v}^F$, we plot the regions (v_H, v_L) where the conditions of Proposition 1.5 hold.

As we can see, in general, the conditions capture reasonably large, implementable regions that, however, need not be convex. Also, the higher the velocity in the free lane, it appears, the larger the area of the parameter space where the optimal solution does include at least one oscillation region.



(a) **(b)** **(c)**

Figure 1.4: A region of parameters v_H, v_L, for which Proposition 1.5 holds, given reasonable functional forms for q_H, q_L, c_H, c_L, for different values of v_F (in normalized values; they correspond to 2.5m/s, 5m/s and 10m/s). The blue area shows the velocity region that satisfies the first part of proposition, while the dotted area the region that satisfies the second part.

Let us note that the conditions of Proposition 1.5 are only sufficient: in Figure 1.5 we present indicative examples of optimal policies that avoid oscillations, despite not satisfying the first part of Proposition 1.5. Specifically, in both examples, the quantity $\mathbb{E}_{k+1}[L] - \mathbb{E}_{k+1}[H]$ lies in Region X for exactly one stage.

Pursuing safety may require sacrificing efficiency, leading to larger delays for the drivers. A natural question to ask is the following: if the central planner were to impose safety constraints via speed limits, satisfying the conditions in Proposition 1.5, what would be the loss in efficiency, measured in aggregate travel time of the drivers? We address this question, numerically, in Figure 1.6: for every such $\overline{v}^F$, we build the "safe" non-oscillating region like the ones in Figure 1.4, compute the best policy within the region, and compare it to the best unconstrained policy.

As expected, the loss is near zero when traffic is relatively free, i.e., $\overline{v}^F \geq 8$ m/s. It also goes to zero as the conditions on the free-lane approaches complete jam, i.e., $\overline{v}^F \approx 0$ m/s. The most problematic is the case of intermediately congested traffic, in which case the increase of travel time, experienced by merging drivers, may exceed 20%.

1.6 Robustness

In this section we study the problem, relaxing some assumptions, to test the robustness of our insights with respect to different modeling features.

1. First, we consider a direct extension to our benchmark model, where the velocity takes values in a compact set. We conduct an extensive numerical

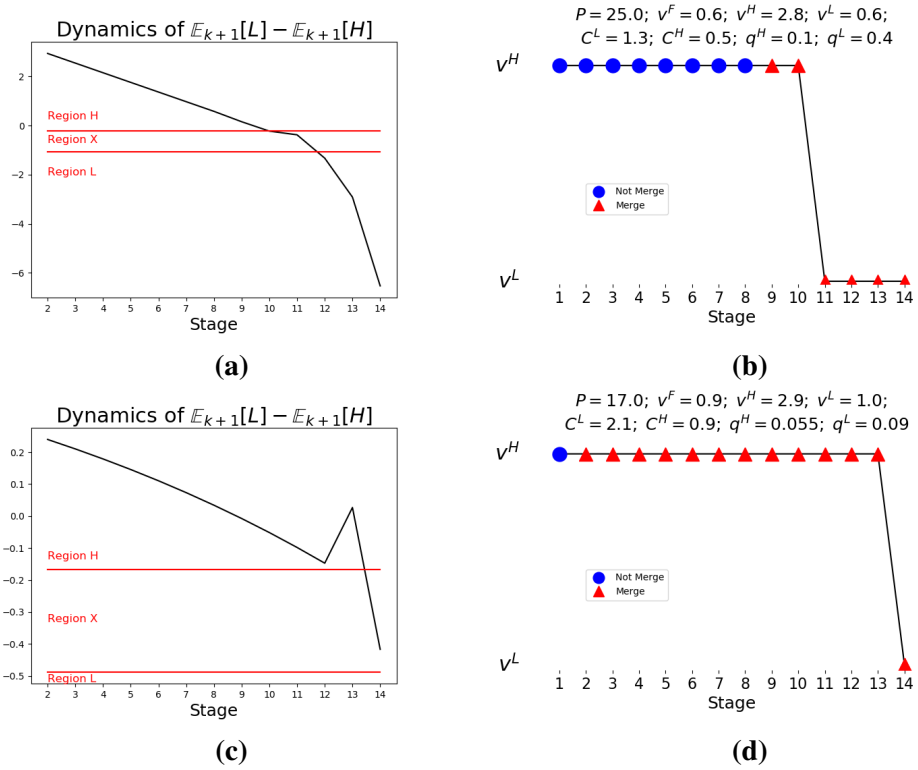


Figure 1.5: Two examples of DP solutions that do not oscillate despite failing to satisfy Proposition 1.5. Top: $\mathbb{E}_{k+1}[L] - \mathbb{E}_{k+1}[H]$ behaves smoothly. Bottom: $\mathbb{E}_{k+1}[L] - \mathbb{E}_{k+1}[H]$ oscillates, but outside of Region X.

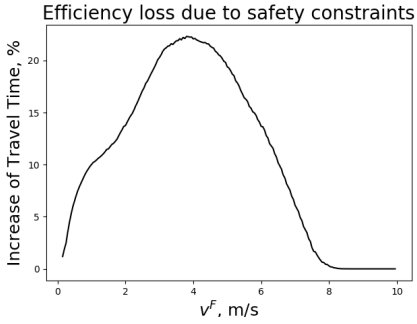


Figure 1.6: Additional delay, experienced by merging drivers due to imposing safety constraints, as a function of the velocity on the target lane.

study, showing that the optimal policies with continuous velocity decisions — instead of just the highest and the lowest velocity, as in our benchmark model — are not too different from the ones in our analysis in Section 1.4.2.

2. Second, we consider a version of our benchmark model, with continuous state and action spaces, and we show that the optimal policy is an instance of bang-bang control. This analysis reinforces our main findings, that regions of maximum acceleration (thus, high velocity) and deceleration (thus, low velocity) are a structural characteristic of the problem, and not the artefact of modeling choices, e.g., discrete vs continuous time and/or state-action spaces.

3. Finally, we consider an extension to our benchmark where the merging points are not predetermined but rather specified endogenously. We identify sufficient conditions such that velocity oscillations persist even then.

Overall, our treatment in this section is informal, geared more towards providing insight rather than formally proving results.

1.6.1 Continuous velocity

We start with a model that is the same as our benchmark one, except for the fact that the velocity is a continuous decision variable, with values in $\left[v_L, v_H\right]$. The DP recursion in this case takes the following form:

$$T_k^B\left(v_k^s\right) = \min_{v_L \le v_k^e \le v_H} \Bigg\{ \frac{2}{v_k^s + v_k^e} + \min \Bigg\{ \underbrace{q(v_k^e)\left(T_{k+1}^F + C(v_k^e)\right) + (1 - q(v_k^e))T_{k+1}^B\left(v_k^e\right)}_{u_k=1}, \underbrace{T_{k+1}^B\left(v_k^e\right)}_{u_k=0} \Bigg\} \Bigg\}. \quad (1.13)$$

The propositions below extend the discrete-velocity results to the case with continuous velocities, and for brevity, we omit their proofs, as they are very similar to the proofs of their counterparts.

Proposition 1.6. *Consider the mandatory lane-changing problem in Eq. (1.13). If it is optimal not to merge at stage k, i.e., $u_k^* = 0$, then it is optimal to have the highest velocity during stage k, i.e., $v_k^{*e} = v_H$.*

Proposition 1.7. *Consider the mandatory lane-changing problem in Eq. (1.13). If it is optimal to merge at stage k, i.e., $u_k^* = 1$, then it is optimal to merge at any later stage, i.e., $u_l^* = 1,\ l = k+1, \ldots, N$.*

Proceeding analytically with Eq. (1.13) is challenging, as the first-order condition for minimizing the expected travel time requires solving a fourth-order polynomial in v_k^e. Thus, we resort to a numerical investigation, by approximating the

compact velocity space by a large number (50) of discrete velocity levels. Also, for the remainder of the section, we adopt a linear merging probability model: $q\left(v^B\right) = \alpha - \beta v^B$, where $\alpha, \beta > 0$ and $\beta v_H < \alpha \leq 1$; and a negative-exponential function for the merging penalty: $c\left(v^B\right) = e^{\gamma - \delta v^B}$, where $\delta > 0$. We refer to Section 1.7.1 for details on selecting these functional forms.

Figure 1.7 presents examples of (approximately) optimal policies in the case of continuous velocities. Note that we use the same parameter sets as in Figure 1.2, so that the two scenarios are directly comparable.

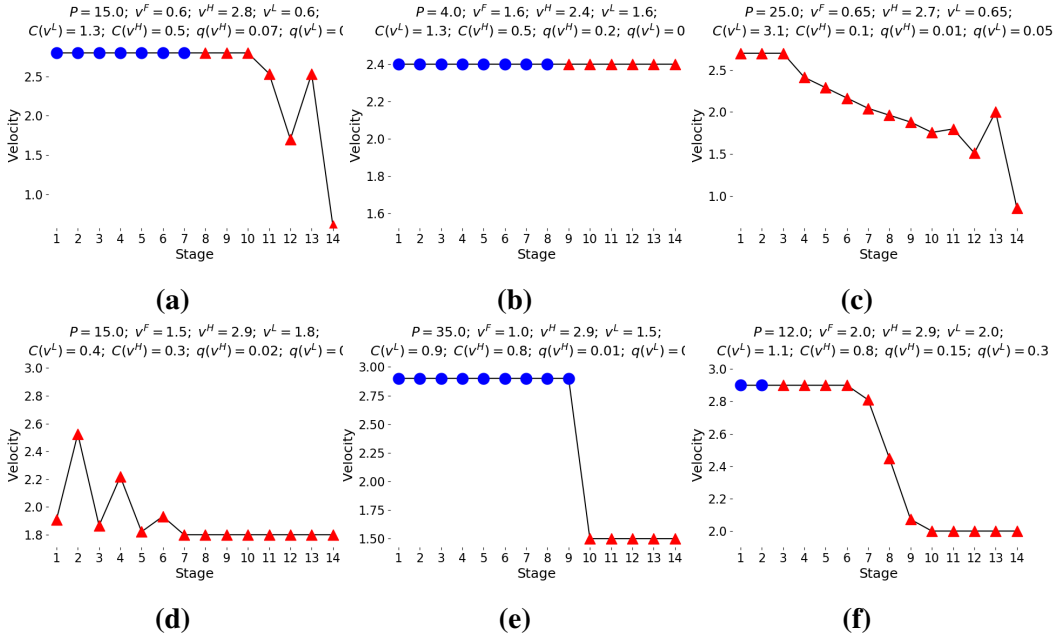


Figure 1.7: Examples of the optimal solution for different choice of parameters P, v_H, v_L, $\overline{v}^F$, $q(v_H)$, $q(v_L)$, $C(v_L)$ and $C(v_H)$ assuming $N = 15$. The velocity space is discretized by dividing the interval evenly into 50 discrete levels.

A first observation is that, overall, the optimal policies in the continuous-velocity scenario do not differ much from those in the discrete-velocity benchmark. In fact, in some cases, the optimal policies are exactly the same, while in the rest, the difference is typically confined in a relatively narrow window during which the driver needs to slow down. To investigate this further, we report on the gap between the optimal expected travel times in the two scenarios in the table below. (Again, in the continuous-velocity case, the corresponding DP is solved numerically by discretizing the velocity space in 50 levels, and the linear probability model is adopted.) The first six rows correspond to the parameter sets in Figures 1.2 and 1.7, and three additional cases are considered.

Table 1.3: Comparison of expected travel time for DP solutions with the same parameters, 2 velocities versus 50 velocities.

$\overline{v}^F$	v_L	v_H	$q(v_L)$	$q(v_H)$	P	$C(v_L)$	$C(v_H)$	Gap
0.6	0.6	2.8	0.14	0.07	15	1.3	0.5	0.331%
1.6	1.6	2.4	0.25	0.2	4	1.3	0.5	0%
0.65	0.65	2.7	0.05	0.01	25	3.1	0.1	1.406%
1.5	1.8	2.9	0.06	0.02	15	0.4	0.3	0.112%
1.0	1.5	2.9	0.45	0.05	35	0.9	0.8	0%
2.0	2.0	2.9	0.3	0.15	12	1.1	0.8	0.092%
0.4	0.4	3.0	0.85	0.15	5	1	0.2	0.749%
0.4	2.4	3.5	0.04	0.03	25	1.5	1	0.027%
2.5	3.0	3.5	0.75	0.6	3	2	1	0%

The reported results show that the gap is less than 1.5% in all cases, and practically nonexistent in many of them. This can be attributed to the fact that the optimal policy is only affected in a small number of stages, and that the travel time seems to be less sensitive to velocity decision in the affected areas, i.e., regions of slowing down.

From a practical standpoint, these findings justify the discrete-velocity scenario being our benchmark. A decision-support tool that is based on this simple model could provide recommendations in real time, something that may be challenging with a continuous-velocity model, whose computational complexity is much higher.

A second observation is that velocity oscillations persist in the continuous-velocity scenario, albeit somewhat damped, wherever they appear in the discrete-velocity experiments. Hence, on the descriptive side, this suggests that oscillations are not an artefact of our modeling choices — to consider just two velocities in our benchmark model — but rather an intrinsic characteristic of the problem.

1.6.2 Connection to bang-bang control

In this section we consider a version of our benchmark model, where state and action spaces are compact, i.e., the driver may attempt to merge at any (continuous) point in time, as well as continuously change the vehicle's velocity by accelerating/decelerating. By deriving the Hamilton-Jacobi-Bellman (HJB) equation of the problem, we establish a connections between the optimal policy and bang-bang control.

As before, we consider a car that needs to merge from lane B to lane F in order to get past a blockage point, which lies at position N of lane B. We denote the position and velocity of the merging car by $x \in [0, N]$ and $v \in [v_L, v_H]$, respec-

tively. The decisions, at each point in time, are whether to accelerate, decelerate, or keep the same speed; and whether to merge or not, given the opportunity. We denote the merging decision by $u \in \{0, 1\}$, and the acceleration/deceleration decision by $\alpha \in [-D, A]$, where A and D are the acceleration and deceleration limits of the car, respectively. If the car moves at velocity v on lane B, it finds a large enough gap to merge in lane F with probability $q(v)$. Conditional on merging successfully, it incurs a penalty $c(v)$, which captures the total effect of the relaxation process, and then continues at a constant velocity $\overline{v}^F$ until it gets past the blockage. Functions $q(\cdot)$ and $c(\cdot)$ are assumed to be continuous and differentiable. If the car reaches the bottleneck without having merged, it incurs a (significant) merging penalty P, as it has to come to a complete stop, and then wait for a very large gap on lane F in order to merge. [3]

Let $T^X\big(x, v\big)$ be the optimal expected travel time until the car gets past the blockage, assuming that the car is on lane $X \in \{B, F\}$, at position x, and moving at velocity v. Clearly,

$$T^F\big(x\big) = \frac{N - x}{\overline{v}^F},$$

since every car on lane F moves at the same velocity $\overline{v}^F$. On the other hand, in the more interesting case where the car is on lane B, we can intuitively express the optimal expected travel time through the following recursion:

$$T^B\big(x, v\big) = \min_{\alpha \in A(v)} \Bigg\{ \delta + \min \Bigg\{ q(v + \delta\alpha)\Big(T^F\big(x + \delta v\big) + c(v + \delta\alpha)\Big) + \big(1 - q(v + \delta\alpha)\big) T^B\big(x + \delta v, v + \delta\alpha\big), T^B\big(x + \delta v, v + \delta\alpha\big) \Bigg\} \Bigg\},$$

where $\delta > 0$ is "small" and $A(v) = [0, A]$, if $v = v_L$; $A(v) = [-D, 0]$, if $v = v_H$; and $A(v) = [-D, A]$, otherwise. Furthermore, we have the boundary condition $T^B\big(N, v\big) = P$, for all $v \geq 0$.

Let us focus on the region of the state space where it is optimal to merge. (In the region where it is not optimal to merge, it is clear that the car should move at the highest possible velocity.) By taking the Taylor expansion and then letting $\delta \to 0$, we derive the HJB equation of the problem, i.e., function $T^B\big(x, v\big)$ is such

[3] This abstract model of merging with continuous state and action spaces has structural similarities to the "rocket-rail car problem" in optimal control theory, where a driver wishes to park a rail car with rocket engines on both ends, with the goal of firing the rockets so as to make the car stop at a precise point in the least amount of time. Importantly, however, there is no merging/optimal stopping aspect to that problem.

that

$$\min_{\alpha \in A(v)} \left\{ 1 + q(v) \frac{\partial T^F}{\partial x} v + \big(1 - q(v)\big) \frac{\partial T^B}{\partial v} + \alpha F\big(x, v\big) \right\} = 0, \tag{1.14}$$

where

$$F\big(x, v\big) \equiv \frac{\partial q}{\partial v} \Big(T^F(x) + c(v) - T^B\big(x, v\big) \Big) + q(v) \frac{\partial c}{\partial v} + \big(1 - q(v)\big) \frac{\partial T^B}{\partial v}.$$

Solving the HJB equation to determine $T^B\big(x, v\big)$ is quite challenging analytically. However, the mere structure of the equation provides useful insight into the optimal policy: in order to minimize the left-hand side in Eq. (1.14), the optimal control needs to be $\alpha \approx -\text{sign}\big\{ F\big(x, v\big) \big\}$. In particular,

$$\alpha = A \cdot 1_{\{v < v_H\}}, \qquad \text{if } F\big(x, v\big) < 0;$$

and

$$\alpha = -D \cdot 1_{\{v > v_L\}}, \qquad \text{if } F\big(x, v\big) > 0;$$

while any feasible α is optimal, otherwise. This is the essence of the phenomenon that the literature terms bang-bang control: the HJB equation decouples in such a way, so that the product of the control and some state-dependent "drift" term needs to be minimized over a compact set. This implies that the optimal control typically takes values on the boundary of the feasible set (the exact value depends on the sign of the drift term), which allows for, and often implies, oscillatory regions in the optimal policy, as we elaborate below.

First, let us extend the definition in Eq. (1.7) to continuous space and velocity:

$$G\big(x, v\big) \equiv q(v) \left(T^F(x) + c(v) \right) + \big(1 - q(v)\big) T^B\big(x, v\big),$$

which denotes the expected remaining travel time at position x and velocity v, right before the merging opportunity is revealed; where $x \in [0, N]$ and $v \in [v_L, v_H]$. Note that

$$\frac{\partial G}{\partial v} = \frac{\partial q}{\partial v} \Big(T^F(x) + c(v) - T^B\big(x, v\big) \Big) + q(v) \frac{\partial c}{\partial v} + \big(1 - q(v)\big) \frac{\partial T^B}{\partial v} = F\big(x, v\big).$$

In essence, the quantity $\mathbb{E}_{k+1}[L] - \mathbb{E}_{k+1}[H]$, on which the analysis of the oscillatory behavior in our benchmark model is largely based, is simply an approximation—in discrete state and action space—to the above partial derivative, and thus to the drift term $F\big(x, v\big)$:

$$\frac{\mathbb{E}_{k+1}[H] - \mathbb{E}_{k+1}[L]}{v_H - v_L} \approx \frac{\partial G}{\partial v}\Big|_{\big(k+1, v_L\big)} = F\big(k+1, v_L\big).$$

The boundaries between regions L, H, and X in the benchmark model are dictated by the differences in traversing a stage at different velocities. In the continuous formulation there is no notion of discrete space (stage) or velocity. Hence, the three regions, in discrete and continuous space and velocity, are as follows:

$$\text{Region L:} \quad \frac{\mathbb{E}_{k+1}[H] - \mathbb{E}_{k+1}[L]}{v_H - v_L} > \frac{1}{v_L(v_L + v_H)} \iff F\big(x, v\big) > 0;$$

$$\text{Region H:} \quad \frac{\mathbb{E}_{k+1}[H] - \mathbb{E}_{k+1}[L]}{v_H - v_L} < \frac{1}{v_H(v_L + v_H)} \iff F\big(x, v\big) < 0;$$

Region X:

$$\frac{1}{v_H(v_L + v_H)} \leq \frac{\mathbb{E}_{k+1}[H] - \mathbb{E}_{k+1}[L]}{v_H - v_L} \leq \frac{1}{v_L(v_L + v_H)} \iff F\big(x, v\big) = 0.$$

The optimal policy in regions L and H is intuitive: if the derivative of the expected remaining travel time, with respect to the velocity, is positive (negative), it is optimal to decelerate (accelerate) as much as possible. In region X on the other hand, every feasible decision is optimal, and oscillations are likely to appear "around" it. For instance, if the solution oscillates between the boundaries of regions L and H, crossing region X but never getting "stuck" there, then we have shifts between maximum acceleration and maximum deceleration, akin to the velocity oscillations that we uncover in our benchmark model.

Furthermore, the above analysis reinforces that regions of maximum acceleration (thus, high velocity) and deceleration (thus, low velocity) are a structural property of the problem, and not the artefact of modeling choices, e.g., discrete vs continuous time and/or state-action spaces. Moreover, it justifies why in the numerical experiments of Section 1.6.1, the performance improvement by considering a continuous velocity spectrum–instead of just the highest and the lowest velocity, as our benchmark model does–is practically negligible.

1.6.3 Endogenous merging points

Finally, we analyze an extension to our benchmark model where merging points are themselves endogenous decisions of the driver, along with the (continuous) velocity. Our aim is to illustrate that velocity oscillations may persist even in this setting.

We continue with one of our main modeling constructs, the merging probability function $q\big(v_k^e, V_k^F, d_k\big)$. However, we fix the driver's belief of the future

free-lane velocity V_k^F at its expected value $\overline{v}^F$. Hence, we can simplify the merging probability to $\tilde{q}\big(v_k^e, d_k\big)$. We assume $\tilde{q}(v,0) = 0$, for all v, and that $\tilde{q}(v,d)$ is increasing in d and decreasing in v. Moreover, we assume that there exists some $\epsilon > 0$, such that

$$\tilde{q}(v,d) = 0, \quad \forall d < \epsilon.$$

This is a standard assumption in dynamic programming, e.g., see Section 11.1 of Puterman (1994), to ensure that we avoid trivial situations with infinite merging attempts. Moreover, it is also realistic: if the driver is unable to merge at some point, the chances of merging in that vicinity are practically zero.

Furthermore, our standing assumption throughout the chapter is that acceleration/deceleration between merging points remains constant.

We formulate the Dynamic Program similarly to our benchmark model, whose solution provides (an optimal) set of merging points and velocities, $\big(x_k^*, v_k^*\big)$, $k = 0, \ldots, N$, with x_{N+1}^* being the blocking point. Let $d_k^* = x_{k+1}^* - x_k^*$.

We seek conditions such that $v_0^* < v_1^* > v_2^*$, which would correspond to velocity oscillations, at optimality. Fix all optimal merge-points and velocities, apart from v_1^*. We have that

$$v_1^* = \arg \min_{v_L \le v \le v_H} T_1(v),$$

where v_H and v_L are the upper and lower limits on feasible velocity, and

$$T_1(v) \equiv \left\{ \frac{d_0^*}{v_0^* + v} + \tilde{q}\big(v, d_0^*\big) T_1^F + \big(1 - \tilde{q}\big(v, d_0^*\big)\big) \left(\frac{d_1^*}{v_2^* + v} + \mathbb{E}\left[T^B\big(x_2^*, v_2^*\big)\right] \right) \right\}.$$

We use the Envelope Theorem in order to compute the derivative of $T_1(v)$, by viewing $x_k^*(v)$ and $v_k^*(v)$, for all $k \neq 1$, as optimized variables for every fixed v:

$$-\frac{d_0^*}{\big(v_0^* + v\big)^2} - \frac{\big(1 - \tilde{q}\big(v, d_0^*\big)\big) d_1^*}{\big(v_2^* + v\big)^2} + \tilde{q}'\big(v, d_0^*\big) \left(T_1^F - \frac{d_1^*}{v_2^* + v} - \mathbb{E}\left[T^B\big(x_2^*, v_2^*\big)\right] \right).$$

Let us assume that the vehicle is already in the Final Merging Zone, i.e., $\left(T_1^F - \frac{d_1^*}{v_2^* + v} - \mathbb{E}\left[T^B\big(x_2^*, v_2^*\big)\right] \right)$ is negative and, for simplicity, that $v_0^* = v_2^* = v_L$. We explore conditions such that the derivative of $T_1(v)$ is negative, for all $v < v_H$, which would imply that the optimal solution to the above constrained optimization problem is v_H. In other words, we require:

$$\tilde{q}'(d_0^*, v) \left(T_1^F - \frac{d_1^*}{v_2^* + v} - \mathbb{E}\left[T^B\big(x_2^*, v_2^*\big)\right] \right) < \frac{d_0^*}{\big(v_0^* + v\big)^2} + \frac{\big(1 - \tilde{q}\big(v, d_0^*\big)\big) d_1^*}{\big(v_2^* + v\big)^2}.$$

For concreteness, let us assume that $\tilde{q}(v,\cdot)$ is a linear function zero slope in the interval $[0,\epsilon]$, and slope $-\alpha$ from then on, for some $\alpha>0$. Since $\tilde{q}'(v,\cdot)$ and $\left(T_1^F-\frac{d_1^*}{v_2^*+v}-\mathbb{E}\left[T^B\left(x_2^*,v_2^*\right)\right]\right)$ are both negative, we have that

$$\alpha\left(\frac{d_1^*}{v_2^*+v}+\mathbb{E}\left[T^B\left(x_2^*,v_2^*\right)\right]-T_1^F\right)<\frac{d_0^*}{\left(v_0^*+v\right)^2}+\frac{\left(1-\tilde{q}\left(v,d_0^*\right)\right)d_1^*}{\left(v_2^*+v\right)^2}.$$

The right-hand side is bounded from below by $\epsilon/\left(v_0^*+v_H\right)^2$, and we obtain a sharper sufficient condition:

$$\alpha\left(\frac{d_1^*}{2v_L}+\mathbb{E}\left[T^B\left(x_2^*,v_2^*\right)\right]-T_1^F\right)<\frac{\epsilon}{\left(v_L+v_H\right)^2}.$$

Summarizing, if T_1^F and $\mathbb{E}\left[T^B\left(x_2^*,v_2^*\right)\right]$ satisfy the inequality above, then we have velocity oscillations similarly to our benchmark model, despite the fact that velocity is continuous and the merging attempt points are optimally decided.

1.7 Monte Carlo discrete-event simulations

In this section, we investigate the performance of optimal policies from our dynamic programming formulation under more realistic traffic situations, and compare it against popular merging guidelines used in practice, e.g., merge early, merge late, merge at a certain point, merge at a random point etc. More specifically, through Monte Carlo discrete-event simulations, we investigate:

1. The average travel time of all the merging vehicles as a function of the merging policy;
2. The throughput-delay tradeoffs of the different merging policies, on the particular road segment.

Our simulations are performed under much more realistic models of driver behavior than the stylized models of our theoretical analysis. We first discuss different aspects of our numerical set-up.

1.7.1 Numerical study set-up

In the stylized, benchmark model presented in Section 1.4 we made a number of simplifying assumptions for analytical tractability. To test its validity in more realisting conditions, we employ the framework of Cellular-Automata (CA), a class

of multi-agent simulations with simple decision rules and their implied dynamics, used in many fields of research where analytical solutions are hard to obtain, including traffic engineering; e.g., see Maerivoet and Moor (2005), Nagel et al. (1998).

More specifically, in the analysis of Sections 1.4,1.5 and 1.6 we assumed that there is a single merging car on the blocked lane; that gaps/merging opportunities are at predetermined positions; that the distance between these positions is the same; and that the velocities of the free-lane cars at different stages are i.i.d random variables. In the CA simulations that follow we relax all the above assumptions: both the blocked and the free lane have multiple cars; the merging ones follow our DP approach (or different heuristics against which we compare our approach), while the ones on the free lane follow a specified set of decision rules that is tuned to mirror characteristics of real vehicular traffic, inducing complicated dynamics about free-lane velocity and merging opportunities.

Gap acceptance and merging probability modeling

While in our theoretical analysis the merging probabilities at different velocities are given exogenously, in order to evaluate the performance of the optimal policy under realistic scenarios, we need to obtain more concrete estimates. The probability of merging depends on the density in the free lane, ρ^F, as a result of the existing gaps in that lane, and the velocities in both lanes. Furthermore, parameter P, which represents the time that the driver has to wait at the blockage point until she finds an appropriate gap, is also a function of the velocity in the free lane. In mathematical terms, in this section we review the existing literature in order to develop functional forms $q(v^B, \overline{v}^F)$ and $P(\overline{v}^F)$, where v^B represents the velocity in the blocked lane, which we can then use, e.g., for blocked-lane velocity optimization.

There is extensive literature of lane-changing behavior based on the *gap-acceptance theory*: a driver who wishes to merge compares the existing gap G at the target lane to a so-called critical gap G_c. If $G > G_c$ then the driver accepts the gap and merges to the target lane, otherwise she stays on the current lane. By adopting the gap-acceptance theory, we define the merging probability as follows:

$$q(v^B, \overline{v}^F) = \mathbb{P}\left[G(\overline{v}^F) > G_c(v^B, \overline{v}^F)\right].$$

Note that this definition is quite flexible, allowing for a broad array of choices for the functional forms of $G_c(v^B, \overline{v}^F)$ and $G(\overline{v}^F)$.

Traffic researchers model the critical gap as made of two components: a lead gap G_c^{lead}, and a lag gap G_c^{lag}. We base our study on the work of Lee (2006),

where the following model is developed:

$$G_c^{lead} = \exp(\alpha_0 + \frac{\alpha_1}{1+\exp(-\max\{0, \Delta V^{lead}\})} + \alpha_2 \min\left\{0, \Delta V^{lead}\right\} + \frac{\alpha_3 d_{nt}}{1+\exp(\alpha_4+\alpha_5\nu)} + \alpha_6\nu + \epsilon^{lead}), \quad (1.15)$$

and

$$G_c^{lag} = \exp(\beta_0 + \beta_1 \max\left\{0, \Delta V^{lag}\right\} + \beta_2 \min\left\{0, \Delta V^{lead}\right\} + \frac{\beta_3 d_{nt}}{1+\exp(\beta_4+\beta_5\nu)} + \beta_6\nu + \beta_7 \max\left\{0, a^{lag}\right\} + \epsilon^{lag}), \quad (1.16)$$

where $\Delta V^{lead} = v^{lead} - v^{subj}$, the difference between a lag vehicle and the subject vehicle velocities; d_{nt} is the distance to the end of the lane; ν is a driver-specific aggressiveness parameter; and a^{lag} is the acceleration of the lag vehicle. Lee (2006) also estimates the coefficients α_i and β_i from US highways data, finding them to be significant. A number of more elaborate models for G_c^{lead} and G_c^{lag} have been proposed recently, as well as a different merging models, including courtesy merging (see Choudhury et al. (2010)). We stick here to a simpler model of forced merging.

Since our model ignores the microstructure of the traffic on the free lane, we are not going to divide the whole required gap to lead gap and lag gap. Instead, we will use the pooled gap G_c:

$$G_c = l + G_c^{lead} + G_c^{lag}, \quad (1.17)$$

where l is the length of the car. (This is due to the fact that in Lee (2006) the lead gap was measured starting from the front bumper, while the lag gap was measured starting from the rear bumper.) To simplify our numerical experiments, we also make the following assumptions:

1. We omit the idiosyncratic aggressiveness term ν, due to the fact that we only have one driver on lane B;
2. We set $d_{nt} = 0$, suggesting that the critical gap does not depend on the distance from the blockage point in our model;
3. We omit the factor regarding a^{lag}, as our model does not capture the acceleration decision explicitly;
4. The random factors ϵ^{lead} and ϵ^{lag} are assumed to be zero-mean normal random variables in the literature, while we set them to zero as an approximation.

Summarizing, G_c is determined by the following equation:

$$\begin{aligned} G_c(v^B, \overline{v}^F) =& 4 + \exp(0.627 + \frac{1.90}{1 + \exp(-\max\left\{0, \overline{v}^F - v^B\right\})} \\ & - 0.314 \min\left\{0, \overline{v}^F - v^B\right\}) + \exp(0.509 + 0.116 \max\left\{0, \overline{v}^F - v^B\right\} \\ & + 0.034 \min\left\{0, \overline{v}^F - v^B\right\}), \end{aligned} \tag{1.18}$$

where we tentatively use $l = 4$m for the length of the car.

And while the critical gap is deterministic in our model, the available gap is a random variable. To find an appropriate distribution for G, we first need to estimate the mean density on the free lane, which is typically a function of the velocity, i.e., $d^F(\overline{v}^F)$. Fundamental relationships between velocity and density/headway have been thoroughly investigated in the literature. Following the seminal paper of Greenshields (1934), many authors have offered functional forms for these relationships and have estimated their parameters, e.g., Underwood (1961) and Pipes (1967). We use one of the popular estimated functional forms (see Del Castillo and Benítez (1995)). If we adjust the units of the function to meters and meters per second, then the function takes the following form:

$$d^F(\overline{v}^F) = 7.48 \exp(\frac{\overline{v}^F}{8.05}) - l, \tag{1.19}$$

where, again, we let $l = 4$m. (We have subtracted the length of the car l as the function determines the headway, i.e., the inter-vehicle distance plus the length of the car.)

It is common to consider headways on a road that are exponentially distributed, e.g., see Miller (1961a) and McNeil and Smith (1969). Under our assumption about a fixed velocity in the free lane $\overline{v}^F$, this is equivalent to modeling car arrivals to the studied road section as a Poisson process. Hence, in our numerical experiments we assume that the available gap is an exponential random variable with parameter $1/d^F(\overline{v}^F)$.

Putting everything together, the probability of merging that we use in our numerical experiments is equal to

$$q(v^B, \overline{v}^F) = \exp(-\frac{G_c(v^B, \overline{v}^F)}{d^F(\overline{v}^F)}) \tag{1.20}$$

where $G_c(v^B, \overline{v}^F)$ and $d^F(\overline{v}^F)$ are given by Eq. (1.18) and (1.19), respectively.

Finally, regarding P, the average delay that the driver experiences when stuck at the blockage point, we propose the following model:

$$P(\overline{v}^F) = m + \frac{1}{\overline{v}^F} \cdot \frac{1}{q(0, \overline{v}^F)} + C(0). \tag{1.21}$$

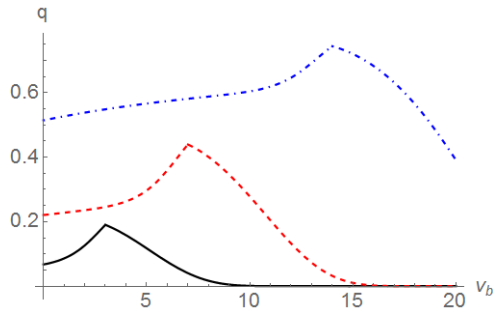


Figure 1.8: The function $q(v^B, \overline{v}^F)$ for $\overline{v}^F = 3$m/s (black solid line), $\overline{v}^F = 7$m/s (red dashed line), and $\overline{v}^F = 14$m/s (blue dot-dashed line).

We interpret the second component as the time that the driver has to wait until she finds an appropriate gap in the free lane, while stuck at the blockage point: $1/\overline{v}^F$ is the average time required to "observe" a new gap, and $1/q(0, \overline{v}^F)$ the average number of attempts needed to find a suitable gap (the mean of corresponding geometric distribution). In our numerical examples we use $m = 2$, as a crude approximation of the time required to actually move from one lane to another. The last component represents the penalty (due to "relaxation") from actual merging to the target lane at 0 velocity. The concrete models that can be used for $C(v^B)$ are discussed below.

Cellular Automata simulation setup

In CA both time and space are discrete. In our simulation setup, we set the time step to 1 second, and we divide the lanes into cells of 5m each. Every cell can be either empty or occupied by a car, and every car has its own velocity, which is an integer number from 0 to v^{max} (although qualitatively similar, not to be confused with the v^{max} in our numerical experiment setup). The velocity indicates how many cells a car travels per time step. For consistency with our theoretical models, we allow lane-changing only for the cars on the blocked lane, and in order for a car to merge to the free lane, there must be enough space both ahead and behind.

Our CA-simulation setup is defined by the following rules:

1. Drivers on the blocked lane who want to merge, check if they have enough space available. For the merging to take place, the cell next to the car on the free lane must be empty. Moreover, there should be v empty cells ahead, where v is the velocity of the car, and v^{back} empty cells behind, where v^{back} is the velocity of the car right behind on the free lane;

2. Every car on the blocked lane accelerates, or it is required to decelerate.

That is, for every car we set $v \leftarrow \min\{v+1, x, v^{max}\}$, where x is a distance to the car ahead on the blocked lane;

3. Every car with a positive velocity decreases the velocity by 1 with probability p. With a reasonable choice of p, this represents various stochastic real-traffic phenomena, such as over-braking or reaction delay at start.

4. Every car moves ahead by v cells.

In our CA simulations we use an open-boundary condition: if the car reaches the end of the road segment while driving on the free lane, it exits the system. On the other hand, car arrivals are generated on the free lane in order to maintain a traffic flow with the constant density. When all blocked-lane drivers exit the bottleneck, the simulation terminates. While we could generate infinite inflow of B-drivers and terminating the simulation after some fixed time, we do not use this approach, since it prohibits us from meaningful use of total travel time metric. Indeed, if the number of B-lane drivers is not fixed, we would not know, whether a higher travel time is attributed to higher delays or simply due to a higher number of exited drivers (which is actually good). Hence, we randomly place a fixed number of B-lane drivers and record how much time they need to leave the bottleneck.

Once we have fixed all the parameters of the CA, we need to evaluate the parameters of the model in Section 1.4.1, so that we can obtain the optimal solution to the corresponding DP, and introduce it as a new rule in the CA framework. We use the CA simulation itself to evaluate these parameters, very similarly to what we would be doing if we were collecting real data. More specifically, the estimation method that we use is to repeat the CA simulations several times, and to take the sample averages of the quantities of interest as their estimates.

Velocity on the free lane, $\overline{v}^F$: we run repeatedly CA simulations (without any cars on the blocked lane), keeping track of the velocity of each car at all times. Averaging over the the time horizon, the different cars, and the various simulation runs, we obtain an estimate for $\overline{v}^F$;

Velocities v_H *and* v_L: as discussed earlier, v_L and v_H are parameters that can be chosen by a blocked-lane driver, or imposed by the traffic regulator. The insights from Section 1.5 suggest that we choose $v_H = v^{max}$ and $v_L \approx \overline{v}^F$, as one of the choices resulting in no oscillations (we set the low velocity, v_L, to the closest integer that is greater or equal than $\overline{v}^F$);

Merging probabilities q_H *and* q_L: to estimate a merging probability, we run repeated CA simulations with a single driver on the blocked lane, moving at the corresponding velocity and trying to merge. Counting the number of attempts per every successful lane-change (at a given velocity), we obtain an estimate for the merging probability.

Late merging penalty P: viewed from a CA perspective, the late-merging penalty, P, in the model of Section 1.4.1 consists of different components. The first is the time spent during unsuccessful attempts to merge, while the car stands still near the blockage point. We denote this by P_0. Similarly to the estimation of q_H and q_L, we can estimate P_0 directly from CA simulations by averaging over many runs. We also introduce two analytically derived components. The final form of the late merging penalty is:

$$L = L_0 + C(0) + \frac{\overline{v}^F}{2} + \sqrt{2d}, \tag{1.22}$$

if we express the distance in CA cells, and the velocity in CA cells per second. The second component is the merging penalty due to "relaxation". The third component is the travel time lost due to the required acceleration from $v = 0$ to $\overline{v}^F$ after merging. This detail is omitted in our theoretical model, but may play an important role when acceleration is limited. To see how we obtain it, note that according to the formula $x = (v_2^2 - v_1^2)/2a$, if we take $a = 1$ (since in CA this is the acceleration we use, expressed in cells per second squared), $v_2^2 = \overline{v}^F$ and $v_1^2 = 0$, we have the distance of acceleration $x = (\overline{v}^F)^2/2$. Trivially, time of acceleration is equal to $\overline{v}^F$ seconds. If the car would move with the constant velocity $\overline{v}^F$ for $\overline{v}^F$ seconds, it would travel $(\overline{v}^F)^2$ cells. The difference between these two distances is also $(\overline{v}^F)^2/2$, which is the distance lost due to the finite acceleration. This corresponds to $\overline{v}^F/2$ seconds of travel time lost due to the acceleration, since we assume that the flow on the free lane is moving with the constant velocity $\overline{v}^F$. The last component is the time required to cover a distance d after merging, and can be derived easily from the formula $at^2/2 = d$, if we again take the value for acceleration $a = 1$.

Merging penalties c_L *and* c_H: we use the theoretical model, described above in this section. As a distance function $d^F(\overline{v}^F)$ for free lane we use the one described by Eq. (1.19) and Eq. (1.18) as the model for critical gap $G_c(v^B, \overline{v}^F)$.

Apart from the aforementioned parameters, we also need to fix d, which determines the distance between subsequent attempts to change lane. Note that there is a discrepancy here, between the structure of our theoretical model and the CA rules. In the latter, a driver can check for merging opportunities once per second. The distance between subsequent attempts clearly depends on the driver's velocity, so it is not constant. On the opposite, there is a known (potentially different) distance between subsequent attempts in our theoretical model. The best we can do, therefore, is to try to solve the single-car discrete-velocity DP for different constant choices of d, which would roughly correspond to the distance the blocked-lane driver covers between CA simulation steps. In the rest of the section we only report on the case where $d = 20$m, since the results turned out to be quite robust to the choice of d.

Finally, we set the slowdown probability p for the CA to $p = 0.3$, which is a typical value in the literature, and we assume the length of the road segment to be equal to 500m.

Merging penalty

A merging car creates a disturbance, and both the merging driver and the followers need to adjust velocity and the distance after completing the manoeuver. This effect is called "relaxation" and may be an important factor for an optimal merging decision.

We first show important qualitative characterization of the merging penalty, that is, how does it changes with increase of the velocity of merging car.

Let us denote the merging penalty to the driver that merges into gap g at the velocity v^B into the flow with velocity $\overline{v}^F$ as $C(g, v^B, \overline{v}^F)$. As before, we count only the pooled gap, rather than discriminating between lead gap and lag gap. Then, the (expected) merging penalty that we use in our DP can essentially be written as follows:

$$C(v^B) = \mathbb{E}[C(g, v^B, \overline{v}^F) | g > G_c(v^B, \overline{v}^F)]. \tag{1.23}$$

To show analytically the required characterization, we take several assumptions:

i. Inter-car distances, and therefore, available gaps g are distributed according to the exponential distribution with parameter λ;

ii. Velocities v^B and $\overline{v}^F$ affect the expected merging penalty only through critical gap $G_c(v^B, \overline{v}^F)$, and therefore the merging penalty $C(g, v^B, \overline{v}^F) = C(g)$ depends only on g;

iii. For $g \geq 0$ the function $C(g)$ maps to the non-negative values and is strictly decreasing. We show results a for negative-exponential function $e^{\gamma - \delta g}$, for parameters $\gamma \geq 0, \delta > 0$. We briefly discuss below also an alternative model with function $e^{\gamma - \delta(g - G_c(v^B, \overline{v}^F))}$. These two functions leads to clean analytical results. Qualitative results for reciprocal function $1/(\gamma + \delta g)$ are similar, although less tractable.

We start from merging penalty function $C(g) = e^{\gamma - \delta g}$. Expected merging penalty, conditional on a merging event, can be written then as (we omit variables of critical gap for brevity):

$$\begin{aligned} C(v^B) = \mathbb{E}[C(g, v^B, \overline{v}^F) | g > G_c] &= \frac{1}{1 - F(G_c)} \int_{G_c}^{\infty} f(g) C(g)\, dg \\ &= \frac{1}{e^{-\lambda G_c}} \int_{G_c}^{\infty} \lambda e^{\gamma - (\lambda + \delta) g}\, dg = \frac{\lambda e^{\gamma - \delta G_c}}{\lambda + \delta}. \end{aligned} \tag{1.24}$$

Since it is known (in particular, it follows from the models we discussed above) that $G_c(v_H, \overline{v}^F) > G_c(v_L, \overline{v}^F)$, for some velocities of the blocked lane driver such that $v_H > v_L$, and hence it follows that $C(v_H) < C(v_L)$. Note that this qualitative result remains unchanged, if we would consider a similar functional form for $C(g, v^B, \overline{v}^F)$, but having the penalty depending on the difference between the actual gap and some constant minimally required gap $g_0 < G_c(v^B, \overline{v}^F)$. Indeed, this leads to the result:

$$C(v^B) = \frac{\lambda e^{\gamma + \delta g_0 - \delta G_c}}{\lambda + \delta}, \tag{1.25}$$

which is the same model with a higher constant.

We consider another special case of the model, when the penalty depends on the difference between the actual gap and the critical gap, i.e. $C(g) = e^{\gamma - \delta(g - G_c)}$. It is straightforward to verify, that in this case the expected penalty would be:

$$C(v^B) = \frac{\lambda e^{\gamma}}{\lambda + \delta},$$

which is independent of v^B. We can conclude, that under reasonable modelling assumptions, ex-post merging penalty in the optimal control and velocity problem is such that $C(v_H) \leq C(v_L)$ for $v_H > v_L$.

We also can use the same model for obtaining the numerical values of $C(v^B)$. As can be seen from relaxation studies, at a high level the merging penalty is the time, lost due to "stretching" leading distance after the lane change to a certain value, normal the for driving conditions on the free lane. Approximately it can be calculated as $(d^F(\overline{v}^F) - g/2)/\overline{v}^F$, that is the time to cover the difference between the gap and the normal for that velocity inter-car distance, at velocity $\overline{v}^F$ (in case $d^F(\overline{v}^F) > g/2$, and zero otherwise). Note that we use only half of gap g, since the merging driver takes into account the lead gap after the merge, and we assume that this is simply the half of the total gap (although it can be not exactly the case in reality).

We can use this (linear) model to fit our exponential model $C(g) = e^{\gamma - \delta g}$. To estimate unknown parameters $\gamma\delta$, we need to take two pairs $g, C(g)$ from described above linear model. The estimates will depend on what two points we take; we use here $g = 0$ and $g = 0.5 d^F(\overline{v}^F)$. It is straightforward to see that this leads to the following estimates:

$$\gamma = \ln \frac{d^F(\overline{v}^F)}{\overline{v}^F}; \qquad \delta = \frac{2 \ln 4/3}{d^F(\overline{v}^F)}.$$

Once we obtained the estimates for fixed $\overline{v}^F$, we can calculate ex-post late-merging penalty $C(v^B)$ according to Eq. (1.24). Note that critical gap $G_c(v^B, \overline{v}^F)$ is needed to calculate merging penalty; any appropriate model can be used for it, we use the model discussed earlier (see Eq. (1.18)).

1.7.2 Travel-time optimization of merging vehicles

We now present the results of Cellular Automata simulation. First, we compare the performance of the proposed DP approach, in terms of average travel time of merging vehicles, against several heuristic or benchmark policies: an uncontrolled "random merging" scenario, "early merging", "late merging," and the parameterized family of "one-threshold policies".

DP policy

The DP policy relies on estimating the parameters of the model introduced in Section 1.4, and then solving (numerically) the corresponding Dynamic Programming problem. The parameters of the model are estimated through training data that we obtain from additional discrete-event simulations, where a blocked-lane driver attempts to merge periodically, without travel-time optimization in mind. While one should aim to estimate the merging model in its full generality, i.e., Eq. (1), in practice this is quite challenging since there are, simply, too many "degrees of freedom." To our knowledge, no estimators for such models, with guaranteed performance, are readily available in the literature. Therefore, through our training data, we estimate the functionally simpler model of Eq. (2), i.e., we assume that merging penalties and merging probabilities only depend on the velocity of the blocked-lane driver, and that only two velocities are available; so, we estimate the parameter values q_H, q_L, c_L, and c_H. In other words, we sacrifice model accuracy for reliable estimation. Yet, the reduced model performs quite well in our discrete-event simulations; at least much better than the various heuristics that we compare it against, as we document below.

Random merging

As a performance benchmark we consider the case where drivers are not provided with any policy recommendation on when and where to merge. Every driver starts attempting to merge at a random distance from the blockage point. We consider the simplest possible case where this distance is uniformly distributed.

In Table 1.4 we summarize the performance of random merging vs. the DP policy, for different values of ρ^B and ρ^F, the density of flow on lane B and F, respectively. These values correspond to a spectrum of traffic conditions, ranging from relatively sparse to dense lane, in both cases. We denote by ΔT the percent decrease of travel time for drivers on the blocked lane under the DP policy, compared to random merging. The parameters $\overline{v}^F$, q_H, q_L, and P are estimated by averaging over 1000 of CA simulations, while the results for ΔT are obtained after averaging over 5000 runs.

Table 1.4: Comparison of DP-based policy and the random merging policy in Cellular Automata simulations. The last column shows gap in expected travel time.

ρ^B	ρ^F	$\overline{v}^F$	v_H	v_L	q_H	q_L	P	c_L	c_H	ΔT
.03	.1	3.87	5	4	.372	.438	8.65	2.36	0.93	3.07%
.06	.1	3.89	5	4	.375	.444	10.8	2.38	0.99	3.68%
.09	.1	3.87	5	4	.379	.435	12.4	2.37	0.95	2.85%
.03	.2	1.91	5	2	.139	.345	6.80	0.96	0	17.05%
.06	.2	1.91	5	2	.138	.340	8.99	0.98	0	15.38%
.09	.2	1.91	5	2	.129	.349	10.6	0.98	0	12.01%
.03	.3	1.21	5	2	.056	.194	6.51	0.43	0	26.6%
.06	.3	1.21	5	2	.051	.208	8.87	0.43	0	22.7%
.09	.3	1.21	5	2	.054	.205	10.1	0.43	0	17.6%

At higher densities of traffic on the free lane, the more interesting case for us, the difference in travel time between random merging and the DP policy becomes greater. Intuitively this makes sense, as the cost of a suboptimal decision is magnified by the congestion in the free lane, and merging closer to the blockage point becomes more appealing.

Moreover, the gap between random merging and the DP policy is quite robust with respect to the density of traffic on the blocked lane, as long as the free lane is relatively dense. We have also found the reported results to be robust to the choice of d, the length of a cell in our simulations, although we do not report on these for brevity. Together, these observations suggest that the most important primitive in terms of its effect on the performance of the DP policy is the density of traffic on the free lane.

One-threshold merging policies

Next, we compare the performance of the DP policy to the class of one-threshold merging policies, which includes intuitive policies such as 'merge early" or "merge late." More specifically, policies included in this class recommend to the driver to start attempting to merge after a given threshold, while the velocity is always kept as high as possible.

We present our simulation results in Figures 1.9 and 1.10, for a sparse and a dense blocked lane, respectively. Note that when the traffic on the free lane is relatively dense, the empirical performance of the DP policy is much better than the "merge early" and "merge late" policies, and very close to the best one-threshold merging policy.[4] The benefit of our framework is that it provides a principled way

[4]As expected, an optimal solution to the single-car problem does not always have the best performance, in simulations, among threshold policies. This could be due to imperfect estimation

to achieve good performance, irrespective of the traffic conditions, in contrast to the aforementioned class of policies, where the best threshold parameter can only be determined empirically, ex post.

The results in Figures 1.9 and 1.10 also confirm one of the insights derived in the comparison to random merging: the performance of the DP policy is quite consistent with respect to the density of the traffic flow on the blocked lane. Correspondingly, the most critical primitive for performance evaluation purposes is the density on the free lane.

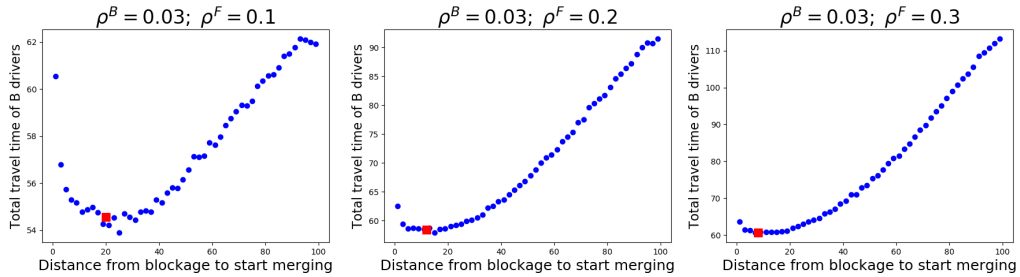


Figure 1.9: Comparison of the DP policy to one-threshold merging policies via Cellular-Automata simulations, for a sparse blocked lane. The merging threshold is presented on the x-axis, and the total travel time (in seconds) on the y-axis. Every point is the average of 5000 simulation runs. The red square corresponds to the proposed DP policy.

1.7.3 Throughput-delay tradeoffs

Finally, we compare the proposed DP policy to the other benchmark policies to compare the bottleneck throughput and delay that they achieve or induce. First we compare the total travel time (delay) of blocked-lane drivers; second, the average flow (throughput) of B-lane drivers through the bottleneck; and third, the total travel time and the flow of all drivers, both on the free and on the blocked lane. In the latter case, effectively, we compare the *bottleneck capacity*, and demonstrate that the proposed approach does not reduce it; in fact, it often improves it. Indeed,

of the model parameters, or the fact that our model does not account for the influence of merging drivers on the free lane. Typically, the DP policy recommends to merge slightly closer to the blockage than the (empirically) best threshold policy, which likely means that either the merging probability is overestimated, or the late merging penalty P is underestimated, or both. However, since the performance of the DP policy for a single car on the blocked lane is even closer to the optimum, the performance gap is likely to be attributed to unaccounted interactions of the blocked-lane cars.

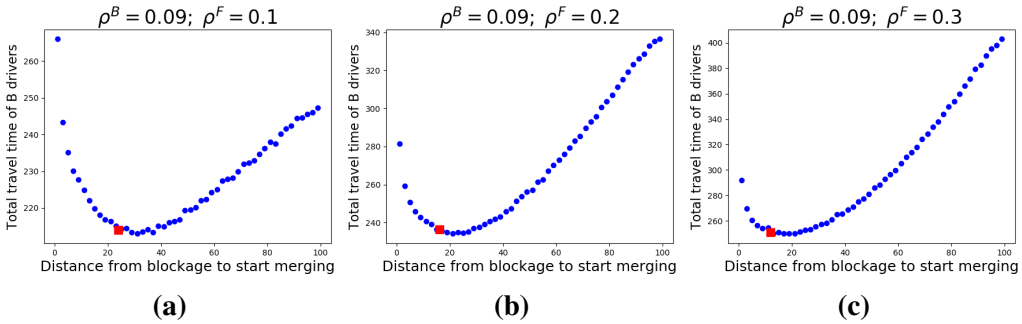


Figure 1.10: Comparison of the DP policy to one-threshold merging policies via Cellular-Automata simulations, for a dense blocked lane. The merging threshold is presented on the x-axis, and the total travel time (in seconds) on the y-axis. Every point is the average of 5000 simulation runs. The red square corresponds to the proposed DP policy.

a policy that improves traffic conditions on one lane at the expense of other lanes, would make little sense. Combined, these results imply that the DP policy, on average, makes more efficient use of the available resources.

In Figures 1.11 and 1.12 we plot the aforementioned measures for the different merging policies, for traffic intensity on B-lane 30% and 60%, respectively, attempting to capture a relatively sparse and a relatively dense scenario. (Simulations at different traffic intensities exhibit, qualitatively, the same behavior.) Note that, in general, increasing the density leads to a steady increase of total travel time. The traffic flow metric behaves differently: for the pooled metric, it grows fast until the bottleneck capacity, and then almost stabilizes. Different policies can lead to slightly different bottleneck capacity. Also, the flow of B-lane drivers can behave differently, for example, it can decrease at very high density, due to the fact that it become harder for B-lane drivers to leave the bottleneck when there are many F-lane drivers. By and large though, the results supports our main thesis: *the DP policy provides a principled way to achieve near-optimal performance under any traffic conditions, whereas the different heuristic policies may or may not perform well depending on the circumstances*.

1.8 Conclusions

Algorithmic-driving and vehicle-to-vehicle communication technologies are reaching maturity, and bring urgency to research on driver behavior, incentives, and control policies for optimal traffic flow. And while decades of research and macroscopic models give good fits to observed phenomena, we still lack a deeper understanding of the underlying causes for fluctuations originated at the driver level,

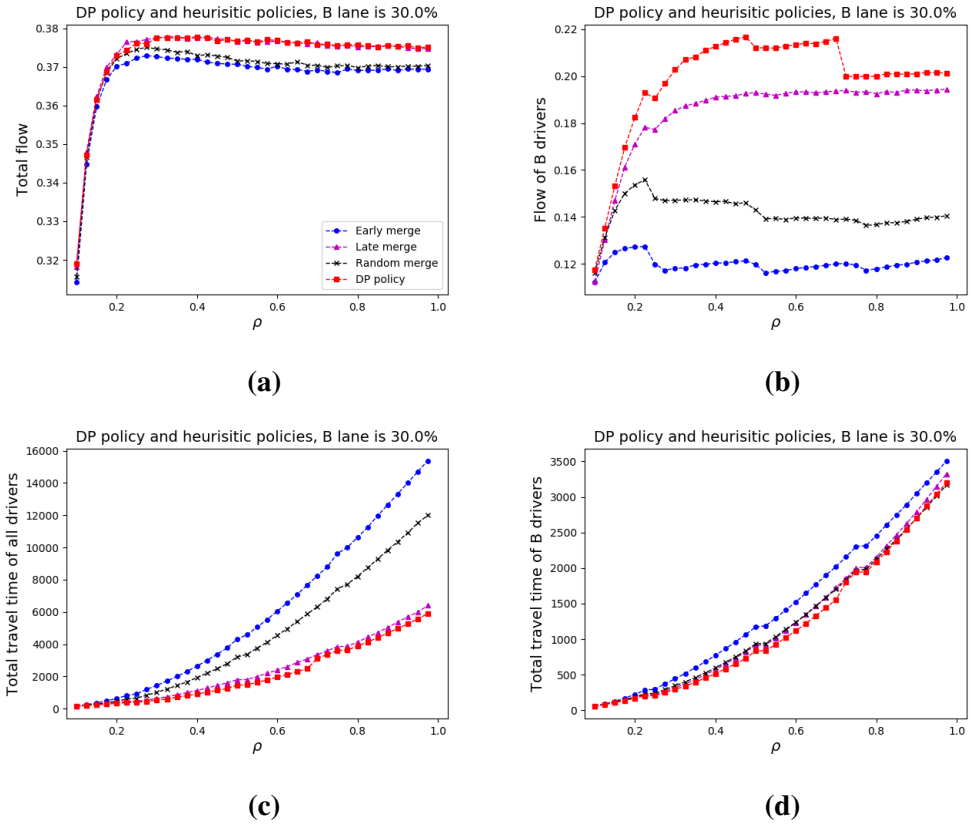


Figure 1.11: Comparison of the DP policy to three other popular heuristic policies, for different traffic densities. Traffic on B-lane is significant (30% of total density)

which makes it difficult to devise good intervention and control policies.

To this end, we isolate a simple traffic situation that is reasonably amenable to analysis. Our stylized model provides insights into both the underlying causes as well as potential management policies for algorithm-assisted drivers, i.e., drivers with clearly defined objectives, whose behavioral or cultural tics play less of a role in traffic modeling.

Our modeling framework and DP formulation allows for a characterization of the optimal merging and velocity control policy, resulting in a somewhat counter-intuitive finding in some parameter regions: in the presence of uncertainty regarding the future state of the target lane, travel-time optimizing drivers may oscillate between high and low velocities while attempting to merge, a perplexing "irrational" behavior often observed in practice. We validate our theoretical analysis via extensive discrete-event simulations under real-life scenarios, where we evaluate the macroscopic impact of the DP policy against various merging heuristics.

Traffic modeling and analysis is extremely challenging as multiple agents

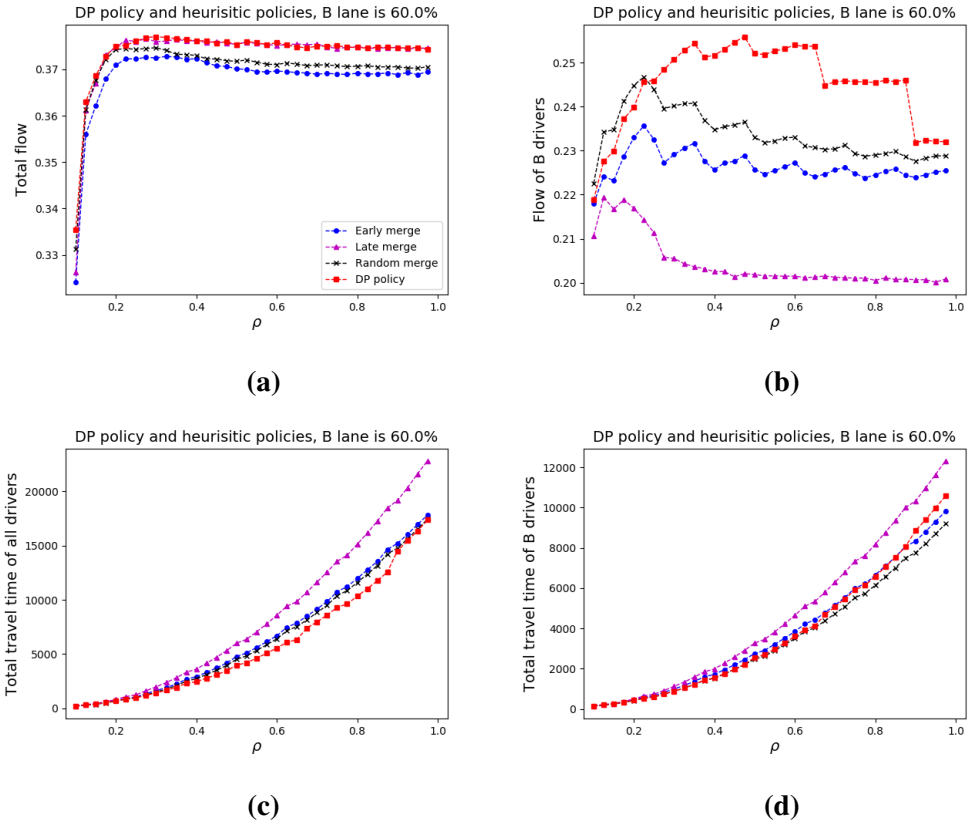


Figure 1.12: Comparison of the DP policy to three other popular heuristic policies, for different traffic densities. Traffic on B-lane is heavy (60% of total density on B-lane)

interact in a dynamically changing environment. However, algorithm-assisted drivers are somewhat easier to model at the microscopic level and the analysis more tractable, potentially leading to actionable insights, and we believe this represents a new and exciting area of research, with a huge potential for making an impact on our daily lives.

For future research, besides rigorous analysis of the case with multiple cars on the blocked lane, there are many interesting questions to consider when one combines our model with free-lane driver behavior, and even more intriguing ones when drivers are modeled as acting strategically.

On the policy and intervention side, apart from setting velocity limits, technology to use prices or set controls on acceleration and distances can lead to significant improvements in traffic control. In the next chapter we consider how setting up an auction or bidding format for the drivers can benefit all driver on the road.

1.9 Appendix

1.9.1 Proof of Proposition 1.5

Our proof strategy relies on establishing sufficient conditions, such that $\Delta E_k \geq 0$, or in other words

$$\mathbb{E}_{k+1}[L] - \mathbb{E}_{k+1}[H] \leq \mathbb{E}_k[L] - \mathbb{E}_k[H]$$

over the three regions of the Final Merging Zone, L, X, and H. To that end, we use the fact that

$$\Delta E_k = \frac{q_L - q_H}{\overline{v}^F} + (1-q_L)\left(T_k^B(v_L) - T_{k+1}^B(v_L)\right) - (1-q_H)\left(T_k^B(v_H) - T_{k+1}^B(v_H)\right). \tag{1.26}$$

Region L: If $\mathbb{E}_{k+1}[L] - \mathbb{E}_{k+1}[H]$ is in region L, and we are in the merging zone then $v_k^{*e} = v_L$, irrespective of the velocity at the start of stage k. Eq. (1.26) implies that

$$\begin{aligned}
\Delta E_k &= \frac{q_L - q_H}{\overline{v}^F} \\
&\quad + (1-q_L)\left(\frac{1}{v_L} + q_L(T_{k+1}^F + c_L) + (1-q_L)T_{k+1}^B(v_L) - T_{k+1}^B(v_L)\right) \\
&\quad - (1-q_H)\left(\frac{2}{v_H + v_L} + q_L(T_{k+1}^F + c_L) + (1-q_L)T_{k+1}^B(v_L) - T_{k+1}^B(v_H))\right) \\
&= \frac{q_L - q_H}{\overline{v}^F} + \frac{1-q_L}{v_L} - \frac{2(1-q_H)}{v_L + v_H} - (1-q_L)q_L T_{k+1}^B(v_L) \\
&\quad - (1-q_H)(1-q_L)T_{k+1}^B(v_L) + (1-q_H)T_{k+1}^B(v_H) + (q_H - q_L)q_L\left(T_{k+1}^F + c_L\right) \\
&= \frac{q_L - q_H}{\overline{v}^F} + \frac{1-q_L}{v_L} - \frac{2(1-q_H)}{v_L + v_H} + (1-q_H)\left(T_{k+1}^B(v_H) - T_{k+1}^B(v_L)\right) \\
&\quad + (q_H - q_L)q_L\left(T_{k+1}^F + c_L - T_{k+1}^B(v_L)\right).
\end{aligned}$$

Note that if it is optimal to merge at v_L, then Eq. (1.2) implies that

$$\frac{2}{v_k^s + v_L} + q_L(T_{k+1}^F + c_L) + (1-q_L)T_{k+1}^B(v_L) \leq \frac{2}{v_k^s + v_L} + T_{k+1}^B(v_L).$$

Rearranging terms, we have that $T_{k+1}^F + c_L \leq T_{k+1}^B(v_L)$, which implies that the last term in the expression above is nonnegative (as $q_H - q_L < 0$). Combining this fact with the lower bound on $T_{k+1}^B(v_H) - T_{k+1}^B(v_L)$ provided by Lemma 1.2, we

have the following lower bound for ΔE_k:

$$\begin{aligned}\Delta E_k &\geq \frac{q_L - q_H}{\overline{v}^F} + \frac{1 - q_L}{v_L} - \frac{2(1 - q_H)}{v_L + v_H} + \frac{(1 - q_H)(v_L - v_H)}{v_L(v_L + v_H)} \\ &= \frac{q_L - q_H}{\overline{v}^F} - \frac{q_L - q_H}{v_L} \geq 0.\end{aligned}$$

Note that if k is in Region L in the merging zone, then $\Delta E_k \geq 0$ as long as $\overline{v}^F \leq v_L$. This also establishes Proposition 1.4.

Region X: If $\mathbb{E}_{k+1}[L] - \mathbb{E}_{k+1}[H]$ is in region X, then it is optimal to merge and $v_k^{*e} \neq v_k^s$. Eq. (1.26) implies that

$$\begin{aligned}\Delta E_k &= \frac{q_L - q_H}{\overline{v}^F} \\ &\quad + (1 - q_L)\left(\frac{2}{v_H + v_L} + q_H(T_{k+1}^F + c_H) + (1 - q_H)T_{k+1}^B(v_H) - T_{k+1}^B(v_L)\right) \\ &\quad - (1 - q_H)\left(\frac{2}{v_H + v_L} + q_L(T_{k+1}^F + c_L) + (1 - q_L)T_{k+1}^B(v_L) - T_{k+1}^B(v_H)\right) \\ &= \frac{q_L - q_H}{\overline{v}^F} - \frac{2(q_L - q_H)}{v_L + v_H} + (q_H - q_L)T_{k+1}^F + (1 - q_L)q_H c_H \\ &\quad - (1 - q_H)q_L c_L + (1 - q_L)(1 - q_H)\big(T_{k+1}^B(v_H) - T_{k+1}^B(v_L)\big) \\ &\quad + (1 - q_H)T_{k+1}^B(v_H) - (1 - q_L)T_{k+1}^B(v_L).\end{aligned}$$

By adding and subtracting $q_L T_{k+1}^B(v_H)$, we can rewrite the expression as follows:

$$\begin{aligned}\Delta E_k &= \frac{q_L - q_H}{\overline{v}^F} - \frac{2(q_L - q_H)}{v_L + v_H} + (q_H - q_L)T_{k+1}^F + (1 - q_L)q_H c_H \\ &\quad - (1 - q_H)q_L c_L + (1 - q_L)(1 - q_H)\big(T_{k+1}^B(v_H) - T_{k+1}^B(v_L)\big) \\ &\quad + (1 - q_L)T_{k+1}^B(v_H) - (1 - q_L)T_{k+1}^B(v_L) + (q_L - q_H)T_{k+1}^B(v_H) \\ &= \frac{q_L - q_H}{\overline{v}^F} - \frac{2(q_L - q_H)}{v_L + v_H} + (1 - q_L)q_H c_H - (1 - q_H)q_L c_L \\ &\quad + (1 - q_L)(2 - q_H)\big(T_{k+1}^B(v_H) - T_{k+1}^B(v_L)\big) + (q_L - q_H)\big(T_{k+1}^B(v_H) - T_{k+1}^F\big).\end{aligned} \tag{1.27}$$

To find a lower bound on the last term, recall that in region X we have that

$$\frac{2}{v_L + v_H} - \frac{1}{v_L} \leq E_{k+1}[L] - E_{k+1}[H] \leq \frac{1}{v_H} - \frac{2}{v_L + v_H}.$$

Note that

$$
\begin{aligned}
&E_{k+1}[L] - E_{k+1}[H] \\
=&(q_L - q_H)T^F_{k+1} + q_L c_L - q_H c_H + (1-q_L)T^B_{k+1}(v_L) - (1-q_H)T^B_{k+1}(v_H) \\
=&(q_L - q_H)T^F_{k+1} + q_L c_L - q_H c_H + (1-q_L)T^B_{k+1}(v_L) \\
&-(1-q_L)T^B_{k+1}(v_H) - (q_L - q_H)T^B_{k+1}(v_H) \\
=&(q_L - q_H)\big(T^F_{k+1} - T^B_{k+1}(v_H)\big) + q_L c_L - q_H c_H \\
&-(1-q_L)\big(T^B_{k+1}(v_H) - T^B_{k+1}(v_L)\big),
\end{aligned}
$$

where, again, we add and subtract $q_L T^B_{k+1}(v_H)$ to get the second equality. Using the upper bound for $E_{k+1}[L] - E_{k+1}[H]$ and the equality above, we establish that

$$
\begin{aligned}
&(q_L - q_H)(T^B_{k+1}(v_H) - T^F_{k+1}) \geq \\
&\qquad \geq \frac{2}{v_L + v_H} - \frac{1}{v_H} + q_L c_L - q_H c_H - (1-q_L)\big(T^B_{k+1}(v_H) - T^B_{k+1}(v_L)\big).
\end{aligned}
$$

Combining the above lower bound with Eq. (1.27), we obtain:

$$
\begin{aligned}
\Delta E_k \geq& \frac{q_L - q_H}{\overline{v}^F} - \frac{2(q_L - q_H)}{v_L + v_H} + q_H q_L (c_L - c_H) \\
&+ \frac{2}{v_L + v_H} - \frac{1}{v_H} + (1-q_L)(1-q_H)\big(T^B_{k+1}(v_H) - T^B_{k+1}(v_L)\big).
\end{aligned}
$$

Finally, using the lower bound on $T^B_{k+1}(v_H) - T^B_{k+1}(v_L)$ provided by Lemma 1.2, we get:

$$
\begin{aligned}
\Delta E_k \geq& \frac{q_L - q_H}{\overline{v}^F} - \frac{2(q_L - q_H)}{v_L + v_H} + q_H q_L (c_L - c_H) \\
&+ \frac{2}{v_L + v_H} - \frac{1}{v_H} + \frac{(v_L - v_H)(1-q_L)(1-q_H)}{v_L(v_L + v_H)}.
\end{aligned} \tag{1.28}
$$

Hence, in region X, the right-hand side of the above expression being nonnegative constitutes a sufficient condition for $\Delta E_k \geq 0$;

Region H: If $\mathbb{E}_{k+1}[L] - \mathbb{E}_{k+1}[H]$ is in region H, then it is optimal to merge and $v^{*e}_k = v_H$, irrespective of the velocity at the start of stage k. Eq. (1.26) implies

that

$$\begin{aligned}\Delta E_k &= \frac{q_L - q_H}{\overline{v}^F} \\ &+ (1-q_L)\left(\frac{2}{v_H+v_L} + q_H(T^F_{k+1}+c_H) + (1-q_H)T^B_{k+1}(v_H) - T^B_{k+1}(v_L)\right) \\ &- (1-q_H)\left(\frac{1}{v_H} + q_H(T^F_{k+1}+c_H) + (1-q_H)T^B_{k+1}(v_H) - T^B_{k+1}(v_H)\right) \\ &= \frac{q_L - q_H}{\overline{v}^F} + \frac{2(1-q_L)}{v_L+v_H} - \frac{1-q_H}{v_H} + q_H(q_L-q_H)\left(T^B_{k+1}(v_H) - T^F_{k+1} - c_H\right) \\ &+ (1-q_L)\left(T^B_{k+1}(v_H) - T^B_{k+1}(v_L)\right) \\ &\geq \frac{q_L - q_H}{\overline{v}^F} + \frac{2(1-q_L)}{v_H+v_L} - \frac{1-q_H}{v_H} + \frac{(1-q_L)(v_L-v_H)}{v_L(v_L+v_H)},\end{aligned} \tag{1.29}$$

where, to derive the last inequality, we use Proposition 1.2 and Lemma 1.2. If the last expression is nonnegative, then $\Delta E_k \geq 0$ in region H.

Summarizing, if both conditions from Eq. (1.28) and Eq. (1.29) hold, which is precisely the requirement of Part (i) of Proposition 1.5, then ΔE_k is nonnegative for every k;

The proof of the second part of Proposition 1.5 builds on the proof of the first part. Specifically, if the solution lies in region X for at most one stage, then the optimal policy does not exhibit velocity oscillations. A sufficient condition for this to hold is if ΔE_k being greater than or equal to the "length" of region X, for every k. Combining Eq. (1.28) with the fact that the length of the region X is equal to

$$\frac{(v_L - v_H)^2}{v_L v_H (v_L + v_H)},$$

we derive the following sufficient condition:

$$\begin{aligned}\frac{q_L - q_H}{\overline{v}^F} - \frac{2(q_L-q_H)}{v_L+v_H} + q_H q_L(c_L - c_H) + \frac{2}{v_L+v_H} \\ - \frac{1}{v_H} + \frac{(v_L-v_H)(1-q_L)(1-q_H)}{v_L(v_L+v_H)} \geq \frac{(v_L-v_H)^2}{v_L v_H (v_L+v_H)}\end{aligned} \tag{1.30}$$

Summarizing, if both conditions from Eq. (1.29) and Eq. (1.30) hold, which is precisely the requirement of Part (ii) of Proposition 1.5, then the solution lies in region X at most once, hence the optimal policy exhibits no oscillations.

Chapter 2

POSITION BIDDING FOR ALGORITHM-ASSISTED DRIVERS

2.1 Introduction

During the whole history of car transportation, we got used to the fact that heterogeneous drivers with potentially different urgency of their trips must share the same infrastructure with the same priority in our daily commuting. The only exception is when the urgency is undoubted and socially agreed upon—such as ambulances and other emergency services. Much research efforts were spent on optimization of aggregate traffic flow, improvements of infrastructure, and increasing safety. Personal incentives of the drivers are often left unattended. Nevertheless, if the drivers who are in a hurry could reach their destinations faster, it could reduce potential economic loss. The information about trip urgency is inherently private and is not known to other drivers or traffic engineers. Therefore, any mechanism that tries to assign priorities on the road must consider the drivers' personal incentives. It is important to understand how the trip's exposed urgency is related to the true one. Are they equal? If they are not, can we clearly understand the dependencies between them? Failing to do so might lead to the wrong conclusions and improper solutions, which would harm society rather than help it.

To make the discussion more concrete, we restrict ourselves to the same forced merge scenario from the Chapter 1, which has all key features of the above discussion and can be frequently observed in practice. A two-lane highway reduces to just one lane—because of an accident, road works, or simply because the highway narrows down. All drivers must merge from the blocked lane to the free lane. Assuming that there is some space for maneuvers and several potential spots

the drivers could merge into, which one should they choose? If the blocked-lane driver is in a hurry (in other words, she has a higher valuation of one unit of time), it would be better to place her ahead of as many drivers as possible. On the other hand, if most of the free-lane drivers have a high valuation of time, while the blocked-lane driver not, we would like to place her behind.

An auction or a similar bargaining mechanism involving money transfers between drivers could be a solution. Many auction formats are known to be an efficient solution in different settings where certain goods are to be allocated among agents with private valuations of these goods. The problem we introduced above shares similar characteristics. If the blocked-lane driver would be bidding for positions, we can expect that higher valuations would result in higher bids, which have a higher chance of being accepted by other drivers, leading to trades that potentially are beneficial for everyone. However, as we detail in Section 2.2, the structure of the problem at hand is not amenable to off-the-shelf mechanism design approaches from Economics (Vickrey-Clarke-Groves mechanism, auctions with externalities, position auctions, sequential bargaining) or classical congestion pricing ideas from Transportation. Hence, our contribution lies in designing mechanisms tailored to the particular setting, which are computationally efficient and achieve very good performance.

There are also good reasons why these interactions were not considered in Traffic Engineering and other research communities in the past. They were impossible to implement in practice: they require complicated communication that needs to be done on the fly, in parallel with the driving operations. To make it safe and usable, merging bidding must be processed in (at least partly) automated manner and rely on advanced wireless communications between the drivers. These components are close to becoming the industry standard: V2V and robotic drivers are being tested and are on the way to become commercially available. It is the right time to develop a better understanding of how these technologies can be used efficiently.

With this in mind, in the current chapter, we study the problem of merging via position bidding. We take the side of a traffic regulator, who is interested in designing the efficient mechanism in the sense of reaching as high social welfare as possible. Moreover, the proposed mechanism must be implementable (that is, efficient computationally) since our target environment is highly dynamical and continually changing, and decisions are to be made quickly. We also aim to develop a budget-balanced mechanism. We do not want the central regulator to collect or inject any money in the system; it would seriously damage the mechanism's practicality.

What makes this problem challenging is that the drivers are inherently selfish. We model them as economic agents. They act strategically; that is, they are individually rational and seek to maximize their objective (in other words, our

mechanism must be incentive compatible). In our setting, it means they prefer shorter trip times for themselves and do not care about trip times of the rest of the drivers.

We should stress that this is a particularly important matter, and in our model, we treat it as carefully as possible. As we discuss in Section 2.2, in many existing attempts of traffic auctions design strategic behavior of the drivers is neglected to a large extent. Auctions or other mechanisms in traffic can lead to a complex environment. Unlike many rather static settings, traffic interactions are dynamic, constantly changing, with many ways of how one agent's actions might affect the rest of the drivers. Analyzing equilibria in such situations might be a challenging task. Nevertheless, not paying enough attention to drivers' incentives might render any mechanism to be useless or even harmful to society. It is the responsibility of the traffic regulator to construct the mechanism in such a way that facilitates understanding drivers' behavior and allows for reliable predictions.

We propose two different mechanisms with different properties that can be useful in different situations. We provide analysis that demonstrates that both these mechanisms lead to the easily predictable behavior of all participating drivers. The first mechanism allows the blocked-lane driver to bid on positions as she moves along the highway, and we hence call it "Tail-to-Head"—because she moves from the tail of the platoon on the free lane to the head of it. All carefully constructed commitments that are prescribed by the mechanism make sure that: (i) incentives on the side of the free-lane drivers are clear, and they simply take the bid if it is above their valuation (ii) it is guaranteed that social utility would not decrease after the termination of the process. We show that the mechanism results in the outcome that is close to the social optimum under the wide range of parameters. Finding optimal bids for the blocked-lane driver in the general case can be time-consuming if the number of drivers is high. We propose heuristics based on the relaxation of DP and intuitively good bidding strategies to overcome this. We show that for uniform valuations, our main heuristic on average loses no more than 2% of total utility, while takes 4 orders of magnitude less time to calculate. For some special cases of cost functions, we provide an exact analytical characterization of equilibrium.

Our second mechanism operates differently. The blocked-lane driver proceeds from head to tail of the platoon (hence we call it "Head-to-Tail"), and takes different commitments. Unlike Tail-to-Head mechanism, the driver commits to merge right away at the first accepted offer. This mechanism does not guarantee positive social utility. However, it does allow for the easily predictable behavior of the free-lane drivers. Moreover, it is much simpler computationally, and it provides a higher expected utility of the blocked-lane driver. In certain situations, such as limited budget or very few drivers on the free lane, the Head-to-Tail mechanism might also be preferable in terms of social utility. Therefore, the traffic regulator

can choose one of these mechanisms depending on the situation and the goals.

The remainder of the chapter is organized as follows. In Section 2.2 we review the literature and discuss why the existing approaches are not suitable for the problem on hand; in Section 2.3 we discuss the modeling assumptions, introduce our Tail-to-Head and Head-to-Tail mechanisms and discuss insights that can be drawn from the solutions; in Section 2.4 we introduce the notion of Partial Information Social Optimum, provide its theoretical comparison with Tail-to-Head mechanism for special case as well as provide an extensive numerical comparison for more general cases; in Section 2.5 we discuss the performance of the mechanisms when the blocked-lane driver has a limited budget; finally, in Section 2.6 we conclude the chapter and discuss future directions.

2.2 Literature Review

The economic literature on auctions, bargaining, and mechanisms is very rich, and many problems theoretically are well-understood. However, many existing approaches do not apply to our scenario due to the complexity of the setting. In this section, we try to outline what are those approaches and why we can not take one of them and directly apply to our scenario.

As our final goal is to develop a mechanism, a field of mechanism design is the first place to look up for a solution. The paper McAfee and McMillan (1987) is a classical review that discusses equilibrium and main results for models such as first-price sealed-bid and key results for mechanisms. The book Rakesh V. Vohra (2011) provides a good theory review of mechanisms design, based on linear programming. Among all mechanisms and approaches to construct them, the most famed is the VCG mechanism (originated from Vickrey (1961)). The mechanism is truthful, efficient, incentive-compatible, and individually rational. Theoretically, VCG has many very desirable properties. However, its practical use remains limited, and there are many reasons why VCG is rarely a choice for practitioners (see Rothkopf (2007)). Unfortunately, in our setting, attempts to apply VCG also face problems. We discuss more detailed examples of VCG use in our setting in Section 2.3.1, but we can summarize two main issues: (i) lack of budget balancedness. Either collecting money or, even worse, paying them to the participants for the efficient outcome is highly undesirable (ii) VCG payments are defined based on the "cost of existence for society" of a participant. Unlike simple and isolated scenarios such as single-item auctions, in the discussed traffic scenarios, this is not well-defined and is ambiguous to interpret. It makes use of VCG problematic, and therefore, we need to look for a different, more specific mechanism or an auction.

One such approach is position auctions. The topic has quickly developed, fol-

lowing its practical application in internet advertising. Papers Varian (2007) and Edelman et al. (2007) are seminal on that side, followed by a significant burst of the literature. As a few notable examples: Gomes and Sweeney (2014) studies numerically the existence and efficiency of Bayes-Nash equilibria; Athey and Ellifason (2011) incorporates the strategical behavior of the customers into the model. There are two main reasons why this type of auctions is not implementable in our setting. Firstly, they are used in environments where interchanging positions essentially requires no time and no cost and thus is always possible. On the contrary, permuting car positions to new ones might be unsafe, costly, and simply not possible because of time constraints (operations need to be finished before hitting the blockage). It might be much more appropriate and feasible not to find a new permutation of positions, but to make an auction format that tries to handle a single driver's merging, which is easier to implement in real road situations. If needed, that can be repeated until there is no time for changes left. The second reason is the problem with drivers who do not want to participate in the auction and instead wish to hold their current position. This is crucially different from online advertising auctions, where the advertiser has to pay continually to the provider of the services. There is no such provider in the traffic scenario, which also makes collecting payment questionable, and again raises the problem with balancedness of the payments (how to redistribute them without destroying the equilibrium?).

Another possible approach is the auctions with externalities. A general treatment of the auctions with externalities can be found in Jehiel et al. (1999). Negative externalities play a significant role in our setting, and they are probably its defining feature, on a technical side. Assume that a free-lane driver competes with others in selling his position. If another driver wins, externalities depend heavily on whether the winning position is behind or ahead. This creates asymmetric externalities. On the contrary, a key assumption of the model in Jehiel et al. (1999) is that these externalities must be symmetric (they assumed to be a random variable drawn from a distribution with the same support, so they are equal in a stochastic sense). A similar restriction is imposed in Jehiel and Moldovanu (2000). On top of that, the mechanisms discussed in these papers involve one seller and many drivers, while we are interested in a reverse auction (with the blocked-lane driver as the only buyer). Generally speaking, the literature on auctions is centered on symmetric settings (because of analytical tractability) and one seller, many buyers (due to richer practical applications). Only recently, a general single-item first-price sealed-bid auction was analyzed for two bidders and uniform distributions (Kaplan and Zamir (2012)), which is by itself a rather simple auction format. This fact emphasizes how hard it is for the researchers to dig deeper into the asymmetric auctions.

Instead of looking at a one-shot traffic auction, another possible way of modeling traffic positions could be a sequence of bargains between the blocked-lane

driver and the free-lane drivers. The classical models considering such trades between two agents are bilateral trading (Myerson and Satterthwaite (1983)), sequential bargaining (Fudenberg and Tirole (1983)), and simultaneous-offers bargaining (Chatterjee and Samuelson (1990)). These papers capture essential features that we want to incorporate, such as private valuations. They can help to analyze a case of one blocked-lane driver and one free-lane driver. However, this case is not attractive from a practical viewpoint, and we want to get reliable predictions for a greater number of agents. As we discussed above, the externalities play a crucial role in our setting and prevent the problem from being decoupled into a sequence of independent bargains. There are, however, the models that specifically tackle bargaining between many agents or between one agent and multiple agents (see De Fontenay and Gans (2014), Segal (1999)). They fit our setting better, and notably, allow for externalities between the agents. Nevertheless, due to being aimed to solve different practical problems (such as agreements between firms) and significant analytical complexity, they tend to take technical assumptions that we can not afford. Examples are the symmetry of the agents or the continuity of utility functions. We discussed the former above; the latter is problematic since the blocked-lane driver can abruptly switch from one merging position to another after an arbitrarily small change in offers, which is not representable by continuous utility functions of corresponding free-lane drivers.

One of the few practically used policies that aim to help congestion and reduce travel time is congestion pricing. Early studies Walters (1961) and Vickrey (1969a) hinted the point that the negative externalities that the drivers cause to others should be corrected in the form of tolls in congested areas. In the following decades, congestion pricing received a solid theoretical basis in the economic literature and numerous practical applications, both implemented and planned for future years. A comprehensive review of the modern state of congestion pricing can be found in de Palma and Lindsey (2011). This pricing mechanism aims to decrease the travel time of all drivers in a transportation link, path or area. It does not specifically tackle the differences in the value of time for the different drivers. Instead, it is concentrated around deriving the average value of time and setting up a socially beneficial price. On the contrary, we aim to increase efficiency by providing a mechanism for position trading to the drivers. In principle, this is perfectly compatible with other measures taken to improve the overall traffic fluency (including congestion pricing). The second known issue with the congestion pricing is the fact that the payment is collected by a central traffic regulator, i.e. there is a lack of budget balance. This results in an extensive discussion, both academic and public, regarding how these funds should be spent, and creates public controversy and aversion (see Small (1992)). The mechanism that allows for voluntary participation and leaves the opportunity to keep the position without negative externalities potentially can achieve a higher level of public support.

Finally, traffic engineers and computer science researchers also try to come up with efficient ways of trading priorities on the road, expecting autonomous vehicles to become prevalent in the nearest future. Carlino et al. (2013) and Vasirani and Ossowski (2012) experiment with auction-like schemes for intersection management. Rewald and Stursberg (2016) is dedicated to the cooperation of autonomous drivers via auctions in a broader range of situations. Notably, in the paper Lin et al. (2019), the authors develop a bargaining scheme with monetary payments in case of a discretionary lane change. However, they assume only two different types of drivers, truthfully reporting their value of time. In general, traffic engineering literature tends to propose algorithmic schemes and formats of the auctions and test them in simulations without careful derivation of equilibria. Fixed bidding functions, a random budget, or other simplifications such as truthful bidding are typically assumed. On the contrary, we aim to develop a mechanism that provides a more solid theoretical basis, and hence we only consider bidding strategies that are incentive-compatible.

2.3 Position Bidding Mechanisms

2.3.1 VCG application to a merging scenario

Before introducing our position bidding mechanism, we want to highlight why the use of VCG mechanism is problematic in traffic. In order to apply VCG to our merging scenario, for ease of illustration, we use a simpler model than the one described later in Section 2.3.2. We assume that there are n drivers on the free-lane and a single driver b on the blocked lane, and all distances between cars on the free lane are equal. Furthermore, assume that the merging of the blocked-lane driver causes all free-lane drivers to lose an equal amount of time (pushes them back equally). We normalize this time to be equal to 1 unit, and all drivers have valuations of this normalized unit of time equal $v_b, v_1, \ldots, v_n$. All free-lane drivers initially posses positions on the free-lane according to their indices, while the blocked-lane driver can have a free and guaranteed merging into the position 0, resulting in zero total utility. If the blocked-lane driver merges into position i, her utility is $u_b = iv_b - q_b$, where q_b is the sum of payments she pays to the free-lane drivers. If the blocked-lane driver merges somewhere ahead of a free-lane driver j, the utility of the latter is $u_j = -v_j + q_j$, where q_j is the payment received.

Our goal is to find such a position (one of the positions $0, 1, \ldots, n$) for the blocked-lane driver that maximizes total utility. VCG is the mechanism that allows for a truthful, incentive-compatible, individually-rational, and efficient way of doing so.

There are, however, two significant issues with the application of VCG in our

setting. The first one is the general lack of balancedness property. All the payments that have to be transferred within the mechanism are not guaranteed to sum up to zero, and there is usually deficit or surplus. The proposed merging bargaining is aimed to be a day-to-day scheme. Therefore, neither collecting nor inserting funds from the central planner is acceptable. One possible way out can be found in Proposition 5.6 in Krishna (2010). According to it, if VCG runs with a surplus in the sum of all payments, then it is possible to create an efficient, incentive-compatible, and individually rational mechanism, which is budget-balanced at the same time. However, as we shortly demonstrate, VCG can run with surplus or with deficit depending on the drivers' valuations. The second issue is ambiguity in interpretations of the "cost to the society" of the driver in this scenario. We discuss it below.

To simplify the discussion further, let us consider the case of $n = 2$. According to VCG, the outcome that maximizes social optimum is always chosen. This makes the problem with equal valuations trivial, since in this case, in the sense of social welfare, any outcome is equal. We therefore consider the case $v_b \neq v_1 \neq v_2$. According to VCG rules, payment of the driver i is determined as:

$$q_i = U_{-i} - W_{-i},$$

where U_{-i} is the sum of outcomes for other drivers if i would not present on the road, and W_{-i} is the sum of outcomes for other drivers in the current outcome.

Here is when the second issue we mentioned above comes into play. The second component is straightforward to determine. However, there are different possible interpretations of the first one. If one free-lane driver disappears, one could bring arguments to claim that any of these interpretations is true:

1. The blocked-lane driver can take opportunity and merge freely into the free spot where the free-lane driver was.

2. All the free-lane drivers advance ahead for 1 spot and gain utility based on it, then we identify best total utility from merging.

3. All the free-lane drivers advance 1 spot ahead, but we consider it as a new "zero-utility" situation (because this additional utility is not related per se to the trading of the positions). However, the outcome of bargaining and the best total utility still might change.

4. The blocked-lane driver does not merge and the free-lane drivers do not advance ahead. However, the distance change, resulting in different merging penalties and advantages for the blocked-lane driver.

Which scenario to choose seems to be a choice of arbitrarily taken assumptions. As a result, analysis using VCG is unreliable. The third scenario seems the most sensible one since it excludes any changes in valuations that are not related to position trading per se. To demonstrate VCG calculations example, below we assume that this is the correct scenario. If one free-lane driver is not present, all the drivers stay in their spots (the blocked lane driver merges to position 0), resulting in zero utility. Then we determine the best possible outcome (but keeping in mind that one driver is missing). There are 3 possible outcomes: driver b merges into positions $0, 1$ or 2. This depends on what reordering gives higher total utility, which in turn depends on valuations. We consider the result in all 3 cases.

1. b merges into positon 0. This happens if $v_b < v_1$. In this case $q_b = q_2 = 0$, since absence of drivers b, f_2 does not change the outcome. However, q_1 depends on whether $v_b < v_2$. If so, then $q_1 = 0$. If $v_b > v_2$, then in the absence of driver f_1, driver b can merge into position 0, and

$$q_1 = v_b - v_2 > 0,$$

   in which case the sum of the payments $Q = q_b + q_1 + q_2 > 0$. Otherwise $Q = 0$.

2. b merges into position 1. This happens if $v_1 < v_b$, but $v_1 + v_2 > v_b$. In this case:

$$\begin{aligned} q_2 &= (v_b - v_1) - (v_b - v_1) = 0; \\ q_1 &= 0 - v_b = -v_b; \\ q_b &= 0 - (-v_1) = v_1. \end{aligned}$$

   In this case, $Q = v_1 - v_b < 0$.

3. b merges into position 2. This happens if $v_1 + v_2 < 2v_b$. Payments are determined as follows:

$$\begin{aligned} q_2 &= \begin{cases} (v_b - v_1) - (2v_b - v_1) = -v_b & \text{if } v_b > v_1; \\ -(2v_b - v_1) = v_1 - 2v_b & \text{if } v_b < v_1; \end{cases} \\ q_1 &= (v_b - v_2) - (2v_b - v_2) = -v_b; \\ q_b &= 0 - (-v_1 - v_2) = v_1 + v_2. \end{aligned}$$

   In this case either $Q = v_1 + v_2 - 2v_b < 0$ or $Q = 2v_1 + v_2 - 3v_b$, and the sign of the expression depends on the valuations.

Summarizing, for the case $n = 2$, we can conclude that it is ambiguous whether VCG runs with surplus or with deficit—it depends on values v_i and v_b. Hence even if we overcome ambiguity in the interpretation of VCG application and choose one, it is still impossible to construct a budget-balanced mechanism using VCG.

2.3.2 Modelling assumptions

We consider the same blocked-lane scenario from Chapter 1. One lane was blocked due to an accident or road works. There is traffic on both lanes, and eventually, drivers from the blocked lane (lane B) must merge into the free lane (lane F). We assume that the free-lane drivers move in a relatively dense platoon, and most of the time, there is not enough space between cars for the blocked-lane driver to merge without delaying any cars behind. However, the exact distances between cars might vary. On the other hand, the blocked lane is relatively sparse, and there are few drivers. As in the previous chapter, we concentrate on a single blocked-lane driver (for example, the closest one to the blockage).

We assume that there is always a position at the end of the platoon where the blocked-lane driver could easily merge. We numerate this position as position 0 or call it *the reserved position*. There are n drivers on the free-lane ahead of the reserved position. We denote them $f_1, \ldots, f_n$, starting from the back of the platoon (closer to the blockage, higher the index). The blocked-lane driver can merge to the corresponding positions in front of each free-lane car. There might be cars on the free lane ahead of car f_n, but given the current flow speed, the blocked-lane driver cannot reach them before hitting the blockage point; hence they are unavailable and excluded from participation.

The distances between cars on the free lane are $d_1, \ldots, d_n$, where d_i stands for a distance between cars f_i and f_{i+1}, excluding the length of the car f_i (i.e., d_i is the the distance from front bumper of f_i to a rear bumper of f_{i+1}). The distances are common knowledge to everyone on the road and remain constant until the bargaining process ends. We illustrate the merging scenario described above in Figure 2.1.

Whenever a merge happens ahead of position j, say at position $i > j$, the car f_j experiences delay that depends not only on d_j but also on all the distances up to d_i. There are two main reasons for this delay. First of all, there might be not enough space at all for the blocked-lane driver to merge. In this case, the free-lane car needs first to create a sufficient gap, and this is effectively is a loss of travel time. Second, even if there is enough space, cars involved in merging adjust (stretch) distances after the merge happens. This is known as the "relaxation" phenomenon and is a well-studied process; see, for example, Zheng et al. (2013). Both effects take place only when the car merges immediately in front of another

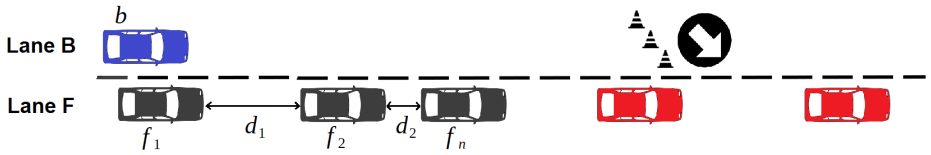


Figure 2.1: The blocked-lane scenario. Blue car is seeking for a merging spot, grey cars are n cars reachable by the blue car, and red cars can not be reached, hence are not considered a part of bargaining process.

car. However, after the merge it propagates upstream, causing loss of travel time for the drivers behind. Eventually, it vanishes since whenever there is a larger gap, the effect is mitigated or completely disappears.

To capture this rather complex dynamic, we use a set of functions

$$C_{ij}(d_j, d_{j+1}, \ldots, d_i),$$

which map distances $d_j, \ldots, d_i$ into a time delay for driver j given that the blocked-lane driver merged into position j. Each function $C_{ij}(\cdot)$ should possess the following properties:

- It is non-negative because it always represents loss of travel time; merging of another car ahead can not create an advantage for the car behind.
- It decreases in each argument (i.e. must decrease when any of the distances increase, given that the rest are the same). This is because the car is affected less by the merge when any of these distances are higher.
- For fixed distances and fixed i, $C_{ij} \leq C_{ik}$ whenever $j < k$. This property is similar to a previous one: the penalty decays over distance, therefore it must be less when there are more cars between the free-lane driver and the merging position.

We do not discuss in detail the exact form of function family $C_{ij}(\cdot)$. Identifying them based on the physics of the traffic and behavior of the drivers can well be a topic for separate research. Moreover, the shift of technologies towards automated drivers can change underlying behavior. In our theoretical analysis, we abstract from these functions and use a general definition of functions $C_{ij}(\cdot)$. For a numerical study, we propose a reasonable and parametric functional form, discussed in Section 2.3.5, and test with different parameters, demonstrating robustness.

We assume that all drivers on road have different urgency of trip, or *valuation of time*. Assuming that all possible valuations are bounded, we normalize them to be real numbers between 0 and 1, which represents a (normalized) monetary

equivalent of one unit of travel time for the driver. Furthermore, assume that valuations per unit of time for all drivers $v_b, v_1, \ldots, v_n$ are independent and identically distributed random variables, according to a distribution with CDF $F(x)$ and PDF $f(x)$. These valuations are private information, and each driver knows only his or her own valuation. The distribution of values are common knowledge.

It should be noted that although we assume i.i.d. valuations throughout the chapter, the identical distribution of valuations can be relaxed at a cost of some computational and notational burden. In principle, different car types might have different distributions of valuations. For example, drivers of trucks or corporate vans is likely to be on duty and potentially have different valuations of time than other drivers. If this is a public information, we can incorporate these considerations into the model. On the contrary, the assumption about the independence of valuations can not be easily revoked. If valuations are not independent, our analysis needs to be adjusted to account possible learning, that the driver can obtain through past interactions. We leave this to future studies for now.

The blocked-lane driver with valuation v_b gains $r_i(v_b)$ units of time from merging into the position i. We discuss exact functional form of these functions in more details in Section 2.3.5. For now, we assume that these functions possess two key properties: (i) they are weakly increasing in i, i.e. $r_i(v_b) \geq r_{i-1}(v_b)$ (i) each $r_i(v_b)$ is weakly increasing in v_b.

Finally, we assume that utility of all drivers is linearly additive with respect to travel time, through the functions $r_i(\cdot)$ and $C_{ij}(\cdot)$, and to the monetary payment. We now try to find mechanisms that can perform well under all stated modeling assumptions.

2.3.3 Tail-to-Head (T2H) Mechanism

As we discuss above, we want to find a mechanism that is incentive compatible, individually rational, and budget-balanced. We can give up truthfulness and we might agree to sacrifice some fraction of efficiency, although we still prefer to have as high total utility (or *social welfare*) as possible. As we discuss in Section 2.2, to our knowledge, there is no obvious or easy way to do so, for example, using general approaches or readily available mechanisms.

There can be different ways of constructing such a mechanism. In any case, it seems to be of high importance to prevent the negative externalities (here, the adverse effect of the merging car on cars behind). One way to do so is to make the blocked-lane driver pay compensations to all drivers involved. If all free-lane drivers agree on payment, it means that their valuations of time are low enough to justify putting the blocked-lane driver ahead in the traffic flow. In our first proposed bargaining scheme, which we call "Tail-to-Head", we not only manage to satisfy this crucial condition but also facilitate analysis by a straightforward

structure of the mechanism. Key features of the mechanism are: (i) sequential (one-by-one) and one-shot bargaining (no chance to re-negotiate); (ii) the free-lane drivers are passive price-takers—greatly decreases potential opportunities for strategic interaction without harming incentive compatibility; (iii) the blocked-lane driver can merge only if all affected parties have agreed upon price.

Specifically, the blocked-lane driver starts from the tail of the platoon, a driver f_1, by proposing payment rate b_1 per unit of travel time lost due to the merge ahead. That is, if f_1 would suffer any delay due to the blocked-lane driver merging ahead (not necessarily immediately in front of f_1), the blocked-lane driver would pay to him according to the rate b_1. If the driver f_1 accepts, the blocked-lane driver can proceed and negotiates with the next driver f_2. The process is repeated exactly in the same manner until the offer is rejected at any point. In this case, the blocked-lane driver has to take the last slot that was agreed upon and cannot proceed with negotiations. Then she pays to all drivers behind according to the delay they incur. Naturally, the blocked-lane diver is interested in violating these restrictions and would like to proceed with negotiations further ahead. However, as we demonstrate later, following this rule is crucial for the mechanism we propose. Therefore, it is of high importance that the traffic regulator imposes such commitment and forces the drivers to follow them, if necessary.

Since both the function and the distances are common knowledge, delay values for each possible pair i, j can be precomputed and stored as numbers. We will denote them as c_j^i (superscript stands for position to which the blocked-lane driver merges). Similarly, we can pre-calculate all $r_i(\cdot)$. We denote them as r_i.

Because the bid is not a fixed amount of money, but rather a rate, incentives on the side of F-lane drivers are especially easy to check. If driver f_i is offered a price b_i per unit of time, expected utilities of accepting and rejecting for him are:

$$u_i^A(b_i) = (b_i - v_i)(\mathbb{P}_0 c_0^i + \mathbb{P}_1 c_1^i + \ldots + \mathbb{P}_N c_N^i);$$
$$u_i^R(b_i) = 0,$$

where $\mathbb{P}_k$ is the probability that the blocked-lane driver would merge into position k at the end of the bargaining process. It is preferable for the driver f_i to accept the offer as long as $u_i^A(b_i) \geq u_i^R(b_i)$. Note the second multiplication term in $u_i^A(b_i)$ is always non-negative (as both probabilities and costs are non-negative), and the decision is determined by the sign of $b_i - v_i$. As a result, the optimal strategy for every driver is straightforward: to accept any offered payment that is higher than his valuation. Then, the problem of finding optimal bids to every position i for the blocked-lane driver can be formulated as a Dynamic Program (DP). The primitives of the problem are as follows:

State: The state of the system consists of all bids (rates per unit of

time) for which the blocked-lane driver has an agreement, prior to the stage i; we denote it as a vector with $i-1$ elements $(b_1, \ldots, b_{i-1})$.

Control: The control/decision at stage i is the bid b_i to offer to a current free-lane driver f_i.

Uncertainty: The uncertainty at stage i is the unknown valuation of time v_i of a free-lane driver f_i.

Dynamics: If a current free-lane driver rejects to be paid at rate b_i in exchange for giving up a slot to the blocked-lane driver, then the blocked-lane driver can not proceed with bargaining. She has to merge to a previously negotiated position $i-1$. The final utility of the blocked-lane driver then:

$$r_{i-1} - \sum_{k=1}^{i-1} b_k c_k^{i-1}.$$

If there is an agreement, then she can proceed to negotiations for position $i+1$, and the vector of bids is updated by adding b_i.

If we denote reward-to-go of the blocked-lane driver at stage i as $J_i(\cdot)$, then the solution can be written as DP recursion:

$$J_i\big(b_1, \ldots, b_{i-1}\big) = \max_{b_i \geq 0} \Bigg\{ J_{i+1}\big(b_1, \ldots, b_i\big) \mathbb{P}\big[f_i \text{ accepts } b_i\big] + \left(r_{i-1} - \sum_{k=1}^{i-1} b_k c_k^{i-1} \right) \mathbb{P}\big[f_i \text{ rejects } b_i\big] \Bigg\}, \tag{2.1}$$

with boundary condition

$$J_{N+1}\big(b_1, \ldots, b_N\big) = r_N - \sum_{k=1}^{N} b_k c_k^N. \tag{2.2}$$

For tractability, we derive our analytical results based on two assumptions:
(i) The merging cost satisfy $c_k^i = c_k,\ i = 1, \ldots, N$;
(ii) The probability distribution function satisfies $F(x) = x,\ x \in [0, 1]$. In other words, the time valuations are uniformly distributed in the $[0, 1]$ interval.

The corresponding Dynamic Programming problem is described by the recursion:

$$J_i\big(b_1, \ldots, b_{i-1}\big) = \max_{0 \leq b_i \leq 1} \left\{ J_{i+1}\big(b_1, \ldots, b_i\big) b_i + \left(r_{i-1} - \sum_{k=1}^{i-1} b_k c_k \right) \big(1 - b_i\big) \right\}. \tag{2.3}$$

We denote by b_i^* the solution to the above optimization problem, i.e., optimal bid that the merging driver makes to driver i on the platoon.

Proposition 2.1. *The optimal expected reward-to-go of the T2H mechanism is equal to*

$$J_i\big(b_1,\ldots,b_{i-1}\big) = r_{i-1} - \sum_{k=1}^{i-1} b_k c_k + \sum_{j=i}^{N} \big(r_j - r_{j-1} - b_j^* c_j\big) \prod_{k=i}^{j} b_k^*,\ i = 1,\ldots,N+1, \tag{2.4}$$

and the optimal bids to drivers $i,\ldots,N$ can be computed through the recursion:

$$b_i^* = \max\left\{\min\left\{\frac{1}{2c_i}\left(r_i - r_{i-1} + \sum_{j=i+1}^{N}\big(r_j - r_{j-1} - b_j^* c_j\big)\prod_{k=i+1}^{j} b_k^*\right), 1\right\}, 0\right\}. \tag{2.5}$$

Proof. The result is proved by induction.

Basis of Induction: For $i = N+1$, Eq. (2.4) implies that

$$J_{N+1}\big(b_1,\ldots,b_N\big) = r_N - \sum_{k=1}^{N} b_k c_k,$$

which holds because it coincides with the boundary condition of the DP problem in Eq. (2.2);

Induction Step: Assume that the statement of Eq. (2.4) holds for $i = l+1$, $l \in \{1,\ldots,N\}$, i.e.,

$$J_{l+1}\big(b_1,\ldots,b_l\big) = r_l - \sum_{k=1}^{l} b_k c_k + \sum_{j=l+1}^{N}\big(r_j - r_{j-1} - b_j^* c_j\big)\prod_{k=l+1}^{j} b_k^*.$$

We show that, in turn, the statement of Eq. (2.4) also holds for $i = l$. According to Eq. (2.3), we have that

$$\begin{aligned} J_l\big(b_1,\ldots,b_{l-1}\big) = \max_{0\le b_l\le 1}\Bigg\{&\left(r_l - \sum_{k=1}^{l} b_k c_k + \sum_{j=l+1}^{N}\big(r_j - r_{j-1} - b_j^* c_j\big)\prod_{k=l+1}^{j} b_k^*\right) b_l \\ &+ \left(r_{l-1} - \sum_{k=1}^{l-1} b_k c_k\right)\big(1 - b_l\big)\Bigg\} \\ = r_{l-1} - \sum_{k=1}^{l-1} b_k c_k + \max_{0\le b_l\le 1}\Bigg\{&\Bigg(r_l - r_{l-1} - b_l c_l \\ &+ \sum_{j=l+1}^{N}\big(r_j - r_{j-1} - b_j^* c_j\big)\prod_{k=l+1}^{j} b_k^*\Bigg) b_l\Bigg\}. \end{aligned} \tag{2.6}$$

We study the function

$$g_l(x) \equiv \left(r_l - r_{l-1} - xc_l + \sum_{j=l+1}^{N} \left(r_j - r_{j-1} - b_j^* c_j \right) \prod_{k=l+1}^{j} b_k^* \right) x.$$

Note that it is continuous, differentiable, and strictly concave in x. Hence, the unconstrained problem of maximizing $g_l(x)$ has a unique solution, which is its stationary point x^*:

$$\begin{aligned} g_l'(x^*) = 0 &\iff r_l - r_{l-1} - x^* c_l + \sum_{j=l+1}^{N} \left(r_j - r_{j-1} - b_j^* c_j \right) \prod_{k=l+1}^{j} b_k^* - x^* c_i = 0 \\ &\iff x^* = \frac{1}{2c_l} \left(r_l - r_{l-1} + \sum_{j=l+1}^{N} \left(r_j - r_{j-1} - b_j^* c_j \right) \prod_{k=l+1}^{j} b_k^* \right). \end{aligned}$$

Now, we wish to maximize the continuous function $g_l(\cdot)$ over the compact set $[0, 1]$. The Extreme Value Theorem implies that the solution to the constrained problem is either the (unique) stationary point x^*, or lies on the boundary of the constraint set. On that end, we distinguish between two cases:

1. If $x^* \geq 0$, then $g_l'(x) \geq 0$, for all $x \in \left[0, x^*\right]$. Consequently, $b_l^* = \min\left\{x^*, 1\right\}$;
2. If $x^* < 0$, then note that $g_l'(x) < 0$, for all $x \in [0, 1]$. Consequently, $b_l^* = 0$.

Combining the analysis of the two cases, we have that

$$b_l^* = \max\left\{\min\left\{x^*, 1\right\}, 0\right\},$$

which verifies Eq. (2.5). Finally, substituting back to Eq. (2.6), we have that

$$\begin{aligned} J_l\left(b_1, \ldots, b_{l-1}\right) &= r_{l-1} - \sum_{k=1}^{l-1} b_k c_k \\ &\quad + \left(r_l - r_{l-1} - b_l^* c_l + \sum_{j=l+1}^{N} \left(r_j - r_{j-1} - b_j^* c_j \right) \prod_{k=l+1}^{j} b_k^* \right) b_l^* \\ &= r_{l-1} - \sum_{k=1}^{l-1} b_k c_k + \sum_{j=l}^{N} \left(r_j - r_{j-1} - b_j^* c_j \right) \prod_{k=l}^{j} b_k^*, \end{aligned}$$

which verifies Eq. (2.4) and completes the induction. ■

It is convenient to study the "normalized" optimal expected reward-to-go of the T2H mechanism, after removing the revenue that has been secured up until the negotiation with driver i:

$$\Delta J_i \equiv J_i\big(b_1,\ldots,b_{i-1}\big) - \left(r_{i-1} - \sum_{k=1}^{i-1} b_k c_k\right).$$

Proposition 2.1 implies that

$$\Delta J_i = \sum_{j=i}^{N} \big(r_j - r_{j-1} - b_j^* c_j\big) \prod_{k=i}^{j} b_k^*, \tag{2.7}$$

so that ΔJ_i does not depend on $b_1,\ldots,b_{i-1}$. Indeed, given the structure of the mechanism and the fact that time valuations of different drivers are independent, a bid that was made to (and accepted by) a driver early on does not have an impact on future bids.

Using this notation, we can rewrite the DP recursion in Eq. (2.3) as follows:

$$\Delta J_i = \max_{0\le b_i\le 1} \big\{\big(\Delta J_{i+1} + r_i - r_{i-1} - b_i c_i\big) b_i\big\}. \tag{2.8}$$

with boundary condition $\Delta J_{N+1} = 0$.

Corollary 2.1. *The optimal bids and the (normalized) optimal expected reward-to-go of the T2H mechanism can be computed via the simple recursive equations:*

1. *If* $\Delta J_{i+1} \le -r_i + r_{i-1} + 2c_i$, *then* $b_i^* = \big(\Delta J_{i+1} + r_i - r_{i-1}\big)/2c_i$ *and* $\Delta J_i = \big(b_i^*\big)^2 c_i$;
2. *Otherwise,* $b_i^* = 1$ *and* $\Delta J_i = \Delta J_{i+1} + r_i - r_{i-1} - c_i \iff J_i\big(b_1,\ldots,b_{i-1}\big) = J_{i+1}\big(b_1,\ldots,b_{i-1},1\big)$.

Proof. Eq. (2.5) and Eq. (2.7) imply directly that

$$b_i^* = \min\left\{\frac{1}{2c_i}\big(\Delta J_{i+1} + r_i - r_{i-1}\big), 1\right\}. \tag{2.9}$$

It is worth noting that the optimal bid to driver i is nonnegative: Eq. (2.8) implies that $\Delta J_i \ge 0$, since the feasible solution $b_i = 0$ results in zero value of the objective function; a value that the optimal solution is no worse than. Hence, $\Delta J_{i+1} + r_i - r_{i-1} \ge 0$, which implies that $b_i^* \ge 0$.

We distinguish between the two cases:

1. If $\Delta J_{i+1} \leq -r_i + r_{i-1} + 2c_i$, then Eq. (2.9) implies that $b_i^* = \frac{1}{2c_i}\big(\Delta J_{i+1} + r_i - r_{i-1}\big)$. In turn, Eq. (2.8) implies that

$$\begin{aligned}
\Delta J_i &= \big(\Delta J_{i+1} + r_i - r_{i-1} - b_i^* c_i\big) b_i^* \\
&= \left(\Delta J_{i+1} + r_i - r_{i-1} - \frac{1}{2c_i}\big(\Delta J_{i+1} + r_i - r_{i-1}\big) c_i\right) \times \\
&\times \frac{1}{2c_i}\big(\Delta J_{i+1} + r_i - r_{i-1}\big) \\
&= \frac{1}{4c_i}\big(\Delta J_{i+1} + r_i - r_{i-1}\big)^2 \\
&= \big(b_i^*\big)^2 c_i.
\end{aligned}$$

2. If $\Delta J_{i+1} > -r_i + r_{i-1} + 2c_i$, then Eq. (2.9) implies that $b_i^* = 1$. In turn, Eq. (2.8) implies that

$$\Delta J_i = \big(\Delta J_{i+1} + r_i - r_{i-1} - b_i^* c_i\big) b_i^* = \Delta J_{i+1} + r_i - r_{i-1} - c_i.$$

   By substitution in the definition of ΔJ_i, we have that the latter expression implies that

$$J_i\big(b_1, \ldots, b_{i-1}\big) = J_{i+1}\big(b_1, \ldots, b_{i-1}, 1\big).$$

■

Proposition 2.2. *The expected Social Welfare under the T2H mechanism, SW_{T2H}, is equal to*

$$SW_{T2H} = \sum_{j=1}^{N} \left(r_j - r_{j-1} - \frac{1}{2} b_j^* c_j\right) b_j^* \prod_{k=1}^{j-1} b_k^*. \tag{2.10}$$

Proof. The expected Social Welfare of the T2H mechanism is equal to the expected reward of the merging driver, J_1, in addition to the externalities that her merging may cause on the free lane drivers. The former is provided directly by Proposition 2.1, hence we start our analysis from the latter. Fix $j \in \{1, \ldots, N\}$. The externality that the merging driver imposes on driver j of the platoon is:

— $\big(b_j^* - v_j\big) c_j$, on the event $\prod_{k=1}^{j} 1_{\{b_k^* \geq v_k\}}$;

— 0, otherwise.

Therefore, the expected externality that the merging driver imposes on driver j of the platoon, X_j, assuming that time valuations are independent and distributed identically to the uniform distribution $U(0,1)$, is equal to

$$\begin{aligned}
X_j &= \mathbb{E}_{v_1,\ldots,v_j}\left[\left(b_j^* - v_j\right)c_j \prod_{k=1}^{j} 1_{\{b_k^* \geq v_k\}}\right] \\
&= \mathbb{E}_{v_j}\left[\left(b_j^* - v_j\right)c_j 1_{\{b_j^* \geq v_j\}}\right] \prod_{k=1}^{j-1} \mathbb{P}\left(b_k^* \geq v_k\right) \\
&= \left(\mathbb{P}\left(b_j^* \geq v_j\right)b_j^* c_j - \mathbb{E}_{v_j}\left[v_j 1_{\{b_j^* \geq v_j\}}\right] c_j\right) \prod_{k=1}^{j-1} b_k^* \\
&= \left(\left(b_j^*\right)^2 c_j - \frac{1}{2}\left(b_j^*\right)^2 c_j\right) \prod_{k=1}^{j-1} b_k^*.
\end{aligned}$$

Consequently, the total expected externality, X_{T2H} that the merging driver imposes on the drivers of the platoon is equal to

$$X_{T2H} = \sum_{j=1}^{N} \left(b_j^*\right)^2 c_j \prod_{k=1}^{j-1} b_k^* - \frac{1}{2}\sum_{j=1}^{N} \left(b_j^*\right)^2 c_j \prod_{k=1}^{j-1} b_k^*. \tag{2.11}$$

Eq. (2.4) and Eq. (2.11) imply that the expected Social Welfare under the T2H mechanism, SW_{T2H}, is equal to

$$\begin{aligned}
SW_{T2H} &\equiv J_1 + X_{T2H} \\
&= \sum_{j=1}^{N} \left(r_j - r_{j-1} - b_j^* c_j\right) \prod_{k=1}^{j} b_k^* + \sum_{j=1}^{N} \left(b_j^*\right)^2 c_j \prod_{k=1}^{j-1} b_k^* - \frac{1}{2}\sum_{j=1}^{N} \left(b_j^*\right)^2 c_j \prod_{k=1}^{j-1} b_k^* \\
&= \sum_{j=1}^{N} \left(r_j - r_{j-1}\right) b_j^* \prod_{k=1}^{j-1} b_k^* - \sum_{j=1}^{N} \left(b_j^*\right)^2 c_j \prod_{k=1}^{j-1} b_k^* + \sum_{j=1}^{N} \left(b_j^*\right)^2 c_j \prod_{k=1}^{j-1} b_k^* \\
&\quad - \frac{1}{2}\sum_{j=1}^{N} \left(b_j^*\right)^2 c_j \prod_{k=1}^{j-1} b_k^* \\
&= \sum_{j=1}^{N} \left(r_j - r_{j-1} - \frac{1}{2} b_j^* c_j\right) b_j^* \prod_{k=1}^{j-1} b_k^*,
\end{aligned}$$

which confirms Eq. (2.10). Note that the internal payments from the merging driver to the drivers of the platoon cancel out in the expression above, and thus, as expected, do not affect the Social Welfare under the T2H mechanism. ■

Note that Eq. (2.10) can be considered as a function of any set of bids $b_1, \ldots, b_N$, not only the bids solving T2H Dynamic Program. The result would be the expected Social Welfare under T2H mechanism when the blocked-lane driver bids $b_1, \ldots, b_N$. Tail-to-Head bids are incentive-compatible with the blocked-lane driver, but not necessarily deliver minimum to the expected social welfare. In Section 2.4.1, we discuss what "socially optimal" bids can perform this, in a special case of homogeneous costs and rewards.

2.3.4 Head-to-Tail (H2T) Mechanism

Tail-to-Head mechanism guarantees no negative externalities and thus is expected to perform well in terms of social welfare. However, is this effect powerful enough to justify a relatively strong commitment on the block-lane driver's side? To check this point in numerical tests, we propose an alternative mechanism. It is constructed with the priority of the blocked-lane driver's utility but might negatively affect the rest of the drivers. For the blocked-lane driver, the most valuable positions are at the head of the platoon. Therefore, it is reasonable for her to start from the head, and just take the position of the first driver who accepts the offer. Given the fact that positions closer to the head provide a strictly higher utility (less travel time), and given the identical distribution of valuations, it can be a reasonable simplification. On the positive side, it greatly facilitates analysis and computational efficiency, as we demonstrate later. Summarizing specific assumptions that make this model different from the Tail-to-Head model: (i) bargaining process starts from driver f_n, then proceeds to the driver f_{n-1}, etc.; (ii) the blocked-lane driver does not commit to stop bargaining if any driver declines; instead, she commits to merge right away if the offer is accepted; (iii) the scheme implies a payment to a single driver, not to all drivers who face externalities. As an important consequence, no free-lane drivers are guaranteed to have positive utility.

This particular bargaining order might be more suitable if the driver has entered the tapper region (near the blockage). If she cannot accelerate more than F-lane drivers, Head-to-Tail order would arise naturally. Another potential use of this scheme is when the traffic regulator prefers maximizing the utility of the blocked-lane drivers instead of total utility—for example, there is identified urgency on the block-lane driver's side.

In this "Head-to-Tail" model, the utility of accepting/rejecting the offer by a free-lane driver remains the same. If there is an agreement, the blocked-lane driver merges immediately, and utility of the free-lane driver i is $(b_i - v_i)c_i^i$. If the free-lane driver declines the offer, there is 0 utility, since the blocked lane driver has already visited the drivers ahead of i and they declined (thus, no potential negative externalities may arise in the future). Hence, the probabilities of accepting and rejecting bid b_i remain the same (a free-lane driver accepts only if

the offered compensation rate per unit of time exceeds valuation). As a result, we can formulate a similar Dynamic Program to solve for optimal bids. We keep the same notation as before (including the numeration of the free-lane drivers). The primitives then can be written as follows:

> *State*: The only state information is the number of stage i (the number of the current driver).
>
> *Control*: The control/decision at stage i is the bid-per-unit-of-time b_i to offer to a current free-lane driver f_i.
>
> *Uncertainty*: The uncertainty at stage i is the unknown valuation of time v_i of free-lane driver f_i.
>
> *Dynamics*: If a current free-lane driver declines to be paid $b_i c_i^i$ in exchange of giving up a slot to the blocked-lane driver, the blocked-lane driver proceeds to bargaining with driver $i-1$. If there is an agreement, then she merges right away and the process is terminated.

The DP recursion can be written as follows:

$$\tilde{J}_i = \max_{\tilde{b}_i \geq 0} \left\{ \big(r_i - \tilde{b}_i c_i\big) \mathbb{P}\big[f_i \text{ accepts } \tilde{b}_i\big] + \tilde{J}_{i-1} \mathbb{P}\big[f_i \text{ rejects } \tilde{b}_i\big] \right\},$$

with boundary condition

$$\tilde{J}_0 = 0. \tag{2.12}$$

For tractability, we derive our analytical results based on two assumptions:
(i) The merging cost satisfy $c_k^i = c_k,\ i = 1, \ldots, N$;
(ii) The probability distribution function satisfies $F(x) = x,\ x \in [0,1]$. In other words, the time valuations are uniformly distributed in the $[0,1]$ interval.

The corresponding Dynamic Programming problem is described by the recursion:

$$\tilde{J}_i = \max_{0 \leq \tilde{b}_i \leq 1} \left\{ \big(r_i - \tilde{b}_i c_i\big) \tilde{b}_i + \tilde{J}_{i-1} \big(1 - \tilde{b}_i\big) \right\}. \tag{2.13}$$

We denote by $\tilde{b}_i^*$ the solution to the above optimization problem, i.e., optimal bid that the merging driver makes to driver i on the platoon.

Proposition 2.3. *The optimal expected reward-to-go of the H2T mechanism is equal to*

$$\tilde{J}_i = \sum_{j=1}^{i} \big(r_j - \tilde{b}_j^* c_j\big) \tilde{b}_j^* \prod_{k=j+1}^{i} \big(1 - \tilde{b}_k^*\big), \qquad i = 0, \ldots, N, \tag{2.14}$$

and the optimal bids to drivers $1, \ldots, i$ *can be computed through the recursion:*

$$\tilde{b}_i^* = \max\left\{\min\left\{\frac{1}{2c_i}\left(r_i - \sum_{j=1}^{i-1}\left(r_j - \tilde{b}_j^* c_j\right)\tilde{b}_j^* \prod_{k=j+1}^{i-1}\left(1-\tilde{b}_k^*\right)\right), 1\right\}, 0\right\}. \tag{2.15}$$

Proof. The result is proved by induction.

Basis of Induction: For $i = 0$, Eq. (2.14) implies that $\tilde{J}_0 = 0$, which holds because it coincides with the boundary condition of the DP problem in Eq. (2.12);

Induction Step: Assume that the statement of Eq. (2.4) holds for $i = l-1$, $l \in \{1, \ldots, N\}$, i.e.,

$$\tilde{J}_{l-1} = \sum_{j=1}^{l-1}\left(r_j - \tilde{b}_j^* c_j\right)\tilde{b}_j^* \prod_{k=j+1}^{l-1}\left(1-\tilde{b}_k^*\right).$$

We show that, in turn, the statement of Eq. (2.14) also holds for $i = l$. According to Eq. (2.13), we have that

$$\tilde{J}_l = \max_{0\le b_l\le 1}\left\{\left(r_l - \tilde{b}_l c_l\right)\tilde{b}_l + \left(\sum_{j=1}^{l-1}\left(r_j - \tilde{b}_j^* c_j\right)\tilde{b}_j^* \prod_{k=j+1}^{l-1}\left(1-\tilde{b}_k^*\right)\right)\left(1-\tilde{b}_l\right)\right\} \tag{2.16}$$

We study the function

$$h_l(x) \equiv r_l x - c_l x^2 + (1-x)\left(\sum_{j=1}^{l-1}\left(r_j - \tilde{b}_j^* c_j\right)\tilde{b}_j^* \prod_{k=j+1}^{l-1}\left(1-\tilde{b}_k^*\right)\right).$$

Note that it is continuous, differentiable, and strictly concave in x. Hence, the unconstrained problem of maximizing $h_l(x)$ has a unique solution, which is its stationary point x^*:

$$\begin{aligned} h_l'(x^*) = 0 &\iff r_l - 2c_l x - \left(\sum_{j=1}^{l-1}\left(r_j - \tilde{b}_j^* c_j\right)\tilde{b}_j^* \prod_{k=j+1}^{l-1}\left(1-\tilde{b}_k^*\right)\right) = 0 \\ &\iff x^* = \frac{1}{2c_l}\left(r_l - \sum_{j=1}^{l-1}\left(r_j - \tilde{b}_j^* c_j\right)\tilde{b}_j^* \prod_{k=j+1}^{l-1}\left(1-\tilde{b}_k^*\right)\right). \end{aligned}$$

Now, we wish to maximize the continuous function $h_l(\cdot)$ over the compact set $[0, 1]$. The Extreme Value Theorem implies that the solution to the constrained problem is either the (unique) stationary point x^*, or lies on the boundary of the constraint set. On that end, we distinguish between two cases:

1. If $x^* \geq 0$, then $h_l'(x) \geq 0$, for all $x \in \left[0, x^*\right]$. Consequently, $\tilde{b}_l^* = \min\left\{x^*, 1\right\}$;

2. If $x^* < 0$, then note that $h_l'(x) < 0$, for all $x \in [0, 1]$. Consequently, $\tilde{b}_l^* = 0$.

Combining the analysis of the two cases, we have that

$$\tilde{b}_l^* = \max\left\{\min\left\{x^*, 1\right\}, 0\right\},$$

which verifies Eq. (2.15). Finally, substituting back to Eq. (2.16), we have that

$$\begin{aligned}\tilde{J}_l &= \left(r_l - \tilde{b}_l^* c_l\right)\tilde{b}_l^* + \left(\sum_{j=1}^{l-1}\left(r_j - \tilde{b}_j^* c_j\right)\tilde{b}_j^* \prod_{k=j+1}^{l-1}\left(1 - \tilde{b}_k^*\right)\right)\left(1 - \tilde{b}_l^*\right) \\ &= \sum_{j=1}^{l}\left(r_j - \tilde{b}_j^* c_j\right)\tilde{b}_j^* \prod_{k=j+1}^{l}\left(1 - \tilde{b}_k^*\right),\end{aligned}$$

which verifies Eq. (2.14) and completes the induction. ■

The result analogical to Corollary 2.1 can be established.

Corollary 2.2. *The optimal bids and the optimal expected reward-to-go of the H2T mechanism can be computed via the simple recursive equations:*

1. *If $\tilde{J}_{i-1} > r_i$, then $\tilde{b}_i^* = 0$ and $\tilde{J}_i = \tilde{J}_{i-1}$;*

2. *If $r_i - 2c_i \leq \tilde{J}_{i-1} \leq r_i$, then $\tilde{b}_i^* = \left(r_i - \tilde{J}_{i-1}\right)/2c_i$ and $\tilde{J}_i = \tilde{J}_{i-1} + \left(\tilde{b}_i^*\right)^2 c_i$;*

3. *Otherwise, $\tilde{b}_i^* = 1$ and $\tilde{J}_i = r_i - c_i$.*

Proof. The result follows directly from the proof of Proposition 2.3. ■

Similar to Tail-to-Head model, it is possible to get an analytical expression for Social Welfare for H2T.

Proposition 2.4. *The expected Social Welfare under the H2T mechanism, SW_{H2T}, is equal to*

$$SW_{H2T} = \sum_{j=1}^{N}\left(r_j - \frac{1}{2}\tilde{b}_j^* c_j - \frac{1}{2}\sum_{l=1}^{j-1} c_l\right)\tilde{b}_j^* \prod_{k=j+1}^{N}\left(1 - \tilde{b}_k^*\right). \tag{2.17}$$

Proof. The expected Social Welfare of the H2T mechanism is equal to the expected reward of the merging driver, $\tilde{J}_N$, in addition to the externalities that her merging may cause on the free lane drivers. The former is provided directly by Proposition 2.3, hence we start our analysis from the latter. Fix $j \in \{1, \dots, N\}$. The externality that the merging driver imposes on driver j of the platoon is:

— $-v_j c_j$, on the event $\sum_{l=j+1}^{N} 1_{\{b_l^* \geq v_l\}} \prod_{k=l+1}^{N} 1_{\{\tilde{b}_k^* < v_k\}}$;

— $\big(\tilde{b}_j^* - v_j\big)c_j$, on the event $1_{\{b_j^* \geq v_j\}} \prod_{k=j+1}^{N} 1_{\{\tilde{b}_k^* < v_k\}}$;

— 0, otherwise.

Therefore, the expected externality that the merging driver imposes on driver j of the platoon, X_j, assuming that time valuations are independent and distributed identically to the uniform distribution $U(0,1)$, is equal to

$$
\begin{aligned}
X_j &= \mathbb{E}_{v_j,\dots,v_N}\Bigg[\big(\tilde{b}_j^* - v_j\big)c_j 1_{\{b_j^* \geq v_j\}} \prod_{k=j+1}^{N} 1_{\{\tilde{b}_k^* < v_k\}} \\
&\quad - v_j c_j \sum_{l=j+1}^{N} 1_{\{\tilde{b}_l^* \geq v_l\}} \prod_{k=l+1}^{N} 1_{\{\tilde{b}_k^* < v_k\}}\Bigg] \\
&= \mathbb{E}_{v_j}\Big[\big(b_j^* - v_j\big)c_j 1_{\{b_j^* \geq v_j\}}\Big] \prod_{k=j+1}^{N} \mathbb{P}\big(\tilde{b}_k^* < v_k\big) \\
&\quad - \mathbb{E}_{v_j}\left[v_j c_j\right] \sum_{l=j+1}^{N} \mathbb{P}\big(\tilde{b}_l^* \geq v_l\big) \prod_{k=l+1}^{N} \mathbb{P}\big(\tilde{b}_k^* < v_k\big) \\
&= \Big(\mathbb{P}\big(b_j^* \geq v_j\big)b_j^* c_j - \mathbb{E}_{v_j}\Big[v_j 1_{\{b_j^* \geq v_j\}}\Big]c_j\Big) \prod_{k=j+1}^{N} \big(1 - \tilde{b}_k^*\big) \\
&\quad - \frac{1}{2}c_j \sum_{l=j+1}^{N} \tilde{b}_l^* \prod_{k=l+1}^{N} \big(1 - \tilde{b}_k^*\big) \\
&= \left(\big(b_j^*\big)^2 c_j - \frac{1}{2}\big(b_j^*\big)^2 c_j\right) \prod_{k=j+1}^{N} \big(1 - \tilde{b}_k^*\big) - \frac{1}{2}c_j \sum_{l=j+1}^{N} \tilde{b}_l^* \prod_{k=l+1}^{N} \big(1 - \tilde{b}_k^*\big).
\end{aligned}
$$

Consequently, the total expected externality, X_{H2T} that the merging driver

imposes on the drivers of the platoon is equal to

$$X_{H2T} = \sum_{j=1}^{N} \left(b_j^*\right)^2 c_j \prod_{k=j+1}^{N} \left(1-\tilde{b}_k^*\right) - \frac{1}{2}\sum_{j=1}^{N} \left(b_j^*\right)^2 c_j \prod_{k=j+1}^{N} \left(1-\tilde{b}_k^*\right) - \frac{1}{2}\sum_{j=1}^{N} c_j \sum_{l=j+1}^{N} \tilde{b}_l^* \prod_{k=l+1}^{N} \left(1-\tilde{b}_k^*\right). \tag{2.18}$$

Eq. (2.14) and Eq. (2.18) imply that the expected Social Welfare under the H2T mechanism, SW_{H2T}, is equal to

$$\begin{aligned} SW_{H2T} &\equiv \tilde{J}_N + X_{H2T} \\ &= \sum_{j=1}^{N} \left(r_j - \tilde{b}_j^* c_j\right)\tilde{b}_j^* \prod_{k=j+1}^{N} \left(1-\tilde{b}_k^*\right) + \sum_{j=1}^{N} \left(b_j^*\right)^2 c_j \prod_{k=j+1}^{N} \left(1-\tilde{b}_k^*\right) \\ &\quad - \frac{1}{2}\sum_{j=1}^{N} \left(b_j^*\right)^2 c_j \prod_{k=j+1}^{N} \left(1-\tilde{b}_k^*\right) - \frac{1}{2}\sum_{j=1}^{N} c_j \sum_{l=j+1}^{N} \tilde{b}_l^* \prod_{k=l+1}^{N} \left(1-\tilde{b}_k^*\right) \\ &= \sum_{j=1}^{N} \left(\left(r_j - \frac{1}{2}\tilde{b}_j^* c_j\right)\tilde{b}_j^* \prod_{k=j+1}^{N} \left(1-\tilde{b}_k^*\right) - \frac{1}{2} c_j \sum_{l=j+1}^{N} \tilde{b}_l^* \prod_{k=l+1}^{N} \left(1-\tilde{b}_k^*\right)\right) \\ &= \sum_{j=1}^{N} \left(r_j - \frac{1}{2}\tilde{b}_j^* c_j - \frac{1}{2}\sum_{l=1}^{j-1} c_l\right)\tilde{b}_j^* \prod_{k=j+1}^{N} \left(1-\tilde{b}_k^*\right), \end{aligned}$$

which confirms Eq. (2.17). Note that the internal payments from the merging driver to the drivers of the platoon cancel out in the expression above, and thus, as expected, do not affect the Social Welfare under the H2T mechanism. ■

2.3.5 Properties and Insights In The General Case

In Sections 2.3.3 and 2.3.4 we show that the exact equilibrium for T2H and H2T can be found under simplifying assumptions. We now relax the assumption about single-indexed cost and consider a more general case.

Numerical solutions of T2H and H2T

The exact analytical solution of DP recursions for Tail-to-Head is not available for general costs c_j^i. We, therefore, as a simple approach, have to resort to an approximate DP—that is, allow the blocked-lane driver to bid only at a discrete step, not continuously. However, the state space dimension grows by one each stage, as there is one additional historic bid each time. This makes use of the

naive approach (exhaustive search, finding an optimal control for every state at every stage) even for an approximate DP be restricted. After dividing each bid into 10 levels, one instance of the problem can be solved reasonably fast for $n = 8$, and becomes too computationally expensive after that. Of course, if higher precision is required, the problem becomes unsolvable even for smaller n.

On the contrary, the Head-to-Tail model can be solved numerically much more efficiently than Tail-to-Head. An approximate DP can solve the problem with high precision: at every stage, the only state is just the index of the stage, and the problem of finding the optimal bid collapses to a single-variable constrained optimization problem.

Solutions to both models share some similarities. The optimal bidding policy is non-monotonic, and bids can increase and decrease as stages proceed, depending on the distances between drivers. Very often, the blocked-lane driver bids more aggressively on the stages where there is a higher inter-car distance since she incurs fewer payments if she merges there. Interestingly, the solution of Tail-to-Head DP in some cases leads to a rather specific policy. If there is a spot with a high enough distance and close enough to the blockage point, it becomes profitable for the blocked lane-driver to bid the maximal value $b_i = 1$ to that driver and all drivers behind him. That way, she can secure a profitable position, while many of these high bids will not be fully paid or not paid at all (because the merging penalty decays with the distance). The second notable feature is that the optimal bidding strategy might be discontinuous in the blocked-lane driver valuation v_b. The blocked-lane driver can switch to aggressive bidding $b_i = 1$ from much smaller bids even if her valuation rises by a tiny value. In the case of Head-to-Tail model, the driver can continue bargaining without the agreement of all drivers behind. Therefore, this specific pattern of bidding 1 does not arise.

To solve the problem numerically, we need first to fix a distribution for valuations $v_b, v_1, \ldots, v_n$, and also the functional forms for r_i and C_{ij}. For the valuations, we use uniform distribution $U(0, 1)$, same as we use for our analytical results earlier. For functions r_i and C_{ij}, we use the following functional forms:

$$r_i(v_b) = \frac{v_b \sum_{k=1}^{i} d_k}{v^F}; \qquad C_{ij}(d_j, \ldots, d_i) = \max\left\{ \frac{D - \alpha \sum_{k=j}^{i} d_k}{v^F}, 0 \right\}, \tag{2.19}$$

where D is measured in meters, and α is a coefficient. These parametric forms allow the following interpretation. D/v^F controls the maximal possible value of the merging penalty that any driver can suffer (higher the value, higher the maximal penalty). α/v^F is responsible for the effect of the distance to merging point to the merging penalty: the higher its value, the faster the penalty decays with distance. The blocked-lane driver's gain $r_i(v_b)$ is straightforward: it is additional travel time she saves from advancing multiplied by her time valuation.

To illustrate the properties of numerical solutions, let us consider two examples. In notation, we superscript "T2H" for the Tail-to-Head model and "H2T" for the Head-to-Tail model.

1. $n = 4, D = 40,\ \alpha = 1,\ v^F = 10, v_b = 0.3$. Optimal expected blocked-driver utilities are $J_0^{T2H} = 0.114,\ J_0^{H2T} = 0.597$. Distances between cars and the optimal bidding policy:

Stage	1	2	3	4
d_i	20	10	15	10
b_i^{T2H}	0.38	0.22	0.3	0.18
b_i^{H2T}	0.3	0.2	0.4	0.32

2. $n = 4, D = 40,\ \alpha = 1,\ v^F = 10, v_b = 0.8$. Optimal expected blocked-driver utilities are $J_0^{T2H} = 2.261,\ J_0^{H2T} = 2.774$. Distances between cars and the optimal bidding policy:

Stage	1	2	3	4
d_i	20	10	15	10
b_i^{t2h}	1.	1.	1.	0.58
b_i^{h2t}	0.8	0.44	0.86	0.56

To emphasize that the block-lane driver's valuation also has a significant effect on the outcome, we vary only v_b in the examples above. Note how the optimal policy jumps from "conservative" bidding to bidding $b_i = 1$ in the second example. That way, the blocked-lane driver secures a valuable position $i = 3$. In other cases, both in Tail-to-Head and Head-to-Tail, we can see that bids are higher for larger gaps. As the valuation of the blocked-lane driver increases, bids and expected utility also increases. Finally, we can see that the H2T format delivers a higher utility to the blocked-lane driver. This is expected, as the blocked-lane driver can easily reach valuable positions and does not need to pay compensations to all drivers involved.

Binary DP and other heuristics for Tail-to-Head

As we note above, Tail-to-Head DP is slow to solve. Can we find a faster, but slightly less precise way to obtain bids for the blocked-lane driver in this format? We develop several different approaches.

The first idea is related to "securing" behavior we observed in the examples above. It seems that often the blocked-lane driver seeks for the most beneficial position (which is usually the position with relatively large distance and close to the blockage), and uses the maximal bid $b_i = 1.$ for all positions up to the target one. It might mean that often the blocked-lane driver does not need much flexibility in bidding to achieve reasonable performance. We, therefore, propose a restricted

form of the Tail-to-Head format. This is the same DP we consider in Eq. (2.3), but possible options for b_i are restricted to be 0 and 1 only. We call it *binary DP*. This approach is faster than simple approximate DP since we only need to check two options at every stage. However, it still has exponential computational complexity $O(2^n)$. The following simple result shows that the optimal bids of binary DP can be computed using an algorithm with linear complexity $O(n)$.

Proposition 2.5. *To find an optimal solution of binary DP, it is enough to consider* n *bidding policies of a threshold form* $1, \dots, 1, 0, \dots, 0$.

Proof. Consider a DP recursion for an arbitrary stage i:

$$J_i(b_1, \dots, b_{i-1}) = \max_{b_i \in \{0,1\}} \Bigg\{ J_{i+1}(b_1, \dots, b_i) F(b_i) + \left(r_{i-1} - \sum_{k=1}^{i-1} b_k c_k^{i-1} \right) (1 - F(b_i)) \Bigg\}.$$

Note that whenever $b_i = 0$, we have:

$$J_i(b_1, \dots, b_{i-1}) = r_{i-1} - \sum_{k=1}^{i-1} b_k c_k^{i-1},$$

and in this case any value of $b_j, j > i$ does not change the resulting cost J_0 of the policy. Since they all are optimal, in particular the choice $b_j = 0, j > i$ is optimal. As the result, the only characteristic that makes the policies different is index of stage i for which $b_i = 0$ for the first time. This is equivalent to what we claimed. ■

Essentially, the binary DP is a stopping problem where the only decision to make is the stage (position) at which the driver wants to stop (merge). There are only n distinct policies, and therefore, computationally, it is very efficient. We further discuss the efficiency of binary DP and other heuristics concerning total and blocked-lane driver's utility in Section 2.4. For now, we emphasize the fact that securing the positions with the largest gap is usually beneficial both for the blocked-lane driver's utility (which is being optimized through DP) and for the total utility (which traffic regulator is mostly interested in). As a result, the binary DP that capitalizes on this behavior can obtain high total utility.

Rather than considering a restricted version of Tail-to-Head DP, we can construct a *myopic bidding policy*. The bid in the original Tail-to-Head DP is trying to balance the cost with future benefits. What we propose is: (i) balance the bid not with optimal future benefit, but with immediate utility improvement (considering

only one future stage); (ii) ignore the fact that current payment is also changing over time due to merging penalty decay. Therefore, we attempt to balance immediate benefits and immediate costs, obtaining the following expression for the stage i:

$$b_i c_i^i = r_i - r_{i-1},$$

resulting in the myopic bidding policy $b_i^m = (r_i - r_{i-1})/c_i^i,\ \forall i$.

This myopic bidding on its own is not producing desirable results since it often overlooks potentially enormous future benefits and underbids in such situations. It might also overbid in many other situations. If the blocked-lane driver perfectly balances benefit and the bid, it leaves her with zero net profit. However, combining the myopic bidding and binary DP should be flexible enough. This approach allows the blocked-lane driver to bid strategically and reach a highly valuable position through binary bidding, and if there is no such single outstanding position, it can resort to myopic bidding. We call this combination *binary-myopic DP*, and it is another restricted version of Tail-to-Head DP, where the blocked-lane driver at every stage can choose her bid out of a three-element set $\{0, b_i^m, 1\}$.

As we show in Section 2.4, this is a powerful approach in terms of social welfare but works reasonably as well for the utility of the blocked-lane driver. However, similar to other DP formulations, computational complexity grows exponentially in n. One possible way to significantly reduce complexity is not to consider all three options each period, but rather to switch between two distinct policies. One policy is the optimal binary DP (which we know can be solved efficiently), and another one is "all-periods-myopic" bidding (which also boils down to calculating n myopic bids). The blocked-lane driver then compares both policies and choose the one that results in a higher expected utility for her. We call the resulting policy *binary-myopic heuristic*.

Below we illustrate example solutions of all three simplified DP and heuristics for the same examples we use above. We use the following abbreviations in notation: BDP - binary DP, BMDP - binary-myopic DP, BMH - binary-myopic heuristic.

1. $n = 4, D = 40,\ \alpha = 1,\ v^F = 10, v_b = 0.3$ Optimal expected blocked-driver utilities are: $J_0^{BDP} = 0$, $J_0^{BMDP} = 0.062$, $J_0^{BMH} = 0.062$. Distances between cars and the optimal bidding policy:

Stage	1	2	3	4
d_i	20	10	15	10
b_i^{BDP}	0.	0.	0.	0.
b_i^{BMDP}	0.6	0.15	0.3	0.15
b_i^{BMH}	0.6	0.15	0.3	0.15

2. $n = 4, D = 40,\ \alpha = 1,\ v^F = 10, v_b = 0.8$ Optimal expected blocked-driver utilities are: $J_0^{BDP} = 1.9$, $J_0^{BMDP} = 2.2$, $J_0^{BMH} = 1.9$. Distances between cars and the optimal bidding policy:

Stage	1	2	3	4
d_i	20	10	15	10
b_i^{BDP}	1.	1.	1.	1.
b_i^{BMDP}	1.	1.	1.	0.4
b_i^{BMH}	1.	1.	1.	1.

From these examples, we can see the expected relation of the utilities: $J_0^{BDP} \leq J_0^{BMH} \leq J_0^{BMDP}$. Furthermore, we can see that sometimes binary-myopic heuristic performs the same as binary DP, and sometimes as myopic-binary DP. For other problems, it can also perform somewhere in between. Comparing myopic bids of BMDP with approximate Tail-to-Head DP from examples earlier, we can observe that sometimes myopic bid exactly equals what recommended by approximate DP, sometimes lower and sometimes higher. Importantly, it follows the correct direction: the bid is higher whenever the distance is greater.

2.4 Individual Utility and Social Welfare

2.4.1 Tail-To-Head and Social Optimum

The traffic regulator and, ultimately, society as a whole, are interested in implementing the bargaining scheme that provides a higher total utility. Tail-to-Head bargaining scheme was explicitly constructed with this in mind. It guarantees that all drivers have non-negative utility in any situation; the same can be said about our simplified solution concepts of Tail-to-Head DP (binary and binary-myopic DP) and the heuristic. In this subsection, we compare the outcome of Tail-to-Head and Head-to-Tail formats to the social optimum.

There could be different ways of defining the social optimum. In particular, different assumptions about the information available to the central planner. As the highest possible standard, one can assume that the central planner perfectly knows the valuations of all drivers. We call this concept *a perfect information social optimum*. For this concept, we can calculate total utility for merging into position i as:

$$U_i = r_i(v_b) - \sum_{j=1}^{i} c_j^i v_j.$$

The perfect information social optimum then can be found as $U^S = \max_i U_i$. We emphasize that this is the best possible outcome that any mechanism can ever

achieve. It requires knowing the exact utilities of all drivers and works directly with them. It is, therefore, can be considered as "not fair" comparison to our proposed mechanisms. Later in the section, we discuss how we can relax this concept to be more feasible.

For the Tail-to-Head (either DP of heuristic) and Head-to-Tail it is also straightforward to find the outcome and total utility. We take the optimal bids of blocked-lane driver and run them using corresponding auction scheme and values $v_1, \ldots, v_n$. Resulting merging agreement allows to calculate the total utility.

Can we prove the existence of universal bounds regarding the relation of total utilities in the perfect information social optimum, Tail-to-Head DP and Head-to-Tail DP? Can we show that Tail-to-Head is universally (meaning for any instance of the problem) better than Head-to-Tail and is close to the social optimum? Unfortunately, as we shortly demonstrate, there are examples for which Tail-to-Head DP performs arbitrarily bad compared to the social optimum and Head-to-Tail. Vice versa, we can easily find examples when Head-to-Tail works arbitrarily poorly compared to the Social Optimum and Tail-to-Head.

Consider the values of parameters that represent a dense, evenly spaced platoon on the free-lane. Assume that advancing one position gives 1 unit of time for the blocked-lane driver, but after the merge, all the free-lane drivers behind are pushed back equally, and each looses the same 1 unit of time. In our model's notation, this means that $C_ij = 1$ if $i \geq j$ and zero otherwise. Also, utility from merge for the blocked-lane driver is $r_i = iv_b$. Then, consider a situation when $v_b = 1$, $v_1 = 1$ and $v_i = 0$ for all $i = 2, \ldots, n$. The Social Optimum utility is $U^S = n - 1$. There is only one free-lane driver with a very high valuation, and the rest have zero valuation. Therefore it is beneficial to put the blocked-lane driver in the head of the platoon. However, the blocked-lane driver would never pass through the very first driver with a high valuation, because she always bids strictly less than v_b (and therefore, less than v_1). For example, for $n = 5$, the bids approximately equal to $b^{TH} = [0.8, 0.7, 0.7, 0.6, 0.5]$. Intuitively, the blocked-lane driver needs to bid $b_1 = 1$ to be able to pass through the first driver, but this means she needs to give up all the utility gained due to advancing. Since she is a rational agent, she prefers to have less chance of accepting (in expectation), but strictly positive utility gain. As a result, the blocked-lane driver stays in the reserved position 0, and total utility is 0.

On the contrary, Head-to-Tail performs optimally in this example. Any positive bid is enough for the blocked-lane driver to merge immediately into the socially optimal position n. In particular, for the same $n = 5$, the bids approximately are $b^{HT} = [0.5, 0.9, 1.0, 1.0, 1.0]$, which achieves the social optimum with total utility $n - 1$.

However, let us say now that v_b is a small positive number, for example, $v_b = 0.1$, and the free-lane valuations are $v_i = 1, i = 1, \ldots, n-1$ and $v_n = 0$. Now the

situation is opposite to the previous one, and it is optimal to put the blocked-lane driver at the tail of the platoon. This results in zero social utility (since merging at position 0 is the reserved case). Any other merging would result in negative total utility. We know that Tail-to-Head always guarantees non-negative total utility, and as a result, it reaches the social optimum. However, the Head-to-Tail mechanism would allow the blocked-lane driver to merge into position n. Indeed, the bid in the approximate DP is $b_n^{HT} = 0.2$, which is accepted since $v_n = 0$. This results in a massive negative utility, specifically, $U^{HT} = 0.1n - n + 1$.

The situations we demonstrate occur not only in exceptional cases. To a less degree, similar situations arise with other parameters, such as ones allowing for merging penalty decay. The existence of such examples implies that it is not possible to make sample-path arguments regarding utility bounds. We can only hope for the results that hold in expectation. However, these results are more challenging to establish analytically. Moreover, they are sensitive to the distributional assumptions that we make.

Nevertheless, later in this section, we provide bounds for Tail-To-Head efficiency for simple homogeneous costs. For the general case, we have to rely on the numerical computations to demonstrate a comparison of Tail-to-Head, Head-to-Tail, and social optimum. As we show in Section 2.4.2, Tail-to-Head DP and related heuristics and relaxations work very well in terms of total utility.

Furthermore, as we mentioned above, the perfect information social optimum is not a fair comparison to Tail-to-Head DP, since it is based on a somewhat unrealistic assumption of revealed valuations. As a much closer to our setting option, we propose a *Partial Information Social Optimum*, or PI-SO. We assume that the central planner possesses the same information that the blocked-lane driver in both our auction formats, i.e. knows only v_b. With that information, the central planner is trying to maximize the total utility. We furthermore restrict ourselves to optimization within the Tail-to-Head format. The problem can be viewed as a purely altruistic blocked-lane driver trying to bid in such a way that everyone would be better off at the end of the process. During the procedure, the driver can adjust her beliefs regarding the valuations of free-lane drivers.

Let us assume for a moment that the blocked-lane driver knows all valuations of the free-lane drivers exactly. Note that finding perfect information social optimum can be formalized as a solution to the following DP:

$$J_i(b_1, \dots, b_{i-1}) = \max_{b_i \geq 0} \Bigg\{ J_{i+1}(b_1, \dots, b_i) F(b_i) + \left(r_{i-1} - \sum_{k=1}^{i-1} v_k c_k^{i-1} \right) (1 - F(b_i)) \Bigg\},$$

$$J_{N+1}(b_1, \ldots, b_N) = r_N - \sum_{k=1}^{N} v_k c_k^N.$$

Here we follow the rules of Tail-to-Head format: at each stage i, the driver offers a bid b_i. If it is accepted, then the blocked-lane driver can proceed. Otherwise, he needs to merge to the last accepted position $i-1$. However, the utility resulting from the merging is the utility of the blocked-lane driver minus the total cost of all free-lane drivers. This equation results in the Perfect Information Social Optimum. The optimal total utility is:

$$\max_i \left\{ r_i - \sum_{k=1}^{i} v_k c_k^i \right\}.$$

If the optimal position is i^*, to achieve this position, the driver needs to bid $b_k = 1, k = 1, \ldots, i^*$ and $b_k = 0$ otherwise. It is easy to see that this is the same social optimum U^S we discussed at the beginning of this subsection.

Now again, assume that the blocked-lane driver does not know all the valuations. However, she can use the information gathered in the previous stages. We cannot use exact valuations v_k, and instead, we replace them with conditional expectations. Equation from the above transforms into the following:

$$J_i(b_1, \ldots, b_{i-1}) = \max_{b_i \geq 0} \Bigg\{ J_{i+1}(b_1, \ldots, b_i) F(b_i) + \left(r_{i-1} - \sum_{k=1}^{i-1} \mathbb{E}\left[v_k | \text{k-th driver accepted } b_k \right] c_k^{i-1} \right) (1 - F(b_i)) \Bigg\}.$$

In the above equation, we replaced v_k with its expectation given the information that k-th driver accepted bid b_k earlier. This is the only information relevant to the blocked-lane driver with regard to v_k. Recall that we show earlier that a free-lane driver accepts bid b_k if $v_k < b_k$ and rejects otherwise. This leads us to the final form of DP recursion:

$$J_i(b_1, \ldots, b_{i-1}) = \max_{b_i \geq 0} \Bigg\{ J_{i+1}(b_1, \ldots, b_i) F(b_i) \tag{2.20}$$

$$+ \left(r_{i-1} - \sum_{k=1}^{i-1} \mathbb{E}[v_k | v_k < b_k] c_k^{i-1} \right) (1 - F(b_i)) \Bigg\}, \tag{2.21}$$

For tractability, we derive our analytical results based on three assumptions:

(i) The merging cost satisfy $c_k^i = c_k,\ i = 1, \ldots, N$;

(ii) The probability distribution function satisfies $F(x) = x,\ x \in [0, 1]$. In other words, the time valuations are uniformly distributed in the $[0, 1]$ interval;

(iii) Time valuations are independent across drivers in the platoon.

The corresponding Dynamic Programming problem is described by the recursion:

$$\bar{J}_i\big(\bar{b}_1, \ldots, \bar{b}_{i-1}\big) = \max_{0 \le \bar{b}_i \le 1} \left\{ \bar{J}_{i+1}\big(\bar{b}_1, \ldots, \bar{b}_i\big)\bar{b}_i + \left(r_{i-1} - \frac{1}{2}\sum_{k=1}^{i-1} \bar{b}_k c_k \right) \big(1 - \bar{b}_i\big) \right\}. \tag{2.22}$$

We denote by $\bar{b}_i^*$ the solution to the above optimization problem, i.e., optimal bid that the merging driver makes to driver i on the platoon.

Proposition 2.6. *The optimal expected reward-to-go of the Partial-Information Social Optimum at stage $i = 1, \ldots, N+1$ is equal to*

$$\bar{J}_i\big(\bar{b}_1, \ldots, \bar{b}_{i-1}\big) = r_{i-1} - \frac{1}{2}\sum_{k=1}^{i-1} \bar{b}_k c_k + \sum_{j=i}^{N} \left(r_j - r_{j-1} - \frac{1}{2}\bar{b}_j^* c_j \right) \prod_{k=i}^{j} \bar{b}_k^*, \tag{2.23}$$

and the optimal bids to drivers $i = 1, \ldots, N$ can be computed through the recursion:

$$\bar{b}_i^* = \max \left\{ \min \left\{ \frac{1}{c_i} \left(r_i - r_{i-1} + \sum_{j=i+1}^{N} \left(r_j - r_{j-1} - \frac{1}{2}\bar{b}_j^* c_j \right) \prod_{k=i+1}^{j} \bar{b}_k^* \right), 1 \right\}, 0 \right\}. \tag{2.24}$$

Proof. The proof is very similar to the proof of Proposition 2.1 and, thus, omitted for brevity. ■

Similarly to the T2H mechanism, it is convenient to study the "normalized" optimal expected reward-to-go of the Partial-Information Social Optimum:

$$\Delta\bar{J}_i \equiv \bar{J}_i\big(\bar{b}_1, \ldots, \bar{b}_{i-1}\big) - \left(r_{i-1} - \frac{1}{2}\sum_{k=1}^{i-1} \bar{b}_k c_k \right).$$

Proposition 2.6 implies that

$$\Delta\bar{J}_i = \sum_{j=i}^{N} \left(r_j - r_{j-1} - \frac{1}{2}\bar{b}_j^* c_j \right) \prod_{k=i}^{j} \bar{b}_k^*, \tag{2.25}$$

so that $\Delta\bar{J}_i$ does not depend on $b_1, \ldots, \bar{b}_{i-1}$. Using this notation, we can rewrite the DP recursion in Eq. (2.22) as follows:

$$\Delta\bar{J}_i = \max_{0\le\bar{b}_i\le 1}\left\{\left(\Delta\bar{J}_{i+1} + r_i - r_{i-1} - \frac{1}{2}\bar{b}_i c_i\right)\bar{b}_i\right\}. \tag{2.26}$$

with boundary condition $\Delta\bar{J}_{N+1} = 0$.

Corollary 2.3. *The optimal bids and the (normalized) optimal expected reward-to-go of the Partial-Information Social Optimum can be computed via the simple recursive equations:*

1. *If* $\Delta\bar{J}_{i+1} \le -r_i + r_{i-1} + c_i$, *then* $\bar{b}_i^* = \big(\Delta\bar{J}_{i+1} + r_i - r_{i-1}\big)/c_i$ *and* $\Delta\bar{J}_i = \frac{1}{2}\big(\bar{b}_i^*\big)^2 c_i$;
2. *Otherwise,* $\bar{b}_i^* = 1$ *and* $\Delta\bar{J}_i = \Delta\bar{J}_{i+1} + r_i - r_{i-1} - \frac{1}{2}c_i \iff \bar{J}_i\big(b_1, \ldots, \bar{b}_{i-1}\big) = \bar{J}_{i+1}\big(b_1, \ldots, \bar{b}_{i-1}, 1\big)$.

Proof. The proof is very similar to the proof of Corollary 2.1 and, thus, omitted for brevity. ■

We formulate Partial Information Social Optimum DP on the intuitive basis, but below we show an argument demonstrating that the resulting bids are the best bids for the society that can be achieved in the Tail-to-Head format.

Proposition 2.7. *The bids of Partial-Information Social Optimum DP maximize Social Welfare of T2H mechanism,* SW_{T2H}.

Proof. We do not have a complete proof of the result, but rather demonstrate some hints on why it holds in the case $0 < \bar{b}_i^* < 1,\ \forall i$.

We want to maximize the following function of N variables $b_1, \ldots, b_N$:

$$f(b_1, \ldots, b_N) \equiv \sum_{j=1}^{N}\left(r_j - r_{j-1} - \frac{1}{2}b_j c_j\right) b_j \prod_{k=1}^{j-1} b_k.$$

Let us fix i and take the partial derivative with respect to b_i:

$$f'_{b_i} = (r_i - r_{i-1} - b_i c_i)\prod_{k=1}^{i-1} b_k + \sum_{j=i+1}^{N}\left(r_j - r_{j-1} - \frac{1}{2}b_j c_j\right)\prod_{k=1}^{i-1} b_k \prod_{p=i+1}^{j} b_p. \tag{2.27}$$

To identify critical points, we need to find points at which $f'_{b_i} = 0$ for any i. Note that in Eq. (2.27) we have $\prod_{k=1}^{i-1} b_k > 0$, and hence we can write

$$r_i - r_{i-1} - b_i c_i + \sum_{j=i+1}^{N}\left(r_j - r_{j-1} - \frac{1}{2}b_j c_j\right)\prod_{k=i+1}^{j} b_k = 0,$$

or, expressing b_i,

$$b_i = \frac{1}{c_i}\left(r_i - r_{i-1} + \sum_{j=i+1}^{N}\left(r_j - r_{j-1} - \frac{1}{2}b_j c_j\right)\prod_{k=i+1}^{j} b_k\right).$$

Note that this expression coincides with $\bar{b}_i^*$ from Eq. (2.24) under our assumption that those bids are always in the interior between 0 and 1. It also should be note that this set of bids has the unique recurrent solution.

■

Some structural properties of the bids under Partial-Information Social Optimum and the T2H mechanism can be shown.

Proposition 2.8. *Assume that $r_{j+1} - r_j = \gamma > 0$ and $c_j = c > 0$, for all $j \in \{1, \ldots, N-1\}$. Then, the following statements are true:*

1. *Bids for T2H and Partial-Information Social Optimum can be found by the following recurrent formulas:*

$$b_i^* = \min\left\{\frac{\gamma}{2c} + \frac{(b_{i+1}^*)^2}{2}, 1\right\}, \qquad \bar{b}_i^* = \min\left\{\frac{\gamma}{c} + \frac{(\bar{b}_{i+1}^*)^2}{2}, 1\right\}.$$

2. *If $\gamma > c/2$, then $b_i^* < 1$ and $\bar{b}_i^* < 1$, $\forall i$.*

Proof. See the proof of Proposition 2.9 ■

Proposition 2.9. *Assume that $r_{j+1} - r_j = \gamma > 0$ and $c_j = c > 0$, for all $j \in \{1, \ldots, N-1\}$. Then, the following statements are true:*

1. *$b_j^* \geq b_{j+1}^*$, for all $j \in \{1, \ldots, N-1\}$, i.e., the bids of the T2H mechanism are monotonically non-increasing;*

2. *$\bar{b}_j^* \geq \bar{b}_{j+1}^*$, for all $j \in \{1, \ldots, N-1\}$, i.e., the bids of Partial-Information Social Optimum are monotonically non-increasing;*

3. *$\bar{b}_j^* \geq b_j^*$, for all $j \in \{1, \ldots, N\}$, i.e., the bids of Partial-Information Social Optimum are greater than or equal to the bids of the T2H mechanism.*

Proof. See Appendix 2.7 ■

As we demonstrate in the following theorem, in some special cases, the difference between total utility of Partial Information Social Optimum and Tail-to-Head mechanism with a rational (self-interested) driver can be bounded from above.

Theorem 2.1. *Let $r_{j+1} - r_j = \gamma > 0$ and $c_j = c > 0$, for all $j \in \{1, \dots, N-1\}$. If $\gamma < c/2$, then the expected Social Welfare under the Partial-Information Social Optimum and the T2H mechanism are related as follows:*

$$SO_{PI} - SW_{T2H} \leq \gamma \left(\frac{\gamma + c/2}{c/2 - \gamma} - \frac{\gamma - \gamma(\gamma/(2c))^N}{2c - \gamma} \right).$$

Proof. Recall that SO_{PI} has the same functional form as SW_{T2H}, but with bids of Partial Information Social Optimum $\bar{b}_i^*$. Then we can write

$$\begin{aligned} SO_{PI} - SW_{T2H} &= \sum_{j=1}^{N} \left(\gamma - \frac{1}{2} \bar{b}_j^* c \right) \bar{b}_j^* \prod_{k=1}^{j-1} \bar{b}_k^* - \sum_{j=1}^{N} \left(\gamma - \frac{1}{2} b_j^* c \right) b_j^* \prod_{k=1}^{j-1} b_k^* \\ &= \gamma \sum_{j=1}^{N} \left(\prod_{k=1}^{j} \bar{b}_k^* - \prod_{k=1}^{j} b_k^* \right) + \frac{c}{2} \left((b_j^*)^2 \prod_{k=1}^{j-1} b_k^* - (\bar{b}_j^*)^2 \prod_{k=1}^{j-1} \bar{b}_k^* \right). \end{aligned}$$

Note that according to Proposition 2.8, the second component is negative, and therefore we can write, using Proposition 2.8 and Proposition 2.9

$$\begin{aligned} SO_{PI} - SW_{T2H} &\leq \gamma \sum_{j=1}^{N} \left(\prod_{k=1}^{j} \bar{b}_k^* - \prod_{k=1}^{j} b_k^* \right) \\ &\leq \gamma \sum_{j=1}^{N} \left((\bar{b}_1^*)^j - (b_N^*)^j \right) \\ &= \gamma \left(\frac{\bar{b}_1^* (1 - (\bar{b}_1^*)^N)}{1 - \bar{b}_1^*} - \frac{b_N^* (1 - (b_N^*)^N)}{1 - b_N^*} \right) \\ &\leq \gamma \left(\frac{\bar{b}_1^*}{1 - \bar{b}_1^*} - \frac{\gamma}{2c} \cdot \frac{(1 - (\gamma/(2c))^N)}{1 - \gamma/(2c)} \right) \\ &= \gamma \left(\frac{\gamma/c + (\bar{b}_2^*)^2/2}{1 - \gamma/c - (\bar{b}_2^*)^2/2} - \frac{\gamma - \gamma(\gamma/(2c))^N}{2c - \gamma} \right) \\ &\leq \gamma \left(\frac{\gamma + c/2}{c/2 - \gamma} - \frac{\gamma - \gamma(\gamma/(2c))^N}{2c - \gamma} \right), \end{aligned}$$

which completes the proof. ■

This bound tends to be small when γ is small comparative to c, showing that T2H mechanism is very efficient in terms of social welfare in those cases.

2.4.2 Social Welfare Comparison in The General Case

In the previous section, we demonstrate that T2H mechanism can be efficient in case of homogeneous costs for some parameters. Can we say the same in case of general costs c_j^i, where costs can differ from one driver to another and fall off with the distance? We would also like to compare T2H, H2T, and heuristics. To make this comparison, we use a Monte-Carlo simulation set-up. We generate all relevant values randomly for one instant of the problem, then calculate the bidding policy, optimal for the blocked-lane driver. Basing on these bids, we calculate the resulting total utility. After averaging over many instances, we obtain an estimation of the gap between the bargaining schemes for the given parameters.

For generating valuations $v_b, v_1, \ldots, v_n$ we use uniform distribution $U(0,1)$. For distances, we use exponential distribution $E(\lambda)$ with mean distance λ. It is a well-known fact that distances on the road tend to be distributed exponentially (see, for example, Miller (1961b)). For functions C_{ij} and r_i we take the forms discussed in Section 2.3.5.

First, we demonstrate several arbitrary examples with different parameters. Each example in the Table 2.1 is calculated by averaging over 2000 instances. The bid range is divided into 10 levels. Each column is the total utility for the corresponding algorithm.

Table 2.1: Examples of bidding problem under different parameters and expected total utility under different solution concepts.

n	D	α	λ	Perf. Soc	Part. Soc	TH	HT	BDP	BMDP	BMH
5	20	1	10	2.024	2.010	1.967	1.947	1.938	1.963	1.949
4	40	1	10	0.956	0.918	0.827	0.658	0.762	0.834	0.827
4	40	0.5	5	0.041	0.033	0.020	-0.651	0.005	0.029	0.029
2	60	0.1	30	0.916	0.888	0.669	0.692	0.448	0.783	0.783

From the examples, we can see that sometimes Tail-to-Head and Head-to-Tail DPs are very close to the social optimum and each other (the first example). The second and third examples demonstrate situations when T2H is beneficial and closer to the optimum. Particularly interesting the third case, where we can see that T2H scheme manages to keep total utility positive and close to the social optimum, while H2T results in a negative number. Finally, the last example shows that there are situations when H2T is better than T2H. We also demonstrate that Perfect Information Social Optimum and Partial Information Social Optimum are persistently close.

Regarding heuristics, generally, all three of them are relatively close to Tail-to-Head DP. Very often, heuristics outperforms it. This is possible because all DPs

and heuristics optimize the utility of the blocked-lane driver, which is typically not the same as the total utility of all drivers. Limiting options of the bidding driver might be beneficial. She would have to choose the bid that might be less beneficial to her but is preferable for the society. If we add additional flexibility, the blocked-lane driver might drift to a bid that delivers higher utility for her but is worse for society. The myopic bid b_i^m and the binary bidding strategy we use in our heuristics seem to be a very good choice for the total utility. They also decrease the computational complexity of the DP.

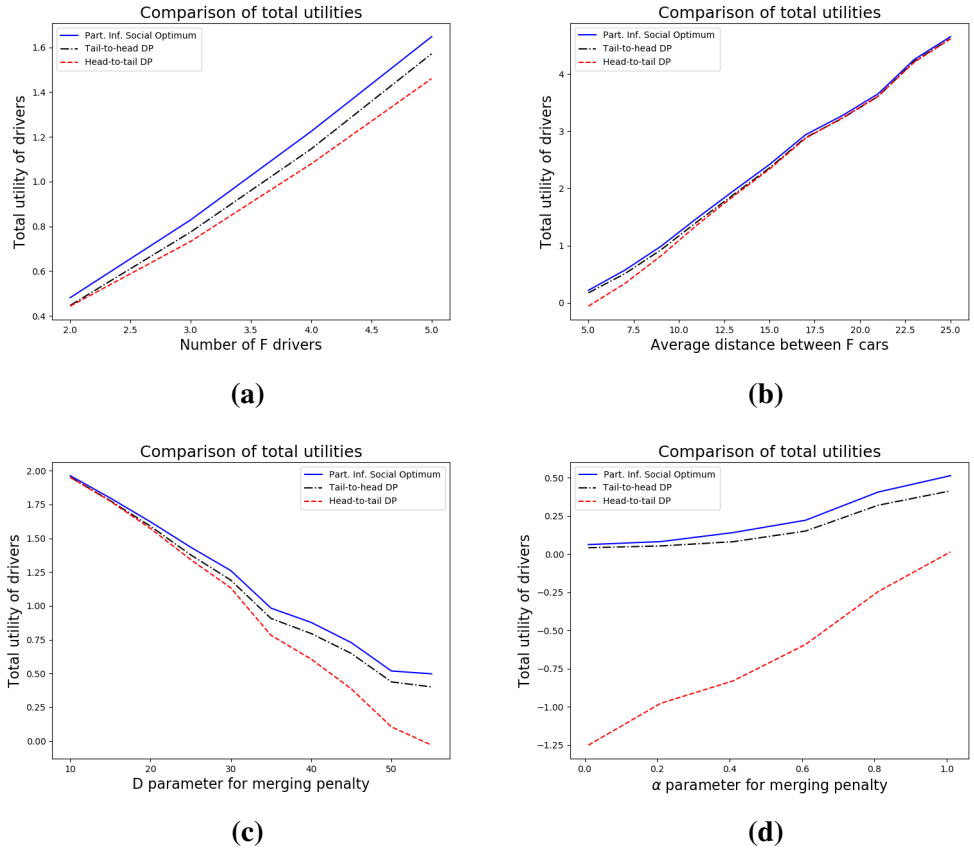


Figure 2.2: Comparison of Partial Information Social Optimum to Tail-To-Head and Head-to-Tail DP. On y-axis is the total utility. On x-axis: *(a)* number of free-lane cars; *(b)* average distance between free-lane cars; *(c)* parameter D of merging penalty function; *(d)* parameter α of merging penalty function.

Specific examples provide some insides, but we also want to see a bigger picture. How do the utilities react to a change in one particular parameter? First, we compare Tail-to-Head and Head-to-Tail. We first pick a base case, for that we use $n = 4, D = 30, \alpha = 1, \lambda = 10, v^F = 10$. This corresponds to a case when there is

relatively dense traffic on the free lane, but not complete congestion; the maximal merging penalty is 3 seconds, and it decays rather fast (3 cars, with an average distance, is enough for the penalty to fall to 0).

We now check how the results are sensitive to the change of one particular parameter, given that the rest of the parameters are fixed to baseline values from the above. In Figure 2.2 we demonstrate 4 different tests to compare Partial Information Social Optimum with Tail-to-Head and Head-to-Tail DPs. First, we test how does performance changes with the change in the number of free-lane drivers. The result implies that Head-to-Tail performs worse when the number of drivers grows, while Tail-to-Head continues to perform relatively well. Note also that in all cases, utility grows in the number of drivers. We can interpret it as an increased potential for beneficial trades between drivers.

In the second test, we evaluate performance under different values of average distance λ on the free lane. We vary λ from 5 (very dense traffic) to 25 (light traffic; very large gaps can appear with high probability). The results imply that as the average distance grows, H2T becomes very close to T2H. At the same time, Tail-to-Head stays close to the social optimum for any distance (note that the difference between T2H and H2T is especially large for low λ, corresponding to high congestion). Note that here again in all cases utility grows—larger average distances make merging penalty less and increase possibilities for beneficial trades.

In the last two test we vary parameters of the merging penalty function. First we change D, from 10 (small maximal penalty; corresponds to 1 second of travel time) to 60 (large maximal penalty, corresponds to 6 seconds). The results shows that for low D all utilities are very close, but for larger values Head-to-Tail performs significantly worse, while Tail-to-Head stays quite close to the social optimum. Similar result can be observed on the last graph, where we vary α, the parameter that controls the rate of merging penalty decay (with distance). Tail-to-Head maintains good performance for all α, while Head-to-Tail performs very poor for low α, resulting in large negative total utility.

We compare the relaxed DP, Partial Information Social Optimum, and heuristics in Figure 2.3. Since heuristics usually are very close to Tail-to-Head DP, for better readability, we rescale the values by the utility of Tail-to-Head DP and report all utilities in percent. In the figure, we can see that both binary-myopic DP and binary-myopic heuristic lose no more than 1-2% compared to Tail-to-Head DP. The binary-myopic DP is almost never worse than Tail-to-Head. In some extreme cases, both heuristics are capable of reaching 140% of Tail-to-Head DP efficiency. Notably, the performance of binary-myopic heuristics often becomes better when the gap between social optimum and Tail-to-Head DP grows, while binary DP performs worse in such cases.

We can conclude that Tail-to-Head DP and heuristics manage to stay relatively close to Partial Information Social Optimum in terms of total utility. On

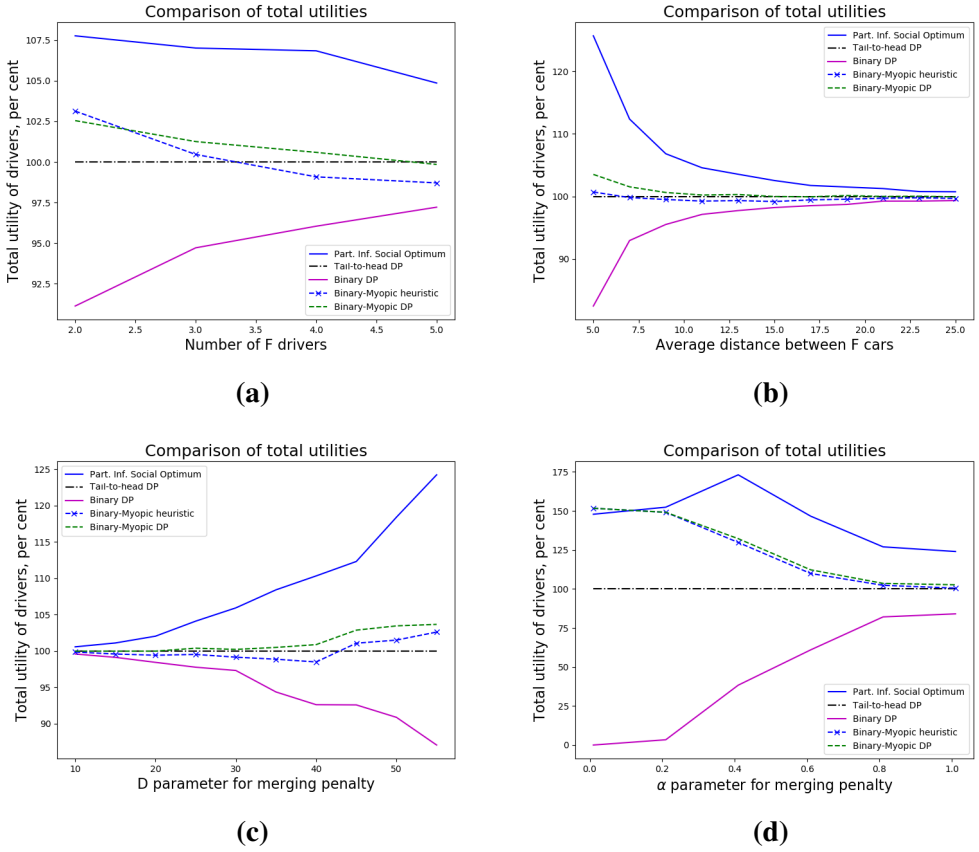


Figure 2.3: Comparison of Partial Information Social Optimum to Tail-To-Head DP, relaxations and heuristics. On y-axis is total utility, rescaled by utility of Tail-to-Head DP, percent. On x-axis: *(a)* number of the free-lane cars; *(b)* average distance between free-lane cars; *(c)* parameter D of merging penalty function; *(d)* parameter α of merging penalty function.

the contrary, Head-to-Tail often provides low or even negative utility, and typically it performs worse than Tail-to-Head. Situations when it performs similarly or slightly better are restricted to low n (no more than 2 drivers), or low merging penalties (which happens when distances are high, or D is high, or α is low). The Tail-to-Head DP relaxations and heuristics perform very well, especially binary-myopic DP and heuristic, and they provide an almost immediate solution. For example, among all examples we considered above, the average performance of binary-myopic heuristic is no worse than 2% compared to the Tail-to-Head DP, and in around 30% cases, it outperforms the latter. Furthermore, the problem with $n = 4$ and 30 levels for bidding can be solved 4 orders of magnitude faster, not speaking of the problems with higher n or greater precision.

To finalize the comparison of total utilities, we present robustness checks. Specifically, we test how robust the result (mostly interesting, performance of Tail-to-Head DP and heuristics) to our assumptions regarding the underlying distribution of values $v_b, v_1, \ldots, v_n$? As we demonstrate, the performance of all Tail-to-Head variations stays reasonable when we assume a different distribution (peaked around certain value). As an example of such distribution, which is general enough, we take normal distribution $N(\mu, \sigma)$, truncated on $[0, 1]$.

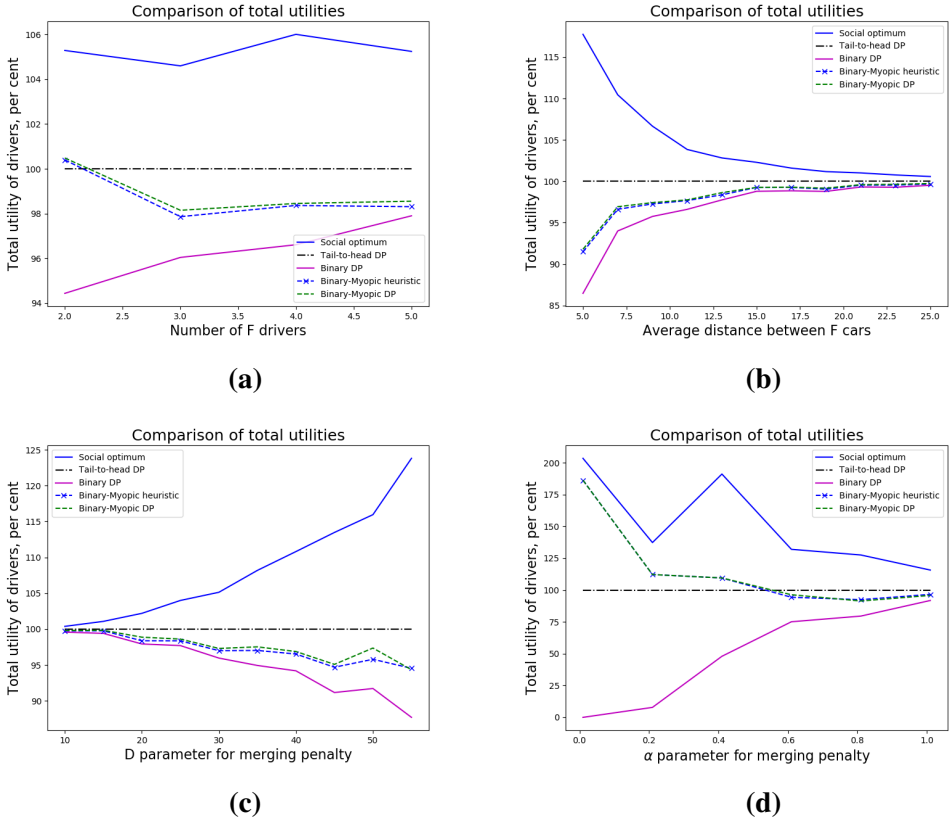


Figure 2.4: Comparison of Perfect Information Social Optimum to Tail-to-Head DP and heuristic for the normal distribution $N(0.5, 0.15)$, truncated on $[0, 1]$. On y-axis is total utility in per cent. On x-axis: *(a)* number of free-lane cars; *(b)* average distance between free-lane cars; *(c)* parameter D of merging penalty function; *(d)* parameter α of merging penalty function.

We show an example in Figure 2.4. We take $\mu = 0.5$, which corresponds to the peak of density function located precisely in the middle of the interval $[0, 1]$. We take a standard deviation $\sigma = 0.15$, so that the density would fall off significantly at the ends of the interval. In this case, density at the peak is around 2.5, and close to 0 at the ends. As a benchmark, we use Perfect Information Social

Optimum since it is much more challenging to derive DP recursion for Partial Information Social Optimum under a truncated normal distribution (computation of conditional expectations is intricate for truncated normal).

As we can see from the results, Tail-to-Head DP and binary DP perform mainly in the same manner as in the case of uniform distribution, in Fig 2.2. At the same time, the gaps between these mechanisms and the social optimum are significantly less. Intuitively, this is easy to understand: as all valuations become closer on average, there is less potential total utility loss, and the cost of a "wrong" decision becomes less profound. Binary-myopic DP and heuristic performs worse than under normal distribution. However, in almost all cases, they stay within 10% from Tail-to-Head DP and are significantly better than binary DP. Therefore, it is still preferable to use binary-myopic heuristic as a fast way of solving Tail-to-Head DP.

2.4.3 Blocked-Lane Driver Utility Comparison

All solution concepts we discussed give the expected utility for the blocked lane driver since it results from selfish optimization by the blocked-lane driver. Even though the traffic regulator is likely to be more interested in total utility, comparing the blocked-lane utility can give useful hints regarding our mechanisms' performance. It can also be useful to observe it to see how it is different from the total utility. Finally, there might be situations when the regulator is interested in maximizing the blocked-lane driver utility instead of the total one.

For specific values of free-lane valuations $v_1, \dots, v_n$, we can obtain actual utility that the driver can obtain using her policy taken from any discussed mechanisms.

Intuitively, we expect Tail-to-Head to perform worse than the Head-to-Tail model since the blocked-lane driver needs to compensate all drivers who suffered the merging penalty in the former and pay to a single driver only in the latter case. The effect might be enormous if the merging penalty is high, and there is no fall-off of the penalty with distance from the merging point. The second factor is that in the Head-to-Tail model, the driver starts bargaining from more beneficial positions and has a chance to negotiate with all drivers, which increases the chance of beneficial trade.

We want to check if this intuition is confirmed in numerical solutions, and how large is this gap, for different parameters. As a secondary goal, we can observe the performance of our DP relaxations and heuristics. We compare these utilities using the same Monte-Carlo simulation set-up as above. We use the same normal distribution $U(0, 1)$, and the same baseline case: $n = 4, D = 30, \alpha = 1, \lambda = 10, v^F = 10$. In Figure 2.5 we demonstrate the results. They imply that the gap increases with the number of free-lane drivers n; decreases in average distance λ;

increases in D; finally, it decreases in α.

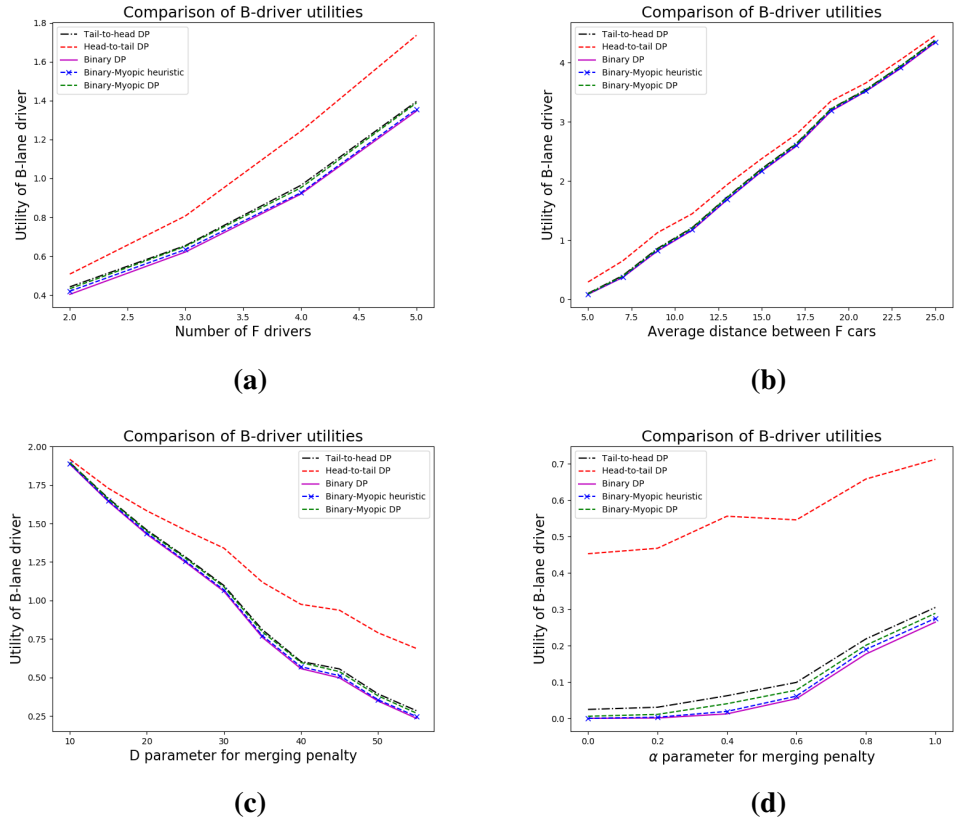


Figure 2.5: Comparison of blocked-lane driver utility in Tail-to-Head (both DP and heuristic) and Head-to-Tail barganing formats. On y-axis is utility of the blocked-lane driver. On x-axis: *(a)* number of free-lane cars; *(b)* average distance between free-lane cars; *(c)* parameter D of merging penalty function; *(d)* parameter α of merging penalty function.

Generally speaking, the gap becomes smaller when the merging penalty becomes smaller in expectation (larger distances, less maximal penalty, or faster decay of the penalty). This is not surprising: the fact that the blocked-lane driver has to compensate for the penalties in Tail-to-Head is the reason for the gap to exist in the first place. However, we should note that the gap is not dramatically large in many cases, except, probably, for a low α.

Regarding relaxations and heuristics, we can see that they all are very close to Tail-to-Head DP. The heuristic can never achieve a higher blocked-lane driver utility than approximate DP (unlike total utility). However, performance is still good. Even the binary DP is very close to the Tail-to-Head DP. Binary-myopic heuristic performs marginally better, and binary-myopic DP is usually impressively close

to Tail-to-Head DP with almost no difference.

2.5 Budget Constraints

The blocked-lane driver might have a maximal budget she is willing to spend on position bidding. Both Tail-to-Head and Head-to-Tail models can be adjusted to incorporate such constraints. Assume that Q is a maximal amount of money the blocked-lane driver is ready to spend. The case of Head-to-Tail is particularly trivial. There is only one bid to pay at any time, and this bid is to be paid immediately. As a result, the blocked-lane driver simply has to bid $0 \leq b_i c_i^i \leq Q$ at every stage i.

For the Tail-to-Head model, the situation is different since the blocked-lane driver needs to pay accordingly to the previous bids and the position she is merging into. In particular, at stage i the driver needs to bid in such a way that she would be able to pay all the payments completely if she would be rejected at stage $i+1$. The only exception is if the probability of rejection equals zero, which happens if the driver bids $b_i = 1$ (the maximal possible valuation). If the driver bids anything less than 1, then the constraints are as follows:

$$0 \leq b_i c_i^i \leq Q - \sum_{k=i}^{i} c_k^i b_k.$$

Important to note that because the merging penalties are changing dynamically with the stage, the amount of money that the blocked-lane driver can bid is not necessarily decreasing. It can happen that at stage $i+1$, the driver has more available funds that she has at stage i. As a result, the problem of budget bidding does not reduce to only allocating the budget on hands among different positions. Since, according to our assumptions c_j^i is non-increasing in j, the blocked-lane driver can at any stage satisfy budget constraints by bidding 0. A situation when she is unable to pay previously agreed bids can never happen. This means that the budget constraints are in a "hard" sense here. One could also think of "soft" budget constraints when the blocked-lane driver can pay above the budget but must respect them in expectation.

How do the blocked-lane driver utility and total utility change when there is a limited budget on the block-lane driver's side? One could expect, that Tail-to-Head is more vulnerable to this since the blocked-lane driver needs to keep bids low enough, in order to be able to pay them off to all drivers involved at the end of the process.

Figure 2.6 partially confirm this intuition. We compare how does the performance of both formats (we also include binary-myopic DP) react to budget limi-

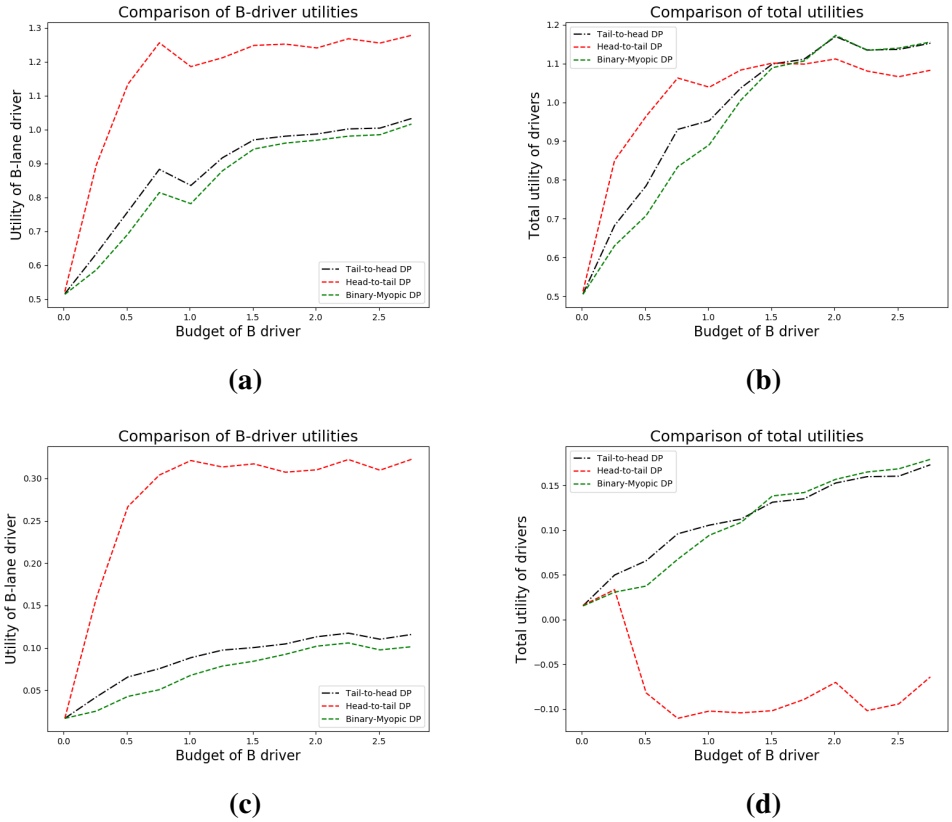


Figure 2.6: Effect of the limited budget on Tail-to-Head DP, Head-to-Tail DP and binary-myopic DP. On the X-axis: *(a, c)* utility of the blocked-lane driver; *(b, d)* total utility. We test for different values of average distance between the cars: *(a, b)*: average distance 10; *(c, d)*: average distance 5.

tations (for the rest of the parameters, we used the baseline from the previous section). When the budget is close to 0, all three policies, naturally, have close to zero utilities. Then values for Head-to-Tail grow faster than for Tail-to-Head, before stabilizing. As a result, there is a range of budgets, for which Head-to-Tail outperforms Tail-to-Head both in terms of total utility and blocked-lane driver utility. Nevertheless, Tail-to-Head performance recovers relatively fast. The performance of binary-myopic DP is typically worse than both DPs and is more vulnerable to budget constraints. This is easy to understand since the binary DP and heuristics have much less flexibility in bidding, and flexibility is important when the budget is limited. When there is not enough budget to make a binary or myopic bids, it must resort to bidding zero, while an approximate Tail-to-Head DP can choose a lower, but positive, bid.

One way to understand the numerical properties of mechanisms in Figure 2.6

is to figure out how much budget the blocked-lane driver needs to bid freely. For Head-to-Tail, we know that budget of 1 is sufficient to pay under any circumstances. Indeed, we can see that utility for Head-to-Tail does not grow after the budget reaches 1. As we can see, even 0.5 is good enough in expectation since the expected utility of the blocked-lane driver is equal to 0.5. Moreover, she often underbids her valuation.

2.6 Conclusions

Driverless technologies and vehicle-to-vehicle communication are coming. Today is a perfect moment to start developing a better highway traffic paradigm, trying to incorporate that different drivers have different urgency of their trips. Prioritizing the drivers properly might achieve social utility improvement via the means that rarely received traffic engineers' attention before. The safety was always the primary consideration (and rightfully so), and thus it was not possible to think of such complicated interactions as on-the-fly auctions performed by humans in parallel with the driving operations. Autonomous vehicles will make such interactions safe, and in this chapter, we tackle the efficiency part of the deal in one specific scenario of a mandatory lane-change.

Unfortunately, there is no easy out-of-the-box mechanism that can solve this problem. Many standard approaches, such as position auctions and the auctions with externalities, can not be utilized because of the complicated dynamics. Neither the standard mechanisms with established properties (such as VCG) can help. This chapter proposes an ad-hoc mechanism that is carefully crafted to deliver high total utility while leading to predictable bidding strategies. We demonstrate that the gap in expected total utility between the Tail-to-Head mechanism and Partial Information Social Optimum can be bounded for homogeneous costs. To provide insights for the general case, we demonstrate in the numerical study that the mechanism performs very well on average and close to the social optimum.

As a potential direction for future research, we can build a better mechanism based on the same idea of guaranteeing zero negative externalities and prove its superiority theoretically. If the Tail-to-Head DP is the best such mechanism, we would like to prove it instead.

2.7 Appendix

2.7.1 Proof of Proposition 2.9

The main focus is on proving Proposition 2.9, we show Proposition 2.8 as a side result.

We start from proof of part 1 of Proposition 2.9. We split the proof of part 1 into two parts, each is proved by induction. First, we show that the result holds when $\gamma < c$. As we shortly see, in this case all the bids are always in the interior between 0 and 1.

Basis of Induction: For $i = N$ Eq. (2.5) implies that

$$b_N^* = \frac{\gamma}{2c}.$$

Note that $0 < b_N^* < 1$. For $i = N - 1$

$$\begin{aligned} b_{N-1}^* &= \max\left\{\min\left\{\frac{1}{2c}\left(\gamma + \gamma b_N^* - (b_N^*)^2 c\right), 1\right\}, 0\right\} \\ &= \max\left\{\min\left\{\frac{\gamma}{2c} + \frac{\gamma^2}{8c^2}, 1\right\}, 0\right\} \\ &= \frac{\gamma}{2c} + \frac{\gamma^2}{8c^2}. \end{aligned}$$

Clearly, $b_{N-1}^* \geq b_N^*$ and $0 < b_{N-1}^* < 1$ holds.

Induction Step: Assume that $0 < b_{i+1}^* < 1$ and the statement of the lemma holds for b_{i+1}^* and b_{i+2}^*, that is $b_{i+1}^* \geq b_{i+2}^*$. We need to show that $b_i^* \geq b_{i+1}^*$ and $0 < b_i^* < 1$.

According to Eq. (2.5), let us denote x_i^* in such way that $b_i^* = \max\{\min\{x_i^*, 0\}, 1\}$. We can write then

$$\begin{aligned} x_i^* &= \frac{1}{2c}\left(\gamma + \sum_{j=i+1}^{N}(\gamma - b_j^* c)\prod_{k=i+1}^{j} b_k^*\right) \\ &= \frac{1}{2c}\left(\gamma + b_{i+1}^*\sum_{j=i+2}^{N}(\gamma - b_j^* c)\prod_{k=i+2}^{j} b_k^* + (\gamma - b_{i+1}^* c)b_{i+1}^*\right) \\ &= \frac{\gamma}{2c} + b_{i+1}^*\left(b_{i+1}^* - \frac{\gamma}{2c}\right) + \frac{(y - b_{i+1}^* c)b_{i+1}^*}{2c} \\ &= \frac{\gamma}{2c} + \frac{(b_{i+1}^*)^2}{2}. \end{aligned}$$

which holds for any i. Note that we used Eq. (2.5) again for b^*_{i+1} in the third line, using the fact that $0 < b^*_{i+1} < 1$. As $\gamma < c$ and $0 < b^*_{i+1} < 1$, it is clear that $0 < x^*_i < 1$ and

$$b^*_i = x^*_i = \frac{\gamma}{2c} + \frac{(b^*_{i+1})^2}{2}.$$

We need to show now that $b^*_i - b^*_{i+1} \geq 0$. We can write

$$b^*_i - b^*_{i+1} = \frac{\gamma}{2c} + \frac{(b^*_{i+1})^2}{2} - \frac{\gamma}{2c} - \frac{(b^*_{i+2})^2}{2} = \frac{1}{2}\left((b^*_{i+1})^2 - (b^*_{i+2})^2\right) \geq 0.$$

which completes the induction.

We now assume that $\gamma \geq c$.

Basis of Induction: For $i = N$ Eq. (2.5) implies that

$$b^*_N = \min\left\{\frac{\gamma}{2c}, 1\right\}.$$

Note that $b^*_N > 0$. For $i = N - 1$, we separate into 2 cases. First, assume that $b^*_N = \gamma/(2c)$. Then

$$\begin{aligned} b^*_{N-1} &= \max\left\{\min\left\{\frac{1}{2c}\left(\gamma + \gamma b^*_N - (b^*_N)^2 c\right), 1\right\}, 0\right\} \\ &= \max\left\{\min\left\{\frac{\gamma}{2c} + \frac{\gamma^2}{8c^2}, 1\right\}, 0\right\} \\ &= \frac{\gamma}{2c} + \frac{\gamma^2}{8c^2}. \end{aligned}$$

In this case $b^*_{N-1} \geq b^*_N$ and $b^*_{N-1} > 0$ holds. Now assume that $b^*_N = 1$; note in this case $\gamma/(2c) \geq 1$.

$$\begin{aligned} b^*_{N-1} &= \max\left\{\min\left\{\frac{1}{2c}\left(\gamma + \gamma b^*_N - (b^*_N)^2 c\right), 1\right\}, 0\right\} \\ &= \max\left\{\min\left\{\frac{\gamma}{c} - \frac{1}{2}, 1\right\}, 0\right\} = 1, \end{aligned}$$

hence $b^*_{N-1} \geq b^*_N$ is satisfied in both cases.

Induction Step: Assume that $b^*_{i+1} \geq b^*_{i+2}$. We need to show that $b^*_i \geq b^*_{i+1}$. We prove this in two parts. First, assume that $0 < b^*_{i+1} < 1$. Induction step of that case is completely identical to the one earlier, so we omit it. Assume now that $b^*_{i+1} = 1$. According to Eq. (2.5), denote x^*_i in such way that $b^*_i = \max\{\min\{x^*_i, 0\}, 1\}$. We can write then

$$
\begin{aligned}
x_i^* &= \frac{1}{2c}\left(\gamma + \sum_{j=i+1}^{N}(\gamma - b_j^* c)\prod_{k=i+1}^{j} b_k^*\right) \\
&= \frac{1}{2c}\left(\gamma + b_{i+1}^* \sum_{j=i+2}^{N}(\gamma - b_j^* c)\prod_{k=i+2}^{j} b_k^* + (\gamma - b_{i+1}^* c)b_{i+1}^*\right) \\
&\geq \frac{\gamma}{2c} + b_{i+1}^*\left(b_{i+1}^* - \frac{\gamma}{2c}\right) + \frac{(y - b_{i+1}^* c)b_{i+1}^*}{2c} \\
&= \frac{\gamma}{2c} + \frac{1}{2} \geq 1.
\end{aligned}
$$

We can conclude that $b_i^* = b_{i+1}^* = 1$, since $\gamma \geq c$. This completes the induction and the proof of part 1 of the Lemma.

The proof for Part 2 of the Lemma is very similar, we therefore omit it for brevity and only provide hints on how it differs. We split the proof into 2 parts: first is assuming that $\gamma < c/2$, and the second part is assuming that $\gamma \geq c/2$. It is possible to show that in the first case the optimal bid $\bar{b}_i^*$ is always in the interior between 0 and 1, and that

$$
\bar{b}_i^* = \frac{\gamma}{c} + \frac{(\bar{b}_{i+1}^*)^2}{2},
$$

then the result follows by induction. In the second case, we can bound $\bar{b}_i^*$ similarly to b_i^*, and the result follows.

Part 3 can be shown by induction.
Basis of Induction: $\bar{b}_N^* \geq b_N^*$ as

$$
\bar{b}_N^* = \min\left\{\frac{\gamma}{c}, 1\right\}, \quad b_N^* = \min\left\{\frac{\gamma}{2c}, 1\right\}.
$$

Induction Step: Assume $\bar{b}_{i+1}^* \geq b_{i+1}^*$. As we show above in the proof of part 1 and part 2, we can express bids $\bar{b}_i^*$ and b_i^* as

$$
\bar{b}_i^* = \min\left\{\frac{\gamma}{c} + \frac{(\bar{b}_{i+1}^*)^2}{2}, 1\right\}, \quad b_i^* = \min\left\{\frac{\gamma}{2c} + \frac{(b_{i+1}^*)^2}{2}, 1\right\}.
$$

We need to show $\bar{b}_i^* \geq b_i^*$. Depending on whether $\bar{b}_i^*, b_i^*$ are in the interior or not, we consider four cases.

1. $\bar{b}_i^* = 1$ and $b_i^* > 1$ — the result holds.

2. Both $\bar{b}_i^* = 1$ and $b_i^* = 1$ — the result holds.

3. Both $\bar{b}_i^* < 1$ and $b_i^* < 1$. In this case we can write

$$\bar{b}_i^* - b_i^* = \frac{\gamma}{2c} + \frac{1}{2}\left((\bar{b}_{i+1}^*)^2 - (b_{i+1}^*)^2\right) \geq 0,$$

where inequality holds because of the inductive hypothesis. Therefore, the result holds.

4. Finally, we need to demonstrate that case $\bar{b}_i^* < 1$ and $b_i^* = 1$ is not possible. Note that if this holds, this would imply that

$$\frac{\gamma}{2c} + \frac{1}{2}\left((\bar{b}_{i+1}^*)^2 - (b_{i+1}^*)^2\right) < 0,$$

which is a contradiction.

Chapter 3

REAL-TIME CONTROL OF TRAFFIC MERGING USING NEURAL NETWORKS TRAINED ON OPTIMAL SOLUTIONS

3.1 Introduction

There are a large number of operational problems that require real-time control over some stochastic configurations. For a fixed configuration, the problem to be solved is often combinatorial, non-convex with stochastic inputs, and difficult; finding an optimal solution may take hours even on powerful machines. The control, however, has to be determined in a fraction of a second. Practitioners, therefore, resort to simple heuristics or policies that may be intuitive and transparent but often do not perform very well compared to the optimal possible solution.

One good example of such a problem is the merging in a blocked-lane scenario discussed in the previous two chapters. Before, we assumed that the drivers are selfish, setting their own (rational) objectives. The central planner's role was merely to provide the general framework or the mechanism and choose its parameters according to some criteria. However, one could imagine a situation when there is a critical need for a central planner to take full control over all involved drivers. For example, there could be an urgency to clear the bottleneck as fast as possible. The problem then boils down to finding a set of optimal control for all the drivers involved.

In this chapter, we combine off-line integer programming and Neural Networks (NN) to offer a solution that adapts to the specific configuration in real-time, yet takes only a fraction of a second. The broad strategy is straightforward: first,

formulate an integer program to find the optimal solution for a given configuration of drivers. Then construct an appropriate Neural Network and train it (off-line) on stochastically generated configurations where the ground-truth is given by the optimal solution for that configuration. Finally, use the NN to recommend a real-time traffic control solution, with the specific observed positional configuration as the input.

The off-line training is highly parallelizable and one can scale to larger problems or more training samples by investing in more computing resources. We note two salient facts about this strategy: First, the main computational cost is a fixed, one-off, expense, and two, the trained neural networks can be reused across multiple road locations.

Generally speaking, this strategy is quite broad and is applicable (naturally with problem-specific integer programming formulations or dynamic programming approximations) to many other problems requiring real-time control with stochastic inputs, such as in job-shop control, air-traffic control, real-time advertising control, and many others. Therefore, we use our traffic merging problem as a test-bed for this strategy.

With sensors, vehicle-to-vehicle communications, and semi-autonomous driving becoming a reality, there is suddenly a great opportunity to bring optimization and control thinking to bear on traffic problems. There is a precedent for this from a different transportation sphere—airline traffic. Air-traffic controllers space and schedule the order of take-offs and landings precisely to maximize throughput and reduce accidents. Such management has greatly improved efficiency and maximized flow at busy airports while maintaining an excellent safety record. Perhaps, similar improvements can be achieved in car traffic.

We investigate how well our methodology performs for the same blocked-lane scenario as before. A central planner wishes to control how the vehicles on the blocked lane merge into the free-flowing lane to minimize the sum of travel times for all the vehicles. The optimal merge points and velocities will change depending on the initial positions, which are stochastic. We first formulate the merging problem via Mixed-Integer Programming (MIP). Given any starting conditions (that is, the positions and velocities of all cars), the resulting optimal solution gives as an output the exact velocities and positions of the cars for any moment until they leave the bottleneck. By numerical simulations, we show that the resulting optimal solution can be as much as 25% better than simple non-state dependent policies such as early-merge or late-merge.

We then train a Neural Network, using a large number of exact MIP solutions for randomly generated inputs. The resulting policy takes the same inputs as MIP and gives recommendations for all the cars, almost in real-time. We conduct extensive simulation studies on how our NN-policy performs, both with respect to the optimal solution and the simple popular policies. In particular, our Neural

Policy performs four times better than simple policies and can be within 1.4% to the optimum. Solving for an exact optimal solution even for a small-size problem requires 10 minutes, while our policy can be computed in a second and takes only the current snapshot of vehicle positions and velocities as inputs.

We also identify various research problems for future study. The output of neural networks may not be directly implementable for a certain proportion of configurations as we do not constrain the outputs to be feasible. So for these configurations, quick adjustments to the NN output have to be made. This points to an interesting new research area on how to modify the NN outputs quickly to obtain good feasible solutions. In this chapter we suggest one that is promising, solving a smaller quicker optimization problem, but this of course takes more computational time than just reading off a NN solution.

While there is a body of literature that deals with this topic (see Section 3.2 for a literature review), we believe our point-of-view and NN-policy is new. To summarize, the contributions of this work are as follows:

1. We formulate a central planner state-dependent optimization problem, that to our knowledge, is new and gives us a new reference for optimization and policy bench-marking and to measure scope for potential improvements.

2. We use the integer program to train a NN with an appropriately chosen architecture on stochastically generated configurations.

3. We devise a policy based on the NN for real-time traffic control. As the NN gives infeasible solutions for a small percentage of configurations we give a post-processing procedure that balances speed with performance.

Note that the only requirement for this overall approach is that we have some idea of the stochastic distribution of traffic. This can usually be estimated quite well based on historical data.

The broad strategy is, of course, applicable to a wide variety of application areas requiring real-time control and where we have sufficient data to form a stochastic model of the input configurations. We are essentially exploiting advances in integer programming technology and software, the universal approximation properties of Neural Networks and the subsequent software and optimization advances in training them, to devise practical and effective real-time controls for a concrete operational situation.

The application-specific research problems are in the post-processing, to harvest as many solutions as possible that are directly feasible, and to adjust those that are not, quickly enough to be implementable. Our application in traffic highlights these issues.

The remainder of the chapter is organized as follows. In Section 3.2 we review the literature with an accent on deep learning techniques and multi-agent systems;

in Section 3.3 we discuss the modeling assumptions; in Section 3.4 we provide a detailed mathematical formulation of the merging problem as a mixed-integer problem; in Section 3.5 we present an architecture of our Neural Network, as well as details on a post-processing procedure and the data set; in Section 3.6 we compare performance of our Neural Network with popular merging heuristics in a simulation; finally, in Section 3.7 we conclude the chapter and discuss directions for future research.

3.2 Literature review

The problem we study is of high relevance; indeed, it touches almost everyone who regularly travels on a highway. Uncoordinated merging is a common observable, relevant phenomenon, and potentially controllable by incentives, prices, or information. Not surprisingly, it was extensively studied by different research communities. For a broader review of the merging literature, we refer the reader to Section 1.3. Note that, to a large extent, what is known about the traffic merging is based on human drivers. In the previous chapters, we assumed that the drivers are algorithm-assisted but rational and self-interested. The situation moves even further if we assume that the drivers are exogenously controlled. This might create severe obstacles to the direct application of principles and models that have been developed during decades.

Theoretical computer scientists and mathematicians (see Olfati-Saber (2006), Cucker and Smale (2007), Vicsek et al. (1995)) have investigated natural self-organizing phenomenon by a large number of autonomous agents, motivated by birds and insects. However, this raises the question on why birds and ants resolve potential congestion efficiently and quickly with decentralized decision-making, while human drivers end up with congestion and chaos[1]. The models that aim to recreate this type of emergent behavior are known as swarm robotics. However, the topic faces many challenges, such as scalability problems (see Barca and Sekercioglu (2013)), and is yet to see major practical applications. This highlights gaps in our understanding of how decentralized multi-agent systems can bring efficient cooperation.

The classical traffic equilibrium models, such as the Waldrop equilibrium model (Wardrop (1952)) or the Braess paradox (Braess (1968)), hint at the inefficiencies in multi-agent systems. Other prominent examples of modeling of Nash equilibria to explain traffic congestion are Arnott et al. (1991), de Palma et al. (1983), Vickrey (1969b), Arnott et al. (1990). However, they are, by and large, macroscopic, population-level models that are good for insight but operationally

[1]To quote Chazelle (2015)): *The emergence of collective structure from the decentralized interaction of autonomous agents remains, with notable exceptions, a mystery*

have not had much impact, at least for resolving local driver-level problems such as traffic flow upon a lane closure.

Neural networks are receiving increasing attention from both researchers and practitioners. A textbook Goodfellow et al. (2016) is a thorough description of state-of-the-art deep learning techniques, both from the practical and academic viewpoint. The great advance of neural networks is one of the key factors that made autonomous vehicles possible in the first place. Researchers apply deep learning techniques to develop robotic cars for more than 30 years by now, starting from early works such as Pomerleau (1989). Deep learning is essential for image processing of vehicle cameras and is also used to train robotic drivers for safe and efficient driving policies.

In a recent paper Amini et al. (2020) the authors develop driving control policies using neural networks and simulation-enhanced data. Henaff (2019) develops a neural network for more general policies of autonomous agents in complex environments. Given that the need to interact with humans is undoubted for autonomous vehicles in the nearest future, policies guaranteeing safety on that side attract the researchers' attention. In the work Sadigh et al. (2016) a deep learning policy is developed, allowing a robotic driver to interact with human drivers on the road. Similarly, in Gupta et al. (2018) a recurrent neural network is trained to help predict pedestrians' maneuvers. We note that it is typical to use human-based data to train policies in the literature. Our work stands out in that sense, as we use pre-computed optimal controls for training, and such optimal solutions are also challenging to find.

There is a recent interest in using modern deep learning as a support for other optimization algorithms, or as a stand-alone way to solve such problems, outside of the traffic. The paper Bengio et al. (2018) is an overview of applications of deep learning to combinatorial optimization. Neural networks can help to pick parameters for a broad class of other methods, exact or approximate. An example is the fine-tuning of branch-and-bounds methods, usually used to solve mixed-integer programming formulations; see Gasse et al. (2019). A different approach related to our work is imitation learning. The idea is to train machine learning algorithms on the decisions provided by human experts; see Hussein et al. (2017) for an overview. Notably, the use of certain types of recurrent networks allows for a flexible generalization. For example, the model trained on solutions with fewer cities in the traveling salesman problem can provide heuristic solutions to a larger problem; see Bello et al. (2016).

3.3 Model details

In this section, we lay out the details of our model. There are two-lanes, the blocked (B) lane and the free lane (F). All cars from both lanes need to exit the system through the free-lane. Time is discrete with 0 representing the start time of the blockage and time T representing the end of the observation horizon. We select a stretch of the highway as our system, and in this section, treat distances, velocities, and accelerations as continuous variables.

3.3.1 The incident and the driver model

The vehicles can accelerate at a maximum rate of A and decelerate at a safe rate of D, and there is a maximum velocity of $v_{\max}$. If a vehicle at velocity v_2 is following a vehicle at velocity v_1 and $v_2 > v_1$, the deceleration rate and safe time-headway α defines when vehicle 2 starts reducing its speed so that by the time it catches up to vehicle 1, its speed should match it.

For safety reasons, a minimal distance between vehicles, which we denote as z, is mandatory. When the distance between the vehicles is less than $x_c = \alpha v_2 + z$, the difference in velocities starts affecting vehicle 2, and it starts decelerating at the rate d. If the distance between 1 and 2 is greater than x_c, vehicle 2 will maintain its speed v_2, or even can accelerate to a maximum as long as the distance to 1 is at least the safety parameters. So the velocity of the vehicle ahead also limits the velocity and acceleration possibilities of a vehicle.

The model is similar to the rules of Cellular Automata simulations, except time headway is a parameter that can be calibrated and continuous space and velocities allow for extra flexibility. It also resembles popular car-following models (see Brackstone and McDonald (1999)), but stylized in a way that allows us to formulate it as a linear optimization problem.

A central planner has the objective of minimizing the *sum* of travel times of all the vehicles in the stretch of study. Alternative objective is to minimize *clearing time*, that is the time until the last vehicle leaves the study area.

The decision variables are the locations where each driver should merge and the velocities at the merges. In notation, a driver at location x_t has a velocity of $v_t = x_t - x_{t-1}$ and their position at $t+1$ is $x_{t+1} = x_t + \min(v_{\max}, v_t + d)$ if they accelerate or $x_{t+1} = x_t + (v_t - d)$ if they decelerate or $x_{t+1} = x_t + v_t$ at constant velocity, where $0 \leq a \leq A$ and $0 \leq d \leq D$.

The incident happens at location L (with the origin 0 representing the start of the stretch of study). A vehicle on the B-lane that reaches location L without merging has to stop completely (velocity 0). So by our deceleration rate, a vehicle on the B-lane traveling at velocity v will start reducing their velocity from a distance of $\alpha^\tau v + z$. Here α^τ is a safe time headway towards the blockage point; it

can be different from α, typically higher since we assume that immobile obstacle leads to more careful driving. This is commonly called the *taper* region. We can visualize merging in the tapper zone on the time-space diagram as in Figure 3.1

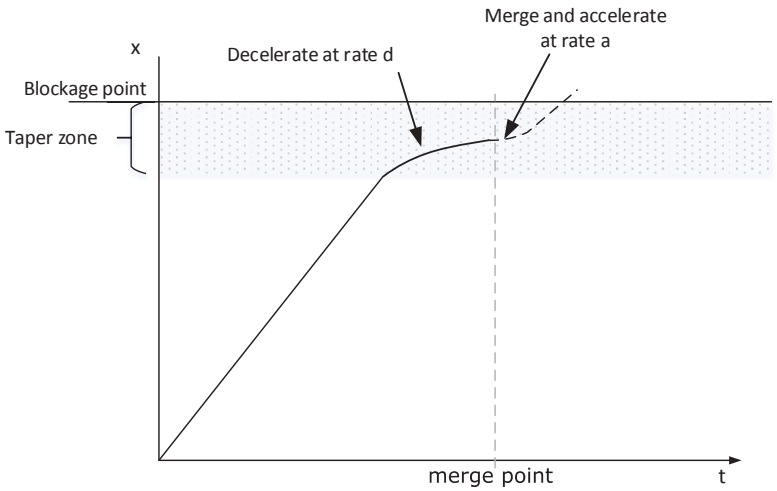


Figure 3.1: The time space diagram of a merge inside the taper region; deceleration due to the taper and acceleration after the merge to the F-lane.

3.3.2 Merging model

Our stylized model of the merging maneuver is as follows: Vehicles on the B-lane have priority of merging, and can merge as long as the safety constraints with respect to the vehicle behind and vehicle ahead are satisfied. However, a merge creates a penalty (because of natural driver behavior that is called relaxation in the literature) that we model as a velocity reduction after they merge, given by κv^F, where $0 < \kappa < 1$ is the merge penalty factor and v^F is velocity of the merging vehicle. This is a stylized model based on many empirical observations (for instance in Hidas (2005)). One could make the penalty κ dependent on the density of vehicles on the F-lane but we take it as a constant for simplicity.

Safety constraints are incorporated as follows. Let us say that the merging vehicle has velocity v, and there are two vehicles on the target lane, one ahead with velocity v^A and one behind, with velocity v^B. Merging drivers need to ensure that the maneuver is safe for both the vehicle ahead and behind. First, there must be a minimum safe distance ahead y^A and behind y^B. Their role is similar to our safety distance z, but their values potentially might be greater, since merging is more dangerous than simple car-following.

In addition, there must be a safe time-headway to all drivers. We denote these

as β^A and β^B, respectively, and they are also not necessarily the same as car-following headway α.

Summarizing, the merging vehicle needs at least a distance behind equal to $\beta^B v^B + y^B$ and a distance ahead $\beta^A v + y^A$, where v^B is the velocity of the vehicle behind on the target lane.

Similarly to the previous chapters, for the sake of simplicity and tractability, we assume that no F-lane to B-lane merges are allowed, so vehicles merge from the blocked B-lane to the F-lane but they do not merge back to the B-lane (for example, to exploit lower densities). This is a plausible assumption as long as the distance to the blockage is not too large as drivers are reluctant to merge to a blocked lane.

3.4 Formulation of the central planner's problem

In this section we give our main formulation to minimize the sum of travel times. The point-of-view is that of a central planner who is able to co-ordinate all the vehicles. However, the planner realizes that individual drivers are interested in minimizing their own travel times. So the formulation incorporates incentive constraints so no driver has an incentive to deviate from the proposed outcome.

There are m vehicles in the system, for every vehicle i we know (i) starting lane, B or F; (ii) starting position x_i, measured from the start of the study area; (iii) starting velocity w_i. Velocities are expressed in meters per second, positions are in meters, and both are real numbers. The distance from the start of the study area until the blockage is L meters.

The time horizon T is known and is divided into a number of time periods of length Δt.

3.4.1 Variables

The primary decision variables are the following:

1. v_{it}: velocity of car i at time period t.

2. ℓ_{it}: binary variable that takes 1 if car i merged by time period t (including t itself), and zero otherwise. Defined only for the cars starting on B lane.

3. o_{ij}: binary variable that defines the final ordering of cars i and j. It takes 1 if i finishes ahead of j and zero otherwise. Clearly, no need to define it for all i, j since $o_{ij} = 1 - o_{ji}$. Moreover, if both i, j are from F-lane, o_{ij} is known from the initial ordering and it can not change.

We also define a number of decision variables derived from the above, i.e., expressed as linear functions of the above primary ones.

1. p_{it}: the position of car i at time period t, counting from the start of the study area. It can be calculated as follows:

$$p_{it} = x_i + \sum_{q=1}^{t} v_{iq}.$$

2. d_{ijt}: the distance between cars i and j at the time period t, it can be calculated as $d_{ijt} = p_{it} - p_{jt}$. Note that the value is positive if car i is ahead of car j.

3. m_{it}: a binary variable representing merge maneuver, which is defined only the for cars that initially were on B-lane. The variable takes 1 if car i performs merge to the F-lane at time t and 0 otherwise. It can be calculated as $m_{it} = \ell_{it} - \ell_{it-1}$.

3.4.2 Objective function and constraints

There are three different ways to define the objective function for the problem:

1. find a minimal time horizon T^*, sufficient for all cars to leave the bottleneck;

2. minimize the sum of leaving times of all individual drivers;

3. maximize the sum of distances traveled by each driver within a fixed horizon T.

The first strategy involves setting a constant objective, then solving numerous integer programs, trying to identify minimal T^* that gives for a feasible solution (say using a binary search strategy). The problem, however, is that solving many MIPs is time-consuming, and the solution time tends to increase greatly when the current T is close to the minimal one. The other problem is that this objective may prescribe a solution where a vehicle can be stuck on one position without moving even when space is available, which is difficult to implement as it wont be compatible with the behavior of the drivers.

The second and third strategies are similar at a high level. They require solving only a single problem and lead to incentive-compatible solutions. However, the second objective function cannot be expressed as a linear function of our decision variables.

We, therefore, choose the third objective, expressed as follows:

$$\min \sum_{i=1}^{n} p_{iT}.$$

The constraints in our MIP are the following:

1. Basic constraints on all variables: They are $0 \leq v_{it} \leq v_{max},\ \forall i, t$. For merging variables, $\ell_{it} \in \{0, 1\},\ \forall i, t$. Moreover, ℓ_{it} must be a step function, so we need to enforce this by $\ell_{it-1} \leq \ell_{it},\ \forall i, t$. For ordering variables, $o_{ij} \in \{0, 1\}\ \forall i, j$.

2. Each vehicle, starting on the B-lane must merge at some point (and can merge only once): $\ell_{iT} = 1$.

3. Each car, starting on B-lane must merge at some point (and can merge only once): $\ell_{iT} = 1$.

4. The change of velocity variables must respect maximum acceleration rate A and maximum deceleration rate D. That is:

$$v_{it} - v_{it-1} \leq A\Delta t, \quad \forall i, t;$$
$$v_{it-1} - v_{it} \leq D\Delta t, \quad \forall i, t;$$

5. All cars by the end of time horizon T must leave the study area in the sense of passing the point of blockage. Mathematically,

$$p_{iT} \geq L, \quad \forall i.$$

6. Final positions of all cars must respect the final ordering variables. We can guarantee this by the following set of constraints:

$$p_{iT} \geq p_{jT} + z - o_{ji}M, \quad \forall i, j,$$

   where z is a minimal required distance between position (i.e. car length), and M - some excessively large constant. The last component ensure that the constraint is not active if $o_{ji} = 1$ (j is ahead of i in that case).

7. All cars must respect the distance to the car ahead and keep a safe distance. The exact form of the constraint depends on the initial lanes of the two cars. Assume first that both cars i, j start on F lane. Then we know exactly, which cars are ahead of another and the constraint is:

$$d_{jit} \geq \alpha v_{it} + z,$$

for all cars, such that i is behind j; α is a coefficient we discussed earlier (safe time headway).

Consider now the case when the car i starts on F and the car j that is ahead of i starts on B. We want car i to take into account the distance to car j, if both conditions are satisfied: (1) car j have merged already (2) car j is ahead of i in the final ordering (and thus, always ahead after merging). The condition can be written as:

$$d_{jit} \geq \alpha v_{it} + z - (1 - \ell_{jt})M - o_{ij}M, \quad \forall t,$$

for all cars i starting on lane F and j starting on lane B.

In case car i is from lane B and j is from lane F, we have very similar constraint (with slightly changed indices):

$$d_{jit} \geq \alpha v_{it} + z - (1 - \ell_{it})M - o_{ij}M, \quad \forall t,$$

Finally, the most complicated case is when both i and j are from B lane. In this case we use two different constraints to capture two situations, when the velocity of car i can be affected by j. First situation is when both cars are currently on lane B and moreover, j is ahead of i. This could be known from the initial ordering, since it could not potentially change before any of cars merge. Let us denote $b_{ij} = 1$ if i was initially ahead of j and zero otherwise. The constraint then is:

$$d_{jit} \geq \alpha v_{it} + z - (\ell_{it} + \ell_{jt})M - b_{ij}M, \quad \forall t.$$

The second case is when they both on lane F, and final ordering is such that j is ahead of i. To account for that case:

$$d_{jit} \geq \alpha v_{it} + z - (2 - \ell_{it} - \ell_{jt})M - o_{ij}M, \quad \forall t.$$

8. A special case is a taper region near blockage on B lane, where drivers should slow down. This can be captured by the following set of constraints:

$$L - p_{it} \geq \alpha^T v_{it} + z - \ell_{it}M, \quad \forall t,$$

and for all i starting on lane B.

9. A merging car should suffer a velocity penalty $0 \leq \kappa \leq 1$ at a time of merge:

$$v_{it} \leq v_{it-1}\kappa + (1 - m_{it})M, \quad \forall t,$$

and for all cars i starting on B lane.

10. A certain distance to a car ahead and a car behind must be satisfied for a car to perform safe merging maneuver. First we check these conditions for all cars ahead on the target lane (in the final ordering). The form of the constraint depends on whether j initially is from F lane, or from B lane. If it is from the F lane:

$$d_{jit} \geq \beta^A v_{it} + y^A - o_{ij}M - (1 - m_{it})M, \quad \forall t,$$

If j is from the B lane, the condition is:

$$d_{jit} \geq \beta^A v_{it} + y^A - o_{ij}M - (1 - m_{it})M - (1 - \ell_{jt})M, \quad \forall t,$$

Similarly we can check conditions for the car behind, accounting for the velocity of that car. If car j is from the lane F, the condition would be:

$$d_{ijt} \geq \beta^B v_{jt} + y^B - o_{ji}M - (1 - m_{it})M, \quad \forall t.$$

If j is from B lane:

$$d_{ijt} \geq \beta^B v_{jt} + y^B - o_{ji}M - (1 - m_{it})M - (1 - \ell_{jt})M, \quad \forall t.$$

3.5 The Policy Neural Network

Solving MIP is in general an NP-hard problem, and we cannot expect to get a solution quickly to our complicated MIP even for a small number of cars. The idea behind a NN-policy is to train a neural network off-line on a large number of configurations with their corresponding MIP solutions as ground-truths (represented in an appropriate way), and then use the resulting trained NN to derive a policy for making merging recommendations for any new configuration occurring on the road.

Feeding a new configuration (a new set of starting positions and velocities) to a trained NN and obtaining an output is almost immediate so we obtain recommendations very quickly. However, the resulting recommendations turn out to be infeasible in a few cases and we have to fix them locally and quickly. We propose one method that seems to work reasonably well.

As a strategy this usage of NN to memorize, learn and generalize from a collection of optimal solutions is promising. However, we note that this is a new approach and further research and experimentation is needed to make it operational.

Our current network is able to predict policy for a fixed number of B and F vehicles (the same as in the training set). We use a standard multi-layer perceptron (MLP) and its implementation in an industry-standard Python library (pytorch).

3.5.1 Architecture

We use a network with 10 hidden layers, with rectified linear activation functions on each of the layers. We use 200 hidden units in all the layers. These choices were made after considerable experimentation.

Consistent with the general experience with deep learning, performance improves significantly with multiple hidden layers. In our experimentation ten hidden layers was sufficient, with acceptable training times and beyond with performance flattens out on our dataset.

As an input to the NN, we use N starting positions and N starting velocities, where N is the total number of vehicles. We use strict ordering (e.g. the blocked-lane vehicles are always the first in the input) and keep the same number of vehicles on each lane.

Rather than trying to predict the full policy, we design a network that predicts the key information and then we use it to derive the policy. Specifically, we want the network to predict merging times of each vehicle and the final ordering of the vehicles. This essentially corresponds to all MIP integer variables ℓ_{it}, o_{ij}. Predicting integer variables, and especially permutations in the latter is much more challenging for the NN.

The trained network predicts not the ordering, but the position (a real number) of each vehicle at a future time T, p_{iT}, $\forall i$. This reduces the number of output units significantly while allowing for a one-to-one transformation of these positions into variables of interest o_{ij} (all positions are real, and thus almost surely different values).

As a final step, we feed these predicted variables back to MIP solver to check feasibility. If feasible, we have a solution very quickly. If not, we solve just an LP without any integer variables. This can be done reasonably quickly (but of course not in real-time); but as we mentioned earlier is probably the critical area for future research. Naturally, feeding the NN prediction into LP solver does not always lead to a feasible solution, but just provides a quick start for the LP. For a small but significant percentage of configurations the solution provided by the neural network ordering is not compatible with merging times, and the resulting LP becomes infeasible. Our heuristic solution strategy in such cases is to solve a partial MIP discarding predicted values of variables ℓ_{it} but using only the predicted ordering variables o_{ij}. This greatly reduces the size of the problem while essentially guaranteeing a feasible solution.

As a result of the entire process, we obtain a policy that provides the information identical to a solution of MIP. A training set is a number of exact MIP solutions. The whole process of forming the neural network policy (including preparation of the training set) is depicted in Figure 3.2.

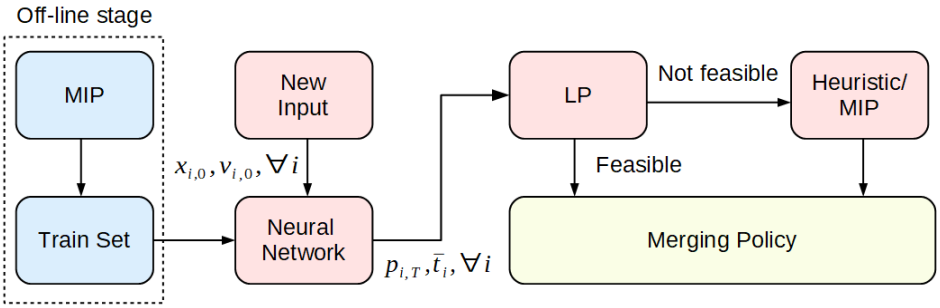


Figure 3.2: The principal scheme of forming Neural Policy. NN inputs are initial positions and velocities; the output is merging times and final positions of the cars, which then transformed into required integer variables and is fed into LP solver. We can solve a reduced size MIP or use heuristic if the prediction of NN does not lead to a feasible solution.

3.5.2 Choice of the hyperparameters and training

As a training set, we use 5000 exact MIP solutions. All solutions have the same fixed number of blocked-lane vehicles $N_B = 4$ and free-lane vehicles $N_F = 8$. We use the road stretch of 150m length, all starting positions and starting velocities are generated randomly. This is a proof-of-concept study done with limited resources, so we did not expand the sizes. We acknowledge realistic situations would require 20 to 30 cars at least (near the critical stage of the road), but the MIP would require more machines to be able to generate solved instances. The important question we are interested is the relative performance of the NN policy compared to the optimal solution.

For learning, we use mean squared error loss function, and the minimization method is adaptive moments with decoupled weight decay (AdamW). The learning rate is set to 0.0001, the rest of hyperparameters are PyTorch defaults. We use full batch optimization and train the network for 30000 epochs.

3.6 Numerical Study

In this section we report a numerical study to see how our policy performs in a simulation comparing it with a typical traffic setting where each driver behaves autonomously making their own decisions on when and where to merge.

3.6.1 Description of the simulation setting

To make the comparison easier, in our simulation model we keep all the modeling assumptions and parameters from Section 3.3 and Section 3.4. We consider the

same continuous-time, discrete-space environment. Similar to the model used in our MIP, we consider a closed system, i.e., without new arrivals, focusing on a snapshot of vehicles during a period of study over a stretch of the road.

The simulation prescribes common-sense rules on how drivers move while respecting all safety conditions and incorporates probabilistic attempts at merges. We study the performance of the various policies by tallying statistics if the drivers followed the policy under study.

In the following we describe in detail driver behavior in our simulation and how they are implemented at each time period:

1. First, the blocked-lane drivers check conditions for the merging. They check that the distance to the blockage is less than merging distance d^M (we discuss the choice of this parameter below). Then, the driver checks that on the target lane there is enough free space, i.e. $d^A - y^A \leq \beta^A v$ and $d^B - y^B \leq \beta^B v^B$, where d^A is the distance to the car ahead (target lane), d^B is the distance to the car behind (target lane), v is the velocity of merging car, v^B is the velocity of the car behind, $y^B, y^A, \beta^A, \beta^B$ are the same parameters we discussed in Section 3.4. If these rules are satisfied, the merge is performed immediately (the drivers are not strategic in their merging). If the driver performs the merge, he or she cannot accelerate this time period, and the velocity is reduced to κv, $0 < \kappa < 1$. As in our MIP, only merges from B to F are allowed.

2. All vehicles respect the maximum acceleration, deceleration limits, maximum velocity and of course the forced stoppage at the blockage point (if the vehicle is on the blocked-lane). At each time-step vehicles on the blocked-lane follow the velocity rule:

$$v \leftarrow \min\{v + A, v_{max}, d/\alpha^\tau, d^A/\alpha\},$$

   where A is the maximum acceleration, d is the distance to the blockage, d^A is the distance to the vehicle ahead (on the same lane), and α, α^τ are the parameters we introduced earlier for the regular and taper zones respectively.

   Similarly, for a driver on the free lane:

$$v \leftarrow \min\{v + A, v_{max}, d^A/\alpha\}.$$

3. All positions are updated at each clock-tick of the simulation. For each driver:

$$p \leftarrow p + v,$$

   where p is the position of the driver.

4. If all driver left the bottleneck (all $p > L$), the simulation terminates, and we record the final positions and time of clearing the bottleneck.

As one could see from the steps, our discrete-event simulation is inspired by CA simulations common in traffic studies (Maerivoet and Moor (2005); Nagel et al. (1998)), but with more detailed mechanics and in continuous time.

3.6.2 Benchmark merging policies

We compare our NN-based policy against three commonly studied policies:

1. *Late Merging*: In this policy we assume drivers merge as late as possible. The motivation for this policy (for practioners, as well as in the academic literature) is that for an unexpected interruption of service such as an unplanned lane-closing, drivers who are quite far from the blockage point cannot observe it, and thus have no incentive to merge from B-lane to the F-lane. Moreover, drivers in general cannot solve an optimization problem on-the-fly, due to lack of information and computational power. Hence the first driver (who can actually observe an approaching obstruction) postpones the merge till the last minute.

   So till reaching a relatively short distance d^M before blockage, no one merges. Within this distance, the first driver checks the opportunity to merge and performs merging if there is a slot. After that, the second driver at B can see the blockage, and if he or she is within d^M from the bottleneck, he or she tries to merge, and so on. The choice for d^M should be reasonably small, we choose $d^M = v_{max}/\alpha^\tau$, which corresponds to reaching the start of our taper point.

2. *Random Merging*: The motivation for this policy is that the drivers are aware of the blockage ahead, but have no information about the distance to the blockage. As a result of different perceptions of different drivers, each one decides when to start merging randomly. For our simulation, we set for each vehicle an individual start-to-merge distance, based on an uniform distribution: $d^M = U(0, L)$. This policy also reflects what one observes in many real-world situations and also serves as a good benchmark policy to represent totally uncoordinated merges.

3. *Early Merging*: This policy represents highly risk-averse drivers who merge as soon as they become aware of a blockage, or when they are either instructed to merge right away. This policy has also been studied in the research literature. All drivers start looking for a merge gap as soon as they enter the study area. Therefore, we set $d^M = L$.

3.6.3 Comparison of MIP, NN and simulation policies

Our main goal for constructing a NN-based policy is to create a fast on-line algorithm that hopefully is capable of learning optimal patterns from a set of MIP solutions of random configurations, and provides a solution that is reasonably close to the global optimum for a particular configuration. Therefore in this section, we examine the performance of our NN-policy vis-a-vis the global optimum, and also the common merging policies described in Section 3.6.2.

NN training and some examples

First, we set all the parameters of the MIP and train the NN with the parameters as follows: Length of road stretch is $L = 150$m; length of vehicle (used also as a minimal distance in checks for available merge space) $z = y^A = y^B = 5$m; discrete time step $\Delta t = 1$sec; maximal velocity $v_{max} = 25$m/sec; maximum acceleration $A = 5$m/sec^2; maximum deceleration $D = 25$m/sec^2. Time headway coefficients are $\alpha = 1$ and $\alpha^\tau = 2$. The former corresponds to keeping 1 second of headway to other drivers, the latter means that driver keeps 2 seconds of headway in the taper zone (drivers are more cautions in the end and also because they keep hoping for a gap to open and it is difficult to accelerate and merge if they are to close to the preceding vehicle). Merging penalty $\kappa = 0.7$, which corresponds to losing 30% of velocity upon merging. All the parameters are chosen to be as close to CA simulations as possible.

In Fig 3.3 we compare 3 solutions of the Neural Policy with their corresponding MIP solutions (all examples are out of the testing set). In the first case (top), both solutions coincide almost exactly (up to the velocity decisions of the blocked-lane vehicles, which do not affect the result). In the second scenario (middle), NN policy looks a bit different from the MIP, and there are clear minor planning errors (note that two drivers on F-lane give the way to a third blocked-lane driver, clearly losing some efficiency). However, even in that case clearing time is not much higher than in MIP solution. Finally, in the third and the most frequent case (bottom) NN policy to a large extent predicts the correct merging pattern, but some details (merging time and positions) can slightly differ, resulting in minor efficiency loss. Overall, the Neural Network learned impressively from MIP solutions. Note how it learns typical patterns: merging is usually performed in a zipper-merge style (B-F-B-F), but often it is optimal for 1-2 blocked-lane vehicles near the blockage to give the way to all passing vehicles from both lanes. Neural Network is capable of incorporating these optimal patterns successfully.

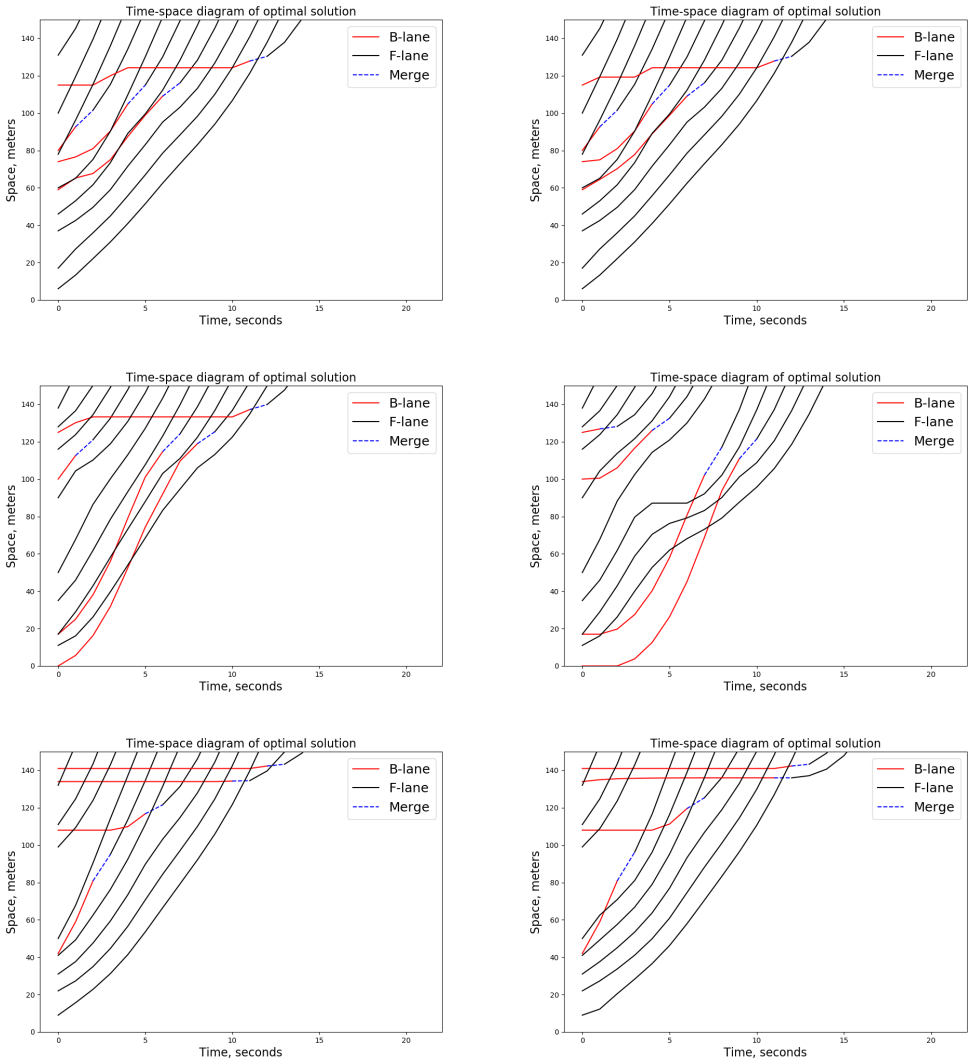


Figure 3.3: Time-space diagrams of the exact solutions provided by MIP (left column) and the same input solutions by Neural Network Policy (right column). All examples are outside of the NN training set.

NN policy on a test set

Ultimately, we are interested in constructing a policy that has low running time, and at the same time outperforms simple policies like the ones in Section 3.6.2 to give more "personalized" recommendations specific to the configuration.

In Table 3.1, we summarize the results for different policies and the exact MIP solution. We test the policies on 100 randomly generated examples (outside of the training test for NN), with $N_B = 4$ and $N_F = 8$, and the rest of the parameters

being the same as we described above. As a metric of efficiency, we use two different options: MIP objective (sum of all distances traveled by the end of time horizon) and clearing time (time required for all drivers to leave the study area). We also calculate the average runtime of each policy.

Table 3.1: Comparison of exact MIP solution, Deep Neural Network Policy and simple Cellular Automata policies.

	MIP	NN Policy	CA (early)	CA (random)	CA (late)
MIP objective gap , %	-	1.422	6.421	6.714	7.827
Clearing time gap , %	-	4.136	24.46	25.98	28.59
Runtime, s	536.9	1.285	0.006	0.009	0.006

As we can see, there is no significant difference in two metrics we use—qualitatively, all policies keep very similar performance in both, which justifies using the sum of positions as an objective for MIP. The NN-policy works remarkably well, providing 3-4 times better performance than the simple policies of Section 3.6.2.

In approximately 25% of cases NN predictions were not feasible in the linear program, and we used the partial MIP post-processing. This clearly affects runtime—NN with LP solver is slower than just reading the output of a trained NN. However, even with this penalty the average run time is only just over a second.

In general, MIP gives an exact solution but does not provide it in a reasonable time. The simple policies of Section 3.6.2 are very fast to compute, but can be significantly off in efficiency. Our NN-Policy is balancing performance and the runtime, providing a good solution in a limited time. CA simulations all perform rather similarly. The Early-Merge seems to be the best policy for this number of vehicles and density, the Random-Merge is slightly worse, and the Late-Merge is the worst of all three.

3.7 Conclusions

In this chapter we investigated a novel form of using traditional Operations Research tools such as LP and MIP which are often not suitable for real-time policy recommendations. We combine the power of these tools to solve to optimality with the ability of NNs to serve as universal function approximators to recall, match and generalize to new configurations. Our initial results are very encouraging.

In that regard, we believe our work shows promise but we wish to emphasize that there are a number of serious research issues that need to be addressed before this general strategy can be applied to operational problems. Feed-forward NNs unfortunately (as they currently stand) do not output structured results that satisfy hard constraints. As is well known, they also have trouble handling variable length inputs and outputs.

So in its current form, our NN policy has two drawbacks: (i) it does not generalize to traffic situations with a different number of vehicles; (ii) partial MIP that we use for infeasible LP cases, is potentially slow, especially for a larger number of vehicles.

The first drawback can be overcome by using a collection of neural networks, each trained for a different number of vehicles. It would require generating a large set of data—and certainly is not the most elegant solution—and is a possible solution within the proposed NN architecture. An alternative is to explore recurrent NNs, which could potentially generalize, based on just a single trained deep learning model.

The second point is not a critical one as long as the size of the problems the central regulator is trying to solve are not significantly greater than ones we use in our numerical comparison. However, developing a faster post-processing heuristic instead of a partial MIP would be more reliable and scale better.

Improving the NN policy along these two directions is a subject for future research.

Bibliography

Ahmed, K. I. 1999. Modeling drivers' acceleration and lane changing behavior. Ph.D. thesis, MIT, Cambridge, MA.

Ahn, S., M. J. Cassidy. 2007. Freeway traffic oscillations and vehicle lane-change maneuvers. Transportation and Traffic Theory 2007.

Ahn, Soyoung, Jorge Laval, Michael J. Cassidy. 2010. Effects of merging and diverging on freeway traffic oscillations: Theory and observation. *Transportation Research Record* **2188**(1) 1–8.

Amini, Alexander, Igor Gilitschenski, Jacob Phillips, Julia Moseyko, Rohan Banerjee, Sertac Karaman, Daniela Rus. 2020. Learning robust control policies for end-to-end autonomous driving from data-driven simulation. *IEEE Robotics and Automation Letters* **5**(2) 1143–1150.

Arnott, R., A. de Palma, R. Lindsey. 1990. Economics of a bottleneck. *Journal of Urban Economics* **27** 111–130.

Arnott, R., A. de Palma, R. Lindsey. 1991. Does providing information to drivers reduce traffic congestion? *Transportation Research A* **25A**(5) 309–318.

Athey, Susan, Glenn Ellifason. 2011. Position auctions with consumer search. *The Quarterly Journal of Economics* **126**(3) 1213–1270.

Barca, Jan Carlo, Y Ahmet Sekercioglu. 2013. Swarm robotics reviewed. *Robotica* **31**(3) 345–359.

Baykal-Gürsoy, M., W. Xiao, K. Ozbay. 2009. Modeling traffic flow interrupted by incidents. *European Journal of Operational Research* **195**(1) 127–138.

Bello, Irwan, Hieu Pham, Quoc V Le, Mohammad Norouzi, Samy Bengio. 2016. Neural combinatorial optimization with reinforcement learning. *arXiv preprint arXiv:1611.09940* .

Bengio, Yoshua, Andrea Lodi, Antoine Prouvost. 2018. Machine learning for combinatorial optimization: a methodological tour d'horizon. *arXiv preprint arXiv:1811.06128* .

Brackstone, Mark, Mike McDonald. 1999. Car-following: a historical review. *Transportation Research Part F: Traffic Psychology and Behaviour* **2**(4) 181–196.

Braess, Dietrich. 1968. Über ein paradoxon aus der verkehrsplanung. *Unternehmensforschung* **12**(1) 258–268.

Buisson, Christine, Mehdi Keyvan-Ekbatani, Peter Wagner. 2018. Impede autonomous vehicles merging at on-ramps? *Conference on Intelligent Transportation Systems,97th Annual Meeting Transportation Research Board*. Washington D.C., 21.

Carlino, Dustin, Stephen D Boyles, Peter Stone. 2013. Auction-based autonomous intersection management. *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. IEEE, 529–534.

Chang, G.-L., Y.-M. Kao. 1991. A realistic two-lane cellular automata traffic model considering aggressive lane-changing behavior of fast vehicle. *Transportation Research A* **25A**(6) 375–389.

Chatterjee, Kalyan, Larry Samuelson. 1990. Perfect equilibria in simultaneous-offers bargaining. *International Journal of Game Theory* **19**(3) 237–267.

Chazelle, Bernard. 2015. An algorithmic approach to collective behavior. *Journal of Statistical Physics* **158**(3) 514–548.

Choudhury, C., M. E. Ben-akiva, T. Toledo, A. Rao, G. Lee. 2007. State dependence in lane changing. *Transportation and Traffic Theory* 711–734.

Choudhury, C. F., M. E. Ben-Akiva. 2013. Modelling driving decisions: a latent plan approach. *Transportmetrica A: Transport Science* **9**(6) 546–566.

Choudhury, Charisma F, Moshe Ben-Akiva, Maya Abou-Zeid. 2010. Dynamic latent plan models. *Journal of Choice Modelling* **3**(2) 50–70.

Chowdhury, D., L. Santen, A. Schadschneider. 2000. Statistical physics of vehicular traffic and some related systems. *Physics Reports* **329**(4) 199–329.

Como, G., E. Lovisari, K. Savla. 2016. Convexity and robustness of dynamical traffic assignment for control of freeway networks. *Transportation Research Part B* **91** 446–465.

Cucker, F., S. Smale. 2007. Emergent behavior in flocks. *Automatic Control, IEEE Transactions on* **52**(5) 852–862.

Daamen, W., M. Loot, S. Hoogendoorn. 2010. Empirical Analysis of Merging Behavior at Freeway On-Ramp. *Transportation Research Record: Journal of the Transportation Research Board* **2188** 108–118.

Daganzo, C.F. 2002a. A behavioral theory of multi-lane traffic flow part I: Long homogeneous freeway sections. *Transportation Research B* **36** 131–158.

Daganzo, C.F. 2002b. A behavioral theory of multi-lane traffic flow part II: Merges and the onset of congestion. *Transportation Research B* **36** 159–169.

De Fontenay, Catherine C, Joshua S Gans. 2014. Bilateral bargaining with externalities. *The Journal of Industrial Economics* **62**(4) 756–788.

de Palma, A., M. Ben-Akiva, C. Lefevre, N. Litinas. 1983. Stochastic equilibrium model of peak period traffic congestion. *Transportation Science* **17** 430–453.

de Palma, André, Robin Lindsey. 2011. Traffic congestion pricing methodologies and technologies. *Transportation Research Part C: Emerging Technologies* **19**(6) 1377–1399.

Del Castillo, J.M., F.G. Benítez. 1995. On the functional form of the speed-density relationship: General theory. *Transportation Research Part B: Methodological* **29**(5) 373 – 389.

DOT, US. 2018. Preparing for the future of transportation: Automated vehicles 3.0. Tech. rep., Department of Transportation.

Duret, A., J. Bouffier, C. Buisson. 2010. Onset of congestion from low-speed merging maneuvers within free-flow traffic stream. *Transportation Research Record: Journal of the Transportation Research Board* **2188** 96–107.

Ebersbach, A., J.J. Schneider. 2004. Two-lane traffic with places of obstruction to traffic. *International Journal of Modern Pysics C* **15**(4) 535–544.

Edelman, Benjamin, Michael Ostrovsky, Michael Schwarz. 2007. Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords. *American economic review* **97**(1) 242–259.

Evans, D.H., R. Herman, G.H. Weiss. 1964. The highway merging and queuing problem. *Operations Research* **12**(6) 832–857.

Fudenberg, Drew, Jean Tirole. 1983. Sequential bargaining with incomplete information. *The Review of Economic Studies* **50**(2) 221–247.

Gasse, Maxime, Didier Chételat, Nicola Ferroni, Laurent Charlin, Andrea Lodi. 2019. Exact combinatorial optimization with graph convolutional neural networks. *Advances in Neural Information Processing Systems*. 15580–15592.

Gipps, P.G. 1986. A model for the structure of lane-changing decisions. *Transp Res B* **35** 107–120.

Gomes, Renato, Kane Sweeney. 2014. Bayes–Nash equilibria of the generalized second-price auction. *Games and Economic Behavior* **86** 421–437.

Goodfellow, Ian, Yoshua Bengio, Aaron Courville. 2016. *Deep learning*. MIT press.

Greenshields, B.D. 1934. A study in highway capacity. *Highway Research Board Proceedings* **14** 448–477.

Gregoire, J., X. Qian, E. Frazzoli, A. de la Fortelle, T. Wongpiromsarn. 2015. Capacity-aware backpressure signal control. *IEEE Transactions on Control of Networked Systems* **2**(2) 164–173.

Grillo, L., T. Datta, C. Hartner. 2008. Dynamic late lane merge system at freeway construction work zones. *Transportation Research Record: Journal of the Transportation Research Board* **2055** 3–10.

Gupta, Agrim, Justin Johnson, Li Fei-Fei, Silvio Savarese, Alexandre Alahi. 2018. Social gan: Socially acceptable trajectories with generative adversarial networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2255–2264.

Han, Y.-S., S.-K. Ko. 2012. Analysis of a cellular automaton model for car traffic with a junction. *Theoretical Computer Science* **450** 54–67.

Heidemann, D. 2001. A queueing theory model of nonstationary traffic flow. *Transportation Science* **35**(4) 405–412.

Helbing, D. 2001. Traffic and related self-driven many-particle systems. *Rev. Mod. Phys.* **73** 1067–1141.

Henaff, Mikael. 2019. Explicit explore-exploit algorithms in continuous state spaces. *Advances in Neural Information Processing Systems*. 9377–9387.

Hidas, P. 2002. Modelling lane changing and merging in microscopic traffic simulation. *Transportation Research Part C: Emerging Technologies* **10** 351–371.

Hidas, P. 2005. Modelling vehicle interactions in microscopic simulation of merging and weaving. *Transportation Research* **13**(1) 37–62.

Hounsell, N.B., S.R. Barnard, M. McDonald, D. Owens. 1992. An investigation of flow breakdown and merge capacity on motorways. TRL Contractor Report 338, Transport and Road Research Laboratory, Crowthorne, Berkshire, UK.

Hussein, Ahmed, Mohamed Medhat Gaber, Eyad Elyan, Chrisina Jayne. 2017. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)* **50**(2) 1–35.

Jain, R., J.M. Smith. 1997. Modeling vehicular traffic flow using M/G/C/C state dependent queueing models. *Transportation Science* **31**(4) 324–336.

Jehiel, Philippe, Benny Moldovanu. 2000. Auctions with downstream interaction among buyers. Tech. Rep. 4.

Jehiel, Philippe, Benny Moldovanu, Ennio Stacchetti. 1999. Multidimensional mechanism design for auctions with externalities. *Journal of economic theory* **85**(2) 258–293.

Jin, W.-L. 2010. A kinematic wave theory of lane-changing traffic flow. *Transportation Research B* **44** 1001–1021.

Kaplan, Todd R, Shmuel Zamir. 2012. Asymmetric first-price auctions with uniform distributions: analytic solutions to the general case. *Economic Theory* **50**(2) 269–302.

Kerner, B.S, S. L. Klenov, A. Hiller, H. Rehborn. 2006. Microscopic features of moving traffic jams. *Phys Rev E Stat Nonlin Soft Matter Phys* .

Kesting, A., M. Treiber, D. Helbing. 2007. General lane-changing model MOBIL for car-following models. *Transportation Research Record: Journal of the Transportation Research Board* **1999** 86–94.

Kesting, Arne, Martin Treiber, Martin Schönhof, Dirk Helbing. 2008. Adaptive cruise control design for active congestion avoidance. *Transportation Research Part C: Emerging Technologies* **16**(6) 668–683.

Keyvan-Ekbatani, Mehdi, Victor L. Knoop, Winnie Daamen. 2016. Categorization of the lane change decision process on freeways. *Transportation Research Part C: Emerging Technologies* **69** 515–526.

Kita, H. 1999. A merging-giveway interaction model of cars in a merging section: a game theoretic analysis. *Transportation Research Part A: Policy and Practice* **33**(3-4) 305–312.

Knoop, V., Mehdi Keyvan-Ekbatani, Marco de Baat, Henk Taale, Serge Hoogendoorn. 2018. Lane change behavior on freeways: An online survey using video clips. *Journal of Advanced Transportation* **2018** 1–11.

Knoop, V. L., S. P. Hoogendoorn, Y. Shiomi, C. Buisson. 2012. Quantifying the number of lane changes in traffic: Empirical analysis. *Transportation Research Record* **2278**(1) 31–41.

Kondyli, Alexandra, Lily Elefteriadou. 2011. Modeling driver behavior at freeway?ramp merges. *Transportation Research Record* **2249**(1) 29–37.

Krishna, Vijay. 2010. *Auction Theory Second edition*.

Laval, J. A., C. F. Daganzo. 2006. Lane-changing in traffic streams. *Transportation Research Part B: Methodological* **40**(3) 251–264.

Le, T., P. Kovacs, N. Walton, H. Vu, L. Andrew, S. Hoogendorn. 2015. Decentralized signal control for urban road networks. *Transportation Research Part C* **58** 431–450.

Lee, G. 2006. Modeling gap acceptance at freeway merges. Ph.D. thesis, Massachusetts Institute of Technology, Dept. of Civil and Environmental Engineering.

Lee, J., M. Park, H. Yeo. 2016. A probability model for discretionary lane changes in highways. *KSCE Journal if Civil Engineering* **20**(7) 2938–2946.

Lee, Seunghyeon, Dong Ngoduy, Mehdi Keyvan-Ekbatani. 2019. Integrated deep learning and stochastic car-following model for traffic dynamics on multi-lane freeways. *Transportation Research Part C: Emerging Technologies* **106** 360 – 377.

Lin, Dianchao, Li Li, Saif Eddin Jabari. 2019. Pay to change lanes: A cooperative lane-changing strategy for connected/automated driving. *Transportation Research Part C: Emerging Technologies* **105** 550–564.

Liu, R., G. Hyman. 2012. Modelling motorway merge: the current practice in the UK and towards establishing general principles. *Transport Policy* **24** 199–210.

Loot, Martijn. 2009. Empirical analysis of merging behaviour. Master thesis, Delft University of Technology, Delft, Netherlands.

Maerivoet, Sven, Bart De Moor. 2005. Cellular automata models of road traffic. *Physics Reports* **419**(1) 1–64.

Mauch, M., M. J. Cassidy. 2002. *Freeway Traffic Oscillations: Observations And Predictions*, chap. 32. Emerald Group Publishing Limited, 653–673.

McAfee, R Preston, John McMillan. 1987. Auctions and Bidding. *Journal of Economic Literature* **25**(2) 699–738.

McCoy, P.T., G. Pesti. 2001. Dynamic late merge-control concept for work zones on rural interstate highways. *Transportation Research Record 1745* 20–26.

McNeil, D.R., J.T. Smith. 1969. A comparison of motorist delays for different merging strategies. *Transportation Science* **3**(3) 239–254.

McQueen, J., R.G. Miller Jr. 1960. Optimal persistence policies. *Operations Research* **8**(3) 362–380.

Miller, A.J. 1961a. A queueing model for road traffic flow. *Journal of the Royal Statistical Society. Series B (Methodological)* **23**(1) 64–90.

Miller, Alan J. 1961b. A queueing model for road traffic flow. *Journal of the Royal Statistical Society: Series B (Methodological)* **23**(1) 64–75.

Myerson, Roger B, Mark A Satterthwaite. 1983. Efficient mechanisms for bilateral trading. *Journal of economic theory* **29**(2) 265–281.

Nagel, K., D.E. Wolf, P. Wagner, P. Simon. 1998. Two-lane traffic rules for cellular automata: A systematic approach. *Physical Review E* **58**(2) 1425–1437.

New York Times. 2016. Cars talking to one another? They could under proposed safety rules. URL https://nyti.ms/2jQkpt4.

Ngoduy, D., S. Lee, M. Treiber, M. Keyvan-Ekbatani, H.L. Vu. 2019. Langevin method for a continuous stochastic car-following model and its stability conditions. *Transportation Research Part C: Emerging Technologies* **105** 599 – 610.

Nilsson, J., J. Silvlin, M. Brannstrom, E. Coelingh, J. Fredriksson. 2016. If, when, and how to perform lane change maneuvers on highways. *IEEE Intelligent Transportation Systems Magazine* **8**(4) 68–78.

Olfati-Saber, R. 2006. Flocking for multi-agent dynamic systems: algorithms and theory. *IEEE Transactions on Automatic Control* **51**(3) 401–420.

Pipes, L.A. 1967. Car-following models and the fundamental diagram of road traffic. *Transportation Research* **1**(1) 21–29.

Pomerleau, Dean A. 1989. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*. 305–313.

Puterman, M. L. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. 1st ed. John Wiley & Sons, Inc., USA.

Rahman, M., M. Chowdhury, Y. Xie, Y. He. 2013. Review of microscopic lane-changing models and future research opportunities. *IEEE Transactions on Intelligent Transportation Systems* **14**(4) 1942–1956.

Rakesh V. Vohra. 2011. *Mechanism Design: A Linear Programming Approach*. Cambridge University Press.

Rewald, Hannes, Olaf Stursberg. 2016. Cooperation of autonomous vehicles using a hierarchy of auction-based and model-predictive control. *2016 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 1078–1084.

Rios-Torres, J., A.A. Malikopoulos. 2017. A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps. *IEEE Transactions on Intelligent Transportation Systems* **18**(5) 1066–1077.

Rothkopf, Michael H. 2007. Thirteen reasons why the Vickrey-Clarke-Groves process is not practical. *Operations Research* **55**(2) 191–197.

Sadigh, Dorsa, Shankar Sastry, Sanjit A Seshia, Anca D Dragan. 2016. Planning for autonomous cars that leverage effects on human actions. *Robotics: Science and Systems*, vol. 2. Ann Arbor, MI, USA.

Schönhof, Martin, Dirk Helbing. 2007. Empirical features of congested traffic states and their implications for traffic modeling. *Transportation Science* **41**(2) 135–166.

Segal, Ilya. 1999. Contracting with externalities. *The Quarterly Journal of Economics* **114**(2) 337–388.

Sequeira, L., A. Szefer, J.Slome, T.Mahmoodi. 2019. A lane merge coordination model for a V2X scenario. *2019 European Conference on Networks and Communications (EuCNC)*. 198–203.

Sheu, J-B. 2013. Microscopic traffic behaviour modelling and simulation for lane-blocking arterial incidents. *Transportmetrica A: Transport Science* **9**(4) 335–357.

Small, Kenneth A. 1992. Using the revenues from congestion pricing. *Transportation* **19**(4) 359–381.

Sun, D., L. Elefteriadou. 2010. Research and implementation of lane-changing model based on driver behavior. *Transportation Research Record: Journal of the Transportation Research Board* **2161**.

Tarko, A.P., D. Shamo, J. Wasson. 1999. Indiana lane merge system for work zones on rural freeways. *Journal of Transportation Engineering* **125**(5) 415–420.

Toledo, T., H. N. Koutsopoulos, M. E. Ben-Akiva. 2003. Modeling integrated lane-changing behavior. *Transportation Research Record* **1857**(1) 30–38.

Underwood, R.T. 1961. *Speed, Volume and Density Relationships*. Yale Bur. Highway Traffic, New Haven, Connecticut.

Van Arem, Bart, Cornelie JG Van Driel, Ruben Visser. 2006. The impact of cooperative adaptive cruise control on traffic-flow characteristics. *IEEE Transactions on intelligent transportation systems* **7**(4) 429–436.

van Beinum, Aries, Haneen Farah, Fred Wegman, Serge Hoogendoorn. 2018. Driving behaviour at motorway ramps and weaving segments based on empirical trajectory data. *Transportation Research Part C: Emerging Technologies* **92** 426–441.

Varian, Hal R. 2007. Position auctions. *International Journal of industrial Organization* **25**(6) 1163–1178.

Vasirani, Matteo, Sascha Ossowski. 2012. A market-inspired approach for intersection management in urban road traffic networks. *Journal of Artificial Intelligence Research* **43** 621–659.

Vickrey, William. 1961. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of finance* **16**(1) 8–37.

Vickrey, William S. 1969a. Congestion theory and transport investment. *The American Economic Review* **59**(2) 251–260.

Vickrey, W.S. 1969b. Congestion theory and transport investment. *American Economic Review* **59** 251–261.

Vicsek, T., A. Czirók, E. Ben-Jacob, I. Cohen, O. Shochet. 1995. Novel type of phase transition in a system of self-driven particles. *Physical Review Letters* **75** 1226–1229.

Walters, Alan A. 1961. The theory and measurement of private and social cost of highway congestion. *Econometrica: Journal of the Econometric Society* 676–699.

Wang, Meng, Winnie Daamen, Serge P Hoogendoorn, Bart van Arem. 2016. Connected variable speed limits control and car-following control with vehicle-infrastructure communication to resolve stop-and-go waves. *Journal of Intelligent Transportation Systems* **20**(6) 559–572.

Wang, Meng, Serge P Hoogendoorn, Winnie Daamen, Bart van Arem, Riender Happee. 2015. Game theoretic approach for predictive lane-changing and car-following control. *Transportation Research Part C: Emerging Technologies* **58** 73–92.

Wardrop, John Glen. 1952. Road paper. some theoretical aspects of road traffic research. *Proceedings of the institution of civil engineers* **1**(3) 325–362.

Wei, H., E. Meyer, J. Lee, C. Feng. 2000. Characterizing and modeling observed lane-changing behavior: Lane-vehicle-based microscopic simulation on urban street network. *Transportation Research Record* **1710**(1) 104–113.

Yang, Q., H.N. Koutsopoulos. 1996. A microscopic traffic simulator for evaluation of dynamic traffic management systems. *Transportations Research C* **4**(3) 113–129.

Zhang, J., X. Li, R. Wang, X. Sun, X. Cui. 2012. Traffic bottleneck characteristics caused by the reduction of lanes in an optimal velocity model. *Physica A: Statistical Mechanics and its Applications* **391**(7) 2381–2389.

Zheng, Fangfang, Can Liu, Xiaobo Liu, Saif Eddin Jabari, Liang Lu. 2020. Analyzing the impact of automated vehicles on uncertainty and stability of the mixed traffic flow. *Transportation Research Part C: Emerging Technologies* **112** 203–219.

Zheng, Z. 2014. Recent developments and research needs in modeling lane changing. *Transportation Research Part B: Methodological* **60** 16–32.

Zheng, Z., S. Ahn, D. Chen, J. Laval. 2013. The effects of lane-changing on the immediate follower: Anticipation, relaxation, and change in driver characteristics. *Transportation Research Part C: Emerging Technologies* **26** 367–379.

Zheng, Z., S. Ahn, C. M. Monsere. 2010. Impact of traffic oscillations on freeway crash occurrences. *Accident Analysis & Prevention* **42**(2) 626–636.