



**HAL**  
open science

# Optimization and game-theoretical methods for transportation systems

David Rey

► **To cite this version:**

David Rey. Optimization and game-theoretical methods for transportation systems. Operations Research [math.OA]. Toulouse 3 Paul Sabatier, 2023. tel-04024464

**HAL Id: tel-04024464**

**<https://hal.science/tel-04024464>**

Submitted on 11 Mar 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



UNIVERSITÉ  
TOULOUSE III  
PAUL SABATIER



# Habilitation à Diriger les Recherches

Université Toulouse III Paul Sabatier

École Doctorale Mathématiques, Informatique, Télécommunications  
de Toulouse

---

## Optimization and game-theoretical methods for transportation systems

---

Presentée par

DAVID REY

*SKEMA Business School, Sophia Antipolis*

Spécialité : RECHERCHE OPÉRATIONNELLE

Soutenue à l'École Nationale de l'Aviation Civile (ENAC) le 6 mars 2023  
devant le jury composé de :

LUCE BROTCORNE	Directrice de Recherche, INRIA Lille	Rapporteure
GUGLIELMO LULLI	Professeur, Lancaster University	Rapporteur
MICHAL TZUR	Professeure, Tel Aviv University	Rapporteure
DIDIER AUSSEL	Professeur, Université de Perpignan	Examineur
MARCEL MONGEAU	Professeur, ENAC	Examineur
JEAN-BAPTISTE HIRIART-URRUTY	Professeur Émérite, Université Toulouse III	Président
SONIA CAFIERI	Professeure, ENAC	Marraine

---

---

# Acknowledgements

I would like to warmly thank Luce Brotcorne, Guglielmo Lulli and Michal Tzur for serving as referees for my *Habilitation à Diriger les Recherches* (HDR). I also thank Didier Aussel, Jean-Baptiste Hiriart-Urruty and Marcel Mongeau for participating in my jury. I am extremely thankful to Sonia Cafieri for her continuous support and dedication throughout this journey.

This manuscript is the fruit of several years of exciting research with my Ph.D. students and my peers. I hope I have not forgotten anyone. First, I would like to thank my former and current Ph.D. students at UNSW Sydney for their efforts and commitment: Mikel Barbara, Shantanu Chakraborty, Fernando H. C. Dias, Ahmed. W. Hammad, Xiaotong (Sharon) Dong, Haoning Xi, Xuefen Liu, Mingye Luan, Zahra Nourmohammadi, Aditya Paul, Esta Qiu, Babak Shahbazi and Xiang Zhang.

I was extremely lucky to spend several years at the Research Centre of Integrated Transport Innovation (rCITI) at UNSW Sydney. This is an outstanding research team where I met many friends and colleagues that have contributed to shaping my academic trajectory. Thank you Sylvia Brohl, Vinayak V. Dixit, Lauren M. Gardner, Hanna Gryzbowska, Sisi Jian, Maria Lee, Wei Liu, Mojtaba Maghrebi, Emily Moylan, Divya J. Nair, Ali Najmi, Ali Nezhad, Taha H. Rashidi, Meead Saberi, Changle Song, Upali Vandebona, S. Travis Waller, Kasun P. Wijayarathna, Zhitao Xiong and Aleksa Zlojutro.

I would also like to thank my colleagues and co-authors: Khaled Almi'ani, Hillel Bar-Gera, Romain Billot, Nour-Eddin El Faouzi, Angelo Furno, Elise Henry, Hassan Hijazi, Michael Levin, Julien Monteil and Jean-Luc Ygnace.

I am grateful for the welcome and the support of SKEMA Business School. Thank you Jan Broekaert, Xavier Brusset, Morteza Davari, Faizal Hafiz, Aïda Jebali, Roland Königgruber, Xuefei Lu, Philippe Monin, Davide La Torre and Bernardo K. Pagnoncelli.

Last, I would like to thank Rachel, Elise, Helene, Caroline, Lucas and my parents Claire and Philippe.

---

# Foreword

This *Habilitation à Diriger les Recherches (HDR)* thesis aims to provide an overview my research activities since I defended my Ph.D. in December 2012. I started working in the field of Operations Research (OR) during my Ph.D. which focused on mixed-integer programming methods for air traffic management. After this first research experience, I explored new applications of OR mostly in the field of, but not restricted to, transportation and logistics. This includes notably network design and management, facility location problems, shared and automated mobility, food rescue logistics, incentive mechanisms and pricing.

While diverse in application topics, these studies share common features: they are largely based on optimization and game-theoretical methods ranging from mixed-integer programming to bilevel optimization and mechanism design. This thesis attempts to show selected samples of my contributions in developing such methodologies and their application to targeted transportation systems.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Bilevel optimization problems in transportation networks</b>	<b>5</b>
2.1	Introduction . . . . .	5
2.2	Literature review . . . . .	6
2.3	Computational benchmarking of exact methods for the DNDP . . . . .	7
2.4	Maintenance scheduling in transportation networks . . . . .	13
2.5	Scheduling for road network disaster recovery . . . . .	29
2.6	Conclusion . . . . .	35
<b>3</b>	<b>Aircraft conflict resolution via mixed-integer nonlinear programming</b>	<b>37</b>
3.1	Introduction . . . . .	37
3.2	Literature review . . . . .	38
3.3	Aircraft conflict resolution formulations . . . . .	39
3.4	Exact solution methods for the aircraft conflict resolution problem . . . . .	50
3.5	Robust aircraft conflict resolution problem . . . . .	56
3.6	Numerical experiments . . . . .	61
3.7	Conclusion . . . . .	71
<b>4</b>	<b>Modeling approaches for autonomous intersection management</b>	<b>73</b>
4.1	Introduction . . . . .	73
4.2	Literature review . . . . .	74
4.3	Conflict point formulation . . . . .	77
4.4	Hybrid network traffic control policy . . . . .	83
4.5	Online mechanism design for traffic intersection auctions . . . . .	92
4.6	Conclusion . . . . .	104
<b>5</b>	<b>Perspectives</b>	<b>105</b>
5.1	Methodological perspectives . . . . .	105
5.2	Perspectives in transportation systems . . . . .	109



# Chapter 1

## Introduction

### Motivation

Transportation systems can be viewed as the blood vessels of a society on the grounds that they provide essential routes for the movements of vital materials that keep every organ alive. They enable society to function by facilitating human interaction and developing economies. Transportation occurs at various levels of human society. It can connect people, provide access to facilities, or ship packages to populations in need. Yet, transportation systems almost always lead to negative externalities. With few exceptions, the act of moving is usually not desirable in itself and it is thus natural to optimize transportation with the goal to minimize such negative externalities. The essence of transportation systems is thus to optimize the movement of such commodities, e.g. people across places or goods between locations. The demand for commodities is linked to human behavior, and it is natural to represent transportation users as utility-maximizing agents. This sets the scene for game-theoretical decision-making frameworks where agents interact with each other through a common medium, i.e. a transportation system.

Several areas of research directly or indirectly connect with transportation systems. For instance, civil infrastructure engineering, urban mobility, logistics, supply chain management may all consider—and even aim to improve—a part of a transportation system relevant to their domain of study. Hence, the optimization of transportation systems encompasses a multitude of efforts across various disciplines including mathematics, economics, computer science, ecology, psychology and humanities. From a methodological standpoint, this thesis does little to address the multiple facets of transportation. The scope of this thesis is largely restricted to operations research (OR) methods for transportation systems with an emphasis on mathematical programming—and more specifically mixed-integer programming—based approaches.

OR has a long history of applications in transportation. Several seminal optimization problems in OR are related to transportation. The assignment and its extension commonly known as the *transportation problem* are typical illustrations of decision problems where a solution is said to be optimal if it minimizes the total cost of connecting two groups of agents. The traveling salesman problem is concerned with the minimization of distance traveled. Facility location aims to identify optimal placements of facilities based on their distance to demand nodes. Network design seeks the optimal configuration of a network so as to minimize the total usage costs—often travel related—induced by commodities. Trajectory optimization



deals with the movements of objects through time and space.

In the OR literature, the traditional treatment of the aforementioned decision problems involves a mix of graph theory, combinatorial optimization, algorithm design and mathematical programming; amongst others. The modeling and the analysis of transportation systems often goes beyond this practice by also embracing a more user-oriented perspective that aims to take into consideration user behavior in the decision-making process. For example, the main difference between a classical network design problem and a transportation network design problem is that the former is purely concerned with cost minimization subject to design constraints while the latter incorporates additional network equilibrium elements to capture the long-term behavior of utility-maximizing—or selfish—users. The incorporation of user behavior in classical OR problems can also be viewed as the problem of finding the optimal incentives to achieve a certain pattern which falls in the realm of mechanism design.

This brief analysis highlights the strong links existing between transportation systems and the fields of optimization and game theory. These links stand at the heart of this thesis which contributes to the development of solutions using OR techniques for decision-making problems that arise in transportation systems. It should be highlighted that while this thesis is not dedicated to the study of classical OR problems, several of the problems discussed therein can be viewed as variants of these seminal decision problems.

## Organization of the thesis

This thesis is organized around three main chapters. Each of these chapters addresses a specific decision problem encountered in transportation systems. The chapters are organized in terms of methodological approaches as well as by the time scale of the decision problems addressed. Chapter 2 summarizes some contributions in the field of bilevel optimization and its applications to transportation network design and maintenance scheduling problems. Such decision-making problems can be viewed as *strategic* in that the decisions involved are based on the long-term behavior of transportation users, which represents a network equilibrium. Chapter 3 focuses on mixed-integer nonlinear programming (MINLP) and its applications to aircraft conflict resolution. This is a *tactical* decision-making problem encountered in air traffic control that seeks short-to-medium term actions to avoid collisions and ensure the safe operations of aircraft in congested airspace. Chapter 4 focuses on modeling approaches for autonomous intersection management in road traffic systems. Autonomous intersection management can be viewed as an *operational* decision-making problem that aims to develop solution approaches for enabling connected and automated mobility at traffic intersections. Various methodologies ranging from mixed-integer linear programming (MILP), decentralized control and mechanism design are explored to enable this future mobility paradigm. An overview of each of these three chapters is provided below, along with some context.

Chapter 2 is concerned with network design problems in transportation. Here the time scale is strategic, that is, long-term decisions are sought. This may include, for example, network capacity addition, link maintenance or repair aspect, or even facility location decisions. In contrast to the classical literature on network design problems, however, network design problems in transportation aim to anticipate the reaction of network users which is typically modeled as a Nash equilibrium problem. This sets the frame for a so-called leader-follower game, also known as a Stackelberg game wherein the leader represents the network planner and the follower represents the collective route choice of network users under some equilibrium conditions. It is critical to emphasize that the strategic nature of the problem is fundamental

---

since user equilibrium conditions aim to represent users' learning of network conditions over repeated experiences. This chapter focuses on the analysis and on the development of bilevel optimization methods for transportation network design problems with a focus on discrete problems wherein design decisions are represented by integer variables. An initial analysis of the state-of-the-art methods for solving bilevel optimization problems in transportation networks is conducted. This is followed by the development of a specific method for a multi-period network maintenance scheduling problem based on a branch-and-price approach. An application to network disaster recovery is also briefly presented.

Chapter 3 addresses a more tactical decision-making problem in the field of air traffic control. It focuses on the aircraft conflict resolution problem, which was also the topic of my Ph.D., but examines an alternative modeling and optimization framework. Specifically, this chapter summarizes recent efforts to build on and extend state-of-the-art formulations and exact solution methods for the aircraft conflict resolution problem. Given a set of aircraft trajectories that may lead to conflicts (loss of separation) or even collisions, the aircraft conflict resolution problem aims to find a set of conflict-free aircraft trajectories that ensure that minimal separation requirements are maintained at all times. The difficulty of this problem largely depends on its decision space and on the modeling assumptions made. This chapter tackles the continuous form of the problem where neither time nor control actions are discretized. This variant of the aircraft conflict resolution problem is particularly challenging since controlling aircraft headings involves manipulating trigonometric functions that are nonlinear and also involves Euclidean distance-based separation constraints that are nonconvex. Throughout this chapter, a new formulation based on a reformulation technique involving complex numbers is proposed and shown to be empirically competitive compared to state-of-the-art algorithms. An extension which aims to incorporate the impact of uncertainty on aircraft predicted trajectories in the conflict resolution problem is also presented. Initial results demonstrate the potential of robust optimization to handle such uncertainty.

Chapter 4 targets an operational problem in the context of autonomous intersection management. Traffic intersections are major bottlenecks of urban transportation networks. Autonomous intersection management is an emerging urban mobility paradigm under which traffic intersections behave autonomously in a signal-free fashion. In this context, users are assumed to be able to reserve space-time trajectories through intersections upon approach. This chapter explores modeling and optimization approaches for autonomous intersection management based on mixed-integer linear programming. A single intersection is first modeled and this formulation is then embedded in a decentralized framework to consider a network of intersections. A stability analysis is conducted to show how a modified max-pressure control policy can accommodate stochastic demands. Autonomous intersection management also opens the door to auction-based mechanisms which hold the potential to improve social welfare by allowing users to bid for priority service. This chapter reports on advances in the application of online mechanism design for developing incentive-compatible mechanisms for traffic intersection auctions. The problem set forth is the design of an online payment mechanism that is able to incentivize utility-maximizing users to reveal their true preferences upon bidding for service. To this end, two Markov chain models are proposed to estimate users' expected waiting time. These models are incorporated into an online payment mechanism that is shown to be incentive-compatible in the dynamic sense. Numerical experiments are reported to show that, in contrast, static incentive-compatible mechanisms (i.e. that do not account for future arrivals) may fail to ensure truthful user behavior. This motivates the necessity to develop online mechanisms for such time-paced game-theoretic applications in transportation systems.

Perspectives are provided in a Chapter 5. Research directions that may build on and extend this thesis are outlined. In addition, an attempt is made to discuss emerging trends in optimization and game theory and their potential to improve transportation systems.

## Other contributions

Throughout my academic career, I have had the opportunity to work on several interesting topics in the field of transportation and logistics, but also in other fields of research such as construction engineering and epidemic modeling. A brief overview of my contributions other than those presented in the core chapters of this thesis is provided hereafter.

At the time I completed my Ph.D., research on shared mobility systems was in great demand. This led me to work on one-way carsharing systems based on MILP formulations [Jian et al., 2016, 2019]. I participated in a couple of studies on rideharing systems that involved real-time computing [Najmi et al., 2017], as well as simulation-based optimization to identify user subsidies in a discrete choice modeling context [Song et al., 2021]. I had the insightful opportunity to participate in an experimental economics study on the design of lottery-based incentive mechanisms for promoting offpeak travel [Rey et al., 2016a]. In a more OR context, we explored the design of demand-aware park-and-ride mobility systems [Henry et al., 2022]. Motivated by industry collaborations, I also had the opportunity to work on vehicle routing problems: first in the context of food rescue and logistics where fairness considerations arise [Nair et al., 2016, 2017, Rey et al., 2018], and later in the context of rich, utility-aware dial-a-ride problems [Dong et al., 2020, 2022].

In the same vein as the works summarized in Chapter 2, I participated in several studies on network design problems, some of which involved bilevel optimization formulations and solution methods. In a series of studies, I contributed to study the impact of electric vehicles and the placement of charging infrastructure on network performance [Zhang et al., 2018, 2019a,b]. The impact of facility location decisions was also studied in a sustainability context using bilevel optimization approaches [Hammad et al., 2017b,a]. Other network design studies focused on transit network with elastic demand [Hossein Rashidi et al., 2016], school network capacity optimization [Barbara et al., 2021] and freeway network design in a mixed automated mobility context [Chakraborty et al., 2021].

In other disciplines, I have developed OR techniques to solve decision-making problems in civil engineering. This includes developing new MINLP formulations and algorithms for construction site layout planning problems [Hammad et al., 2016a,b, 2017c], as well as new modeling approaches for job allocation in construction organizations [Shahbazi et al., 2019]. I started working on outbreak modeling and control nearly a decade ago by contributing to risk analysis and modeling [Gardner et al., 2014] and also by proposing combinatorial optimization approaches to solve a variant of the Steiner tree problem [Rey et al., 2016b]. Other efforts in this research direction have been focused on developing multiscale network models to evaluate and optimize the impact of control strategies for mitigating global outbreaks [Chen et al., 2016, 2017, Zlojutro et al., 2019].

The rest of the thesis follows the plan outlined above which includes three core chapters and a final chapter wherein research perspectives are discussed. For conciseness, all mathematical proofs of propositions, lemmas and theorems are omitted, and readers are referred to the associated publications for these details.

## Chapter 2

# Bilevel optimization problems in transportation networks

This chapter is based on [Rey et al. \[2019\]](#), [Rey \[2020\]](#), [Rey and Bar-Gera \[2020\]](#).

### 2.1 Introduction

We first consider the discrete network design problem (DNNDP) which is a challenging problem in transportation, introduced by [Leblanc \[1975\]](#). The DNNDP can be formulated as a bilevel optimization problem where the leader problem aims to identify the optimal network design to minimize network travel time and the follower problem represents network users' reaction, typically as a static traffic assignment problem (TAP) under user equilibrium [[Wardrop, 1952](#)].

The DNNDP can be defined on a network with nodes  $N$  and directed links  $A$  as a multi-commodity network flow problem with nonlinear link travel time functions. Let  $D$  be the set of destination nodes and  $d_{is}$  be the demand from node  $i \in N$  to destination node  $s \in D \subseteq N$ . If the pair  $(i, s)$  is not an Origin-Destination (OD) pair in the network then  $d_{is} = 0$  and to ensure flow conservation we set  $d_{ss} = -\sum_{i \in N} d_{is}$ . We denote  $x_{ij,s}$  the flow of on link  $(i, j) \in A$  travelling to destination  $s \in D$ , and  $x_{ij}$  the total flow on link  $(i, j) \in A$ . Let  $t_{ij}$  represent the travel time on link  $(i, j) \in A$ , typically modelled as a strictly convex function of the total link flow  $x_{ij}$  to ensure the uniqueness of the equilibrium link flows. Let  $A_1$  be the set of existing links and  $A_2$  be the set of candidate links to improve the network,  $A = A_1 \cup A_2$ . For each link  $(i, j) \in A_2$ , let  $g_{ij}$  be the cost of adding this link to the network and let  $y_{ij} \in \{0, 1\}$  be the variable representing this choice. Let  $B$  be the available budget for optimization. In the resulting formulation DNNDP,  $\mathcal{L}$  is the *leader* problem and  $\mathcal{F}$  is the *follower* problem.

The leader problem  $\mathcal{L}$  aims to minimize the total system travel time (TSTT) defined as the sum of  $x_{ij}t_{ij}(x_{ij})$  over all links, subject to a budget constraint capturing the cost of link addition decisions  $\mathbf{y}$ —hereby referred to as the leader variable. The link flow pattern variable  $\mathbf{x} = [x_{ij}]_{(i,j) \in A}$  is optimized in the follower problem  $\mathcal{F}$ , which is the traditional link-based TAP formulation under UE [[Beckmann et al., 1956](#), [Leblanc, 1975](#), [Magnanti and Wong, 1984](#)]. The impact of the leader variable  $\mathbf{y}$  in the follower is achieved through the constraint  $x_{ij} \leq y_{ij}M$  wherein  $M$  is an upper-bound on the total link flow  $x_{ij}$ .

$$\begin{aligned}
(\mathcal{L}) \min_{\mathbf{y}} \quad & \sum_{(i,j) \in A} x_{ij} t_{ij}(x_{ij}), \\
\text{s.t.} \quad & \sum_{(i,j) \in A_2} y_{ij} g_{ij} \leq B, \\
& y_{ij} \in \{0, 1\}, \quad \forall (i, j) \in A_2, \\
(\mathcal{F}) \quad \mathbf{x} \in \arg \min_{\mathbf{x}} \quad & \sum_{(i,j) \in A} \int_0^{x_{ij}} t_{ij}(v) dv, \\
\text{s.t.} \quad & \sum_{j \in N: (i,j) \in A} x_{ij,s} - \sum_{j \in N: (j,i) \in A} x_{ji,s} = d_{is}, \quad \forall i \in N, \forall s \in D, \\
& \sum_{s \in D} x_{ij,s} = x_{ij}, \quad \forall (i, j) \in A, \\
& x_{ij} \leq y_{ij} M, \quad \forall (i, j) \in A_2, \\
& x_{ij,s} \geq 0, \quad \forall (i, j) \in A, \forall s \in D.
\end{aligned}$$

(DNDP)

The goal of this chapter is threefold. A summary of the state-of-the-art of exact methodologies for the DNDP is first presented and a computational benchmark is conducted to highlight existing computational bottlenecks (Sections 2.2-2.3.3). An application of the DNDP for a network maintenance scheduling problem is then presented (Section 2.4) along with its extension to a network disaster recovery problem (Section 2.5).

## 2.2 Literature review

Several efforts have been proposed to solve the DNDP to global optimality. The seminal work of [Leblanc \[1975\]](#) introduced a Branch-and-Bound (B&B) algorithm for the DNDP which used the system-optimum (SO) relaxation of the TAP to find lower bounds. This relaxation requires fixing all unfixed  $\mathbf{y}$  variables to 1 to avoid Braess' paradox effects [[Braess, 1968](#)], and may thus lead to poor lower bounds. [Gao et al. \[2005\]](#) introduced a mixed-integer nonlinear programming (MINLP) approach based on generalized Benders' decomposition. The authors proposed a benchmark network with a single origin-destination (OD) pair, 12 nodes, 17 existing links and 6 candidate links, commonly referred to as *Gao's instance*. They also provided results for an instance based on the Sioux Falls network which contains 24 zones and nodes, 76 existing links and 5 candidate projects (which may involve one or more links) but computation time is not reported. It was noted by [Farvareh and Sepehri \[2013\]](#) that despite claims of global optimality, the proposed approach of [Gao et al. \[2005\]](#) may converge to local optimums.

Bilevel optimization problems with a convex follower problem can be reformulated into single-level formulations by representing the follower problem using its Karush-Kuhn-Tucker (KKT) conditions and introducing binary variables to model complementarity slackness conditions [[Bard, 2013](#)]. This applies to the DNDP since the follower problem is a TAP which can be represented as a convex nonlinear program (NLP) [[Beckmann et al., 1956](#)]. [Farvareh and Sepehri \[2011\]](#) proposed a mixed-integer linear programming (MILP), single-level reformulation of the DNDP obtained using piecewise linear approximations of link travel time functions. Numerical results are only reported for Gao's instance and an extended network based on Gao's instance which contains 16 nodes, 17 existing links and 25 candidate links, and

---

2 OD pairs. Farvaresh and Sepehri [2013] proposed a more scalable approach which extends the seminal B&B of Leblanc [1975] by solving the SO relaxation of the DNDP—instead of the TAP—as a mixed-integer nonlinear program (MINLP) to provide tighter lower bounds. This extended B&B algorithm was shown capable to solve instances with up to 100 nodes, 387 links and 15 candidate projects.

Luathep et al. [2011] proposed an SO-relaxation based approach wherein variational inequalities (VIs) are iteratively added to ensure UE conditions. Numerical results are reported for Gao’s instance, as well as for an instance on Sioux Falls network with 5 candidate projects, and a variation with 10 candidate projects for a mixed (discrete and continuous) case. Fontaine and Minner [2014] proposed a Benders’ decomposition of a single-level MILP reformulation of a linearized DNDP with piecewise linear approximations of link travel time functions. Using this decomposition, the authors are able to solve this linearized DNDP on a network of Berlin’s centre containing 36 zones, 398 nodes, 871 links and 10 candidate projects.

Wang et al. [2013] studied an extended DNDP where the capacity of candidate links must also be decided. They present two global optimization algorithms which are based on the SO-relaxation of the DNDP, as well as a dynamic outer approximation of link travel time functions to derive lower bounds. Numerical results for a Sioux Falls network instance with up to 10 candidates links and 3 levels of capacity are reported. Wang et al. [2015] extended the DNDP to a variant involving both discrete and continuous decisions variables for adding links and determining their capacity, respectively. The authors used the VI formulation of Luathep et al. [2011] but combine it with an outer approximation of link travel time functions. Numerical results are only reported for Gao’s instance. Bagloee et al. [2017] proposed a B&B algorithm which uses a generalized Benders’ decomposition approach to solve the SO-relaxation of the DNDP at each node of the tree. Results on Sioux Falls and Winnipeg’s network are reported with up to 20 candidate projects.

Although several efforts have been proposed to solve the DNDP or a linear approximation of this problem, the literature on exact or near-exact approaches remains scarce. Further, there is no reference datasets for benchmarking solution methods which undermines the research on this challenging bilevel optimization problem.

## 2.3 Computational benchmarking of exact methods for the DNDP

In this section, we discuss formulations and algorithms to solve the DNDP to optimality. We start by discussing the role of link travel time functions and attempt to categorize existing solution methodologies for the DNDP.

A major computational challenge inherent to the DNDP is the nonlinearity of the link travel time functions  $t_{ij}(x_{ij})$ , typically Bureau of Public Roads (BPR) functions. The consensus on link travel time functions is to use strictly convex functions of the form  $t_{ij}(x_{ij}) = T_{ij} + c_{ij}x_{ij}^{e_{ij}}$  where  $T_{ij}$  represents link free-flow travel time,  $c_{ij}$  is a coefficient which captures link capacity and  $e_{ij} \geq 1$  is an exponent; to ensure uniqueness of the link flow solution. The vast majority of existing data for link travel time functions of this form assumes  $e_{ij} = 4$  for all  $(i, j) \in A$ . Link travel time functions are present in the objective function of both leader and follower problems in the form  $x_{ij}t_{ij}(x_{ij})$  which involves the nonlinear term  $x_{ij}^{e_{ij}+1}$ .

This term, and sometimes term  $x_{ij}^{e_{ij}}$ , have been either approximated using piecewise linear functions [Farvareh and Sepehri, 2011, Luathep et al., 2011, Fontaine and Minner, 2014] or using linear outer approximation schemes [Wang et al., 2013, 2015].

Piecewise linear approximations require discretizing the domain of link flow variable  $\mathbf{x}$  into  $m$  disjunctive segments and requires  $\mathcal{O}(m)$  auxiliary variables to adjust the approximated nonlinear term according to the segment activated. The domain of link flows is typically defined as  $x_{ij} \in [0, \bar{x}_{ij}]$  where  $\bar{x}_{ij}$  is the maximum flow that can travel on link  $(i, j)$ . Determining tight values for the upper bound  $\bar{x}_{ij}$  is not trivial and using a conservative value such as the total demand of the network may lead to poor approximation schemes. However, since the nonlinear terms  $x_{ij}^{e_{ij}}$  and  $x_{ij}^{e_{ij}+1}$  are convex on  $[0, \bar{x}_{ij}]$ , piecewise linear approximations of these terms do not require any integer variables, as noted by Farvareh and Sepehri [2011] and in Fontaine and Minner [2014]. Hence, while piecewise linear approximation may require a significant amount of additional variables and linear constraints to achieve high-quality solutions, their impact on computational performance can be moderate.

Outer approximations of link travel time functions attempt to derive compact convex envelopes of nonlinear terms  $x_{ij}^{e_{ij}}$  and  $x_{ij}^{e_{ij}+1}$  using integer-linear constraints. These schemes are notoriously computationally challenging but guarantee that the original (non-linearized) DNDP is solved to optimality. Only in rare cases have link travel time functions been incorporated without any direct approximation scheme [Farvareh and Sepehri, 2013, Bagloee et al., 2017].

We next discuss exact methodologies for the DNDP which have adopted one of the above approaches to handle link travel time functions. Since piecewise linear approximations of link travel time functions cannot be guaranteed to converge to optimal solutions unless  $m \rightarrow \infty$ , such methodologies are referred to as linearized DNDP approaches.

### 2.3.1 SO-relaxation based approaches

Despite its potential weak initial lower bound, the SO-relaxation of the DNDP has emerged as a powerful mechanism to conceive iterative solution methods for the DNDP. The SO-relaxation of the DNDP is a single-level optimization problem which ignores the follower objective function as summarized in SO-DNDP.

$$\begin{aligned}
\min_{\mathbf{x}, \mathbf{y}} \quad & \sum_{(i,j) \in A} x_{ij} t_{ij}(x_{ij}), \\
\text{s.t.} \quad & \sum_{(i,j) \in A_2} y_{ij} g_{ij} \leq B, \\
& \sum_{j \in N: (i,j) \in A} x_{ij,s} - \sum_{j \in N: (j,i) \in A} x_{ji,s} = d_{is}, & \forall i \in N, \forall s \in D, \\
& \sum_{s \in D} x_{ij,s} = x_{ij}, & \forall (i,j) \in A, \\
& x_{ij} \leq y_{ij} M, & \forall (i,j) \in A_2, \\
& y_{ij} \in \{0, 1\}, & \forall (i,j) \in A_2, \\
& x_{ij,s} \geq 0, & \forall (i,j) \in A, \forall s \in D.
\end{aligned} \tag{SO-DNDP}$$

Solving SO-DNDP yields a lower bound on the optimum of DNDP which can serve as starting point of iterative schemes. Leblanc [1975] was the first to propose a customized

B&B algorithm which branches on unfixed  $\mathbf{y}$  variables to yield subproblems that refine the initial lower bound. At each node of the B&B tree, [Leblanc \[1975\]](#) approach further relaxed [SO-DNDP](#) by temporarily fixing all unfixed  $\mathbf{y}$  variables and solving the resulting SO-TAP using a convex programming algorithm, e.g. Frank-Wolfe. This B&B scheme was refined and extended by [Farvareh and Sepehri \[2013\]](#) which proposed to solve [SO-DNDP](#) at each node of the tree using a global MINLP algorithm. Upper bounds are obtained by solving the TAP under UE conditions after obtaining  $\mathbf{y}$  from [SO-DNDP](#).

[Bagloee et al. \[2017\]](#) proposes a B&B algorithm which uses generalized Benders' decomposition approach to solve the SO-relaxation of the DNDP at each node of the tree. The proposed algorithm is parameterized using a value (denoted  $\alpha$ ) which influences the solution of the TAP solved therein. The algorithm is not guaranteed to find global optimal solutions if  $\alpha \neq 1$ .

[Luathep et al. \[2011\]](#) proposed a cutting-plane algorithm to complement formulation [SO-DNDP](#) with VIs that characterizes UE conditions [[Dafermos, 1980](#)]. Let  $\Omega_{\mathcal{F}}$  be the polyhedron of the feasible region of the follower problem  $\mathcal{F}$  and let  $\mathbf{x}^*$  be the UE link flow pattern, the following VI holds:

$$\sum_{(i,j) \in A} t_{ij}(x_{ij}^*)(x_{ij} - x_{ij}^*) \geq 0, \quad \forall \mathbf{x} \in \Omega_{\mathcal{F}}. \quad (2.1)$$

The approach of [Luathep et al. \[2011\]](#) is based on the observation that VIs corresponding to the set of extreme points of the TAP polyhedron is sufficient and necessary to characterize UE conditions. At each iteration, [SO-DNDP](#) is solved with a restricted set of VIs and shortest path problems are solved for each OD pair to identify violated VIs of the form (2.1), which are then added as cuts to [SO-DNDP](#) until none can be found.

[Wang et al. \[2013\]](#) observed that it was sufficient to iteratively forbid the last  $\mathbf{y}$  solution found to construct a UE solution from the SO-relaxation of the DNDP. At each iteration, the proposed algorithm first solves [SO-DNDP](#) before calculating the UE cost of the corresponding  $\mathbf{y}$  solution using any TAP algorithm to obtain an upper bound. The process is then repeated by adding an interdiction constraint to identify the next-best SO network design. Let  $\mathbf{y}^k$  be the optimal solution of [SO-DNDP](#) at iteration  $k$ . At iteration  $k + 1$ , [SO-DNDP](#) is solved with the interdiction constraint:

$$\sum_{(i,j) \in A_2} \left( y_{ij}(1 - y_{ij}^n) + (1 - y_{ij})y_{ij}^n \right) \geq 1, \quad \forall n \in \{1, \dots, k\}. \quad (2.2)$$

The iterative process is repeated until the resulting objective value is greater than or equal to the upper bound.

### 2.3.2 KKT conditions-based approaches

Since the TAP can be formulated as a convex optimization problem, an intuitive approach to solve the DNDP is to replace the follower problem  $\mathcal{F}$  by its KKT conditions. [Farvareh and Sepehri \[2011\]](#) proposed such a direct approach wherein the KKT conditions of the TAP are represented with auxiliary variables. Let  $\pi_{is} \geq 0$  be the travel time or path travel time from node  $i \in N$  to destination  $s \in D$ . The UE conditions of the TAP require:



Paper	Method	Link travel time function approximation	Type(s) of optimization problem solved
Leblanc [1975]	SO-relaxation and B&B	N/A	Convex NLP
Gao et al. [2005]	Generalized Benders' decomposition	N/A	MINLP and Convex NLP
Farvareh and Sepehri [2011]	KKT conditions	Piecewise linear	MILP
Luathep et al. [2011]	SO-relaxation and VIs	Piecewise linear	MILP and LP
Farvareh and Sepehri [2013]	SO-relaxation and B&B	N/A	MINLP and Convex NLP
Wang et al. [2013]	SO-relaxation and interdiction cuts	Outer linear	MILP and Convex NLP
Fontaine and Minner [2014]	KKT conditions and Benders' decomposition	Piecewise linear	MILP and LP
Wang et al. [2015]	SO-relaxation and VIs	Outer linear	MILP and LP
Bagloee et al. [2017]	B&B and Generalized Benders' decomposition	N/A	MINLP and Convex NLP

Table 2.1: Summary of exact methodologies for the DNDP.

$$t_{ij}(x_{ij}) - \pi_{is} + \pi_{js} \geq 0, \quad \forall (i, j) \in A, \forall s \in D, \quad (2.3a)$$

$$x_{ij}(t_{ij}(x_{ij}) - \pi_{is} + \pi_{js}) = 0, \quad \forall (i, j) \in A, \forall s \in D. \quad (2.3b)$$

The complementarity slackness conditions (2.3b) is nonlinear and typical mathematical programming approaches require additional binary variables to obtain an integer-linear form suitable for MILP. Farvareh and Sepehri [2011] observe that this surplus of binary variables yields considerable computational challenge. Fontaine and Minner [2014] proposed an alternative approach to incorporate the KKT conditions of the TAP which consists of replacing the objective of the follower problem  $\mathcal{F}$  with a primal-dual constraint which requires a null duality gap. This method is applied to a linearized DNDP, which results in a single-level MILP composed of leader, primal follower and dual follower constraints and variables, with the addition of the primal-dual constraint. This MILP does not require any additional binary variable and the authors propose a Benders' decomposition approach.

A summary of the solution methods discussed is provided in Table 2.1. Most of these methodologies can be adapted to work with piecewise linear or outer- approximations of link travel time functions. While the former only solves the so-called linearized DNDP, piecewise linear approximations are often capable of identifying global optimal solutions of the original problem and are, at the same time, substantially easier to implement.

### 2.3.3 Numerical experiments for the linearized DNDP

We implement three solution methods for the linearized DNDP: i) the B&B algorithm of Farvareh and Sepehri [2013] which is an extension of Leblanc [1975]—SOBB; ii) the SO-relaxation based algorithm of Wang et al. [2013] with interdiction cuts—SOIC, and iii) the primal-dual formulation of Fontaine and Minner [2014] as a single-level MILP (without Benders' decomposition)—MKKT. All three solution methods, SOBB, SOIC and MKKT, are implemented using the piecewise linear approximation of link travel time functions proposed

by Farvaresh and Sepehri [2011]. Hence, in all cases, the optimization problems solved are MILPs and LPs (TAPs within SOBB and SOIC are solved in their linearized form). This differs from the original implementation of Farvaresh and Sepehri [2013] and Wang et al. [2013], wherein outer approximation schemes are used. In addition, the method MKKT is implemented as direct MILP approach unlike the Benders' decomposition scheme proposed in Fontaine and Minner [2014]. To measure the quality of the approximated solutions, the flow pattern corresponding to the best (lowest leader objective value)  $\mathbf{y}$  solution among all three methods is calculated by solving the TAP as a convex problem.

All methods are implemented in Python. All MILPs and LPs are solved using CPLEX 12.8. Convex TAPs are solved using the Pyomo module and IPOPT. All solution methods were tested and implemented on the same Windows 7 machine with 16Gb of RAM and a CPU of 2.7Ghz, in a single-thread mode with a time limit of 10 minutes. The upper bound on link flows  $\bar{x}_{ij}$  was set to  $1e^5$  and this value is also used for  $M$ . The number of segments used in the piecewise linear approximations of link travel time functions is  $m = 100$ . A scaling factor of  $1e^{-3}$  is used to scale travel demand and link capacities as it was found to improve computational performance. For reproducibility purposes, all implemented optimization formulations and codes are provided at <https://github.com/davidrey123/DNDP>.

New instances for the DNDP based on Sioux Falls network<sup>1</sup> have been designed to test the implemented solution methods. For these new instances, a total of 15 pairs of new links (total of 30 links) have been created, along with their performance characteristics and addition costs. Among these new links, 5 of the 15 pairs are identical to the ones used by Luathep et al. [2011]. A total of 20 DNDP instances have been created, 10 of these contain 10 new links (5 pairs) and are named SF\_DNDP\_10, and the remaining contain 20 new links (10 pairs) and are named SF\_DNDP\_20. In all our numerical experiments, link addition variables are not paired, i.e. the size of  $A_2$  is equal to the total number of new links which may or may not be added in pairs; hence instances SF\_DNDP\_10 require 10 binary variables and instances SF\_DNDP\_20 require 20 binary variables. All instance data is provided in the public repository available at <https://github.com/davidrey123/DNDP> in the TNTP format. An extra column has been appended to provide link cost data ( $g_{ij}$ ) and a value of 0 indicates that the link is part of the original network (before optimization).

Two numerical experiments are conducted on instances SF\_DNDP\_10 and SF\_DNDP\_20: a budget sensitivity analysis wherein the available budget  $B$  is equal to 25%, 50% and 75% of the total cost  $\sum_{(i,j) \in A_2} g_{ij}$ , and a demand sensitivity analysis wherein the budget is fixed to 50% of the total cost and travel demand is set to 50%, 100% and 150% of the base demand. The average performance of all three methods implemented, i.e. SOBB, SOIC and MKKT, is reported in Tables 2.2 and 2.3 for budget and demand sensitivity analysis, respectively.

We find that 10-project instances SF\_DNDP\_10 can present considerable computational challenges, especially for SO-relaxation based methods when the budget available is relatively high ( $B_{\%} = 75\%$ ). Increasing the number of candidate links to 20 results in substantial difficulties in most cases, regardless of the method used, since most of the time the methods implemented were not able to converge within the 10-minute time limit. The results of the budget sensitivity analysis on SF\_DNDP\_10 instances show that MKKT is often faster than other methods at medium and high budgets ( $B_{\%} = 50\%, 75\%$ ). In turn, SOIC and SOBB tend to time-out more frequently at these budgets, which is likely due to the more substantial

<sup>1</sup>These instances are based on Sioux Falls network as available at <https://github.com/bstabler/TransportationNetworks>

Method	SF_DNDP_10									SF_DNDP_20								
	$B_{\%} = 25\%$			$B_{\%} = 50\%$			$B_{\%} = 75\%$			$B_{\%} = 25\%$			$B_{\%} = 50\%$			$B_{\%} = 75\%$		
	AT	AG	TO	AT	AG	TO	AT	AG	TO	AT	AG	TO	AT	AG	TO	AT	AG	TO
SOBB	56.8	0.00	0	333.8	0.39	20	484.9	1.75	60	600	2.48	100	600	4.76	100	600	6.07	100
SOIC	33.8	0.00	0	283.4	0.31	20	357.5	0.69	40	600	3.76	100	600	4.44	100	600	5.26	100
MKKT	57.2	0.00	0	170.6	0.00	0	155.8	0.00	0	600	5.68	100	600	6.46	100	600	6.21	100

Table 2.2: Budget sensitivity experiment. Methods SOBB, SOIC and MKKT are implemented for instances SF\_DNDP\_10 and SF\_DNDP\_20 for a budget  $B$  equal to  $B_{\%} = 25\%$ ,  $50\%$  and  $75\%$  of the total cost  $\sum_{(i,j) \in A_2} g_{ij}$ . The time limit is 10 minutes. AT is the average runtime in seconds, AG is the average relative optimality gap upon termination in % and TO is the proportion of time-outs in %.

Method	SF_DNDP_10									SF_DNDP_20								
	$D_{\%} = 50\%$			$D_{\%} = 100\%$			$D_{\%} = 150\%$			$D_{\%} = 50\%$			$D_{\%} = 100\%$			$D_{\%} = 150\%$		
	AT	AG	TO	AT	AG	TO	AT	AG	TO	AT	AG	TO	AT	AG	TO	AT	AG	TO
SOBB	537.7	0.54	80	336.6	0.40	20	79.2	0.00	0	600	2.38	100	600	4.75	100	432.2	0.24	40
SOIC	513.8	0.56	80	284.6	0.34	20	14.6	0.00	0	600	2.14	100	600	4.44	100	254.9	0.12	30
MKKT	87.9	0.00	0	172.1	0.00	0	65.9	0.00	0	600	2.58	100	600	6.46	100	519.8	1.43	50

Table 2.3: Demand sensitivity experiment. Methods SOBB, SOIC and MKKT are implemented for instances SF\_DNDP\_10 and SF\_DNDP\_20 for a budget  $B$  equal to 50% of the total cost  $\sum_{(i,j) \in A_2} g_{ij}$  and a demand of  $D_{\%} = 50\%$ ,  $100\%$  and  $150\%$  of the base demand. The time limit is 10 minutes. AT is the average runtime in seconds, AG is the average relative optimality gap upon termination in % and TO is the proportion of time-outs in %.

gap between UE and SO flow patterns when a high budget is available. At low budget ( $B_{\%} = 25\%$ ), SOIC outperforms other methods. Instances SF\_DNDP\_20 result systematically in time-outs and the average optimality gap upon termination tends to increase with the budget. The performance of all three methods is of the same order of magnitude, with optimality gaps in the range of 3%-6%. The demand sensitivity analysis highlights the good performance of MKKT at low demand ( $D_{\%} = 50\%$ ) compared to other methods on SF\_DNDP\_10 instances. Increasing the demand to 150% of the base demand yields substantially easier problems, notably for SF\_DNDP\_20 instances, and SOIC is found to outperform other methods in terms of number of time-outs. To assess the quality of the linear approximation of link travel time functions, we compare the network travel time obtained by solving the TAP in its convex NLP form for the best  $\mathbf{y}$  solution found among all three methods. The deviation observed is on average of 0.28% with a standard deviation of 0.25%.

Overall, this study shows that even medium-size instances for the linearized DNDP can present considerable computational challenges. The numerical experiments revealed that some 10-project instances and most 20-project instances could not be solved within the imposed time limit (10 minutes). Although this time limit is relatively small, all methods were implemented using a piecewise linear approximation of link travel time functions which is expected to outperform outer approximations approaches.

---

## 2.4 Maintenance scheduling in transportation networks

Road maintenance and repair operations are conducted regularly in major cities. This is due to the numerous links that connect the transport network as well as to the high travel demand which contributes in deteriorating road quality. In addition, the organization of urban areas often relies on the road network to incorporate utility networks, such as electricity, telecommunications and water networks. As a result, lane or road closures—hereby referred to as maintenance projects—are frequent and can lead to significant delays within the city if they are conducted without any coordination across the network. In this section, we address the problem of determining an optimal schedule for the conduction of maintenance projects in urban transportation networks. The impact of link capacity reductions induced by maintenance projects onto network performance is measured by the TSTT. We propose a novel solution method for the Network Maintenance Scheduling Problem (NMSP) that finds a schedule which minimizes the TSTT over a planning period subject to project schedule constraints.

The first effort to account for traffic dynamics in network maintenance scheduling can be attributed to [Chang et al. \[2001\]](#) who proposed a tabu search approach to solve the optimization problem. The authors implement their approach on a 6-project scenario derived from the Columbus, Ohio network which comprises 5,441 nodes and 12,658 links. All projects are of equal duration and at most 2 projects can be conducted in parallel (i.e. 2 teams are available), thus considerably reducing the number of possible patterns. Subsequent works have mainly focused on meta-heuristics approaches to solve this bilevel scheduling problem. [Ng et al. \[2009\]](#) used a cell transmission model to account for users' reaction and proposed a genetic algorithm-based solution method. The authors implemented the proposed methodology on a test network with 13 nodes and 24 links where each link can be eligible for yearly maintenance over a 3-year planning period. [Lee \[2009\]](#) considered a framework that combines ant colony optimization with a microscopic traffic simulation model to represent traffic delays caused by maintenance projects. The framework is implemented on a realistic case study derived from a sewer system construction project on a network with 292 nodes and 178 links. A total of 35 maintenance projects are considered with a maximum of 5 teams over a planning period of 270 days. A similar approach is used by [Xiong et al. \[2014\]](#) where an ant colony optimization approach is used along with a static traffic assignment model to account for users' reaction. The proposed approach is tested on a 6-project scenario derived from the Chicago Sketch network which includes 933 nodes and 2,950 links over a planning period of 36 months. [Chu and Chen \[2012\]](#) proposed a hybrid dynamic model to simulate threshold-based maintenance based on pavement life-cycle. The bilevel optimization formulation is solved using a modified tabu search algorithm on a network with 70 nodes and 270 links. A 10-year planning period is considered over which all links are eligible for maintenance operations. [Zheng et al. \[2012\]](#) introduced a detailed modeling framework for representing maintenance project features, such as day/night time construction strategies, and used an approximation of the UE travel times to account for traffic delays. [Fontaine and Minner \[2017\]](#) proposed an exact approach for the bilevel network maintenance planning problem which is based on a Benders' decomposition method of a single-level reformulation of the original problem. The optimality conditions of the TAP are used as equilibrium constraints. In their formulation, they assume that a budget is available for maintenance costs. Numerical results on the Sioux Falls network and an extended Sioux Falls network of 100 nodes of 317 links are reported for planning periods of 10 and 8 time steps, respectively.

We propose a new bilevel programming formulation and develop a solution methodol-

ogy based on maintenance project patterns enumeration. This representation of the NMSP involves a number of decision variables which grows exponentially with the number of maintenance projects. We propose a new Branch-and-Price (B&P) algorithm to solve this bilevel problem. Namely, we introduce customized branching and bounding rules, and develop a Column Generation (CG) procedure to solve the Linear Programming (LP) relaxation of a restricted problem at each node of the tree. Problem-specific pricing algorithms are proposed to iteratively introduce profitable project patterns in the formulation. In addition, we introduce a statistical regression model to approximate congestion effects and derive approximated lower bounds within the optimization framework.

### 2.4.1 Network maintenance scheduling

The coordination of maintenance projects within a transport network must satisfy operational constraints (e.g. spatial and temporal restrictions) and should have a minimal impact onto network performance. Specifically, we focus on a planning period (e.g. one year) during which all projects must be completed in consecutive time periods (e.g. one week). The consecutiveness assumption is consistent with the operations of traffic management centers in which road maintenance projects are conducted without interruption from their starting date. It is important to stress that we are mostly concerned with maintenance projects that have an impact during peak-hour traffic. Strictly overnight work and other off-peak road maintenance projects with no impact on traffic conditions may be scheduled without accounting for their influence on network performance. Hence, in this section, we focus on maintenance projects that span one or more time periods and reduce the capacity of the associated links during the entire duration of the project.

The proposed network maintenance scheduling problem can be defined on a directed graph  $G = (N, A)$  where the set of nodes  $N$  represents travel demand centroids or intersections, and the set of arcs  $A$  represents links among nodes. We assume that the set of projects is known at the beginning of the planning horizon and we assume that the recovery time of each project is known deterministically. The network travel time at each time period is determined based on network link states and its associated traffic assignment pattern determined under the UE principle. We denote  $Q_w$  the demand for a specific OD pair  $w \in W$ ,  $\Pi_w$  its set of paths and  $f_k^w$  the flow on path  $k \in \Pi_w$ . In addition, we denote  $\delta_{a,k}^w$  the fixed binary value indicating if link  $a \in A$  belongs to path  $k \in \Pi_w$  (1) or not (0). Finally, we denote  $x_a$  the flow on link  $a \in A$ . Let  $m_a$  denote the state of link  $a \in A$ . The TAP corresponding to a network state vector  $\mathbf{m}$  is summarized in (2.4). Note that bold symbols are used to denote vectors throughout the chapter.

$$\mathbf{x}^*(\mathbf{m}) \in \arg \min \quad \sum_{a \in A} \int_0^{x_a} t_a(v, m_a) dv, \quad (2.4a)$$

$$\text{s.t.} \quad \sum_{k \in \Pi_w} f_k^w = Q_w, \quad \forall w \in W, \quad (2.4b)$$

$$\sum_{w \in W} \sum_{k \in \Pi_w} f_k^w \delta_{a,k}^w = x_a, \quad \forall a \in A, \quad (2.4c)$$

$$f_k^w \geq 0, \quad \forall w \in W, k \in \Pi_w, \quad (2.4d)$$

$$x_a \geq 0, \quad \forall a \in A. \quad (2.4e)$$

---

Observe that the link flow pattern  $\mathbf{x}^*(\mathbf{m})$  depends on the network state vector  $\mathbf{m}$ . The network travel time under UE, also known as the total system travel time, corresponding to network state  $\mathbf{m}$ , denoted  $TSTT(\mathbf{m})$ , is thus calculated as follows:

$$TSTT(\mathbf{m}) = \mathbf{x}^*(\mathbf{m})^\top \mathbf{t}(\mathbf{x}^*(\mathbf{m}), \mathbf{m}) = \sum_{a \in A} x_a^*(\mathbf{m}) t_a(x_a^*(\mathbf{m}), m_a). \quad (2.5)$$

Let  $P$  be the set of network maintenance projects that must be conducted and let  $T$  be the set of time periods within the planning period considered. Each maintenance project  $p \in P$  is defined by the following attributes: the set of links affected by the project  $A_p \subseteq A, \forall p \in P$ ; link capacity reduction ratios  $R_a \in [0, 1], \forall a \in A_p$ ; and project duration  $D_p \in \{1, \dots, |T|\}$ . The set of links affected by each project is typically a subset of the network links and we assume that each link can only be affected by at most one project during the planning period. The link capacity reduction  $R_a$  is assumed to be a percentage of the maximal capacity of link  $a \in A$ . A reduction of 0% for all links of the network ( $R_a = 0, \forall a \in A$ ) corresponds to the base scenario in which no project is *active* and all links operate at maximal capacity. If a project  $p$  is active, then all links  $a \in A_p$  see their capacity reduced by  $100 \times R_a\%$ . Each project  $p$  must be active for  $D_p$  consecutive time periods and completed within the planning period. Hence, for each project  $p \in P$ , the set of feasible start times  $F_p$  can be defined as:

$$F_p \equiv \{0, \dots, |T| - D_p\}. \quad (2.6)$$

We introduce binary decision variables to represent project start times:

$$g_{s,p} \equiv \begin{cases} 1 & \text{if project } p \text{ starts at time period } s, \\ 0 & \text{otherwise,} \end{cases} \quad \forall p \in P, \forall s \in F_p. \quad (2.7)$$

and impose the project schedule constraints:

$$\sum_{s \in F_p} g_{s,p} = 1, \quad \forall p \in P. \quad (2.8)$$

To connect maintenance projects with the network performance metric (TSTT) we propose to enumerate all the capacity patterns  $\mathbf{c}$  resulting from all possible maintenance project combinations. The number of possible project combination is  $2^{|P|}$ . For each project combination  $\bar{P} \subseteq P$ —i.e. all projects in  $\bar{P}$  are active—the capacity of the links  $\bigcup_{p \in \bar{P}} A_p$  is reduced. We refer to the corresponding link capacity vector  $\mathbf{c}$  as a *capacity pattern*. Since links are associated with maintenance projects, it is more adequate to handle *project patterns* than capacity patterns. Let  $\Sigma \equiv \{0, \dots, 2^{|P|} - 1\}$  be the set of all project patterns, for each pattern  $\sigma \in \Sigma$ , we denote  $TSTT(\sigma)$  its associated TSTT under UE conditions. That is,  $TSTT(\sigma)$  represents the total network travel time corresponding to the unique TAP link flow solution under UE if pattern  $\sigma$  is active. Observe that each project pattern  $\sigma$  corresponds to a unique network state vector  $\mathbf{m}$  with  $TSTT(\sigma) = TSTT(\mathbf{m})$ . Further, each pattern can be represented as a binary vector where each component indicates if the corresponding project is active in this pattern or not. For instance, consider a scenario with 10 projects:  $P = \{1, \dots, 10\}$ ; the pattern consisting of all odd numbered projects is  $\sigma = \{1, 3, 5, 7, 9\}$  and this pattern can be encoded as the binary vector  $\boldsymbol{\varphi} = [1, 0, 1, 0, 1, 0, 1, 0, 1, 0]$  where  $\varphi_p \in \{0, 1\}$  indicate the activeness of  $p \in P$ —we henceforth refer to  $\boldsymbol{\varphi}$  as the binary representation of  $\sigma$ . Throughout this chapter we use alternatively  $\mathbf{m}$ ,  $\sigma$  and  $\boldsymbol{\varphi}$  to denote a *pattern* whenever convenient.

The objective of the NMSP is to minimize the overall TSTT over the planning period  $T$ . Hence a possible formulation for the NMSP is to view the decision problem as that of selecting the optimal sequence of patterns subject to project schedule constraints. We introduce binary decision variables to represent the pattern selection at each time interval:

$$k_{\sigma}^t \equiv \begin{cases} 1 & \text{if pattern } \sigma \text{ is selected at time period } t, \\ 0 & \text{otherwise,} \end{cases} \quad \forall \sigma \in \Sigma, \forall t \in T. \quad (2.9)$$

and impose the pattern selection constraints:

$$\sum_{\sigma \in \Sigma} k_{\sigma}^t = 1, \quad \forall t \in T. \quad (2.10)$$

The objective function of the upper level of the NMSP can then be formulated as:

$$\min \sum_{t \in T} \sum_{\sigma \in \Sigma} TSTT(\sigma) k_{\sigma}^t. \quad (2.11)$$

To link the project schedule variables with the pattern selection variables, we introduce two binary matrices  $[\gamma_{\sigma,p}]$  and  $[\zeta_{s,p}^t]$  defined as:

$$\gamma_{\sigma,p} \equiv \begin{cases} 1 & \text{if project } p \text{ is active in pattern } \sigma, \\ 0 & \text{otherwise,} \end{cases} \quad \forall p \in P, \forall \sigma \in \Sigma, \quad (2.12)$$

$$\zeta_{s,p}^t \equiv \begin{cases} 1 & \text{if } s \leq t \leq s + D_p - 1, \\ 0 & \text{otherwise,} \end{cases} \quad \forall p \in P, \forall s \in F_p, \forall t \in T. \quad (2.13)$$

Matrix  $[\gamma_{\sigma,p}]$  maps maintenance projects to patterns while matrix  $[\zeta_{s,p}^t]$  maps a project start time  $s \in F_p$  to  $D_p$  consecutive time periods. The linking constraints are:

$$\sum_{\sigma \in \Sigma} k_{\sigma}^t \gamma_{\sigma,p} = \sum_{s \in F_p} g_{s,p} \zeta_{s,p}^t, \quad \forall p \in P, \forall t \in T. \quad (2.14)$$

Let  $\mathbf{g}$  denote the vector of variables  $g_{s,p}, \forall p \in P, \forall s \in F_p$  and  $\mathbf{k}$  denote the vector of variables  $k_{\sigma}^t, \forall \sigma \in \Sigma, \forall t \in T$ . The space delimited by constraints (2.8), (2.10) and (2.14) represents the feasible region for the binary variables  $\mathbf{g}$  and  $\mathbf{k}$ . We show that, under these constraints, the integrality constraints on variables  $\mathbf{k}$  can be relaxed.

**Proposition 2.1.** *The integrality constraints on variables  $\mathbf{k}$  can be relaxed.*

Proposition 2.1 allows us to relax the integrality constraints on variables  $k_{\sigma}^t$  and helps in significantly reducing the number of subproblems in a Branch-and-Bound (B&B) framework. Further, because of the pattern selection constraint (2.10), the upper bound on variables  $k_{\sigma}^t$  can be dropped. The resulting formulation for the NMSP is summarized in Model 2.1.

**Model 2.1** (NMSP).

$$z_{NMSP} = \min \sum_{t \in T} \sum_{\sigma \in \Sigma} TSTT(\sigma) k_{\sigma}^t, \quad (2.15)$$

$$\text{s.t. } \sum_{s \in F_p} g_{s,p} = 1, \quad \forall p \in P, \quad (2.16)$$

$$\sum_{\sigma \in \Sigma} k_{\sigma}^t = 1, \quad \forall t \in T, \quad (2.17)$$

$$\sum_{\sigma \in \Sigma} k_{\sigma}^t \gamma_{\sigma,p} = \sum_{s \in F_p} g_{s,p} \zeta_{s,p}^t, \quad \forall p \in P, \forall t \in T, \quad (2.18)$$

$$k_{\sigma}^t \geq 0, \quad \forall \sigma \in \Sigma, \forall t \in T, \quad (2.19)$$

$$g_{s,p} \in \{0, 1\}, \quad \forall p \in P, \forall s \in F_p. \quad (2.20)$$

Model 2.1 is a bilevel MIP with an exponential number of variables (recall that  $|\Sigma| = 2^{|P|}$ ). Further each weight  $TSTT(\sigma)$  in the objective function must be obtained by solving the TAP for the corresponding pattern. Being a combination of MIP and bilevel programming, both of which are difficult optimization problems, the NMSP is intractable for large instances and an efficient solution methodology is required to solve the problem on realistic scenarios.

To solve the NMSP using the representation summarized in Model 2.1 we propose to work with a restricted set of pattern variables and use a Column Generation (CG) algorithm to introduce additional project patterns as required. Let  $\lambda^t$  be the dual variable associated to constraint (2.17) and let  $\mu_p^t$  be the dual variable associated to constraint (2.18). To price unused pattern variables  $k_{\sigma}^t$  we need to determine their reduced-cost  $r_{\sigma}^t$ , given by:

$$r_{\sigma}^t = TSTT(\sigma) - \lambda^t - \sum_{p \in P} \mu_p^t \gamma_{\sigma,p}. \quad (2.21)$$

Therefore determining the reduced-cost of a pattern variable requires solving the TAP for this particular pattern.

A solution to the NMSP is globally optimal if and only if the reduced cost of all variables is positive. To efficiently explore the feasible region of Model 2.1, we introduce the concept of *1-distance optimality*. Let  $d_1(\sigma)$  be the *1-distance neighborhood* of pattern  $\sigma$ , i.e. the set of patterns obtained by modifying each bit (one at a time) of the binary representation of  $\sigma$  to its complementary value. We say that a solution is *1-distance optimal* if the reduced cost of all patterns in the 1-distance neighborhood of the solution is positive. Let  $\Sigma^* \equiv \{\sigma \in \Sigma : \exists t \in T, k_{\sigma}^t > 0\}$  be the set of patterns used in the solution; and let  $T_{\sigma} \equiv \{t \in T : k_{\sigma}^t > 0\}, \forall \sigma \in \Sigma$  be the set of time periods at which pattern  $\sigma$  is used in the solution. For any solution, the condition of 1-distance optimality can be checked efficiently using Proposition 2.2.

**Proposition 2.2.** *Let  $\sigma \in \Sigma^*$  and let  $\sigma' \in d_1(\sigma)$  where  $\sigma'$  is obtained from  $\sigma$  by modifying the activeness of project  $p' \in P$ . Define:*

$$\mu_{p'}^* = \begin{cases} + \max\{\mu_{p'}^t : t \in T_{\sigma}\}, & \text{if } \gamma_{\sigma,p'} = 0, \\ - \min\{\mu_{p'}^t : t \in T_{\sigma}\}, & \text{otherwise.} \end{cases} \quad (2.22)$$



There exists  $t^* \in T$  such that  $r_{\sigma'}^{t^*} < 0$  if and only if  $TSTT(\sigma) - TSTT(\sigma') + \mu_{p'}^* > 0$ .

Since the NMSP is a mixed-integer program, the CG-based approach must be integrated within a B&P algorithm to guarantee integral solutions. Further, the lower-level must be accounted for within the optimization framework. The proposed B&P methodology is composed of several components. Specifically, the B&P algorithm requires a method to construct an initial feasible solution and initialize the columns pool, branching and node selection rules, a CG procedure to solve the LP relaxation of the Restricted Master Problem (LRMP) at each node of the B&P tree, pricing algorithms to identify negative reduced-cost pattern variables. In addition, a statistical regression model to approximate congestion effects will be integrated within the B&P algorithm to derive approximated lower bounds during the optimization.

We next introduce the concept of *first-order TSTT effects* of maintenance projects and discuss how this can be used to inform the solution method for the NMSP.

### 2.4.2 First-order TSTT effects of maintenance projects

To measure the level of interaction among projects we examine the variation of the TSTT when projects are active and inactive. This approach is highly combinatorial: for instance, to measure the variation of the TSTT when  $n_p$  projects are either active or inactive, we need to consider  $2^{|P|-n_p}$  subpatterns and, for each of them, compare the resulting TSTT values when the  $n_p$  projects are either active or inactive. To investigate the interaction among maintenance projects, we thus focus on low interaction levels. Measuring project interaction also plays a central role in the proposed solution methodology for the NMSP.

For presentation purposes, we introduce the following notation: we denote  $\varphi_p \in \{0, 1\}$  the activeness of project  $p \in P$ . We introduce the concatenation operator  $\oplus$  to represent the concatenation of mutually exclusive projects in a pattern. Further, we abuse notation and use  $\tilde{\sigma}$  to denote a subpattern, i.e. a pattern with a subset of projects, whenever necessary. For instance,  $\sigma = [\varphi_p = 1, \varphi_q = 0] \oplus \tilde{\sigma}$  is a project pattern where  $p$  is active,  $q$  is inactive and the activeness of the remaining projects is given by  $\tilde{\sigma}$  (which is a subpattern of the subset of projects  $P \setminus \{p, q\}$ ).

Using the above notation, we define the *first order TSTT effects* of a project  $p$  with regards to subpattern  $\tilde{\sigma}$ ,  $\Delta_{p,\tilde{\sigma}}$ :

$$\Delta_{p,\tilde{\sigma}} \equiv TSTT([\varphi_p = 1] \oplus \tilde{\sigma}) - TSTT([\varphi_p = 0] \oplus \tilde{\sigma}). \quad (2.23)$$

We denote  $\Delta_p$  the vector of first order TSTT effects of project  $p$  for all subpatterns of the subset of projects  $P \setminus \{p\}$ , i.e.  $\Delta_p = [\Delta_{p,\tilde{\sigma}}]_{\tilde{\sigma}}$ .  $\Delta_p$  represents the influence of a single project onto the TSTT experienced in the network.

We next present a tailored experimental design based on first-order TSTT information to construct a statistical regression model that can be used to approximate TSTT values.

### 2.4.3 Statistical approximation of TSTT effects

Throughout this section, we assume that a set of observed TSTT values and their corresponding project patterns is available. This data will be eventually obtained in a pre-processing

step prior to the execution of the B&P algorithm which is presented at the end of the section. We denote  $\Sigma_{obs} \subset \Sigma$  the set of observed patterns.

The TSTT corresponding to a project pattern  $\sigma$  can be approximated using regression analysis by using project activeness variables  $\varphi_p, \forall p \in P$  as explanatory variables. Due to the nonlinear congestion effects inherent to traffic networks, we propose an exponential regression model of the form:

$$TSTT(\varphi) = e^{b_y + \sum_{p \in P} b_p \varphi_p + \epsilon_\varphi}. \quad (\text{E})$$

where  $\varphi = [\varphi_p]_{p \in P}$  is the binary representation of a pattern,  $b_y, b_p \forall p \in P$  are regression coefficients for each project and a y-intercept, and  $\epsilon_\varphi$  is a normally distributed error on the prediction of pattern  $\varphi$ .

We use an Ordinary Least Squares (OLS) regression to fit Model (E) to observed TSTT data. The resulting statistical model can then provide some information on the TSTT corresponding to a non-observed pattern, i.e. for which the TAP has not been solved. This procedure can be helpful in the pricing subproblem within the CG algorithm since the reduced-cost of a pattern variable depends on its TSTT value. To increase the likelihood that negative reduced-cost pattern variables are correctly identified, we propose to use a scaling factor  $\bar{\psi}$  to approximate the prediction error  $\epsilon_\varphi$  in the statistical model. We define the approximated TSTT of pattern  $\varphi$  as:

$$aTSTT(\varphi) = \bar{\psi} e^{b_y + \sum_{p \in P} b_p \varphi_p}. \quad (2.24)$$

The value of  $\bar{\psi}$  should be chosen such that  $aTSTT(\varphi)$  is a lower bound on  $TSTT(\varphi)$  for most patterns. Assuming that  $\bar{\psi}$  is chosen accordingly, this approximation can be used to efficiently determine lower bounds within the resolution of the NMSP. This stems from the functional form of  $aTSTT(\varphi)$  which is convex if the integrality constraints on variables  $\varphi$  are relaxed.

**Proposition 2.3.** *Let  $\varphi \in [0, 1]^{|P|}$ . The function  $aTSTT(\varphi)$  is convex with regards to  $\varphi$ .*

Using approximated TSTT values within the calculation of patterns' reduced-cost gives:

$$\hat{r}^t(\varphi) = aTSTT(\varphi) - \lambda^t - \sum_{p \in P} \mu_p^t \varphi_p, \quad \forall t \in T. \quad (2.25)$$

If  $\hat{r}^t(\varphi) \geq 0$ , then there is a high likelihood that the true reduced-cost of the corresponding pattern variable  $k_\sigma^t$  (i.e.  $\varphi$  is the binary representation of  $\sigma$ ) is positive.

To determine an initial value for  $\bar{\psi}$ , we determine the 5th percentile of the cumulative distribution function of random variable  $\frac{TSTT(\varphi)}{e^{b_y + \sum_{p \in P} b_p \varphi_p}}$  over the observed TSTT data  $\Sigma_{obs}$  and denote this value  $\psi_{5th}$ . This initial value for  $\bar{\psi}$  provides a conservative lower bound on the TSTT values since it guarantees that the approximated TSTT is a lower bound on TSTT for 95% of the observed patterns. In order to refine this bound, we introduce an ordered set of scaling factors  $\Psi$  and initialize this set with  $\psi_{5th}$ . The set  $\Psi$  will be updated during the execution of the B&P algorithm as new TSTT data becomes available in order to provide a less conservative bound. At any point, the scaling factor  $\bar{\psi}$  used in (2.24) is the average of all scaling factors in  $\Psi$ .

We now explain how we select a subset of the patterns and use their TSTT—obtained by solving the TAP—to build a robust statistical model. Project patterns can be characterized

by the number of active projects which follows a binomial distribution, i.e. the number of patterns with  $i$  active projects is  $\binom{|P|}{i}$ . To span the entire spectrum of patterns, we iterate over the number of active projects  $i$  and, for each  $i$ , we randomly select a few patterns. Since the number of patterns with  $i$  active projects follows a binomial distribution, we wish to sample patterns accordingly. The ratio  $\binom{|P|}{i}/2^{|P|}$  gives the proportion of patterns with  $i$  active projects over the total number of patterns. We propose to pick an initial number of patterns  $n_S$  with a varying density of active projects and explore their 1-distance neighborhood. Let  $S_i = \left\lceil \frac{n_S \binom{|P|}{i}}{2^{|P|}} \right\rceil$  be the number of randomly selected patterns with  $i$  active projects and let  $\Sigma_S \subset \Sigma$  be the set of randomly selected patterns, i.e.  $|\Sigma_S| = \sum_{i=0..|P|} S_i$ . We initialize the set of observed patterns as  $\Sigma_{obs} = \Sigma_S \cup (\cup_{\sigma \in \Sigma_S} d_1(\sigma))$ , where  $d_1(\sigma)$  is the 1-distance neighborhood of pattern  $\sigma$ . We solve the TAP for each pattern in  $\Sigma_{obs}$  and use the resulting TSTT values to fit Model (E). We then used the resulting TSTT values to estimate projects' first order TSTT effects based on the data gathered. Since not all TAP problems are expected to be solved during the execution of Algorithm 1, we denote  $\tilde{\Delta}_p$  the vector of first order TSTT effects obtained for Project  $p \in P$ . The objective of this process is twofold: (i) it provides an initial set of patterns of reasonable size to build a robust statistical model; (ii) it provides data on projects' first-order TSTT effects which is shown to have a high impact on branching operations. The pseudo-code of this pre-processing step is summarized in Algorithm 1.

---

**Algorithm 1:** Pre-processing algorithm

---

**Input:**  $N, A, P, A_p \forall p \in P, R_a \forall a \in A, n_S$   
**Output:**  $b_y, b_p \forall p \in P, \Delta_p, \forall p \in P, \Psi, \Sigma_{obs}$

- 1  $\Sigma_{obs} \leftarrow \emptyset$
- 2 **for**  $i = 0..|P|$  **do**
- 3      $S_i \leftarrow \left\lceil \frac{n_S \binom{|P|}{i}}{2^{|P|}} \right\rceil$
- 4     **for**  $1..S_i$  **do**
- 5          $\sigma \leftarrow$  randomly select a pattern with  $i$  active projects
- 6          $\Sigma_{obs} \leftarrow \Sigma_{obs} \cup (\sigma \cup d_1(\sigma))$
- 7 **for**  $\sigma \in \Sigma_{obs}$  **do**
- 8      $TSTT(\sigma) \leftarrow$  Solve  $TAP(\sigma)$
- 9 Fit Model (E) on observed TSTT values
- 10  $\tilde{\Delta}_p, \forall p \in P \leftarrow$  first-order TSTT effects
- 11  $\psi_{5th} \leftarrow$  5th percentile of  $\frac{TSTT(\varphi)}{e^{b_y + \sum_{p \in P} b_p \varphi_p}}$
- 12  $\Psi \leftarrow \{\psi_{5th}\}$

---

The pre-processing algorithm solves a number of TAP which grows quadratically with  $|P|$ . The exact number of TAP solved depends on  $n_S$  and on the number of times a modified pattern has already been processed by the algorithm. The accuracy of this method is discussed in Section 2.4.7.

#### 2.4.4 Optimization framework and numerical experiments

In this section, we present the optimization framework developed to solve the bilevel NMSB. This framework ultimately results in a B&P algorithm that integrates a collection of models and algorithms. B&P algorithms are used to solve mixed integer programs with a large number of variables [Barnhart et al., 1998] and have only recently been used in game-theoretic

---

contexts such as zero-sum Bayesian games [Halvorson et al., 2009] and Stackelberg security games [Jain et al., 2010]. In B&P, instead of solving the LP relaxation of the integer program at each node of an enumeration tree—as in traditional B&B frameworks—a CG algorithm is used to solve the LRMP and iteratively add columns to the Master Problem.

Implementing a B&P algorithm involves the resolution of a pricing subproblem at each iteration of the CG algorithm. To guarantee global optimality, this pricing subproblem should be solved to optimality or at least a valid lower bound should be produced and passed to the children nodes of the current node in the B&P tree. This is typically achieved by solving an optimization problem that determines the minimal reduced cost value. Deriving such a valid lower bound is difficult since the existence of Braess’ Paradox effects in the TAP invalidates a traditional optimization approach aiming to minimize the reduced cost  $r_\sigma^t$  of pattern variables  $k_\sigma^t$  as defined by Eq. (2.21).

A global lower bound on  $r_\sigma^t$  can be derived by solving the System Optimum (SO) TAP for the base case configuration, i.e. all projects are inactive: since the resulting SO-TSTT value is a lower bound on the TSTT of any pattern  $\sigma \in \Sigma$ , this can be used to derive a valid lower bound on the minimal reduced cost of pattern variables. However, this global lower bound is expected to be very weak since there is often a substantial gap between the TSTT values at SO and at UE in congested networks. Extensive numerical experiments have shown that this bound systematically results in negative bounds on reduced costs, thus invalidating its use for bounding purposes. To overcome this challenge, we revert to an approximation approach.

The convex approximation of TSTT values outlined in Section 2.4.3 provides a method to determine a lower bound on the approximated TSTT of a non-observed pattern. This bounding technique can be used in two ways: (i) to determine an approximated lower bound at each node of the B&P tree by minimizing  $aTSTT(\varphi)$  with regards to project schedule constraints; (ii) to determine an approximated lower bound on the price of pattern variables within the CG algorithm by minimizing  $\hat{r}^t(\varphi)$ . In both cases the optimization problem can be solved using a convex algorithm for which efficient implementations are readily available—recall that we relax the integrality constraints on project activeness variables.

Formally, Model 2.2 represents the Approximated NMSP (ANMSP). Since the objective function of Model 2.2 is convex with regards to the decision variables and all constraints are linear, the ANMSP is a convex optimization problem. Observe that the objective function of Model 2.2 is expressed in terms of variables  $y_p^t$  which indicate project activeness at each time period. Although it is not guaranteed that  $z_{ANMSP} \leq z_{NMSP}$ , it is highly likely due to the presence of the scaling factor  $\bar{\psi}$ .

**Model 2.2** (Approximated NMSP).

$$z_{ANMSP} = \min \sum_{t \in T} \bar{\psi} e^{b_y + \sum_{p \in P} b_p y_p^t}, \quad (2.26)$$

$$\text{s.t. } \sum_{s \in F_p} g_{s,p} = 1, \quad \forall p \in P, \quad (2.27)$$

$$\sum_{s \in F_p} g_{s,p} \zeta_{s,p}^t = y_p^t, \quad \forall p \in P, \forall t \in T, \quad (2.28)$$

$$g_{s,p} \in [0, 1], \quad \forall p \in P, \forall s \in F_p, \quad (2.29)$$

$$y_p^t \in [0, 1], \quad \forall p \in P, \forall t \in T. \quad (2.30)$$

Similarly, Model 2.3 finds a lower bound on the reduced-cost of pattern variables  $k_\sigma^t$  by solving a separable convex optimization problem. For each time period  $t \in T$ , the integrality constraints in the formulation of the approximated reduced cost  $\hat{r}^t(\varphi)$  are relaxed and the minimum over all time periods  $\hat{\underline{r}}$  is a likely lower bound on the true minimal reduced-cost of pattern variables  $k_\sigma^t$ .

**Model 2.3** (Lower bound on approximated reduced-cost).

$$\hat{\underline{r}} = \min_{t \in T} \min \left\{ \bar{\psi} e^{b_y + \sum_{p \in P} b_p \varphi_p} - \lambda^t - \sum_{p \in P} \mu_p^t \varphi_p : \varphi \in [0, 1]^{|P|} \right\}. \quad (2.31)$$

If  $\hat{\underline{r}} > 0$ , then the lower bound on the approximated reduced-cost is positive and the CG algorithm can be terminated with a statistical guarantee. Otherwise, this bound on the reduced-cost provides a classical lower bound on the objective value of the LRMP which may be used to prematurely terminate the CG algorithm [Lübbecke and Desrosiers, 2005]. Using constraint (2.17), the sum of all pattern variables  $k_\sigma^t$  is:  $\sum_{t \in T} \sum_{\sigma \in \Sigma_R} k_\sigma^t = |T|$ . Let  $z$  be the current value of objective function (2.11) and let  $z^*$  be its optimal value:

$$\hat{\underline{z}} = z + |T| \hat{\underline{r}} \leq z^* \leq z. \quad (2.32)$$

The approximated lower bound  $\hat{\underline{z}}$  provides an approximated optimality gap within the CG algorithm.

### 2.4.5 Overview of the B&P algorithm for the NMSP

An overview of the proposed B&P algorithm is summarized in Algorithm 2.

The proposed B&P algorithm starts with a pre-processing step during which the TAP is solved for a subset of the patterns selected such that projects' first-order TSTT effects can be estimated. A MILP approximating the NMSP is then solved to obtain an upper bound and initialize the set of columns (pattern variables) in the B&P framework. The algorithm then enters a **while** loop that iterates over the nodes in the enumeration tree. At first pass, the

Network	Nb of Zones	Nb. of Nodes	Nb. of Links	Nb. of Trips	OD Flow Factor
Sioux Falls	24	24	76	360,000	1
BerlinMPF	98	975	2,184	23,648.5	10
Barcelona	110	1,020	2,522	184,679.5	2

Table 2.4: Network Characteristics.

ANMSP is solved (Model 2.2) at the root node and if its solution is not integral, the LRMP of the NMSP is solved using CG. The best of the two lower bounds obtained is stored. If the solution of the LRMP is also not integral, a rounding heuristic is triggered and the algorithm branches on the solution providing the highest lower bound. This procedure is repeated until the enumeration tree has been fully explored or the approximated optimality gap is closed.

---

**Algorithm 2:** Overview of the Branch-and-Price Algorithm

---

- 1 Pre-processing step: get observed patterns and first-order TSTT effects
  - 2 Solve MILP to obtain an upper bound for the NMSP and an initial set of columns
  - 3 **while** *optimal* =False **do**
  - 4     Select node in the tree
  - 5     Solve ANMSP at current node
  - 6     **if** *solution is not integral* **then**
  - 7         Solve LRMP by CG at current node
  - 8         **if** *solution is not integral* **then**
  - 9             Select solution with highest lower bound between ANMSP and CG
  - 10            Try rounding heuristic to find an improved upper bound
  - 11            Branch on solution which provides the highest lower bound
- 

Details on each component of Algorithm 2 can be found in Rey et al. [2019].

## 2.4.6 Numerical experiments

Testing the proposed models and algorithms for the NMSP requires the following input data: a transport network, flow-dependent link travel time functions, maintenance projects with their characteristics (duration, links affected and capacity reduction) and a planning period. Given the many possibilities we focused on specific instances for the NMSP that we find representative of competitive realistic scenarios. All the results are presented for three transport networks, namely Sioux Falls, BerlinMPR and Barcelona, available at <https://github.com/bstabler/TransportationNetworks> which characteristics are presented in Table 2.4. The Sioux Falls is a classical benchmark network in traffic equilibrium whereas BerlinMPF and Barcelona are larger, more realistic transport networks. We use an OD flow factor on the latter two networks to increase the baseline congestion effects.

For each network, three network maintenance scenarios are used to evaluate the models and the algorithms with respectively 10, 15 and 20 projects. Maintenance projects are composed of spatially clustered links (i.e. all links in  $A_p$  are connected) and, when a project is active, each affected link has its capacity reduced by 50%. The list of links affected by each maintenance project in each of the three networks considered can be found at [https://github.com/davidrey123/Network\\_Maintenance\\_Scheduling](https://github.com/davidrey123/Network_Maintenance_Scheduling).

In the 10-project scenario each project affects 100 links and this scenario induces  $2^{10} = 1,024$  patterns on the network. Given that the number of pattern variables in Model 2.1 is  $|T|2^{|P|}$ , NMSP instances based on 10-project scenarios can be handled by most commercial MILP solvers without resorting to column generation, assuming that all 1,024 TSTT values can be pre-processed and that  $|T|$  is of reasonable size. Hence 10-project scenarios, are used to explore problem behavior and benchmark some components of the B&P algorithm that do not require any column pricing. We implicitly make the assumption that these components will perform similarly when the number of projects increases and a CG algorithm is required to manage the LP relaxation of the master problem.

In the 15- and 20-project scenarios, each project affects 50 links and these scenarios induce 32,768 and 1,048,576 patterns, respectively. Extensive testing on these scenarios indicates that solving Model 2.1 with all pattern variables poses memory management and runtime issues. Further, pre-processing all TSTT values may be prohibitive, even for medium-size transport networks. Hence these scenarios are used to benchmark the proposed B&P algorithm. All the numerical results are presented in terms of relative TSTT (rTSTT) defined as the ratio of the TSTT by the base TSTT (when all projects are inactive).

We consider two planning periods:  $|T| = 25$  and  $|T| = 50$ , which can be thought of as half-year-long and year-long planning periods if the time unit is assumed to represent a week. We assume that the duration of projects is normally distributed with a mean of  $|T|/3$  and a standard deviation of  $|T|/10$ ; and for all the tests conducted in the optimization framework we randomly generate NMSP instances by sampling project durations from this distribution.

The computing environment is as follows: all the models and algorithms are implemented in Python on a Windows machine with 8Gb of RAM and an Intel i7 processor at 2.9GHz. The linear and integer linear optimization problems are solved with CPLEX v12.7.1 [CPLEX, 2014] in single-threaded mode: CPLEX’s primal Simplex algorithm is used for the LP relaxations of Model 2.1 and CPLEX’s MIP algorithm is used for solving the proposed MILPs to find an initial feasible solution. In addition, we use IPOPT [Wächter and Biegler, 2006] to solve the convex constrained optimization problems that arise within the CG algorithm. Finally, we use the TAPAS algorithm [Bar-Gera, 2010] with an accuracy corresponding to average excess cost of  $1e^{-10}$  to solve all TAPs in our tests.

#### 2.4.7 Testing of the statistical regression model

We test the statistical model to approximate TSTT values on all three networks for three sets of projects of size 10, 15 and 20 using the proposed pre-processing procedure (Algorithm 1) with a varying parameter  $n_S = 1, 10$  and 100. For each combination of network, project size and  $n_S$  value, we fitted the proposed statistical model and tested its performance with an out-of-sample validation. For 10-project instances, the number of patterns is relatively small ( $2^{10} = 1,024$ ) and using  $n_S = 1$  in Algorithm 1 results in using 10.74% of the patterns. We use all remaining patterns for the out-of-sample validation of the 10-project models. Since 10.74% is already a significant proportion of patterns, higher values of  $n_S$  are not tested for 10-project instances. For 15-project instances, 0.75% ( $n_S = 1$ ) to 5.23% ( $n_S = 100$ ) of the patterns are used, whereas for 20-project instances these figures are reduced to 0.04% ( $n_S = 1$ ) to 0.23% ( $n_S = 100$ ). We use 10,000 patterns for the out-of-sample validation of 15- and 20-project models. The R-square ( $R^2$ ) and Root-Mean-Square-Error (RMSE) in terms of relative TSTT values are summarized in Table 2.5 wherein the values in parenthesis indicate the corresponding proportion of patterns used for fitting the statistical model.

$ P $	$n_S$ (% patterns)	Sioux Falls		BerlinMPF		Barcelona	
		$R^2$	RMSE	$R^2$	RMSE	$R^2$	RMSE
10	1 (10.74)	0.977	0.209	0.961	0.283	0.997	0.051
15	1 (0.75)	0.968	0.194	0.974	0.152	0.984	0.035
15	10 (0.95)	0.960	0.219	0.978	0.140	0.979	0.041
15	100 (5.23)	0.975	0.169	0.983	0.105	0.985	0.034
20	1 (0.04)	0.943	0.224	0.946	0.397	0.945	0.528
20	10 (0.05)	0.947	0.222	0.963	0.296	0.961	0.423
20	100 (0.23)	0.961	0.186	0.978	0.191	0.963	0.404

Table 2.5: Validation of the exponential statistical regression model (E). The  $R^2$  and RMSE values are expressed in terms of relative TSTT and are obtained from an out-of-sample validation of the exponential statistical regression model (E). The patterns used for fitting the models are obtained using Algorithm 1 with different values of the tuning parameter  $n_S$ . The proportion of patterns used for fitting the models with regards to the entire sample space ( $2^{|P|}$ ) is provided in parenthesis.

The results highlight that the fitness of the proposed statistical models is of good quality with an R-square value consistently greater than 0.94, even if only a small fraction of patterns are used for fitting the models (e.g.  $n_S = 1$ ). Increasing the number of samples through the  $n_S$  parameter tends to increase R-square values and reduce RMSE values. We find that model fitness is impacted by the network and the associated maintenance projects: larger networks, in terms of number of nodes and links (such as BerlinMPF and Barcelona), appear to be more sensitive to the number of projects compared to smaller networks (Sioux Falls). Although increasing the number of projects appears to reduce R-square values and to increase RMSE values, the fitness of the statistical models remains significant. This suggests that the proposed exponential statistical regression model together with the proposed pre-processing procedure (Algorithm 1) which explore the 1-distance neighborhood of selected patterns, are appropriate to estimate TSTT effects on congested networks. Indeed the proposed exponential approximation cannot capture various effects that occur in transportation networks, such as cases of Braess' paradox. However, recent findings suggest that the existence of Braess' paradox in real transport networks may be scarce [Bekhor and Sorani, 2017], thus not diminishing the usefulness of the proposed approximation too much.

### B&P vs B&B

To show the performance of the proposed B&P algorithm, we compare the solution returned by Algorithm 2 with that obtained when solving the same problem using an exact approach for 10-project scenarios. To solve the latter, all TSTT values of the instance patterns are pre-calculated and Model 2.1 is then solved using the proposed B&B procedure. The relative gap between these two solutions are then calculated and the histograms of their distribution is plotted for each network and number of time periods. For this benchmark, 10.74% of the patterns are used for fitting Model (E). For each network and each number of time periods, we solve 50 randomly generated instances. The results are summarized in Figure 2.1. Over the 300 generated instances, we find that the highest optimality gap observed never exceeds



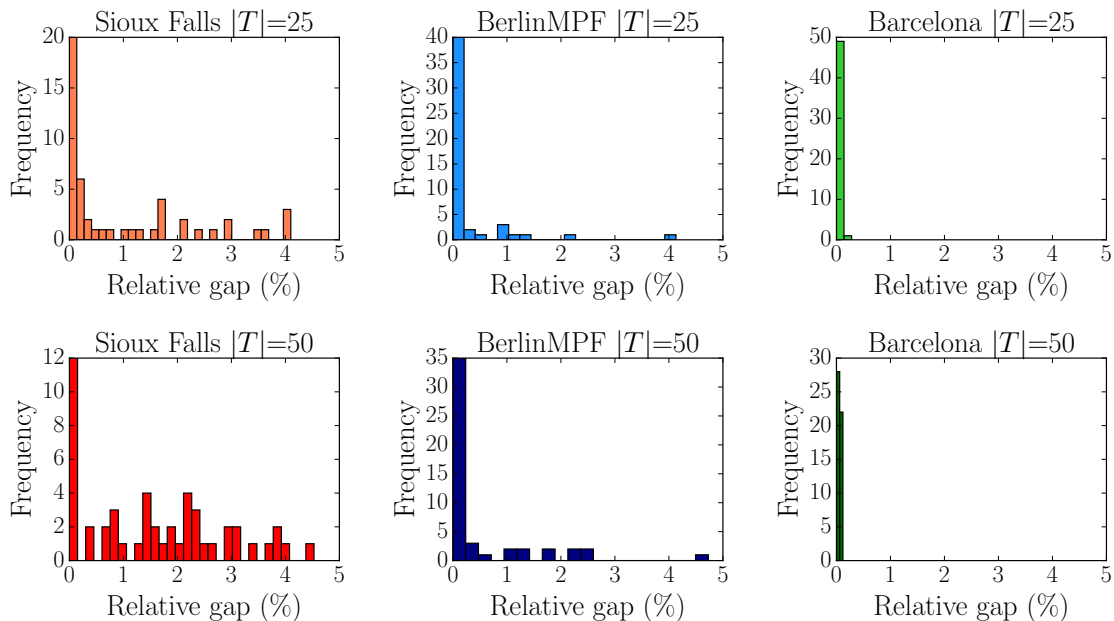


Figure 2.1: Benchmark of the B&B and B&P algorithms: relative gap after solving 50 randomly generated instances from three 10-project scenarios (Sioux Falls, BerlinMPF and Barcelona) with  $|T| = 25, 50$  and normally distributed project durations  $[D_p]_{p \in P} \sim \mathcal{N}(\frac{|T|}{3}, (\frac{|T|}{10})^2)$ .

5% and the average optimality gap is 0.57%. In addition, the proposed B&P algorithm is able to find the optimal solution in 35% of the cases. Finally, the average proportion of patterns used within the B&P algorithm is 19.9%.

This benchmark shows that the proposed B&P is capable of finding optimal or near-optimal solutions for the bilevel network maintenance scheduling problem using only a fraction of the patterns. Without accounting for the time spent solving the TAPs, the average runtime of the proposed B&B approach is 18.2s and that of the B&P is 56.9s. This gap in runtime is expected for such small instances since the number of columns remains manageable for a direct problem resolution using B&B which obviates the need for column generation. However, extensive implementation reveals that a direct resolution approach by B&B is intractable for larger instances.

### Results on large instances

We now present numerical results for the proposed B&P algorithm on instances derived from the Barcelona network. Results for the Sioux Falls and the Berlin MPF networks are available in [Rey et al. \[2019\]](#). For each network, we consider four experiments consisting of instances with 15 and 20 maintenance projects and with 25 and for 50 time periods. For each experiment, we generate 20 random instances with project durations following a Normal distribution with a mean of  $|T|/3$  and a standard deviation of  $|T|/10$ . Hence a total of 240 instances are tested regrouped in four experiments per network. The objective of these tests is to assess the performance of the proposed solution algorithm with regards to the input data.

---

The results are summarized in Table 2.6 which give the average performance of the B&P algorithm for each experiment (i.e. column in the tables). For each experiment, the mean value over the 20 instances is given together with the standard deviation in parenthesis. The header of the rows is described from top to bottom. The first and second rows give the number of projects and the number of patterns in each scenario. The third row gives the number of time periods in each experiment. The next nine rows provide runtime information. Since the runtime for solving TAPs may change considerably depending on the size of the network considered, we distinguish the runtime of the B&P algorithm spent solving the TAPs problems from the rest, hereby referred to as the upper-level problem and denoted U-NMSP. The next seven rows are expressed in % of the total runtime spent solving U-NMSP while the following two rows give the total runtime spent solving the U-NMSP and the TAP (lower-level problem) in hours. FEAS is the total time spent to find an initial and feasible solutions; ANMSP is the total time spent solving Model 2.2 to find lower bounds; LRMP is the total time spent solving the LRMP; PH is the total time consumed by the pricing heuristic; ARC is the total time spent solving convex approximations on reduced-costs (Model 2.3); ES is the total time spent explicitly searching for negative reduced-cost columns; RH is the total time spend in the rounding heuristic. The next six rows give statistics (in %) for the proportion of patterns used within the B&P algorithm, the proportion of TAP solved, the proportion of false negative approximations, the mean value for  $\bar{\psi}$  used in the convex approximations, the optimality gap after solving the root node in the B&P tree and the proportion of time spent solving the root node in the B&P algorithm (with regards to the total solve time). The last two rows indicate the number of iterations and the value of the objective function  $z_{NMSP}$  in terms of relative TSTT—defined as the ratio of the TSTT by the base TSTT—upon termination of the B&P algorithm.

We find that the mean total runtime of the upper level problem (U-NMSP) increases with the number of projects and the number of time periods but that it is robust across network topologies. Instances with 15 projects require on average 5 ( $|T| = 25$ ) to 20 ( $|T| = 50$ ) minutes; whereas instances with 20 projects require on average 40 ( $|T| = 25$ ) to 90 ( $|T| = 50$ ) minutes for the upper-level problem, i.e. solving the LRMP, the convex approximation problems and the pricing algorithms. Overall, most of the runtime for the upper-level problem is spent solving the LRMP and within the explicit search (ES) pricing algorithm. The pricing heuristic (PH) is seen to be robust to the number of time periods but less so to the number of projects. For 20-project instances, a significant amount of time is also spent exploring the 1-distance neighborhood of the patterns within the PH algorithm. The robustness with regards to the number of time periods is a direct consequence of Proposition 2.2 which obviates the exploration of all time periods to price patterns in the 1-distance neighborhood of the current solution. Although not as robust, the ES procedure is more sensitive to the number of projects than that of time periods.

The time spent solving the TAP varies considerably based on the network size and demand. The total TAP solve time is correlated with the number of TAP solved. For 15-project instances, the implementation on the three networks considered reveals that approximately 15% to 25% of the TAPs are typically solved (i.e. 5,000 to 8,000 TAPs), whereas for 20-project instances these figures change to 1.2% to 1.5% (i.e. 12,000 to 16,000 TAPs). We also find that the number of pattern variables used within the LRMP is reduced by a factor of 2 to 5 compared to the number of TAP solved. This is reflected in the high number of false negative approximations observed (often in the 60% - 80% range) and this can be attributed to the conservative lower bound which is used to initialize  $\bar{\psi}$ .

	$ P $	15		20	
	$ \Sigma $	3e+4		1e+6	
	$ T $	25	50	25	50
Time (% U-NMSP)	FEAS	0.38 (0.23)	0.25 (0.45)	0.02 (0.04)	0.03 (0.04)
	ANMSP	2.30 (1.02)	2.88 (1.32)	0.23 (0.07)	0.81 (0.16)
	LRMP	50.45 (11.41)	76.55 (8.88)	32.54 (6.98)	38.11 (5.52)
	PH	5.62 (1.78)	1.52 (0.49)	30.06 (8.21)	28.50 (6.87)
	ARC	23.59 (5.78)	11.64 (4.48)	1.77 (0.58)	1.66 (0.29)
	ES	15.07 (4.61)	6.79 (3.27)	30.18 (10.70)	25.66 (6.22)
	RH	2.37 (9.23)	0.27 (0.46)	4.49 (14.85)	3.10 (5.18)
Time (h)	U-NMSP	0.03 (0.03)	0.15 (0.10)	0.64 (0.41)	1.07 (0.48)
	TAP	18.25 (7.57)	18.38 (6.36)	95.99 (19.98)	103.91 (21.44)
(in %)	Patterns used	3.02 (1.17)	3.58 (1.32)	0.39 (0.09)	0.60 (0.18)
	TAP solved	16.56 (5.70)	17.37 (4.98)	1.28 (0.21)	1.33 (0.28)
	False negative	82.79 (4.34)	83.14 (5.16)	69.93 (7.59)	61.81 (4.08)
	Mean $\bar{\psi}$	99.91 (0.77)	100.39 (0.78)	101.48 (2.43)	99.18 (2.22)
	Root node gap	1.49 (0.54)	1.53 (0.49)	2.36 (0.71)	5.91 (6.35)
	Root node time	52.96 (17.72)	50.73 (24.24)	22.11 (27.28)	21.58 (22.96)
	Nb. iterations	3.05 (1.02)	3.35 (1.01)	4.10 (1.58)	3.45 (1.20)
	$z_{NMSP}$	32.64 (1.20)	65.16 (2.69)	42.73 (4.19)	92.88 (7.30)

Table 2.6: Performance of the B&P Algorithm on the Barcelona network. Average performance of the B&P algorithm on the Barcelona network with 15 and 20 projects and with 25 and 50 time periods. In each experiment, 20 instances with random project durations ( $[D_p]_{p \in P} \sim \mathcal{N}(\frac{|T|}{3}, (\frac{|T|}{10})^2)$ ) are solved. The mean of each experiment is given with the standard deviation in parenthesis.

The root node gap is on average relatively tight, with low values of 1.5% observed for the 15-project instances on the Barcelona network and high values of 5.7 to 6.2% observed for 20-project instances on all three networks tested. The time spent in the B&P algorithm at the root node is often substantial and varies between 18% and 61% of the total solve time for the Sioux Falls network. Since this accounts for the time solve TAPs, this proportion changes considerably based on the network. For the Barcelona network, on average, half of the total solve time in 15-project instances is spent at the root node whereas this represents only 22% of the total solve time in 20-project. This is due to the greater time required to solve TAPs on the Barcelona network compared to the smaller networks tested.

We find that the number of iterations required by the B&P algorithm only slightly increases with the number of projects and time periods, with an average of 3 to 5 iterations frequently achieved. This low number of iterations together with the analysis of the B&P algorithm at the root node suggests that most of the progress is performed in the CG process at the root node. Subsequent iterations then attempt to close the gap by branching but may also offer the possibility to generate additional patterns and further improve the solution.

## 2.5 Scheduling for road network disaster recovery

The scope of this section is the long-term problem of planning the reconstruction of a road network which was damaged by a disaster. The recovery of the entire road network is assumed to require a substantial amount of resources, e.g. budget, working crew, equipment; which are limited in availability. We assume that the recovery of the road network will take place over a pre-defined planning horizon (e.g. 12-24 months), which is divided into time periods (e.g. 2-4 weeks). We assume that the time periods are long enough so that travelers have the time to adjust their route choice preferences under the UE principle. We assume that the reconstruction of the road network can be organized into *recovery projects*, each of which represents a set of roads damaged by the disaster. The network recovery problem of interest is that of optimizing the order in which the recovery projects should be completed subject to resource availability constraints and assuming UE conditions. We build on and extend the pattern formulation of the NMSP presented in Section 2.4.1 to account for network recovery effects, and introduce customized scheduling heuristics.

### 2.5.1 Formulation of the network recovery problem

We use the same notation system as in Section 2.4.1. We extend this framework to characterize the evolution of links of the network beyond two states (which is the case for the NMSP). Let  $\gamma_{p,m,\sigma}$  be a binary parameter indicating if project  $p$  is in state  $m$  in pattern  $\sigma$  (1) or not (0):

$$\gamma_{p,m,\sigma} \equiv \begin{cases} 1 & \text{if project } p \text{ is in state } m \text{ in pattern } \sigma, \\ 0 & \text{otherwise,} \end{cases} \quad \forall p \in P, \forall m \in M, \forall \sigma \in \Sigma. \quad (2.33)$$

Let  $\omega_{p,m}^{s,t}$  be a binary parameter indicating if project  $p$  is in state  $m$  at time period  $t$ , if  $p$  started at time period  $s$  (1) or not (0):

$$\omega_{p,m}^{s,t} \equiv \begin{cases} 1 & \text{if project } p \text{ is in state } m \text{ at time period } t, \\ & \text{if } p \text{ started at time period } s, \\ 0 & \text{otherwise,} \end{cases} \quad \forall p \in P, \forall m \in M, \forall s \in F_p, \forall t \in T. \quad (2.34)$$

For each project  $p$ , for each state  $m$  and for each time period  $t$ , the following linking constraints link variables  $g_{s,p}$  and  $k_{\sigma}^t$ :

$$\sum_{\sigma \in \Sigma} k_{\sigma}^t \gamma_{p,m,\sigma} = \sum_{s \in F_p} g_{s,p} \omega_{p,m}^{s,t}, \quad \forall p \in P, \forall m \in M, \forall t \in T. \quad (2.35)$$

Constraints (2.35) combine two representations of the concept of project activeness, i.e. both its left-hand-side and its right-hand-side take value 1 if project  $p$  is in state  $m$  at time  $t$ , and 0 otherwise. To model recovery resources availability, e.g. work crew or equipment, we assume that each project  $p \in P$  in state  $m \in M$  requires a known amount of recovery resources per time period denoted  $R_{p,m}$ . The limited availability in recovery resources can be represented by the total amount of recovery resources available at time period  $t \in T$  denoted  $R_t$ . Thus, we impose the following resource availability constraint:

$$\sum_{p \in P} \sum_{m \in M} \sum_{s \in F_p} g_{s,p} \omega_{p,m}^{s,t} R_{p,m} \leq R_t, \quad \forall t \in T. \quad (2.36)$$

The objective function of the leader problem is to minimize the total network travel time over the planning horizon. For each project pattern  $\sigma$  there exists a unique corresponding network link state vector  $\mathbf{m}$ . Formally, if pattern  $\sigma$  is selected at time  $t$ , i.e.  $k_\sigma^t = 1$ , then the set of links  $\{a \in A_p : p \in P\}$  are in state  $m$  if  $\gamma_{p,m,\sigma} = 1$ .

We can now present the complete formulation of the proposed bilevel network recovery problem. The leader problem is summarized in (2.37). The leader variables are project start times  $g_{s,p}$  and pattern selection variables  $k_\sigma^t$ . The follower problem is only implicitly represented by the network travel time  $TSTT(\sigma)(\sigma)$  in the objective function of the leader.

$$\min \quad \sum_{t \in T} \sum_{\sigma \in \Sigma} k_\sigma^t TSTT(\sigma)(\sigma), \quad (2.37a)$$

$$\text{s.t.} \quad \sum_{s \in F_p} g_{s,p} = 1, \quad \forall p \in P, \quad (2.37b)$$

$$\sum_{\sigma \in \Sigma} k_\sigma^t = 1, \quad \forall t \in T, \quad (2.37c)$$

$$\sum_{\sigma \in \Sigma} k_\sigma^t \gamma_{p,m,\sigma} = \sum_{s \in F_p} g_{s,p} \omega_{p,m}^{s,t}, \quad \forall p \in P, \forall m \in M, \forall t \in T, \quad (2.37d)$$

$$\sum_{p \in P} \sum_{m \in M} \sum_{s \in F_p} g_{s,p} \omega_{p,m}^{s,t} R_{p,m} \leq R_t, \quad \forall t \in T, \quad (2.37e)$$

$$g_{s,p} \in \{0, 1\}, \quad \forall p \in P, s \in F_p, \quad (2.37f)$$

$$k_\sigma^t \in \{0, 1\}, \quad \forall t \in T, \sigma \in \Sigma. \quad (2.37g)$$

We detail now how the required model parameters, i.e. binary matrices  $\gamma_{p,m,\sigma}$  and  $\omega_{p,m}^{s,t}$ , and recovery resources  $R_{p,m}$ , can be determined based on the number of states  $|M|$ . We consider a three-states case with the states *damaged*, *repair* and *restored*. In *damaged* state, links are deteriorated and perform accordingly. Then, links evolve to the *repair* state for  $D_p$  time periods. Finally, links evolve to the *restored* state in which they recover their original link performance characteristics. In this three-states case, we assume that  $R_{p,\text{damaged}} = R_{p,\text{restored}} = 0$  and that  $R_{p,\text{repair}} > 0$ , thus ensuring that recovery resources are only consumed during the  $D_p$  time periods when  $p$  is active. The matrix  $\omega_{p,m}^{s,t}$  can then be determined by partitioning the time line in three segments corresponding to the three states of the project lifecycle.

$$\omega_{p,\text{damaged}}^{s,t} \equiv \begin{cases} 1 & \text{if } 0 \leq t \leq s - 1, \\ 0 & \text{otherwise,} \end{cases} \quad \forall p \in P, \forall s \in F_p, \forall t \in T. \quad (2.38)$$

$$\omega_{p,\text{repair}}^{s,t} \equiv \begin{cases} 1 & \text{if } s \leq t \leq s + D_p - 1, \\ 0 & \text{otherwise,} \end{cases} \quad \forall p \in P, \forall s \in F_p, \forall t \in T. \quad (2.39)$$

$$\omega_{p,\text{restored}}^{s,t} \equiv \begin{cases} 1 & \text{if } s + D_p \leq t \leq |T|, \\ 0 & \text{otherwise,} \end{cases} \quad \forall p \in P, \forall s \in F_p, \forall t \in T. \quad (2.40)$$

The choice regarding the number of states used in the model depends on the ability to estimate the impacts during each state at the time that the scheduling of recovery projects is determined. In our numerical experiments, we consider a three-state case where  $M =$

---

{damaged, repair, restored} but due to the lack of data on link performance characteristics post-disaster and during reconstruction, we assume that links in either state *damaged* or *repair* behave identically. Thus, in our numerical experiments, although  $|M| = 3$ , only  $2^{|P|}$  project patterns are used to model the network recovery problem.

## 2.5.2 Scheduling heuristics and numerical experiments

We propose three heuristic algorithms for the bilevel network recovery problem that build on existing methods in the scheduling literature [Smith, 1956]. All three heuristics are greedy algorithms that differ in the criterion used to sort recovery projects. Once sorted, recovery projects are scheduled as soon as possible while respecting the resource availability constraint (2.36). It should be noted that these heuristics are not guaranteed to find a feasible solution if one exists.

### Shortest Processing Time (SPT)

A natural scheduling heuristic is to sort recovery projects by increasing duration  $D_p$ . This is known as the Shortest Processing Time (SPT) rule [Smith, 1956]. Using the SPT rule is intuitive in that by completing shortest projects first, the benefits of recovering the associated nominal link characteristics are expected to yield network-wide improvements early in the planning horizon. This heuristic does not require any knowledge of the congestion effects associated with damaged links and their reconstruction.

### Largest Average First-Order (LAFO)

To incorporate congestion effects within the scheduling process, we can estimate the average first-order effects of each recovery project. The first-order effects of a recovery project represent the network travel time reduction observed before and after this project is completed. These effects depend on the completion of other recovery projects, hence calculating exactly the average first-order effect of a recovery project requires comparing network travel time for all  $|M|^{|P|-1}$  combinations of other recovery projects. We denote  $\Delta_p$  the distribution of first-order effects for project  $p \in P$  and we denote  $\bar{\Delta}_p$  the average first-order effect of  $p$ . The Largest Average First-Order (LAFO) heuristic consists of sorting projects by decreasing average first-order effects  $\bar{\Delta}_p$ . This heuristic is expected to complete the most impactful (in terms of network travel time reduction) projects first, thus benefiting the entire network for subsequent time periods.

### Largest Approximated Smith's Ratio (LASR)

Smith's ratio, also known as weighted-SPT, consists of sorting projects by decreasing "weight"-to-duration ratio [Smith, 1956]. This rule is optimal for the problem of minimizing the total weighted completion time in a single-machine scheduling problem. In the context of the bilevel network recovery scheduling problem, the "weight" of each project is unknown a priori due to the nonlinear congestion effects inherent to traffic equilibrium. A possible adaptation of Smith's ratio consists of using projects' average first-order network travel time effects to approximate the "weight" of each project. Hence, we sort projects by decreasing  $\frac{\bar{\Delta}_p}{D_p}$  values.

The LASR heuristic requires as much computational effort as the LAFO heuristic but captures both projects impact and duration in the decision process. Both LAFO and LASR heuristics require the calculation of project’s average first-order effects, which require the solution of  $|M|^{|P|-1}$  TAPs. To avoid the possibly resource-intensive computational effort, heuristics algorithms can be used to approximate the exact average first-order effects by only solving a subset of the required TAPs. We use Algorithm 1 to sample subsets of the  $|M|^{|P|-1}$  combinations and estimate first-order effects. In our numerical experiments, we compare the behavior of the LAFO and LASR heuristics where all project patterns are used to determine the average first-order effects; and the more practical case where only a subset of project patterns is used for this calculation.

We next report numerical results for the proposed formulation and solution methods. All numerical experiments are conducted on a desktop computer with 16Gb of RAM and a CPU of 3.6GHz. The optimal solution of the bilevel network recovery scheduling problem is obtained by solving the leader problem (2.37) using Mixed-Integer Linear Programming (MILP) commercial software after having solved all  $|M|^{|P|}$  TAPs using TAPAS [Bar-Gera, 2010]. The MILPs are solved with CPLEX 12.8 Python API.

### 2.5.3 Numerical results

We conduct numerical experiments on two types of disaster scenario: a **corridor scenario** where all links of a corridor of the network are damaged, and a **crossing scenario** where all links of cross-section of the network are damaged. The motivation behind both disaster scenarios is that a natural disaster such as an earthquake may damage the network within a specific topological region. In the proposed corridor and crossing scenarios, the topology of the damaged region is assumed to follow a fault-line, i.e. longitudinal, pattern. We use the Berlin Mitte Centre network which contains 36 centroids, 398 nodes and 871 links to benchmark the proposed heuristics. This network is available at <https://github.com/bstabler/TransportationNetworks>.

For the corridor scenario, damaged links are organized into 10 recovery projects and all damaged links are assumed to be affected by a 95% reduction in link capacity. For the crossing scenario, damaged links are organized into 9 recovery projects and all damaged links are assumed to be affected by a 90% reduction in link capacity. In both scenarios, the free-flow travel times of damaged links are ten times greater than their default values (additional link characteristics are provided with the transport network data). Two maps depicting the damaged and recovery projects in the corridor and crossing scenarios are provided in Figure 2.2.

The planning horizon is set to 52 time periods (in weeks) and we assume that project durations are uniformly distributed in the range  $\{1, \dots, 26\}$ , i.e. the maximum project duration is half of the planning horizon. We assume that all projects require a unit resource per time period, i.e.  $R_p = 1$  for all  $p \in P$ . The available recovery resources  $R_t$  are assumed to remain constant across time periods.

For both corridor and crossing scenarios, we randomly generate 10 instances (total of 20 instances). For each instance, we vary the available amount of recovery resources  $R_t$  from 3 to 9 (corridor) or 8 (crossing). Note that in this case,  $R_t$  represents the maximum number of projects that can be *under repair* in parallel. The lower bound 3 was determined by observing that the problem is often infeasible if  $R_t < 3, \forall t \in T$ .

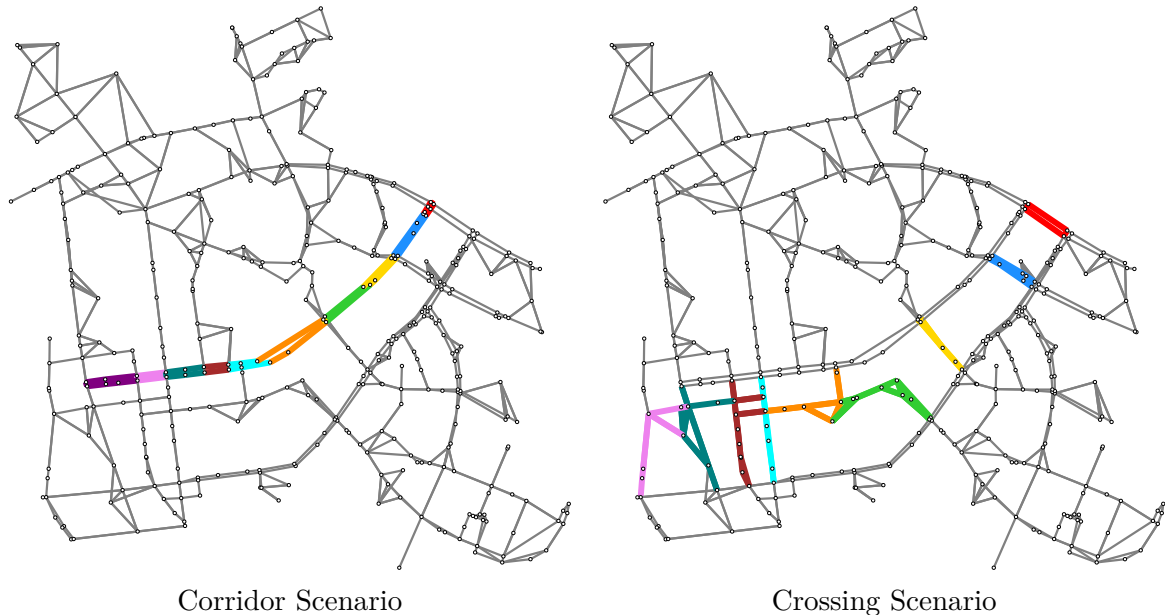


Figure 2.2: Map of the Berlin Mitte Centre network with colored recovery projects for the corridor and crossing scenarios.

TAPs on the Berlin Mitte Centre network solved with TAPAS require on average 1s per TAP, hence the pre-processing time required to implement the MILP and to exactly calculate projects' first-order effects is on average  $2^{10} = 1,024s$  for the Corridor scenario, and  $2^9 = 512s$  for the Crossing scenario. In contrast, implementing heuristics LAFO and LASR using the sampling algorithm of [Rey et al. \[2019\]](#) to approximate project's average first-order effects requires the solution of 113 and 88 TAPs, for the Corridor and Crossing scenarios, respectively. Further, in all our numerical experiments, the run time of the heuristics (without pre-processing) is significantly less than 0.1 s. This can be expected, since the heuristics essentially consists of sorting a list of  $|P|$  values by a provided criterion (SPT, LAFO or LASR) and creating a schedule based on this sorted list.

All numerical results are presented in terms of normalized network travel time. Let  $TSTT(\sigma)(\sigma_0)$  be the network travel time corresponding to the pattern  $\sigma_0$  wherein all recovery projects have been completed, i.e.  $\gamma_{p,\text{restored},\sigma_0} = 1$ ,  $\gamma_{p,\text{damaged},\sigma_0} = \gamma_{p,\text{repair},\sigma_0} = 0$ ,  $\forall p \in P$ . The normalized network travel time of pattern  $\sigma$  is  $TSTT(\sigma)(\sigma) = \frac{TSTT(\sigma)(\sigma)}{TSTT(\sigma)(\sigma_0)}$ .

The average performance of the heuristics is summarized in [Table 2.7](#). For each type of disaster (Corridor and Crossing) and for each level of recovery resources ( $R_t$ ), we report the minimum (min), average (avg) and maximum (max) optimality gap over all 10 instances of this group. We also report how frequently each heuristic provides the best optimality gap (best), and how frequently each heuristic is unable to produce a feasible solution (inf). Average performance over all levels of recovery resources is reported in the last three rows.

LAFO-all is the heuristic which most frequently provides the best optimality gap for Corridor instances: this heuristic dominates other heuristics 62.86% of the time. For Crossing instances, LASR-all dominates 63.33% of them time. LAFO-all is the heuristic less frequently infeasible in Crossing scenario instances, whereas LASR-all outperforms other heuristics in this regard on Corridor scenario instances. However, we find that, on average, LASR is the most robust heuristic and achieves an overall average optimality gap of 0.52% for the Corridor



$R_t$	Heuristic	Type	Corridor					Crossing				
			min	avg	max	best	inf	min	avg	max	best	inf
3	SPT		1.51	81.84	100.00	60.00	80.00	0.16	41.16	100.00	50.00	30.00
	LAFO	All	0.71	76.02	100.00	60.00	70.00	0.83	59.17	100.00	30.00	20.00
		(sampling)	(0.85)	(81.13)	(100.00)	(80.00)	(80.00)	(1.23)	(26.74)	(100.00)	(90.00)	(10.00)
4	LASR	All	0.43	60.38	100.00	100.00	60.00	0.22	30.61	100.00	60.00	30.00
		(sampling)	(0.06)	(60.29)	(100.00)	(70.00)	(60.00)	(0.65)	(38.21)	(100.00)	(50.00)	(30.00)
	SPT		0.33	2.75	9.65	30.00	0.00	0.11	8.70	28.07	30.00	0.00
5	LAFO	All	0.16	12.33	36.48	30.00	0.00	0.36	15.21	52.42	0.00	0.00
		(sampling)	(0.07)	(12.31)	(36.39)	(60.00)	(0.00)	(0.84)	(6.83)	(18.98)	(50.00)	(0.00)
	LASR	All	0.22	1.02	3.23	40.00	0.00	0.00	0.46	0.87	70.00	0.00
6		(sampling)	(0.07)	(1.33)	(8.00)	(50.00)	(0.00)	(0.32)	(4.96)	(24.20)	(10.00)	(0.00)
	SPT		0.08	1.68	5.78	20.00	0.00	0.00	3.90	20.13	50.00	0.00
	LAFO	All	0.26	1.35	8.36	60.00	0.00	0.17	3.40	30.32	30.00	0.00
7		(sampling)	(0.14)	(1.32)	(8.49)	(60.00)	(0.00)	(0.24)	(0.70)	(2.07)	(30.00)	(0.00)
	LASR	All	0.09	0.57	1.36	40.00	0.00	0.00	0.35	0.73	40.00	0.00
		(sampling)	(0.14)	(1.36)	(8.36)	(50.00)	(0.00)	(0.03)	(3.52)	(16.16)	(50.00)	(0.00)
8	SPT		0.17	0.74	1.72	20.00	0.00	0.00	1.79	16.00	50.00	0.00
	LAFO	All	0.10	0.82	3.85	70.00	0.00	0.00	3.11	29.85	50.00	0.00
		(sampling)	(0.14)	(0.82)	(4.18)	(40.00)	(0.00)	(0.09)	(0.28)	(0.75)	(30.00)	(0.00)
9	LASR	All	0.10	0.57	1.54	10.00	0.00	0.00	0.17	0.39	40.00	0.00
		(sampling)	(0.14)	(0.85)	(3.95)	(60.00)	(0.00)	(0.01)	(0.25)	(0.57)	(50.00)	(0.00)
	SPT		0.12	0.35	0.82	30.00	0.00	0.00	0.13	0.45	20.00	0.00
10	LAFO	All	0.05	0.43	1.34	60.00	0.00	0.00	0.06	0.36	60.00	0.00
		(sampling)	(0.15)	(0.82)	(3.94)	(40.00)	(0.00)	(0.00)	(0.06)	(0.36)	(100.00)	(0.00)
	LASR	All	0.06	0.42	1.34	20.00	0.00	0.00	0.01	0.10	90.00	0.00
11		(sampling)	(0.00)	(0.41)	(1.40)	(70.00)	(0.00)	(0.00)	(0.13)	(0.36)	(60.00)	(0.00)
	SPT		0.08	0.23	0.61	30.00	0.00	0.00	0.03	0.17	30.00	0.00
	LAFO	All	0.02	0.27	1.08	70.00	0.00	0.00	0.00	0.01	80.00	0.00
12		(sampling)	(0.00)	(0.48)	(3.74)	(70.00)	(0.00)	(0.00)	(0.00)	(0.01)	(100.00)	(0.00)
	LASR	All	0.07	0.31	1.08	20.00	0.00	0.00	0.00	0.01	80.00	0.00
		(sampling)	(0.00)	(0.21)	(0.90)	(90.00)	(0.00)	(0.00)	(0.00)	(0.01)	(100.00)	(0.00)
13	SPT		0.02	0.11	0.18	40.00	0.00	-	-	-	-	-
	LAFO	All	0.00	0.06	0.28	90.00	0.00	-	-	-	-	-
		(sampling)	(0.00)	(0.06)	(0.28)	(100.00)	(0.00)	-	-	-	-	-
14	LASR	All	0.00	0.07	0.28	80.00	0.00	-	-	-	-	-
		(sampling)	(0.00)	(0.07)	(0.28)	(90.00)	(0.00)	-	-	-	-	-
	SPT		<b>0.02</b>	<b>1.24</b>	<b>16.86</b>	<b>32.86</b>	<b>11.43</b>	<b>0.00</b>	<b>4.51</b>	<b>43.91</b>	<b>38.33</b>	<b>5.00</b>
avg	LAFO	All	<b>0.00</b>	<b>3.38</b>	<b>36.48</b>	<b>62.86</b>	<b>10.00</b>	<b>0.00</b>	<b>10.51</b>	<b>82.11</b>	<b>41.67</b>	<b>3.33</b>
		(sampling)	(0.00)	(2.73)	(36.39)	(64.29)	(11.43)	(0.00)	(4.17)	(65.87)	(66.67)	(1.67)
	LASR	All	<b>0.00</b>	<b>0.52</b>	<b>3.23</b>	<b>44.29</b>	<b>8.57</b>	<b>0.00</b>	<b>0.28</b>	<b>1.86</b>	<b>63.33</b>	<b>5.00</b>
	(sampling)	(0.00)	(0.71)	(8.36)	(68.57)	(8.57)	(0.00)	(3.00)	(28.21)	(53.33)	(5.00)	

Table 2.7: Summary of the performance of heuristics SPT, LAFO and LASR. All data is in percentage. For each type of disaster scenario (Corridor and Crossing) and each level of recovery resources ( $R_t$ ) the minimum (min), average (avg) and maximum (max) optimality gap over all 10 instances of this group is reported. Columns “best” indicate the frequency that this heuristic was the best (least optimality gap), and columns “inf” indicate the frequency that this heuristic did not produced a feasible solution. The bottom five rows indicate average performance over all values of recovery resources  $R_t$ .

---

scenario instances, and of 0.28% for the Crossing scenario instances, followed by SPT. Hence while LAFO may frequently perform best, it also is more prone to poor performance. This is reflected in the average maximum optimality gaps observed which are substantially higher for this heuristic compared to that of SPT and LASR. If only a subset of patterns are used to approximate project’s first-order effects, we find that the maximum optimality gaps increase but average performance is marginally impacted for LASR. In contrast, we find that LAFO on average benefits from this approximation, which suggests while first-order effects are critical as highlighted by the good performance of LASR, their calculation and incorporation in the proposed heuristics require further research.

## 2.6 Conclusion

The DNDP is a challenging bilevel optimization problem with critical implications for network design in urban transport systems. Over the past decade, exact methods have become more capable to solve non-trivial decisions problems. We synthesized the literature on exact methodologies and discussed the main approaches developed to solve the DNDP to optimality. In total, nine papers have been examined. Although some of these papers have not addressed the DNDP in its present form, they have presented a methodology for a related problem which can be reduced to the DNDP. The characteristics of the solution methods examined have been categorized in two types of approaches: SO-relaxation based and KKT conditions based methods. A computational benchmarking of the DNDP was conducted by adapting three solution methods and comparing their performance using the same piecewise linear link travel time function approximation. For this benchmark, a total of 20 new instances for the DNDP have been created. For reproducibility purposes, all implementation formulations, codes and benchmarking instances are available at <https://github.com/davidrey123/DNDP>.

The problem of scheduling maintenance projects in congested transport networks can be viewed as a multi-period application of the DNDP. TSTT can be used as a metric to represent the impact of road capacity reductions onto network performance by accounting for users’ reaction in the schedule optimization. A novel pattern-based formulation has been proposed which extends previous work by providing a compact MIP representation based on the enumeration of project patterns. An exponential regression model was introduced to approximate network congestion effects based on maintenance projects’ first order TSTT effects. This model is used in combination with a pre-processing algorithm to explore the project pattern space based on the concept of 1-distance neighborhood which proved to be very effective in capturing projects’ impact on network congestion. We showed that the integer-relaxation of the proposed statistical model is convex and that this relaxation provides an efficient method to approximate TSTT values. MILP models to find feasible solutions as well as customized branching and node selection rules are proposed and calibrated. A CG algorithm was then presented to solve the NMSP on large instances where pre-processing all TSTT values is intractable. The proposed CG algorithm relies on a fast pricing heuristic based on 1-distance optimality, approximate lower bounds derived from convex approximations of the NMSP and an explicit search procedure. The proposed B&P algorithm was tested on instances derived from 15- and 20-project scenarios on three network topologies with 25 and 50 time periods.

Our implementation demonstrated that the proposed B&P algorithm is able to solve all instances in reasonable time and by using only a fraction of the project patterns during the exploration. This is an important result since it suggests that competitive solutions can be

found efficiently even on large instances. From a runtime perspective, the bottleneck of the algorithm is the time spent solving the TAP and the time spent in the explicit search procedure. While the former depends on the network size, the latter depends on the number of projects considered. Both of these steps may be challenging from a computational standpoint and additional work is necessary to identify more efficient approaches to manage this complexity.

The pattern-based formulation for the NMSP was then extended to incorporate network recovery effects. Three heuristics employed from the scheduling literature were adapted for this bilevel network recovery problem. Numerical experiments were designed on a realistic transport network wherein two disaster scenarios were considered. Both disaster scenarios aim to represent the possible damage caused by disasters that exhibit a fault-line topology, such as earthquakes. A total of 20 instances based on these disaster scenarios were randomly generated and used to benchmark the proposed solution methods. The numerical results obtained highlight the performance of the proposed adaptation of Smith's ratio for the bilevel network recovery scheduling problem at hand, but also reveal that the proposed heuristic methods may fall short of optimal solutions when the amount of recovery resources is low compared to the demand of recovery projects. This research provides a basis to support decision-making in the context of disaster recovery. The proposed formulation focused on the reconstruction of the road network under limited recovery resources availability which can help in informing planning policies.

## Chapter 3

# Aircraft conflict resolution via mixed-integer nonlinear programming

This chapter is based on [Rey and Hijazi \[2017\]](#), [Dias et al. \[2022\]](#), [Dias and Rey \[2022\]](#).

### 3.1 Introduction

Air traffic control (ATC) is an extremely dynamic and constrained environment where many decisions need to be taken in a short amount of time. Adopting automation within such an environment can be vital to reduce controller workload and improve airspace capacity [[Durand et al., 1997](#), [Barnier and Allignol, 2009](#), [Rey et al., 2016c](#)]. Traditional methods for air traffic control have been exhaustively used and are reaching their limits, hence automated approaches are receiving a significant and growing attention in the field [[Vela et al., 2009b](#)]. The International Civil Aviation Organization (ICAO) determines all regulations related to civil aviation [[ICAO, 2010](#)]. One of its main roles is to set separation standards for commercial aviation. We focus on aircraft separation for en-route traffic. During cruise stage, separation conditions require a minimum of 5 Nautical Miles (NM) horizontally or 1000 feet (ft) vertically between any pair of aircraft. A conflict between two or more aircraft is a loss of separation among these aircraft. Air traffic networks are organised in flight levels which are separated by at least 1000 ft, hence during cruise stage, most conflicts occur among aircraft flying at the same flight level. Congested air traffic networks can lead to loss of separation between aircraft which impairs flight safety and may result in collisions. The aircraft conflict resolution problem (ACRP) can be formulated as an optimization problem in which the objective is to find least-deviating conflict-free trajectories for a set of aircraft. Different strategies have been used to address this problem based on the type of deconfliction manoeuvres available, namely: speed control (acceleration or deceleration), heading control, vertical control (flight level reassignment) or a combination of these manoeuvres.

This chapter presents the complex number formulation for aircraft conflict resolution introduced by [Rey and Hijazi \[2017\]](#) and further developed by [Dias et al. \[2022\]](#) for the deterministic case, and by [Dias and Rey \[2022\]](#) for the robust case. Nonconvex mixed-integer programming (MIP) formulations for ACRPs and exact solution methods are proposed

for the two-dimensional (2D) conflict resolution problem with speed and heading control represented as continuous variables; as well as for an extended problem wherein flight level (FL) changes are also available. Theoretical and numerical contributions are provided, and extensive numerical results on benchmarking instances are reported. For reproducibility purposes, all formulations and instances are made available online at the public repository <https://github.com/acrp-lib/acrp-lib>.

The chapter is organised as follows. We review the state-of-the-art on aircraft conflict resolution, highlight existing research gaps and position our contributions to the field in Section 3.2. We next characterise pairwise aircraft separation conditions and present nonconvex aircraft conflict resolution formulations for the 2D problem, as well as for an extension of the 2D problem with FL changes, hereby referred to as 2D+FL in Section 3.3. We present exact solution methods for the 2D and 2D+FL problems that build on existing convex relaxations and propose novel decomposition and constraint generation algorithms in Section 3.4. The robust aircraft conflict resolution problem is discussed in Section 3.5. Numerical results are provided in Section 3.6 and concluding remarks are discussed in Section 3.7.

## 3.2 Literature review

The first exact global optimization approach for aircraft conflict resolution problems was proposed by Pallottino et al. [2002], who introduced two MIP formulations: a first model based on speed control only and a second model based on heading control only which required that all aircraft fly at the same speed. In the proposed MIP formulation for conflict resolution with speed control, the authors derived linear pairwise aircraft separation constraints based on the geometric construction introduced by Bilimoria [2000]. These separation conditions are obtained by projecting the shadow of an intruder aircraft onto the trajectory of a reference aircraft, thus we henceforth refer to these separation conditions as the shadow separation conditions. Frazzoli et al. [2001] were the first to observe that this geometric construction provided a basis to characterise the set of aircraft pairwise conflict-free trajectories via linear half-planes in the relative velocity (speed and heading) plane. The authors introduced a nonconvex formulation for the conflict resolution problem with speed and heading control, and proposed a convex relaxation based on semi-definite programming as well as a heuristic algorithm to find feasible solutions on problems with up to 10 aircraft.

The shadow separation conditions were subsequently used in several formulations. Alonso-Ayuso et al. [2011] proposed a mixed-integer linear programming (MILP) formulation for conflict resolution by speed and altitude control and reported solving instances with up to 50 aircraft in competitive time. Alonso-Ayuso et al. [2014] proposed a two-step approach in which only heading control is available for deconfliction and the available angle changes are discretised. The same group of authors also proposed a nonconvex formulation involving trigonometric functions based on the shadow separation conditions that enables speed, heading and altitude control [Alonso-Ayuso et al., 2016]. The authors used a mixed-integer nonlinear programming (MINLP) solver to solve the resulting nonconvex formulations and reported results for the 2D ACRP with up to 7 aircraft on structured instances and up to 20 aircraft on un-structured instances. Alternative representations of pairwise aircraft separation based on conflict points have been proposed by several authors. In Vela et al. [2009c,b,a], the authors proposed several MILP formulations which aim to minimise fuel consumption, incorporate air traffic controller workload in the objective function, and account for the impact of uncertainty on trajectory prediction due to wind effects. Omer [2015] proposed a

---

space-discretised MILP formulation involving a finite set of turning angles. In contrast to most other approaches, the heading control manoeuvres consist of two actions: a first heading change for collision avoidance and a second turn to recover aircraft’s initial heading. [Rey et al. \[2012, 2016c\]](#) proposed linear upper bounds for the ACRP with speed control only and the resulting MILP formulations can solve realistic large-scale instances to optimality within a few seconds.

More recently, nonlinear global optimization approaches received an increasing attention in the literature. [Omer and Farges \[2013\]](#) proposed a hybrid algorithm which uses the optimal solution of a MILP as the starting point for solving a nonlinear formulation of the same problem. [Cafieri \[2012\]](#) introduced a MINLP approach that provides a compact nonlinear separation condition for conflict resolution. This formulation was then used by [Cafieri and Rey \[2017\]](#) for conflict resolution with speed control only which highlights that subliminal speed control alone may not be sufficient to resolve all conflicts in dense traffic scenarios. Using a similar framework, [Cafieri and Omheni \[2017\]](#) presented a two-step approach where a maximum number of conflicts are first solved using speed control only and outstanding conflicts are solved by heading control. [Cerulli et al. \[2021\]](#) proposed a formulation based on bi-level optimization with multiple follower problems, each of which representing a two-aircraft separation problem. The authors presented two formulations, one using speed control only and another using heading control only. A cut generation algorithm is proposed to solve the corresponding bi-level optimization problems. Recently, [Pelegrín and d’Ambrosio \[2022\]](#) conducted a review of the literature on mathematical programming approaches for aircraft conflict resolution and have shown that the disjunctive linear separation conditions introduced by [Rey and Hijazi \[2017\]](#) and the shadow separation conditions are equivalent. This result combined with the theorem provided in this study – linking the disjunctive separation conditions with the definitional nonlinear separation conditions (see [Theorem 3.1](#)) – further motivates the comparison of both separation conditions in terms of computational performance which we address in our numerical experiments.

This review of the literature highlights that despite recent improvements in the development of optimization approaches for aircraft conflict resolution, there remains significant open challenges in the design of scalable and exact global optimization approaches. The rest of the chapter is organized as follows. Formulations based on disjunctive linear separation conditions for the deterministic aircraft conflict resolution problem are first presented ([Section 3.3](#)) and exact solution methods are then developed ([Section 3.4](#)). An extension to the robust aircraft conflict resolution problem is presented ([Section 3.5](#)) before numerical experiments and results are summarized ([Section 3.6](#)).

### 3.3 Aircraft conflict resolution formulations

We present mixed-integer formulations for the ACRP with speed, heading and altitude control. We first focus on the 2D problem under velocity, i.e. speed and heading, control which aims to represent a single flight level during cruise stage air traffic conditions. We present disjunctive linear separation conditions in [Section 3.3.1](#). We then characterise 2D separation conditions for a pair of aircraft based on velocity control bounds in [Section 3.3.2](#), before introducing a compact nonconvex formulation in [Section 3.3.3](#). We then extend this nonconvex formulation to the case of multiple separated flight levels (FL), denoted 2D+FL in [Section 3.3.4](#).

### 3.3.1 Disjunctive linear separation conditions

Our goal is to find least-deviating conflict-free trajectories for a set of aircraft in cruise stage, i.e. flying at a fixed altitude and at constant speed. We denote  $t = 0$  the time instant representing aircraft current positions at the time of decision, also referred to as aircraft initial positions. We assume that all aircraft are separated at  $t = 0$  and seek to derive separation conditions to ensure that aircraft trajectories are separated for any time  $t \geq 0$ . Let  $\mathcal{A}$  be the set of aircraft. For each  $i \in \mathcal{A}$ ,  $[\hat{x}_i, \hat{y}_i]^\top$  is the aircraft initial position in the 2D plane,  $\hat{v}_i$  is its nominal speed (in NM/h) and  $\hat{\theta}_i$  is its heading angle. Assuming uniform motion laws, aircraft motion can be described as:  $p_i(t) = [x_i(t), y_i(t)]^\top$ , where  $x_i(t) = \hat{x}_i + q_i \hat{v}_i \cos(\hat{\theta}_i + \theta_i)t$  and  $y_i(t) = \hat{y}_i + q_i \hat{v}_i \sin(\hat{\theta}_i + \theta_i)t$ . In this model, the decision variables are  $q_i$ , which is the speed control variable that determines the acceleration or deceleration with regards to the initial speed  $\hat{v}_i$  ( $q_i$  equals to 1 means no speed variation) and  $\theta_i$ , which is the heading control variable that determines the deviation with regards to the initial trajectory ( $\theta_i$  equal to 0 means no deviation in heading angle).

Let  $\mathcal{P} \equiv \{i, j \in \mathcal{A} : i < j\}$  be the set of aircraft pairs, the relative motion of  $(i, j) \in \mathcal{P}$  is denoted  $p_{ij}(t) = p_i(t) - p_j(t)$ . Let  $d$  be the minimum separation distance (e.g. 5 NM). We next define the notion of 2D separation for a pair of aircraft.

**Definition 3.1** (2D separated trajectories). *The trajectories of a pair of aircraft  $(i, j) \in \mathcal{P}$  are said to be 2D separated, i.e. conflict-free, if and only if:*

$$\|p_{ij}(t)\| \geq d, \quad \forall t \geq 0. \quad (3.1)$$

Let  $v_{ij} = v_i - v_j$  be the 2D relative velocity vector of  $(i, j) \in \mathcal{P}$ , i.e.  $v_{ij} = [v_{ij,x}, v_{ij,y}]^\top$ :

$$v_{ij,x} = q_i \hat{v}_i \cos(\hat{\theta}_i + \theta_i) - q_j \hat{v}_j \cos(\hat{\theta}_j + \theta_j), \quad (3.2a)$$

$$v_{ij,y} = q_i \hat{v}_i \sin(\hat{\theta}_i + \theta_i) - q_j \hat{v}_j \sin(\hat{\theta}_j + \theta_j). \quad (3.2b)$$

Aircraft relative velocity equations are linear with regards to speed control variables  $q_i$  and  $q_j$ , but nonlinear with regards to heading control variables  $\theta_i$  and  $\theta_j$ . Expanding the expression in Eq. (3.1) and denoting  $\hat{p}_{ij}$  the relative initial position of aircraft pair  $(i, j) \in \mathcal{P}$ , we obtain a second-order polynomial function:

$$f_{ij}(t) \equiv \|v_{ij}\|^2 t^2 + 2\hat{p}_{ij} \cdot v_{ij} t + \|\hat{p}_{ij}\|^2 - d^2 \geq 0. \quad (3.3)$$

Since the coefficient of the second-order term is positive,  $f_{ij}(t)$  admits a minimum which corresponds to the time instant of minimum separation between aircraft  $i$  and  $j$ . Let  $t_{ij}^{\min}(v_{ij})$  be the function corresponding to this time instant parameterised by aircraft relative velocity vector, then:

$$t_{ij}^{\min}(v_{ij}) = \frac{-\hat{p}_{ij} \cdot v_{ij}}{\|v_{ij}\|^2}. \quad (3.4)$$

Evaluating Eq. (3.3) at  $t_{ij}^{\min}(v_{ij})$  yields a time-independent separation condition [Rey and Hijazi, 2017, Cafieri and Rey, 2017, Cafieri and Omhni, 2017]:

$$f_{ij}(t_{ij}^{\min}(v_{ij})) = \frac{-(\hat{p}_{ij} \cdot v_{ij})^2}{\|v_{ij}\|^2} + \|\hat{p}_{ij}\|^2 - d^2 \geq 0. \quad (3.5)$$

Let  $g_{ij}(v_{ij}) \equiv \|v_{ij}\|^2 f_{ij}(t_{ij}^{\min}(v_{ij}))$ . We abuse notation and rewrite  $g_{ij}(\cdot)$  and  $t_{ij}^{\min}(\cdot)$  as functions of aircraft velocity components  $v_{ij,x}$  and  $v_{ij,y}$ . After expanding and factorising, we obtain:

$$g_{ij}(v_{ij,x}, v_{ij,y}) = v_{ij,x}^2 (\hat{y}_{ij}^2 - d^2) + v_{ij,y}^2 (\hat{x}_{ij}^2 - d^2) - v_{ij,x} v_{ij,y} (2\hat{x}_{ij} \hat{y}_{ij}) \geq 0. \quad (3.6)$$

Assuming aircraft are initially separated, if  $t_{ij}^{\min}(v_{ij,x}, v_{ij,y}) \leq 0$ , then they are diverging and do not incur any risk of future conflict. If  $t_{ij}^{\min}(v_{ij,x}, v_{ij,y}) > 0$  and  $g(v_{ij,x}, v_{ij,y}) \geq 0$ , aircraft are converging but separation is ensured. Otherwise, if  $t_{ij}^{\min}(v_{ij,x}, v_{ij,y}) > 0$  and  $g(v_{ij,x}, v_{ij,y}) < 0$ , there is a loss of separation and aircraft trajectories should be adjusted to avoid it. Hence, pairwise aircraft separation conditions for  $(i, j) \in \mathcal{P}$  can be written as:

$$g_{ij}(v_{ij,x}, v_{ij,y}) \geq 0 \vee t_{ij}^{\min}(v_{ij,x}, v_{ij,y}) \leq 0. \quad (3.7)$$

To linearise the separation condition (3.6) with regards to variables  $v_{ij,x}$  and  $v_{ij,y}$ , we adopt the approach proposed by [Rey and Hijazi \[2017\]](#) and recall the main steps hereafter. Observe that the solutions of the equation  $g_{ij}(v_{ij,x}, v_{ij,y}) = 0$  can be identified by alternatively fixing variable  $v_{ij,x}$  and  $v_{ij,y}$ , and calculating the roots of the resulting single-variable quadratic equations. Isolating each variable, we obtain the discriminants:

$$\begin{cases} \Delta_{v_{ij,x}} = 4d^2 v_{ij,y}^2 (\hat{x}_{ij}^2 + \hat{y}_{ij}^2 - d^2), \\ \Delta_{v_{ij,y}} = 4d^2 v_{ij,x}^2 (\hat{x}_{ij}^2 + \hat{y}_{ij}^2 - d^2). \end{cases} \quad (3.8)$$

Assuming aircraft are initially separated, then  $\hat{x}_{ij}^2 + \hat{y}_{ij}^2 - d^2 \geq 0$  and the discriminants are positive, and the roots of equation  $g_{ij}(v_{ij,x}, v_{ij,y}) = 0$  are the lines defined by the system of equations:

$$(\hat{y}_{ij}^2 - d^2)v_{ij,x} - (\hat{x}_{ij}\hat{y}_{ij} + d\sqrt{\hat{x}_{ij}^2 + \hat{y}_{ij}^2 - d^2})v_{ij,y} = 0, \quad (3.9a)$$

$$(\hat{y}_{ij}^2 - d^2)v_{ij,x} - (\hat{x}_{ij}\hat{y}_{ij} - d\sqrt{\hat{x}_{ij}^2 + \hat{y}_{ij}^2 - d^2})v_{ij,y} = 0, \quad (3.9b)$$

$$(\hat{x}_{ij}^2 - d^2)v_{ij,y} - (\hat{x}_{ij}\hat{y}_{ij} + d\sqrt{\hat{x}_{ij}^2 + \hat{y}_{ij}^2 - d^2})v_{ij,x} = 0, \quad (3.9c)$$

$$(\hat{x}_{ij}^2 - d^2)v_{ij,y} - (\hat{x}_{ij}\hat{y}_{ij} - d\sqrt{\hat{x}_{ij}^2 + \hat{y}_{ij}^2 - d^2})v_{ij,x} = 0. \quad (3.9d)$$

Let us emphasise that if all coefficients in (3.9a)-(3.9d) are non-zero, then (3.9a) is identical to (3.9c) and (3.9b) is identical to (3.9d). Observe that

$$\begin{aligned} & \hat{x}_{ij}\hat{y}_{ij} \pm d\sqrt{\hat{x}_{ij}^2 + \hat{y}_{ij}^2 - d^2} = 0, \\ \Rightarrow & \quad d^2(\hat{x}_{ij}^2 + \hat{y}_{ij}^2 - d^2) = \hat{x}_{ij}^2 \hat{y}_{ij}^2, \\ \Leftrightarrow & \quad \hat{x}_{ij}^2 \hat{y}_{ij}^2 - d^2(\hat{x}_{ij}^2 + \hat{y}_{ij}^2 - d^2) = 0, \\ \Leftrightarrow & \quad (\hat{x}_{ij}^2 - d^2)(\hat{y}_{ij}^2 - d^2) = 0. \end{aligned}$$

Eqs. (3.9a), (3.9b), (3.9c) and (3.9d) define two lines, denoted  $R_1$  and  $R_2$ , in the plane  $\{(v_{ij,x}, v_{ij,y}) \in \mathbb{R}^2\}$  and the sign of  $g_{ij}(v_{ij,x}, v_{ij,y})$  can be characterised based on the position of  $(v_{ij,x}, v_{ij,y})$  relative to these lines. Recall that according to Eq. (3.4), the sign of the dot



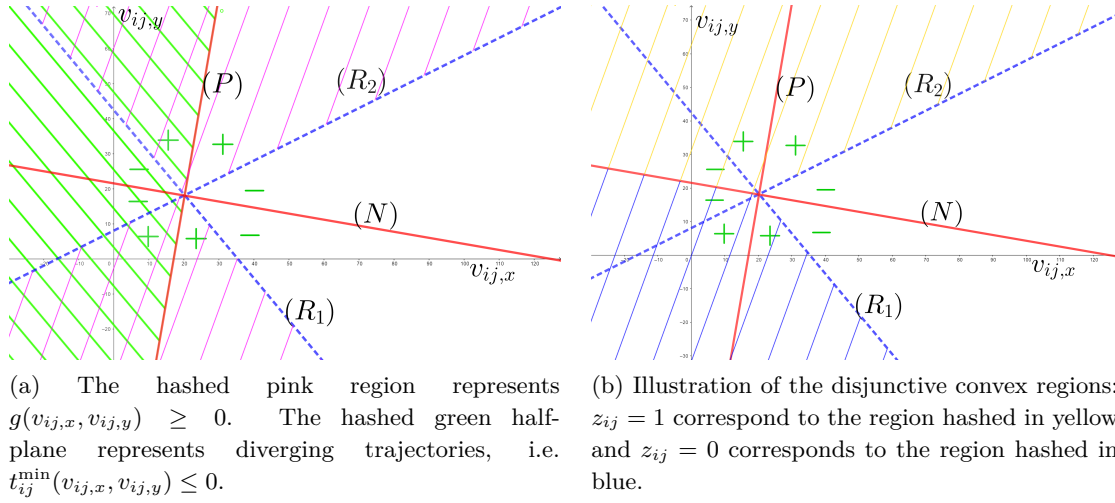


Figure 3.1: Illustration of a two-aircraft conflict in the plane  $\{(v_{ij,x}, v_{ij,y}) \in \mathbb{R}^2\}$ . The red lines represent the lines  $P$  and  $N$ . The dashed blue lines correspond to the linear equations  $R_1$  and  $R_2$  that are the roots of  $g_{ij}(v_{ij,x}, v_{ij,y}) = 0$ . The sign of  $g_{ij}(v_{ij,x}, v_{ij,y})$  is shown by the + and - green symbols.

product  $\hat{p}_{ij} \cdot v_{ij}$  indicates aircraft convergence or divergence. Let  $(P)$  be the equation of the line corresponding to  $\hat{p}_{ij} \cdot v_{ij}$ :

$$v_{ij,x}\hat{x}_{ij} + v_{ij,y}\hat{y}_{ij} = 0. \quad (P)$$

The line defined by  $(P)$  splits the plane  $\{(v_{ij,x}, v_{ij,y}) \in \mathbb{R}^2\}$  in two half-planes, each of which representing converging and diverging trajectories, respectively. This is illustrated in Figure 3.1a which depicts a two-aircraft conflict in the plane  $\{(v_{ij,x}, v_{ij,y}) \in \mathbb{R}^2\}$ . The sign of  $g_{ij}(v_{ij,x}, v_{ij,y})$  is shown by the + and - green symbols and the hashed pink region corresponds to  $g_{ij}(v_{ij,x}, v_{ij,y}) \geq 0$ . The hashed green half-plane delimited by  $(P)$  represents diverging trajectories, i.e.  $t_{ij}^{\min}(v_{ij,x}, v_{ij,y}) \leq 0$ .

Consider the line normal to  $(P)$ , denoted  $(N)$ :

$$v_{ij,y}\hat{x}_{ij} - v_{ij,x}\hat{y}_{ij} = 0. \quad (N)$$

Recall that any point  $(v_{ij,x}, v_{ij,y})$  such that  $t_{ij}^{\min} \leq 0$  or  $g_{ij}(v_{ij,x}, v_{ij,y}) \geq 0$  corresponds to a pair of conflict-free trajectories. Hence, the conflict-free region is nonconvex and is represented by the union of the green and pink hashed regions in Figure 3.1a. Reciprocally, the conflict region, corresponding to conflicting trajectories is convex and represented by the non-hashed region in Figure 3.1a. An equivalent expression of Eq. (3.9) was proposed by Frazzoli et al. [2001] which observed that the set of conflict-free trajectories could be characterised by the union of two half-planes. We next show through Lemmas 3.1 and 3.2 that  $(N)$  is a bisector of the angle formed by lines  $R_1$  and  $R_2$  in the conflict region ( $g_{ij}(v_{ij,x}, v_{ij,y}) \leq 0$ ) and can be used to generate two disjunctive but convex conflict-free regions.

**Lemma 3.1.** *The lines  $(P)$  and  $(N)$  are bisectors of the angles formed by the two lines  $R_1$  and  $R_2$  representing the solutions of  $g_{ij}(v_{ij,x}, v_{ij,y}) = 0$ .*

**Lemma 3.2.**  *$g_{ij}(v_{ij,x}, v_{ij,y}) \leq 0$  for all points  $(v_{ij,x}, v_{ij,y})$  of the normal line  $(N)$ .*

Lemmas 3.1 and 3.2 assert that (N) can be used to split the conflict-free region into two convex but disjunctive regions. We model this disjunction using variable  $z_{ij} \in \{0, 1\}$  defined as:

$$v_{ij,y}\widehat{x}_{ij} - v_{ij,x}\widehat{y}_{ij} \leq 0, \quad \text{if } z_{ij} = 1, \quad \forall (i, j) \in \mathcal{P}, \quad (3.10a)$$

$$v_{ij,y}\widehat{x}_{ij} - v_{ij,x}\widehat{y}_{ij} \geq 0, \quad \text{if } z_{ij} = 0, \quad \forall (i, j) \in \mathcal{P}. \quad (3.10b)$$

In each convex sub-region, the lines defined by (3.9a)-(3.9d) delineate the conflict-free region. The expressions of these lines depend on aircraft initial positions, i.e.  $\widehat{x}_{ij}, \widehat{y}_{ij}$ . Recall that we denote  $R_1$  and  $R_2$  the equation of these lines. Integer-linear separation conditions with regards to aircraft velocity components can be derived as follows:

$$v_{ij,y}\gamma_{ij}^l - v_{ij,x}\phi_{ij}^l \leq 0, \quad \text{if } z_{ij} = 1, \quad \forall (i, j) \in \mathcal{P}, \quad (3.11a)$$

$$v_{ij,y}\gamma_{ij}^u - v_{ij,x}\phi_{ij}^u \geq 0, \quad \text{if } z_{ij} = 0, \quad \forall (i, j) \in \mathcal{P}, \quad (3.11b)$$

where  $\gamma_{ij}^l, \phi_{ij}^l$  and  $\gamma_{ij}^u, \phi_{ij}^u$  are coefficients of the lines (3.9a)-(3.9d) corresponding to the roots of  $g_{ij}(v_{ij,x}, v_{ij,y}) = 0$ . The proposed linear disjunction is illustrated in Figure 3.1b which depicts the resulting convex sub-regions corresponding to the disjunction  $z_{ij} \in \{0, 1\}$  for a two-aircraft conflict. This leads to the following result.

**Theorem 3.1.** *The disjunctive linear separation conditions (3.10)-(3.11) fully characterises the set of aircraft pairwise conflict-free trajectories as given by Eq. (3.1)*

Theorem 3.1 asserts that the disjunctive separation conditions (3.10)-(3.11) are equivalent to the definitional nonlinear separation conditions. Further, these disjunctive separation conditions are linear with regards to aircraft velocity variables  $v_{ij,x}$  and  $v_{ij,y}$ , and only require a single binary variable per pair of aircraft. This is expected to improve on the so-called shadow separation conditions which are also linear with regards to aircraft velocity variables, but require four binary variables per pair of aircraft [Pallottino et al., 2002, Alonso-Ayuso et al., 2011, 2016].

### 3.3.2 Characterisation of conflict-free trajectories based on velocity control bounds

To characterise the set of 2D conflict-free trajectories, we examine the relative velocity vector  $v_{ij}$  as a function of aircraft trajectory control bounds. For each aircraft  $i \in \mathcal{A}$ , we assume that the speed rate variable is lower bounded by  $\underline{q}_i$  and upper bounded by  $\bar{q}_i$ , i.e.

$$\underline{q}_i \leq q_i \leq \bar{q}_i, \quad \forall i \in \mathcal{A}. \quad (3.12)$$

We assume that the heading deviation is lower and upper bounded by  $\underline{\theta}_i$  and  $\bar{\theta}_i$ , i.e.

$$\underline{\theta}_i \leq \theta_i \leq \bar{\theta}_i, \quad \forall i \in \mathcal{A}. \quad (3.13)$$

To derive lower and upper bounds on relative velocity components  $v_{ij,x}$  and  $v_{ij,y}$ , we re-arrange Eq. (3.2) using trigonometric identities:

$$v_{ij,x} = q_i \widehat{v}_i \cos(\widehat{\theta}_i) \cos(\theta_i) - q_i \widehat{v}_i \sin(\widehat{\theta}_i) \sin(\theta_i) - q_j \widehat{v}_j \cos(\widehat{\theta}_j) \cos(\theta_j) + q_j \widehat{v}_j \sin(\widehat{\theta}_j) \sin(\theta_j), \quad (3.14a)$$

$$v_{ij,y} = q_i \widehat{v}_i \sin(\widehat{\theta}_i) \cos(\theta_i) + q_i \widehat{v}_i \cos(\widehat{\theta}_i) \sin(\theta_i) - q_j \widehat{v}_j \sin(\widehat{\theta}_j) \cos(\theta_j) - q_j \widehat{v}_j \cos(\widehat{\theta}_j) \sin(\theta_j). \quad (3.14b)$$

Let  $\underline{v}_{ij,x}, \bar{v}_{ij,x}$  and  $\underline{v}_{ij,y}, \bar{v}_{ij,y}$  be the lower and upper bounds for  $v_{ij,x}$  and  $v_{ij,y}$ , respectively. These bounds can be determined using Eq. (3.14) and the bounds on speed and heading control provided in Eqs. (3.12) and (3.13). The derived bounds on the relative velocity components can be used to define a box in the plane  $\{(v_{ij,x}, v_{ij,y}) \in \mathbb{R}^2\}$ .

**Definition 3.2** (Relative velocity box). *Consider a pair of aircraft  $(i, j) \in \mathcal{P}$ . Let  $\mathcal{B}_{ij}$  be the subset of  $\mathbb{R}^2$  defined as*

$$\mathcal{B}_{ij} \equiv \left\{ (v_{ij,x}, v_{ij,y}) \in \mathbb{R}^2 : \underline{v}_{ij,x} \leq v_{ij,x} \leq \bar{v}_{ij,x}, \underline{v}_{ij,y} \leq v_{ij,y} \leq \bar{v}_{ij,y} \right\}. \quad (3.15)$$

$\mathcal{B}_{ij}$  is the relative velocity box of  $(i, j) \in \mathcal{P}$ .

The relative velocity box  $\mathcal{B}_{ij}$  characterises all possible trajectories for the pair  $(i, j) \in \mathcal{P}$  based on the available 2D deconfliction resources, i.e. speed and heading controls. To characterise the set of conflict-free trajectories of a pair of aircraft  $(i, j) \in \mathcal{P}$ , we compare the relative position of the relative velocity box  $\mathcal{B}_{ij}$  with the conflict region of this pair of aircraft. Observe that the conflict region is convex and can be defined by reversing the inequalities (3.11) and omitting the disjunction  $z_{ij} \in \{0, 1\}$ .

**Definition 3.3** (Conflict region). *Consider a pair of aircraft  $(i, j) \in \mathcal{P}$ . Let  $\mathcal{C}_{ij}$  be the subset of  $\mathbb{R}^2$  defined as*

$$\mathcal{C}_{ij} \equiv \left\{ (v_{ij,x}, v_{ij,y}) \in \mathbb{R}^2 : v_{ij,y}\gamma_{ij}^l - v_{ij,x}\phi_{ij}^l \geq 0 \wedge v_{ij,y}\gamma_{ij}^u - v_{ij,x}\phi_{ij}^u \leq 0 \right\}. \quad (3.16)$$

$\mathcal{C}_{ij}$  is the conflict region of  $(i, j) \in \mathcal{P}$ .

The conflict region of a pair of aircraft represents the set of relative velocity vectors which corresponds to a conflict. The relative positions of the relative velocity box  $\mathcal{B}_{ij}$  and the conflict region  $\mathcal{C}_{ij}$  can be examined to determine the existence or not of a potential conflict. For any pair  $(i, j) \in \mathcal{P}$ , if  $\mathcal{B}_{ij} \cap \mathcal{C}_{ij} = \emptyset$ , then aircraft  $i$  and  $j$  are separated for any combination of controls; conversely if  $\mathcal{B}_{ij} \subset \mathcal{C}_{ij}$  then  $i$  and  $j$  cannot be separated via speed or heading control within the assumed control bounds; otherwise,  $\mathcal{B}_{ij}$  and  $\mathcal{C}_{ij}$  intersect but do not completely overlap. This is illustrated in Figure 3.2 which illustrates the three possible cases. Figure 3.2a illustrates the case where aircraft  $i$  and  $j$  are separated for any combination of speed and heading control—we say that such pairs are conflict-free. Figure 3.2b depicts the case where  $\mathcal{B}_{ij}$  and  $\mathcal{C}_{ij}$  only partially intersect—we say that such pairs are separable. Last, Figure 3.2c illustrates the case where  $\mathcal{B}_{ij} \subset \mathcal{C}_{ij}$ —we say that such pairs are non-separable. The following propositions provide methods to efficiently determine if, given controls bounds on speed and heading, a pair of aircraft is either conflict-free or non-separable.

**Proposition 3.1** (Conflict-free aircraft pair). *Consider a pair of aircraft  $(i, j) \in \mathcal{P}$ , and let  $LP(i, j)$  be the feasibility problem defined as:*

$$LP(i, j) : \begin{cases} v_{ij,y}\gamma_{ij}^l - v_{ij,x}\phi_{ij}^l \geq 0, \\ v_{ij,y}\gamma_{ij}^u - v_{ij,x}\phi_{ij}^u \leq 0, \\ (v_{ij,x}, v_{ij,y}) \in \mathcal{B}_{ij}. \end{cases}$$

*The pair of aircraft  $(i, j)$  is conflict-free for any 2D control if and only if the feasibility problem  $LP(i, j)$  is infeasible.*

---

**Proposition 3.2** (Non-separable aircraft pair). *Consider a pair of aircraft  $(i, j) \in \mathcal{P}$ . The pair  $(i, j)$  is non-separable if and only if each of the four extreme points of  $\mathcal{B}_{ij}$  corresponds to a conflict.*

Using Propositions 3.1 and 3.2, we can design an efficient pre-processing algorithm to partition the set of aircraft pairs  $\mathcal{P}$  of a 2D ACRP instance into three categories: conflict-free pairs denoted  $\mathcal{P}_F$ , separable pairs denoted  $\mathcal{P}_S$  and non-separable pairs  $\mathcal{P}_I$ .

---

**Algorithm 3:** Pre-processing of aircraft pairs

---

**Input:**  $\mathcal{A}, \hat{\theta}, \hat{v}, \mathbf{q}, \bar{\mathbf{q}}, \boldsymbol{\theta}, \bar{\boldsymbol{\theta}}$   
**Output:**  $\mathcal{P}, \mathcal{P}_F, \mathcal{P}_S, \mathcal{P}_I$

- 1  $\mathcal{P} \leftarrow \{i, j \in \mathcal{A} : i < j\}$
- 2  $\mathcal{P}_F, \mathcal{P}_S, \mathcal{P}_I \leftarrow \emptyset$
- 3 **for**  $(i, j) \in \mathcal{P}$  **do**
- 4     Solve  $LP(i, j)$
- 5     **if**  $LP(i, j)$  is infeasible **then**
- 6          $\mathcal{P}_F \leftarrow \mathcal{P}_F \cup \{(i, j)\}$
- 7     **else**
- 8          $k \leftarrow 0$
- 9         **for**  $(v_{ij,x}, v_{ij,y}) \in E(\mathcal{B}_{ij})$  **do**
- 10             **if**  $g_{ij}(v_{ij,x}, v_{ij,y}) < 0 \wedge t_{ij}^{min}(v_{ij,x}, v_{ij,y}) > 0$  **then**
- 11                  $k \leftarrow k + 1$
- 12             **if**  $k = 4$  **then**
- 13                  $\mathcal{P}_I \leftarrow \mathcal{P}_I \cup \{(i, j)\}$
- 14     **else**
- 15          $\mathcal{P}_S \leftarrow \mathcal{P}_S \cup \{(i, j)\}$

---

To identify conflict-free pairs,  $LP(i, j)$  is solved and a pair of aircraft is conflict-free if and only if the LP is infeasible. Observe that the feasibility problem  $LP(i, j)$  is linear and can be solved by enumerating all extreme points of its feasible region and test if this corresponds to a conflict or not. Since  $LP(i, j)$  consists of four bound constraints and two shared constraints, there is a total of 13 extreme points to test (the combinations of the bound constraints of a variable can be excluded). To identify non-separable pairs, we denote  $E(\mathcal{B}_{ij})$  the set of extreme points of the relative velocity box  $\mathcal{B}_{ij}$  for any pair  $(i, j) \in \mathcal{P}$  and use the separation condition (3.3) to determine if all extreme points are conflicts or not. This procedure is summarized in Algorithm 3 (we use boldface to denote vectors). Observe that pairwise variables and constraints need only to be indexed by the set of separable pairs  $\mathcal{P}_S$  since pairs in  $\mathcal{P}_F$  are always conflict-free. Further, any 2D conflict resolution problem such that  $|\mathcal{P}_I| > 0$  is infeasible.

The nonconvex formulation of the 2D ARCP is summarized in Model 3.1.

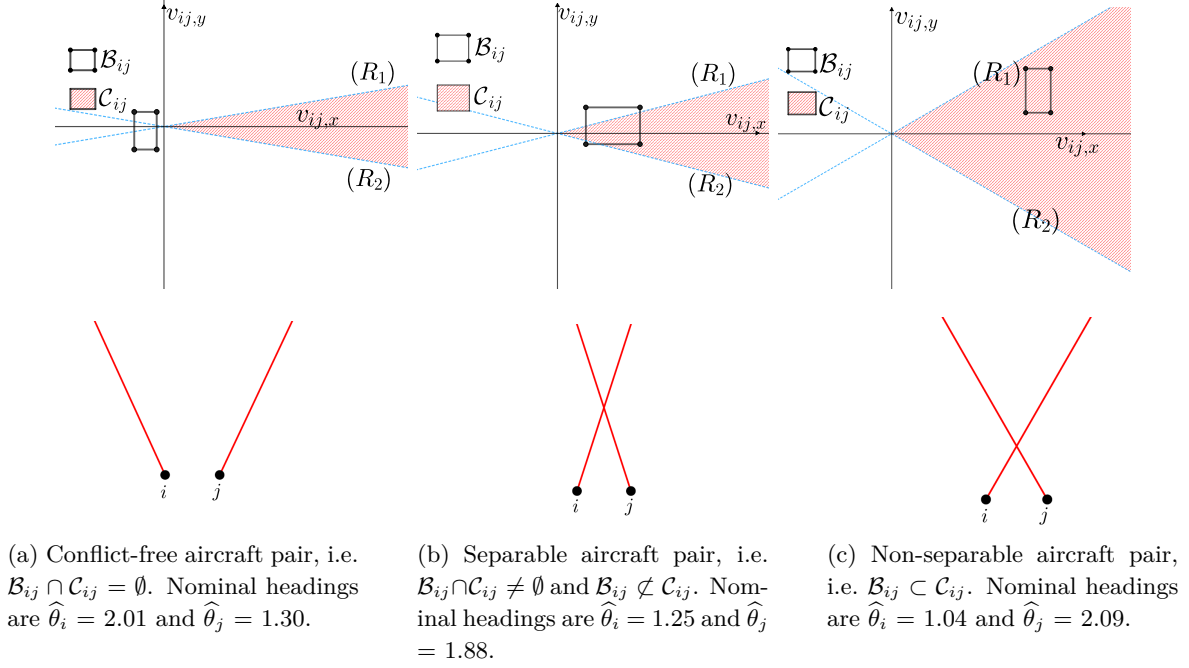


Figure 3.2: Three aircraft configurations illustrating conflict-free (3.2a), separable (3.2b) and non-separable (3.2c) pairs. Aircraft initial positions and trajectories are depicted at the bottom part of each sub-figure, where black dots correspond to initial positions and red lines represent initial trajectories. The top part of each sub-figure shows a graph of the relative velocity box  $\mathcal{B}_{ij}$  represented by a black rectangle, and the conflict region  $\mathcal{C}_{ij}$  represented by a red hashed area. Dashed blue lines represent the solutions of  $g(v_{ij,x}, v_{ij,y}) = 0$  ( $R_1$  and  $R_2$ ). In all cases, aircraft  $i$  and  $j$  have nominal speeds  $\hat{v}_i = \hat{v}_j = 500$  NM/h and are initially separated by 30 NM in the x-axis direction.

**Model 3.1** (Nonconvex 2D Formulation).

$$\min \sum_{i \in \mathcal{A}} (1 - w)(1 - q_i)^2 + w\theta_i^2, \quad (3.17a)$$

s.t.

$$v_{ij,x} = q_i \hat{v}_i \sin(\hat{\theta}_i + \theta_i) - q_j \hat{v}_j \sin(\hat{\theta}_j + \theta_j), \quad \forall (i, j) \in \mathcal{P}, \quad (3.17b)$$

$$v_{ij,y} = q_i \hat{v}_i \cos(\hat{\theta}_i + \theta_i) - q_j \hat{v}_j \cos(\hat{\theta}_j + \theta_j), \quad \forall (i, j) \in \mathcal{P}, \quad (3.17c)$$

$$v_{ij,y} \hat{x}_{ij} - v_{ij,x} \hat{y}_{ij} \leq 0, \quad \text{if } z_{ij} = 1, \quad \forall (i, j) \in \mathcal{P}, \quad (3.17d)$$

$$v_{ij,y} \hat{x}_{ij} - v_{ij,x} \hat{y}_{ij} \geq 0, \quad \text{if } z_{ij} = 0, \quad \forall (i, j) \in \mathcal{P}, \quad (3.17e)$$

$$v_{ij,y} \gamma_{ij}^l - v_{ij,x} \phi_{ij}^l \leq 0, \quad \text{if } z_{ij} = 1, \quad \forall (i, j) \in \mathcal{P}, \quad (3.17f)$$

$$v_{ij,x} \gamma_{ij}^u - v_{ij,y} \phi_{ij}^u \geq 0, \quad \text{if } z_{ij} = 0, \quad \forall (i, j) \in \mathcal{P}, \quad (3.17g)$$

$$v_{ij,x}, v_{ij,y} \in \mathcal{B}_{ij}, \quad \forall (i, j) \in \mathcal{P}, \quad (3.17h)$$

$$z_{ij} \in \{0, 1\}, \quad \forall (i, j) \in \mathcal{P}, \quad (3.17i)$$

$$\underline{q}_i \leq q_i \leq \bar{q}_i, \quad \forall i \in \mathcal{A}, \quad (3.17j)$$

$$\underline{\theta}_i \leq \theta_i \leq \bar{\theta}_i, \quad \forall i \in \mathcal{A}. \quad (3.17k)$$

Model 3.1 provides a compact formulation for the ACRP with speed and heading control

which requires a single binary variable per pair of aircraft. Note that coefficients  $\gamma_{ij}^l$ ,  $\phi_{ij}^l$ ,  $\gamma_{ij}^u$  and  $\phi_{ij}^u$  (present in (3.10)) can be pre-processed based on the sign of  $\hat{x}_{ij}$  and  $\hat{y}_{ij}$ . For implementation details, a fully reproducible formulation can be found at: <https://github.com/acrp-lib/acrp-lib>.

### 3.3.3 Complex formulation for the 2D aircraft conflict resolution problem

An alternative way to represent aircraft motion equation is via the complex number formulation introduced by [Rey and Hijazi \[2017\]](#). Let  $V_i$  be the complex number defined as:

$$V_i = q_i(\cos(\theta_i) + i \sin(\theta_i)), \quad \forall i \in \mathcal{A}. \quad (3.18)$$

Representing  $V_i$  in its polar form with  $\delta_{i,x} = q_i \cos(\theta_i)$  and  $\delta_{i,y} = q_i \sin(\theta_i)$ , yield:

$$V_i = \delta_{i,x} + i\delta_{i,y}, \quad \forall i \in \mathcal{A}. \quad (3.19)$$

The magnitude of  $V_i$  is  $|V_i| = \sqrt{\delta_{i,x}^2 + \delta_{i,y}^2} = q_i$  and its argument is  $\arg(V_i) = \arctan \frac{\delta_{i,y}}{\delta_{i,x}} = \theta_i$ . This approach is inspired by complex number formulation for the optimal power flow problem in power systems, such as in [Hijazi et al. \[2017\]](#) and [Coffrin et al. \[2015\]](#). Accordingly, the relative motion equations of a pair of aircraft can be rewritten as:

$$v_{ij,x} = \delta_{i,x} \hat{v}_i \cos(\hat{\theta}_i) - \delta_{i,y} \hat{v}_i \sin(\hat{\theta}_i) - \delta_{j,x} \hat{v}_j \cos(\hat{\theta}_j) + \delta_{j,y} \hat{v}_j \sin(\hat{\theta}_j), \quad \forall (i, j) \in \mathcal{P}_S, \quad (3.20a)$$

$$v_{ij,y} = \delta_{i,y} \hat{v}_i \cos(\hat{\theta}_i) - \delta_{i,x} \hat{v}_i \sin(\hat{\theta}_i) - \delta_{j,y} \hat{v}_j \cos(\hat{\theta}_j) + \delta_{j,x} \hat{v}_j \sin(\hat{\theta}_j), \quad \forall (i, j) \in \mathcal{P}_S. \quad (3.20b)$$

The variables  $\delta_{i,x}$  and  $\delta_{i,y}$  are used as the main control variables in this formulation and their bounds are derived from the bounds of the original control variables  $q_i$  and  $\theta_i$ :

$$\underline{q}_i \cos(\max\{|\bar{\theta}_i|, |\underline{\theta}_i|\}) \leq \delta_{i,x} \leq \bar{q}_i, \quad \forall i \in \mathcal{A}, \quad (3.21a)$$

$$\bar{q}_i \sin(\underline{\theta}_i) \leq \delta_{i,y} \leq \bar{q}_i \sin(\bar{\theta}_i), \quad \forall i \in \mathcal{A}. \quad (3.21b)$$

The speed control constraint (3.12) can be reformulated in quadratic form as:

$$\underline{q}_i^2 \leq \delta_{i,x}^2 + \delta_{i,y}^2, \quad \forall i \in \mathcal{A}, \quad (3.22a)$$

$$\bar{q}_i^2 \geq \delta_{i,x}^2 + \delta_{i,y}^2, \quad \forall i \in \mathcal{A}. \quad (3.22b)$$

The heading control constraint (3.13) can be reformulated in linear form as:

$$\delta_{i,x} \tan(\underline{\theta}_i) \leq \delta_{i,y} \leq \delta_{i,x} \tan(\bar{\theta}_i), \quad \forall i \in \mathcal{A}. \quad (3.23a)$$

To design the objective function, we introduce a preference weight  $w \in ]0, 1[$  to balance the trade-offs among velocity controls, i.e. speed and heading. We extend the objective function proposed by [Rey and Hijazi \[2017\]](#) as follows:

$$\text{minimise } \sum_{i \in \mathcal{A}} w \delta_{i,y}^2 + (1 - w)(1 - \delta_{i,x})^2. \quad (3.24)$$

We next show that the objective function (3.24) attains its minimum value when aircraft have deviation-free trajectories, i.e.  $q_i = 1$  and  $\theta_i = 0$  for all  $i \in \mathcal{A}$ .

**Proposition 3.3.** *The 2D objective function (3.24) is convex with regards to decision variables  $\delta_{i,x}$  and  $\delta_{i,y}$  for any value  $w \in ]0, 1[$ , and is minimal for  $q_i = 1$  and  $\theta_i = 0$  for all aircraft  $i \in \mathcal{A}$ .*

Proposition 3.3 shows that the proposed 2D objective function achieves an optimal value for deviation-free aircraft trajectories for any preference weight  $w \in ]0, 1[$ . We will show in Section 3.6.2 that increasing  $w$  increases the penalisation of heading deviations whereas decreasing  $w$  increases the penalisation of speed deviations. The complex number nonconvex formulation for the 2D ACRP is summarized in Model 3.2.

**Model 3.2** (Complex number nonconvex 2D formulation).

$$\min \sum_{i \in \mathcal{A}} w \delta_{i,y}^2 + (1-w)(1 - \delta_{i,x})^2, \quad (3.25a)$$

s.t.

$$v_{ij,x} = \delta_{i,x} \hat{v}_i \cos(\hat{\theta}_i) - \delta_{i,y} \hat{v}_i \sin(\hat{\theta}_i) - \delta_{j,x} \hat{v}_j \cos(\hat{\theta}_j) + \delta_{j,y} \hat{v}_j \sin(\hat{\theta}_j), \quad \forall (i, j) \in \mathcal{P}_S, \quad (3.25b)$$

$$v_{ij,y} = \delta_{i,y} \hat{v}_i \cos(\hat{\theta}_i) - \delta_{i,x} \hat{v}_i \sin(\hat{\theta}_i) - \delta_{j,y} \hat{v}_j \cos(\hat{\theta}_j) + \delta_{j,x} \hat{v}_j \sin(\hat{\theta}_j), \quad \forall (i, j) \in \mathcal{P}_S, \quad (3.25c)$$

$$v_{ij,y} \hat{x}_{ij} - v_{ij,x} \hat{y}_{ij} \leq 0, \quad \text{if } z_{ij} = 1, \quad \forall (i, j) \in \mathcal{P}_S, \quad (3.25d)$$

$$v_{ij,y} \hat{x}_{ij} - v_{ij,x} \hat{y}_{ij} \geq 0, \quad \text{if } z_{ij} = 0, \quad \forall (i, j) \in \mathcal{P}_S, \quad (3.25e)$$

$$v_{ij,y} \gamma_{ij}^l - v_{ij,x} \phi_{ij}^l \leq 0, \quad \text{if } z_{ij} = 1, \quad \forall (i, j) \in \mathcal{P}_S, \quad (3.25f)$$

$$v_{ij,y} \gamma_{ij}^u - v_{ij,x} \phi_{ij}^u \geq 0, \quad \text{if } z_{ij} = 0, \quad \forall (i, j) \in \mathcal{P}_S, \quad (3.25g)$$

$$\underline{q}_i^2 \leq \delta_{i,x}^2 + \delta_{i,y}^2 \leq \bar{q}_i^2, \quad \forall i \in \mathcal{A}, \quad (3.25h)$$

$$\delta_{i,x} \tan(\underline{\theta}_i) \leq \delta_{i,y} \leq \delta_{i,x} \tan(\bar{\theta}_i), \quad \forall i \in \mathcal{A}, \quad (3.25i)$$

$$\underline{q}_i \cos(\max\{|\bar{\theta}_i|, |\underline{\theta}_i|\}) \leq \delta_{i,x} \leq \bar{q}_i, \quad \forall i \in \mathcal{A}, \quad (3.25j)$$

$$\bar{q}_i \sin(\underline{\theta}_i) \leq \delta_{i,y} \leq \bar{q}_i \sin(\bar{\theta}_i), \quad \forall i \in \mathcal{A}, \quad (3.25k)$$

$$\underline{v}_{ij,x} \leq v_{ij,x} \leq \bar{v}_{ij,x}, \quad \forall (i, j) \in \mathcal{P}_S, \quad (3.25l)$$

$$\underline{v}_{ij,y} \leq v_{ij,y} \leq \bar{v}_{ij,y}, \quad \forall (i, j) \in \mathcal{P}_S, \quad (3.25m)$$

$$z_{ij} \in \{0, 1\}, \quad \forall (i, j) \in \mathcal{P}_S, \quad (3.25n)$$

$$\delta_{i,x}, \delta_{i,y} \in \mathbb{R}, \quad \forall i \in \mathcal{A}. \quad (3.25o)$$

Model 3.2 provides a compact formulation for the 2D ACRP with continuous speed and heading control variables which requires a single binary variable per pair of aircraft. This formulation is nonconvex due to the speed lower bound constraint (3.22a) which is nonconvex quadratic. Note that we use the convex hull reformulation of the linear On/Off constraints (3.10) and (3.11) as derived in Hijazi et al. [2014]. This reformulation does not require the introduction of auxiliary variables and is proved to provide the tightest continuous relaxation for each On/Off constraint. Coefficients  $\gamma_{ij}^l$ ,  $\phi_{ij}^l$  and  $\gamma_{ij}^u$ ,  $\phi_{ij}^u$  (present in (3.10) and (3.11)) can be pre-processed based on the sign of  $\hat{x}_{ij}$  and  $\hat{y}_{ij}$ .

### 3.3.4 Complex number formulation for the 2D+FL aircraft conflict resolution problem

To model FL changes, we assume that each aircraft  $i \in \mathcal{A}$  is initially assigned to a base FL denoted  $\hat{\rho}_i$ . We assume that adjacent FLs are vertically separated (e.g. by 1000 ft). Thus, we

only need to impose separation constraints on pairs of aircraft which share the same FL. Let  $\mathcal{Z}_i$  denote the set of available FLs for each aircraft  $i \in \mathcal{A}$ , and consider the binary variable  $\rho_{ik}$  defined as:

$$\rho_{ik} \equiv \begin{cases} 1 & \text{if aircraft } i \in \mathcal{A} \text{ is assigned to FL } k \in \mathcal{Z}_i, \\ 0 & \text{otherwise.} \end{cases} \quad (3.26)$$

By design,  $\hat{\rho}_i \in \mathcal{Z}_i$  and we require that each aircraft  $i \in \mathcal{A}$  be assigned to exactly one FL in its reachable set  $\mathcal{Z}_i$  via constraint (3.27):

$$\sum_{k \in \mathcal{Z}_i} \rho_{ik} = 1, \quad \forall i \in \mathcal{A}. \quad (3.27)$$

Let  $\mathcal{P}_S^3$  be the set of aircraft pairs which are not conflict-free and may share the same FL, i.e.  $\mathcal{P}_S^3 \equiv \{(i, j) \in \mathcal{P}_S \cup \mathcal{P}_I : \mathcal{Z}_i \cap \mathcal{Z}_j \neq \emptyset\}$ . Let  $\varphi_{ij}$  be the binary variable defined as:

$$\varphi_{ij} \equiv \begin{cases} 1 & \text{if aircraft } i \text{ and } j \text{ are assigned to the same FL,} \\ 0 & \text{otherwise.} \end{cases} \quad (3.28)$$

Variable  $\varphi_{ij}$  can be linked to binary variables  $\rho_{ik}$  and  $\rho_{jk}$  via the constraint:

$$\rho_{ik} + \rho_{jk} \leq \varphi_{ij} + 1, \quad \forall (i, j) \in \mathcal{P}_S^3, k \in \mathcal{Z}_i \cap \mathcal{Z}_j. \quad (3.29)$$

The separation conditions (3.10)-(3.11) are rewritten to account for altitude separation as follows:

$$v_{ij,y}\hat{x}_{ij} - v_{ij,x}\hat{y}_{ij} \leq 0, \quad \text{if } z_{ij} = 1 \quad \text{and} \quad \varphi_{ij} = 1, \quad \forall (i, j) \in \mathcal{P}_S^3, \quad (3.30a)$$

$$v_{ij,y}\hat{x}_{ij} - v_{ij,x}\hat{y}_{ij} \geq 0, \quad \text{if } z_{ij} = 0 \quad \text{and} \quad \varphi_{ij} = 1, \quad \forall (i, j) \in \mathcal{P}_S^3, \quad (3.30b)$$

$$v_{ij,y}\gamma_{ij}^l - v_{ij,x}\phi_{ij}^l \leq 0, \quad \text{if } z_{ij} = 1 \quad \text{and} \quad \varphi_{ij} = 1, \quad \forall (i, j) \in \mathcal{P}_S^3, \quad (3.30c)$$

$$v_{ij,y}\gamma_{ij}^u - v_{ij,x}\phi_{ij}^u \geq 0, \quad \text{if } z_{ij} = 0 \quad \text{and} \quad \varphi_{ij} = 1, \quad \forall (i, j) \in \mathcal{P}_S^3. \quad (3.30d)$$

FL changes are typically less desirable compared to other deconfliction manoeuvres such as speed or heading control [Bilimoria et al., 1996, Hu et al., 2002, Alonso-Ayuso et al., 2011]. This is due to several practical considerations including an increase in fuel consumption, passenger discomfort due to climbing or descending and the need for extended monitoring. Under these considerations, we propose a lexicographic optimization approach for the 2D+FL ACRP. The number of FL changes is first minimised; before the total 2D deviation of flights is minimised. The proposed objective function for minimising the number of FL changes is:

$$\text{minimise } \sum_{i \in \mathcal{A}} \left| \sum_{k \in \mathcal{Z}_i} k\rho_{ik} - \hat{\rho}_i \right|. \quad (3.31)$$

The resulting nonconvex lexicographic 2D+FL formulation is summarized in Model 3.3.

**Model 3.3** (Complex number nonconvex lexicographic 2D+FL formulation).

$$1. \quad \min \sum_{i \in \mathcal{A}} \left| \sum_{k \in \mathcal{Z}_i} k\rho_{ik} - \hat{\rho}_i \right|, \quad (3.32a)$$



$$2. \min \sum_{i \in \mathcal{A}} w \delta_{i,y}^2 + (1-w)(1 - \delta_{i,x})^2, \quad (3.32b)$$

s.t.

$$v_{ij,x} = \delta_{i,x} \hat{v}_i \cos(\hat{\theta}_i) - \delta_{i,y} \hat{v}_i \sin(\hat{\theta}_i) - \delta_{j,x} \hat{v}_j \cos(\hat{\theta}_j) + \delta_{j,y} \hat{v}_j \sin(\hat{\theta}_j), \quad \forall (i,j) \in \mathcal{P}_S^3, \quad (3.32c)$$

$$v_{ij,y} = \delta_{i,y} \hat{v}_i \cos(\hat{\theta}_i) - \delta_{i,x} \hat{v}_i \sin(\hat{\theta}_i) - \delta_{j,y} \hat{v}_j \cos(\hat{\theta}_j) + \delta_{j,x} \hat{v}_j \sin(\hat{\theta}_j), \quad \forall (i,j) \in \mathcal{P}_S^3, \quad (3.32d)$$

$$v_{ij,y} \hat{x}_{ij} - v_{ij,x} \hat{y}_{ij} \leq 0, \quad \text{if } z_{ij} = 1 \quad \text{and} \quad \varphi_{ij} = 1, \quad \forall (i,j) \in \mathcal{P}_S^3, \quad (3.32e)$$

$$v_{ij,y} \hat{x}_{ij} - v_{ij,x} \hat{y}_{ij} \geq 0, \quad \text{if } z_{ij} = 0 \quad \text{and} \quad \varphi_{ij} = 1, \quad \forall (i,j) \in \mathcal{P}_S^3, \quad (3.32f)$$

$$v_{ij,y} \gamma_{ij}^l - v_{ij,x} \phi_{ij}^l \leq 0, \quad \text{if } z_{ij} = 1 \quad \text{and} \quad \varphi_{ij} = 1, \quad \forall (i,j) \in \mathcal{P}_S^3, \quad (3.32g)$$

$$v_{ij,y} \gamma_{ij}^u - v_{ij,x} \phi_{ij}^u \geq 0, \quad \text{if } z_{ij} = 0 \quad \text{and} \quad \varphi_{ij} = 1, \quad \forall (i,j) \in \mathcal{P}_S^3, \quad (3.32h)$$

$$\underline{q}_i^2 \leq \delta_{i,x}^2 + \delta_{i,y}^2 \leq \bar{q}_i^2, \quad \forall i \in \mathcal{A}, \quad (3.32i)$$

$$\delta_{i,x} \tan(\theta_i) \leq \delta_{i,y} \leq \delta_{i,x} \tan(\bar{\theta}_i), \quad \forall i \in \mathcal{A}, \quad (3.32j)$$

$$\underline{q}_i \cos(\max\{|\bar{\theta}_i|, |\theta_i|\}) \leq \delta_{i,x} \leq \bar{q}_i, \quad \forall i \in \mathcal{A}, \quad (3.32k)$$

$$\bar{q}_i \sin(\theta_i) \leq \delta_{i,y} \leq \bar{q}_i \sin(\bar{\theta}_i), \quad \forall i \in \mathcal{A}, \quad (3.32l)$$

$$\sum_{k \in \mathcal{Z}_i} \rho_{ik} = 1, \quad \forall i \in \mathcal{A}, \quad (3.32m)$$

$$\rho_{ik} + \rho_{jk} \leq \varphi_{ij} + 1, \quad \forall (i,j) \in \mathcal{P}_S^3, k \in \mathcal{Z}_i \cap \mathcal{Z}_j \quad (3.32n)$$

$$\underline{v}_{ij,x} \leq v_{ij,x} \leq \bar{v}_{ij,x}, \quad \forall (i,j) \in \mathcal{P}_S^3, \quad (3.32o)$$

$$\underline{v}_{ij,y} \leq v_{ij,y} \leq \bar{v}_{ij,y}, \quad \forall (i,j) \in \mathcal{P}_S^3, \quad (3.32p)$$

$$z_{ij} \in \{0, 1\}, \quad \forall (i,j) \in \mathcal{P}_S^3, \quad (3.32q)$$

$$\delta_{i,x}, \delta_{i,y} \in \mathbb{R}, \quad \forall i \in \mathcal{A}, \quad (3.32r)$$

$$\rho_{ik} \in \{0, 1\}, \quad \forall i \in \mathcal{A}, k \in \mathcal{Z}_i. \quad (3.32s)$$

Compared to Model 3.2, Model 3.3 requires additional binary decision variables  $\rho_{ik}$  and  $\varphi_{ij}$ . The former is used to assign aircraft to separated FLs and the latter ensures that aircraft sharing the same FL are separated via the 2D separation conditions.

### 3.4 Exact solution methods for the aircraft conflict resolution problem

We present exact solution methods for the 2D and the 2D+FL ACRPs that build on and extend the convex relaxations presented by [Rey and Hijazi \[2017\]](#). We first present a convex relaxation of the 2D ACRP that fully relaxes the speed control constraint in the complex number formulation (Section 3.4.1). This relaxation yields a Mixed-Integer Quadratic Program (MIQP) which solution may violate the speed control bounds of the problem. To eliminate these potential violations, we incorporate convex quadratic constraints and piecewise linear outer approximations in a Mixed-Integer Quadratically Constrained Program (MIQCP). We then propose a constraint generation algorithm to iteratively refine the piece-wise linear approximation and show that our approach converges to optimal solutions of the 2D ACRP (Section 3.4.2).

To solve the lexicographic optimization formulation for the 2D+FL ACRP, we propose a

two-step decomposition approach. We first solve a restricted flight assignment problem that only implicitly accounts for aircraft trajectories and yields an optimal solution with regards to the first objective function (total FL deviation). We then use the optimal solution of the flight assignment formulation to assign aircraft to FLs and solve a series of 2D problems (one per FL) to construct an optimal solution with regards to the second objective function (total 2D deviation). Both steps are iterated until a global solution is found (Section 3.4.3). We next present in detail the proposed exact solution methods.

### 3.4.1 Mixed-integer quadratic relaxation for the 2D ACRP

An initial convex relaxation of Model 3.2 was proposed by [Rey and Hijazi \[2017\]](#) by relaxing the speed control constraint (3.22). The resulting formulation is a MIQP summarized in Model 3.4.

**Model 3.4** (MIQP 2D formulation).

$$\min \sum_{i \in \mathcal{A}} w \delta_{i,y}^2 + (1-w)(1 - \delta_{i,x})^2, \quad (3.33a)$$

s.t.

$$v_{ij,x} = \delta_{i,x} \hat{v}_i \cos(\hat{\theta}_i) - \delta_{i,y} \hat{v}_i \sin(\hat{\theta}_i) - \delta_{j,x} \hat{v}_j \cos(\hat{\theta}_j) + \delta_{j,y} \hat{v}_j \sin(\hat{\theta}_j), \quad \forall (i, j) \in \mathcal{P}_S, \quad (3.33b)$$

$$v_{ij,y} = \delta_{i,y} \hat{v}_i \cos(\hat{\theta}_i) - \delta_{i,x} \hat{v}_i \sin(\hat{\theta}_i) - \delta_{j,y} \hat{v}_j \cos(\hat{\theta}_j) + \delta_{j,x} \hat{v}_j \sin(\hat{\theta}_j), \quad \forall (i, j) \in \mathcal{P}_S, \quad (3.33c)$$

$$v_{ij,y} \hat{x}_{ij} - v_{ij,x} \hat{y}_{ij} \leq 0, \quad \text{if } z_{ij} = 1, \quad \forall (i, j) \in \mathcal{P}_S, \quad (3.33d)$$

$$v_{ij,y} \hat{x}_{ij} - v_{ij,x} \hat{y}_{ij} \geq 0, \quad \text{if } z_{ij} = 0, \quad \forall (i, j) \in \mathcal{P}_S, \quad (3.33e)$$

$$v_{ij,y} \gamma_{ij}^l - v_{ij,x} \phi_{ij}^l \leq 0, \quad \text{if } z_{ij} = 1, \quad \forall (i, j) \in \mathcal{P}_S, \quad (3.33f)$$

$$v_{ij,y} \gamma_{ij}^u - v_{ij,x} \phi_{ij}^u \geq 0, \quad \text{if } z_{ij} = 0, \quad \forall (i, j) \in \mathcal{P}_S, \quad (3.33g)$$

$$\delta_{i,x} \tan(\underline{\theta}_i) \leq \delta_{i,y} \leq \delta_{i,x} \tan(\bar{\theta}_i), \quad \forall i \in \mathcal{A}, \quad (3.33h)$$

$$\underline{q}_i \cos(\max\{|\bar{\theta}_i|, |\underline{\theta}_i|\}) \leq \delta_{i,x} \leq \bar{q}_i, \quad \forall i \in \mathcal{A}, \quad (3.33i)$$

$$\bar{q}_i \sin(\underline{\theta}_i) \leq \delta_{i,y} \leq \bar{q}_i \sin(\bar{\theta}_i), \quad \forall i \in \mathcal{A}, \quad (3.33j)$$

$$\underline{v}_{ij,x} \leq v_{ij,x} \leq \bar{v}_{ij,x}, \quad \forall (i, j) \in \mathcal{P}_S, \quad (3.33k)$$

$$\underline{v}_{ij,y} \leq v_{ij,y} \leq \bar{v}_{ij,y}, \quad \forall (i, j) \in \mathcal{P}_S, \quad (3.33l)$$

$$z_{ij} \in \{0, 1\}, \quad \forall (i, j) \in \mathcal{P}_S, \quad (3.33m)$$

$$\delta_{i,x}, \delta_{i,y} \in \mathbb{R}, \quad \forall i \in \mathcal{A}. \quad (3.33n)$$

Model 3.4 yields a lower bound on the optimal objective value of Model 3.2 and a solution which is a global optimum if the relaxed constraint (3.22) is not violated.

### 3.4.2 Mixed-integer quadratically constrained relaxation and constraint generation algorithm

To tighten the MIQP relaxation given in Section 3.4.1, we build on and extend the MIQCP relaxation proposed by [Rey and Hijazi \[2017\]](#) by incorporating the speed control constraint (3.22) using convex quadratic and piecewise linear constraints. Observe that the speed upper bound constraint (3.22b) is convex quadratic, hence it can be incorporated directly in the MIQCP formulation.

To incorporate the speed lower bound constraint (3.22a), for each aircraft  $i \in \mathcal{A}$ , we introduce real variables  $\tilde{\delta}_{i,x}$  and  $\tilde{\delta}_{i,y}$  to approximate  $\delta_{i,x}^2$  and  $\delta_{i,y}^2$ , respectively, via convex quadratic constraints:

$$\tilde{\delta}_{i,x} \geq \delta_{i,x}^2, \quad \forall i \in \mathcal{A}, \quad (3.34a)$$

$$\tilde{\delta}_{i,y} \geq \delta_{i,y}^2, \quad \forall i \in \mathcal{A}. \quad (3.34b)$$

Accordingly, we require:

$$\underline{q}_i^2 \leq \tilde{\delta}_{i,x} + \tilde{\delta}_{i,y}, \quad \forall i \in \mathcal{A}. \quad (3.35)$$

To impose upper bounds on  $\tilde{\delta}_{i,x}$  and  $\tilde{\delta}_{i,y}$ , we use their McCormick envelopes [McCormick, 1976]:

$$\tilde{\delta}_{i,x} \leq (\bar{q}_i + \underline{q}_i \cos(\max\{|\underline{\theta}_i|, |\bar{\theta}_i|\}))\delta_{i,x} - \bar{q}_i \underline{q}_i \cos(\max\{|\underline{\theta}_i|, |\bar{\theta}_i|\}), \quad \forall i \in \mathcal{A}. \quad (3.36a)$$

$$\tilde{\delta}_{i,y} \leq \bar{q}_i (\sin(\underline{\theta}_i) + \sin(\bar{\theta}_i))\delta_{i,y} - \bar{q}_i^2 \sin(\bar{\theta}_i) \sin(\underline{\theta}_i), \quad \forall i \in \mathcal{A}. \quad (3.36b)$$

Constraints (3.34)-(3.36) restrict variables  $\tilde{\delta}_{i,x}$  and  $\tilde{\delta}_{i,y}$  to convex regions thus providing an initial relaxation of the speed lower bound constraint (3.22a). This initial relaxation is illustrated in Figure 3.3 which depicts the variation of  $\delta_{i,x}^2$  and  $\delta_{i,y}^2$  (in red) over the domain of  $\delta_{i,x}$  and  $\delta_{i,y}$  for realistic speed and heading control bounds. The green lines in Figures 3.3a and 3.3b represent the initial McCormick envelopes. This relaxation may still yield infeasible aircraft speeds. To refine this convex relaxation, we introduce mixed-integer cuts that can be generated on-the-fly in a constraint generation algorithm.

We next present the general structure of the proposed mixed-integer cuts before discussing how these cuts are generated iteratively. Let  $\Gamma_{i,x}$  and  $\Gamma_{i,y}$  be partitions of the domain of variables  $\delta_{i,x}$  and  $\delta_{i,y}$ , respectively. Since  $\delta_{i,x}^2$  and  $\delta_{i,y}^2$  are convex, any line joining two extremities of a segment in  $\Gamma_{i,x}$  and  $\Gamma_{i,y}$  refines the initial McCormick upper envelopes. Let  $\alpha_{i,x}^k$  (resp.  $\alpha_{i,y}^k$ ) and  $\beta_{i,x}^k$  (resp.  $\beta_{i,y}^k$ ) be the slope and the intercept corresponding to segment  $k \in \Gamma_{i,x}$  (resp.  $k \in \Gamma_{i,y}$ ). Further, let  $s_{i,x}^k$  (resp.  $s_{i,y}^k$ ) be a binary variable taking value 1 if  $\delta_{i,x}$  (resp.  $\delta_{i,y}$ ) belongs to segment  $k$  of partition  $\Gamma_{i,x}$  (resp.  $\Gamma_{i,y}$ ). The proposed mixed-integer cuts take the form of:

$$\tilde{\delta}_{i,x} \leq \alpha_{i,x}^k \delta_{i,x} + \beta_{i,x}^k, \quad \text{if } s_{i,x}^k = 1, \quad \forall i \in \mathcal{A}, k \in \Gamma_{i,x}, \quad (3.37a)$$

$$\tilde{\delta}_{i,y} \leq \alpha_{i,y}^k \delta_{i,y} + \beta_{i,y}^k, \quad \text{if } s_{i,y}^k = 1, \quad \forall i \in \mathcal{A}, k \in \Gamma_{i,y}. \quad (3.37b)$$

Let  $\underline{\delta}_{i,x}^k$  and  $\bar{\delta}_{i,x}^k$  (resp.  $\underline{\delta}_{i,y}^k$  and  $\bar{\delta}_{i,y}^k$ ) be the extremities of segment  $k \in \Gamma_{i,x}$  (resp.  $k \in \Gamma_{i,y}$ ). Binary variables  $s_{i,x}^k$  and  $s_{i,y}^k$  are defined as:

$$s_{i,x}^k \equiv \begin{cases} 1 & \text{if } \delta_{i,x} \in [\underline{\delta}_{i,x}^k, \bar{\delta}_{i,x}^k[, \\ 0 & \text{otherwise,} \end{cases} \quad \forall i \in \mathcal{A}, k \in \Gamma_{i,x}, \quad (3.38a)$$

$$s_{i,y}^k \equiv \begin{cases} 1 & \text{if } \delta_{i,y} \in [\underline{\delta}_{i,y}^k, \bar{\delta}_{i,y}^k[, \\ 0 & \text{otherwise.} \end{cases} \quad \forall i \in \mathcal{A}, k \in \Gamma_{i,y}. \quad (3.38b)$$

We require the following cut selection constraints:

$$\sum_{k \in \Gamma_{i,x}} s_{i,x}^k = 1 \quad \forall i \in \mathcal{A}, \quad (3.39a)$$

$$\sum_{k \in \Gamma_{i,y}} s_{i,y}^k = 1 \quad \forall i \in \mathcal{A}. \quad (3.39b)$$

The proposed mixed-integer cuts are illustrated in Figures 3.3a and 3.3b for the case of  $|\Gamma_{i,x}| = 2$  and  $|\Gamma_{i,y}| = 4$  segments, respectively. The shaded purple region represents the feasible region of  $\tilde{\delta}_{i,x}$  and  $\tilde{\delta}_{i,y}$  after imposing the mixed-integer cuts. The resulting formulation is a MIQCP with mixed-integer cuts summarized in Model 3.5. This formulation can be solved by off-the-shelf commercial optimization software and provides a relaxation of the Model 3.2 which can be tightened as desired by refining the partitions  $\Gamma_{i,x}$  and  $\Gamma_{i,y}$  of the domain of variables  $\delta_{i,x}$  and  $\delta_{i,y}$ , respectively.

**Model 3.5** (MIQCP 2D formulation).

$$\min \sum_{i \in \mathcal{A}} w \delta_{i,y}^2 + (1-w)(1-\delta_{i,x})^2, \quad (3.40a)$$

s.t.

$$v_{ij,x} = \delta_{i,x} \hat{v}_i \cos(\hat{\theta}_i) - \delta_{i,y} \hat{v}_i \sin(\hat{\theta}_i) - \delta_{j,x} \hat{v}_j \cos(\hat{\theta}_j) + \delta_{j,y} \hat{v}_j \sin(\hat{\theta}_j), \quad \forall (i,j) \in \mathcal{P}_S, \quad (3.40b)$$

$$v_{ij,y} = \delta_{i,y} \hat{v}_i \cos(\hat{\theta}_i) - \delta_{i,x} \hat{v}_i \sin(\hat{\theta}_i) - \delta_{j,y} \hat{v}_j \cos(\hat{\theta}_j) + \delta_{j,x} \hat{v}_j \sin(\hat{\theta}_j), \quad \forall (i,j) \in \mathcal{P}_S, \quad (3.40c)$$

$$v_{ij,y} \hat{x}_{ij} - v_{ij,x} \hat{y}_{ij} \leq 0, \quad \text{if } z_{ij} = 1, \quad \forall (i,j) \in \mathcal{P}_S, \quad (3.40d)$$

$$v_{ij,y} \hat{x}_{ij} - v_{ij,x} \hat{y}_{ij} \geq 0, \quad \text{if } z_{ij} = 0, \quad \forall (i,j) \in \mathcal{P}_S, \quad (3.40e)$$

$$v_{ij,y} \gamma_{ij}^l - v_{ij,x} \phi_{ij}^l \leq 0, \quad \text{if } z_{ij} = 1, \quad \forall (i,j) \in \mathcal{P}_S, \quad (3.40f)$$

$$v_{ij,y} \gamma_{ij}^u - v_{ij,x} \phi_{ij}^u \geq 0, \quad \text{if } z_{ij} = 0, \quad \forall (i,j) \in \mathcal{P}_S, \quad (3.40g)$$

$$\delta_{i,x} \tan(\underline{\theta}_i) \leq \delta_{i,y} \leq \delta_{i,x} \tan(\bar{\theta}_i), \quad \forall i \in \mathcal{A}, \quad (3.40h)$$

$$\tilde{\delta}_{i,x} \geq \delta_{i,x}^2, \quad \forall i \in \mathcal{A}, \quad (3.40i)$$

$$\tilde{\delta}_{i,y} \geq \delta_{i,y}^2, \quad \forall i \in \mathcal{A}, \quad (3.40j)$$

$$\underline{q}_i^2 \leq \tilde{\delta}_{i,x} + \tilde{\delta}_{i,y}, \quad \forall i \in \mathcal{A}, \quad (3.40k)$$

$$\tilde{\delta}_{i,x} \leq (\bar{q}_i + \underline{q}_i \cos(\max\{|\underline{\theta}_i|, |\bar{\theta}_i|\})) \delta_{i,x} - \bar{q}_i \underline{q}_i \cos(\max\{|\underline{\theta}_i|, |\bar{\theta}_i|\}), \quad \forall i \in \mathcal{A}, \quad (3.40l)$$

$$\tilde{\delta}_{i,y} \leq \bar{q}_i (\sin(\underline{\theta}_i) + \sin(\bar{\theta}_i)) \delta_{i,y} - \bar{q}_i^2 \sin(\bar{\theta}_i) \sin(\underline{\theta}_i), \quad \forall i \in \mathcal{A}, \quad (3.40m)$$

$$\tilde{\delta}_{i,x} \leq \alpha_{i,x}^k \delta_{i,x} + \beta_{i,x}^k, \quad \text{if } s_{i,x}^k = 1, \quad \forall i \in \mathcal{A}, k \in \Gamma_{i,x}, \quad (3.40n)$$

$$\tilde{\delta}_{i,y} \leq \alpha_{i,y}^k \delta_{i,y} + \beta_{i,y}^k, \quad \text{if } s_{i,y}^k = 1, \quad \forall i \in \mathcal{A}, k \in \Gamma_{i,y}, \quad (3.40o)$$

$$s_{i,x}^k = 1 \text{ if } \delta_{i,x} \in [\underline{\delta}_{i,x}^k, \bar{\delta}_{i,x}^k], \quad \forall i \in \mathcal{A}, k \in \Gamma_{i,x}, \quad (3.40p)$$

$$s_{i,y}^k = 1 \text{ if } \delta_{i,y} \in [\underline{\delta}_{i,y}^k, \bar{\delta}_{i,y}^k], \quad \forall i \in \mathcal{A}, k \in \Gamma_{i,y}, \quad (3.40q)$$

$$\sum_{k \in \Gamma_{i,x}} s_{i,x}^k = 1 \quad \forall i \in \mathcal{A}, \quad (3.40r)$$

$$\sum_{k \in \Gamma_{i,y}} s_{i,y}^k = 1 \quad \forall i \in \mathcal{A}, \quad (3.40s)$$

$$\underline{q}_i^2 \leq \delta_{i,x}^2 + \delta_{i,y}^2, \quad \forall i \in \mathcal{A}, \quad (3.40t)$$

$$\underline{q}_i \cos(\max\{|\bar{\theta}_i|, |\underline{\theta}_i|\}) \leq \delta_{i,x} \leq \bar{q}_i, \quad \forall i \in \mathcal{A}, \quad (3.40u)$$

$$\bar{q}_i \sin(\underline{\theta}_i) \leq \delta_{i,y} \leq \bar{q}_i \sin(\bar{\theta}_i), \quad \forall i \in \mathcal{A}, \quad (3.40v)$$

$$\underline{v}_{ij,x} \leq v_{ij,x} \leq \bar{v}_{ij,x}, \quad \forall (i,j) \in \mathcal{P}_S, \quad (3.40w)$$

$$\underline{v}_{ij,y} \leq v_{ij,y} \leq \bar{v}_{ij,y}, \quad \forall (i,j) \in \mathcal{P}_S, \quad (3.40x)$$

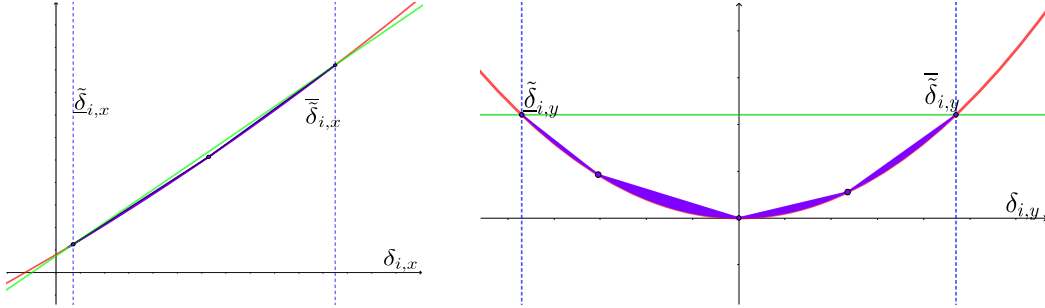
$$z_{ij} \in \{0, 1\}, \quad \forall (i,j) \in \mathcal{P}_S, \quad (3.40y)$$

$$\delta_{i,x}, \delta_{i,y} \in \mathbb{R}, \quad \forall i \in \mathcal{A}, \quad (3.40z)$$

$$\tilde{\delta}_{i,x}, \tilde{\delta}_{i,y} \in \mathbb{R}, \quad \forall i \in \mathcal{A}, \quad (3.40aa)$$

$$s_{i,x}^k \in \{0, 1\}, \quad \forall i \in \mathcal{A}, k \in \Gamma_{i,x}, \quad (3.40ab)$$

$$s_{i,y}^k \in \{0, 1\}, \quad \forall i \in \mathcal{A}, k \in \Gamma_{i,y}. \quad (3.40ac)$$



(a) Graph of  $\delta_{i,x}^2$  (in red) over the domain given by Eq. (3.21a). The mixed-integer cuts (3.37a) are illustrated for a partition of  $|\Gamma_{i,x}| = 2$  segments.

(b) Graph of  $\delta_{i,y}^2$  (in red) over the domain given by Eq. (3.21b). The mixed-integer cuts (3.37b) are illustrated for a partition of  $|\Gamma_{i,y}| = 4$  segments.

Figure 3.3: Piecewise linear approximation of  $\delta_{i,x}^2$  and  $\delta_{i,y}^2$  (in red). The green lines represent the initial McCormick relaxation given by (3.36). The purple shaded regions represent the refined feasible region of  $\tilde{\delta}_{i,x}$  and  $\tilde{\delta}_{i,y}$  after imposing the mixed-integer cuts (3.37). Control bounds are  $\underline{q}_i = 0.94$ ,  $\bar{q}_i = 1.03$ ,  $\underline{\theta}_i = -\frac{\pi}{6}$  and  $\bar{\theta}_i = +\frac{\pi}{6}$ .

We propose a constraint generation algorithm which starts from a relaxed formulation and iteratively refines the piecewise linear outer approximation of the quadratic terms  $\delta_{i,x}^2$  and  $\delta_{i,y}^2$  via mixed-integer cuts (3.37). At each iteration, the proposed constraint generation algorithm examines the solution  $(\delta_{i,x}, \delta_{i,y})$  of each aircraft  $i \in \mathcal{A}$  for speed lower bound violations. If  $\delta_{i,x}^2 + \delta_{i,y}^2 < \underline{q}_i^2$ , then at least one of the relaxed auxiliary variables  $\tilde{\delta}_{i,x}$  or  $\tilde{\delta}_{i,y}$  must be such that  $\tilde{\delta}_{i,x} > \bar{\delta}_{i,x}^2$  or  $\tilde{\delta}_{i,y} > \bar{\delta}_{i,y}^2$ . To eliminate the current infeasible solution, the partition(s)  $\Gamma_{i,x}$  or  $\Gamma_{i,y}$  of the violating variable(s) is augmented by dividing the segment(s) corresponding to  $\delta_{i,x}$  or  $\delta_{i,y}$  into two segments which meet at  $\delta_{i,x}$  or  $\delta_{i,y}$  and corresponding binary variable(s)  $s_{i,x}^k$  or  $s_{i,y}^k$  are added to the formulation. The process is repeated until all aircraft have feasible speed profiles which corresponds to a global optimum of Model 3.2. To further improve the convergence of the solution algorithm, after each solve of the relaxed formulation 3.5, Model 3.2 is solved as a nonlinear program (NLP) by fixing variable  $z_{ij}$ . The pseudo-code of the resulting algorithm is summarized in Algorithm 4.

---

**Algorithm 4:** Solution algorithm for the 2D ACRP

---

**Input:**  $\mathcal{A}, \hat{\theta}, \hat{v}, \underline{q}, \bar{q}, \underline{\theta}, \bar{\theta}, \epsilon$   
**Output:**  $q^*, \theta^*, \text{LB}, \text{UB}$

- 1  $\mathcal{P}, \mathcal{P}_I, \mathcal{P}_F, \mathcal{P}_S \leftarrow$  Algorithm 3
- 2  $\text{LB} \leftarrow 0, \text{UB} \leftarrow +\infty$
- 3  $q, \theta, \delta_x, \delta_y, z, \text{LB} \leftarrow$  Solve MIQP 3.4 and calculate  $q$  and  $\theta$  from  $\delta_x$  and  $\delta_y$
- 4 **if** MIQP 3.4 is infeasible **then**
- 5 |   Return INFEASIBLE
- 6 **if**  $q$  is feasible (no speed violation) **then**
- 7 |    $\text{UB} \leftarrow \text{LB}, q^* \leftarrow q, \theta^* \leftarrow \theta$
- 8 **else**
- 9 |    $\text{CONVERGED} \leftarrow \text{False}$
- 10 **while**  $\text{CONVERGED} = \text{False}$  **do**
- 11 |    $q', \theta', \text{UB-NLP} \leftarrow$  Solve Model 3.2 as NLP with fixed  $z$
- 12 |   **if**  $\text{UB-NLP} < \text{UB}$  **then**
- 13 |   |    $\text{UB} \leftarrow \text{UB-NLP}, q^* \leftarrow q', \theta^* \leftarrow \theta'$
- 14 |   **for**  $i \in \mathcal{A}$  **do**
- 15 |   |   **if**  $\delta_{i,x}^2 + \delta_{i,y}^2 > \bar{q}_i^2$  **then**
- 16 |   |   |   Add constraint (3.22b)
- 17 |   |   **else if**  $\delta_{i,x}^2 + \delta_{i,y}^2 < \underline{q}_i^2$  **then**
- 18 |   |   |   **if**  $\tilde{\delta}_{i,x} > \delta_{i,x}^2$  **then**
- 19 |   |   |   |   Add segment to  $\Gamma_{i,x}$  at  $\delta_{i,x}$ , variable  $s_{i,x}^k$ , constraints (3.37a),  
20 |   |   |   |   (3.38a), and update (3.39a)
- 21 |   |   |   **if**  $\tilde{\delta}_{i,y} > \delta_{i,y}^2$  **then**
- 22 |   |   |   |   Add segment to  $\Gamma_{i,y}$  at  $\delta_{i,y}$ , variable  $s_{i,y}^k$ , constraints (3.37b),  
23 |   |   |   |   (3.38b), and update (3.39b)
- 24 |    $q, \theta, \delta_x, \delta_y, \tilde{\delta}_x, \tilde{\delta}_y, z, \text{LB} \leftarrow$  Solve MIQCP 3.5 and calculate  $q$  and  $\theta$  from  $\delta_x$   
25 |   and  $\delta_y$
- 26 |   **if** MIQCP 3.5 is infeasible **then**
- 27 |   |   Return INFEASIBLE
- 28 |   **if**  $q$  is feasible (no speed violation) **then**
- 29 |   |    $\text{UB} \leftarrow \text{LB}, q^* \leftarrow q, \theta^* \leftarrow \theta$
- 30 |   |    $\text{CONVERGED} \leftarrow \text{True}$
- 31 |   **if**  $(\text{UB-LB})/\text{UB} \leq \epsilon$  **then**
- 32 |   |    $\text{CONVERGED} \leftarrow \text{True}$

---

### 3.4.3 Decomposition algorithm for the 2D+FL ACRP

We next introduce a two-step decomposition approach for the nonconvex lexicographic 2D+FL conflict resolution problem represented by Model 3.3. The first objective function (3.31) focuses on minimising the number of FL changes. This objective function is null and minimal if all aircraft can remain at their initial FL. Since minimising aircraft FL re-assignment is the highest priority, we need only to identify combinations of aircraft which are non-separable in 2D and ensure that such combinations are not assigned to the same FL. Let  $\Omega_I \subseteq 2^{\mathcal{A}}$  be the

set of aircraft combinations which are 2D non-separable, i.e. for any  $\omega \in \Omega_I$ , the subset of aircraft  $\omega$  cannot be separated in 2D. We require:

$$\sum_{i \in \omega} \rho_{ik} \leq |\omega| - 1, \quad \forall \omega \in \Omega_I, \forall k \in \bigcap_{i \in \omega} \mathcal{Z}_i. \quad (3.41)$$

Let  $\Delta\rho_i \geq 0$  be a variable representing the absolute FL deviation for aircraft  $i \in \mathcal{A}$ . Objective function (3.31) can be linearised using traditional mathematical programming techniques. Combining the FL separation constraint (3.41) and the linearised objective function yields a compact FL assignment formulation summarized in Model 3.6 which is a MILP with an exponential number of constraints.

**Model 3.6** (FL assignment formulation).

$$\min \sum_{i \in \mathcal{A}} \Delta\rho_i \quad (3.42a)$$

s.t.

$$\sum_{k \in \mathcal{Z}_i} \rho_{ik} = 1, \quad \forall i \in \mathcal{A}, \quad (3.42b)$$

$$\sum_{i \in \omega} \rho_{ik} \leq |\omega| - 1, \quad \forall \omega \in \Omega_I, \forall k \in \bigcap_{i \in \omega} \mathcal{Z}_i, \quad (3.42c)$$

$$\Delta\rho_i \geq \sum_{k \in \mathcal{Z}_i} k\rho_{ik} - \hat{\rho}_i, \quad \forall i \in \mathcal{A}, \quad (3.42d)$$

$$\Delta\rho_i \geq \hat{\rho}_i - \sum_{k \in \mathcal{Z}_i} k\rho_{ik}, \quad \forall i \in \mathcal{A}, \quad (3.42e)$$

$$\rho_{ik} \in \{0, 1\}, \quad \forall i \in \mathcal{A}, k \in \mathcal{Z}_i, \quad (3.42f)$$

$$\Delta\rho_i \geq 0, \quad \forall i \in \mathcal{A}. \quad (3.42g)$$

We propose to solve Model 3.6 by first restricting Constraint (3.41) to subsets of size two, i.e. 2D non-separable aircraft pairs, which can be efficiently identified using Algorithm 3 in pre-processing. We then decompose the 2D+FL problem into a series of 2D conflict resolution problems, one per FL, based on the optimal solution  $\boldsymbol{\rho}^*$  of the relaxed Model 3.6. For each FL, we solve the corresponding 2D problem using the exact constraint generation approach of Algorithm 4 with the aircraft set  $\mathcal{A}_k \equiv \{i \in \mathcal{A} : \rho_{ik}^* = 1\}$ . If Algorithm 4 returns INFEASIBLE for FL  $k$ , then the corresponding FL separation constraint (3.41) with  $\omega = \mathcal{A}_k$  is generated and Model 3.6 is re-solved with the additional constraint(s). The process is repeated until all 2D problems are feasible. Let  $\mathcal{Z}$  be the set of all FLs, i.e.  $\mathcal{Z} \equiv \cup_{k \in \mathcal{Z}_i: i \in \mathcal{A}} \mathcal{Z}_i$ , we denote  $\mathbf{q}_k$  and  $\boldsymbol{\theta}_k$  the vectors of speed and heading controls for aircraft assigned to FL  $k \in \mathcal{Z}$ , respectively. The proposed approach for the lexicographic 2D+FL conflict resolution problem is summarized in Algorithm 5. Note that this decomposition approach offers the possibility to solve all 2D problems in parallel via the **for** loop at Line 8.

### 3.5 Robust aircraft conflict resolution problem

In this section, we introduce formulations for the robust aircraft conflict resolution problem. We first define the uncertainty model before discussing how it can be incorporated within the ACRP. We then propose a tractable robust counterpart formulation for the robust ACRP.

---

**Algorithm 5:** Solution algorithm for the lexicographic 2D+FL conflict resolution problem

---

**Input:**  $\mathcal{A}, [\mathcal{Z}_i]_{i \in \mathcal{A}}, \hat{\theta}, \hat{v}, \hat{\rho}, \underline{q}, \bar{q}, \underline{\theta}, \bar{\theta}$   
**Output:**  $q^*, \theta^*, \rho^*$

- 1  $\mathcal{P}, \mathcal{P}_F, \mathcal{P}_S, \mathcal{P}_I \leftarrow$  Algorithm 3
- 2  $\Omega_I \leftarrow \{(i, j) \in \mathcal{P}_I : \mathcal{Z}_i \cap \mathcal{Z}_j \neq \emptyset\}$
- 3  $\mathcal{Z} \leftarrow \cup_{k \in \mathcal{Z}_i : i \in \mathcal{A}} \mathcal{Z}_i$
- 4 **CONVERGED**  $\leftarrow$  **False**
- 5 **while** **CONVERGED** = **False** **do**
- 6      $\rho^* \leftarrow$  Solve MILP 3.6
- 7     **CONVERGED**  $\leftarrow$  **True**
- 8     **for**  $k \in \mathcal{Z}$  **do**
- 9          $\mathcal{A}_k \leftarrow \{i \in \mathcal{A} : \rho_{i_k}^* = 1\}$
- 10          $q_k^*, \theta_k^* \leftarrow$  Algorithm 4 with  $\mathcal{A} = \mathcal{A}_k$
- 11         **if** Algorithm 4 returns **INFEASIBLE** **then**
- 12              $\Omega_I \leftarrow \Omega_I \cup \mathcal{A}_k$
- 13             **CONVERGED**  $\leftarrow$  **False**
- 14  $q^*, \theta^* \leftarrow [q_k^*]_{k \in \mathcal{Z}}, [\theta_k^*]_{k \in \mathcal{Z}}$

---

### 3.5.1 Uncertainty model

We assume that each aircraft has a source of randomness and this affects its current velocity and position. Let  $\epsilon_i = [\epsilon_{i,x}, \epsilon_{i,y}]^\top$  be a vector of random variables representing the uncertainty on the velocity components of aircraft  $i \in \mathcal{A}$ . We next use this vector of random variables to define aircraft-based uncertainty sets.

Given aircraft  $i \in \mathcal{A}$ , let  $v_{i,x}$  and  $v_{i,y}$  its velocity components.

$$v_{i,x} = q_i \hat{v}_i \cos(\hat{\theta}_i + \theta_i), \quad (3.43)$$

$$v_{i,y} = q_i \hat{v}_i \sin(\hat{\theta}_i + \theta_i). \quad (3.44)$$

**Definition 3.4** (Uncertainty set of aircraft). *The uncertainty set of aircraft  $i \in \mathcal{A}$ , is defined as:*

$$\mathcal{U}_i \equiv \{\epsilon_i \in \mathbb{R}^2 \mid -\bar{\epsilon}_{i,x} \leq \epsilon_{i,x} \leq \bar{\epsilon}_{i,x}, -\bar{\epsilon}_{i,y} \leq \epsilon_{i,y} \leq \bar{\epsilon}_{i,y}\}, \quad (3.45)$$

where  $\bar{\epsilon}_{i,x} \geq 0$  and  $\bar{\epsilon}_{i,y} \geq 0$  represent the maximum perturbations on the velocity components  $v_{i,x}$  and  $v_{i,y}$ , respectively, of aircraft  $i$ .

We denote  $[\tilde{v}_{i,x}, \tilde{v}_{i,y}]^\top$  the vector of random aircraft velocity components where the random variables  $\tilde{v}_{i,x}$  and  $\tilde{v}_{i,y}$  take values in  $\tilde{v}_{i,x} \in [-v_{i,x}(1 + \bar{\epsilon}_{i,x}), v_{i,x}(1 + \bar{\epsilon}_{i,x})]$  and  $\tilde{v}_{i,y} \in [-v_{i,y}(1 + \bar{\epsilon}_{i,y}), v_{i,y}(1 + \bar{\epsilon}_{i,y})]$ , respectively. Accordingly, for each pair of aircraft  $(i, j) \in \mathcal{P}$ , the random relative velocity components  $\tilde{v}_{ij,x}$  and  $\tilde{v}_{ij,y}$  are:

$$\tilde{v}_{ij,x} = v_{i,x}(1 + \epsilon_{i,x}) - v_{j,x}(1 + \epsilon_{j,x}) = v_{ij,x} + v_{i,x}\epsilon_{i,x} - v_{j,x}\epsilon_{j,x}, \quad (3.46a)$$

$$\tilde{v}_{ij,y} = v_{i,y}(1 + \epsilon_{i,y}) - v_{j,y}(1 + \epsilon_{j,y}) = v_{ij,y} + v_{i,y}\epsilon_{i,y} - v_{j,y}\epsilon_{j,y}. \quad (3.46b)$$



Let  $\tilde{v}_{ij,x}, \bar{v}_{ij,x}$  and  $\tilde{v}_{ij,y}, \bar{v}_{ij,y}$  be the lower and upper bounds for  $\tilde{v}_{ij,x}$  and  $\tilde{v}_{ij,y}$ , respectively. These bounds can be determined using Eq. (3.46) and the bounds on speed and heading control provided in Eqs. (3.12) and (3.13). The derived bounds on the random relative velocity components can be used to define the random relative velocity box which characterizes all possible trajectories for the pair  $(i, j) \in \mathcal{P}$  under uncertainty.

**Definition 3.5** (Random relative velocity box). *Consider a pair of aircraft  $(i, j) \in \mathcal{P}$ . Let  $\mathcal{U}_i$  and  $\mathcal{U}_j$  be the uncertainty sets of aircraft  $i$  and  $j$ , respectively. Let  $\tilde{\mathcal{B}}_{ij}(\mathcal{U}_i, \mathcal{U}_j)$  be the parametric subset of  $\mathbb{R}^2$  defined as*

$$\tilde{\mathcal{B}}_{ij}(\mathcal{U}_i, \mathcal{U}_j) \equiv \left\{ (\tilde{v}_{ij,x}, \tilde{v}_{ij,y}) \in \mathbb{R}^2 : \tilde{v}_{ij,x} \leq \tilde{v}_{ij,x} \leq \bar{v}_{ij,x}, \tilde{v}_{ij,y} \leq \tilde{v}_{ij,y} \leq \bar{v}_{ij,y} \right\}. \quad (3.47)$$

$\tilde{\mathcal{B}}_{ij}(\mathcal{U}_i, \mathcal{U}_j)$  is the random relative velocity box of  $(i, j) \in \mathcal{P}$  under the uncertainty sets  $\mathcal{U}_i$  and  $\mathcal{U}_j$ .

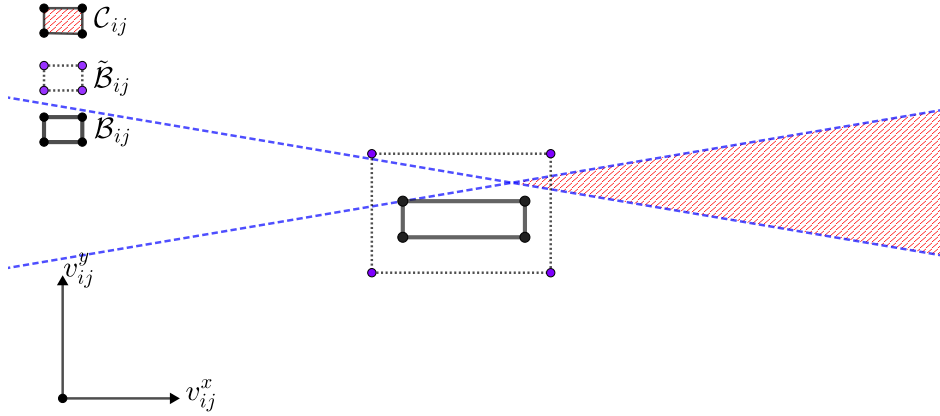


Figure 3.4: Illustration of a two-aircraft conflict in the plane  $\{(v_{ij,x}, v_{ij,y}) \in \mathbb{R}^2\}$ . The inner box with black lines corresponds to the velocity bounds  $\mathcal{B}_{ij}$  in the deterministic scenario while the box with purple dots  $\tilde{\mathcal{B}}_{ij}(\mathcal{U}_i, \mathcal{U}_j)$  is the random relative velocity box. The region is hashed in red corresponds to the conflict region  $\mathcal{C}_{ij}$ . If the uncertainty sets  $\mathcal{U}_i$  and  $\mathcal{U}_j$  are empty, i.e. aircraft trajectories are deterministic, then aircraft  $i$  and  $j$  are conflict-free. In turn, if the uncertainty sets  $\mathcal{U}_i$  and  $\mathcal{U}_j$  are such that the random relative velocity  $\tilde{\mathcal{B}}_{ij}$  intersects with the conflict region  $\mathcal{C}_{ij}$ , then there exists a risk of conflict.

The random relative velocity box is illustrated in Figure 3.4 for a two-aircraft conflict. To ensure that a pair of aircraft  $(i, j) \in \mathcal{P}$  is separated for any realization of the random variables  $\epsilon_i \in \mathcal{U}_i$  and  $\epsilon_j \in \mathcal{U}_j$ , we redefine (3.17e)-(3.17h) using the random velocity variables  $\tilde{v}_{ij,x}$  and  $\tilde{v}_{ij,y}$ :

$$\tilde{v}_{ij,y}\hat{x}_{ij} - \tilde{v}_{ij,x}\hat{y}_{ij} \leq 0, \quad \text{if } z_{ij} = 1, \quad \forall (i, j) \in \mathcal{P}, \quad (N_1)$$

$$\tilde{v}_{ij,y}\hat{x}_{ij} - \tilde{v}_{ij,x}\hat{y}_{ij} \geq 0, \quad \text{if } z_{ij} = 0, \quad \forall (i, j) \in \mathcal{P}, \quad (N_0)$$

$$\tilde{v}_{ij,y}\gamma_{ij}^l - \tilde{v}_{ij,x}\phi_{ij}^l \leq 0, \quad \text{if } z_{ij} = 1, \quad \forall (i, j) \in \mathcal{P}, \quad (S_1)$$

$$\tilde{v}_{ij,y}\gamma_{ij}^u - \tilde{v}_{ij,x}\phi_{ij}^u \geq 0, \quad \text{if } z_{ij} = 0, \quad \forall (i, j) \in \mathcal{P}. \quad (S_0)$$

To find robust aircraft trajectories under uncertainty sets  $\mathcal{U}_i$  and  $\mathcal{U}_j$ , we require that the pairwise separation constraints  $(N_0)$ ,  $(N_1)$ ,  $(S_1)$  and  $(S_0)$  hold for any  $(\tilde{v}_{ij,x}, \tilde{v}_{ij,y}) \in \tilde{\mathcal{B}}_{ij}(\mathcal{U}_i, \mathcal{U}_j)$ . We next use state-of-the-art approaches in robust optimization to integrate these constraints into a robust counterpart formulation.

### 3.5.2 Robust counterpart formulation

The constraints  $(N_0)$ ,  $(N_1)$ ,  $(S_1)$  and  $(S_0)$  are function of the random variables  $\tilde{v}_{ij,x}$  and  $\tilde{v}_{ij,y}$ . We use the approach of Bertsimas and Sim [2004] to reformulate these robust separation constraints as integer-linear constraints with regards to deterministic relative velocity variables  $v_{ij,x}$  and  $v_{ij,y}$ . The constraints  $(N_0)$ ,  $(N_1)$ ,  $(S_1)$  and  $(S_0)$  are of the form  $a\tilde{v}_{ij,x} + b\tilde{v}_{ij,y} \leq 0$  (omitting the disjunction), and can be rearranged by separating deterministic and random elements as follows:

$$a(v_{i,x} - v_{j,x}) - b(v_{i,y} - v_{j,y}) + av_{i,x}\epsilon_{i,x} - av_{j,x}\epsilon_{j,x} - bv_{i,y}\epsilon_{i,y} + bv_{j,y}\epsilon_{j,y} \leq 0. \quad (3.49)$$

We introduce new variables  $\nu_{i,x} \geq 0$  and  $\nu_{i,y} \geq 0$  for each  $i \in \mathcal{A}$  through constraints (3.50) to impose artificial bounds on aircraft velocity components  $v_{i,x}$  and  $v_{i,y}$ :

$$-\nu_{i,x} \leq v_{i,x} \leq \nu_{i,x}, \quad \forall i \in \mathcal{A}, \quad (3.50a)$$

$$-\nu_{i,y} \leq v_{i,y} \leq \nu_{i,y}, \quad \forall i \in \mathcal{A}. \quad (3.50b)$$

Let  $\mathcal{RS} \equiv \{N_1, N_0, S_1, S_0\}$  be a set of indices corresponding to constraints  $(N_0)$ ,  $(N_1)$ ,  $(S_1)$ ,  $(S_0)$ , respectively. Further, let  $\alpha_k = 1$  if  $k = N_1$  or  $k = S_1$ , and let  $\alpha_k = 0$  if  $k = N_0$  or  $k = S_0$ . Let  $\Gamma$  be a real parameter that takes values in the range  $[0,4]$ , where the upper bound is given by the number of decision variables in constraint (3.49). The parameter  $\Gamma$  determines the level of robustness for each robust separation constraint. To link the level of robustness  $\Gamma$  with each robust separation constraint  $k \in \mathcal{RS}$  and each aircraft pair  $(i, j) \in \mathcal{P}$ , we introduce real decision variables  $\psi_{ij}^k \geq 0$ . Further, each constraint of the form (3.49) involves four decision variables, hence for each constraint  $k \in \mathcal{RS}$  and  $(i, j) \in \mathcal{P}$ , we introduce associated variables  $\rho_{ij,x}^{l,k} \geq 0$  and  $\rho_{ij,y}^{l,k} \geq 0$  for  $l \in \{i, j\}$ . These artificial variables reflect the consumption of robustness resources of each velocity variable in (3.49). The following constraints link variables  $\psi_{ij}^k$ ,  $\rho_{ij,x}^{l,k}$  and  $\rho_{ij,y}^{l,k}$  with  $\nu_i^x$  and  $\nu_j^x$ :

$$\psi_{ij}^k + \rho_{ij,x}^{l,k} \geq \nu_{i,x}\bar{\epsilon}_{i,x}, \quad \text{if } z_{ij} = \alpha_k, \quad \forall (i, j) \in \mathcal{P}, \forall l \in \{i, j\}, \forall k \in \mathcal{RS}, \quad (3.51a)$$

$$\psi_{ij}^k + \rho_{ij,y}^{l,k} \geq \nu_{i,y}\bar{\epsilon}_{i,y}, \quad \text{if } z_{ij} = \alpha_k, \quad \forall (i, j) \in \mathcal{P}, \forall l \in \{i, j\}, \forall k \in \mathcal{RS}. \quad (3.51b)$$

Constraints of the form (3.49) can then be rewritten as:

$$a(v_{i,x} - v_{j,x}) - b(v_{i,y} - v_{j,y}) + \psi_{ij}^k \Gamma + \sum_{l \in \{i, j\}} (\rho_{ij,x}^{l,k} + \rho_{ij,y}^{l,k}) \leq 0, \quad \text{if } z_{ij} = \alpha_k, \forall (i, j) \in \mathcal{P}, \forall k \in \mathcal{RS}. \quad (3.52)$$

Combining the above constraints and artificial real variables, we obtain a tractable formulation of the robust separation constraints.

**Proposition 3.4.** *The set of robust separation constraints  $(N_0)$ ,  $(N_1)$ ,  $(S_1)$  and  $(S_0)$  are*

equivalent to the following set of integer-linear constraints and real variables:

$$a(v_{i,x} - v_{j,x}) - b(v_{i,y} - v_{j,y}) + \psi_{ij}^k \Gamma + \sum_{l \in \{i,j\}} (\rho_{ij,x}^{l,k} + \rho_{ij,y}^{l,k}) \leq 0, \quad \text{if } z_{ij} = \alpha_k, \quad \forall (i,j) \in \mathcal{P}, \forall k \in \mathcal{RS}, \quad (3.53a)$$

$$\psi_{ij}^k + \rho_{ij,x}^{l,k} \geq \nu_{i,x} \bar{\epsilon}_{i,x}, \quad \text{if } z_{ij} = \alpha_k, \quad \forall (i,j) \in \mathcal{P}, \forall l \in \{i,j\}, \forall k \in \mathcal{RS}, \quad (3.53b)$$

$$\psi_{ij}^k + \rho_{ij,y}^{l,k} \geq \nu_{i,y} \bar{\epsilon}_{i,y}, \quad \text{if } z_{ij} = \alpha_k, \quad \forall (i,j) \in \mathcal{P}, \forall l \in \{i,j\}, \forall k \in \mathcal{RS}, \quad (3.53c)$$

$$\rho_{ij,x}^{l,k}, \rho_{ij,y}^{l,k} \geq 0, \quad \text{if } z_{ij} = \alpha_k, \quad \forall (i,j) \in \mathcal{P}, \forall l \in \{i,j\}, \forall k \in \mathcal{RS}, \quad (3.53d)$$

$$\psi_{ij}^k \geq 0, \quad \text{if } z_{ij} = \alpha_k, \quad \forall (i,j) \in \mathcal{P}, \forall k \in \mathcal{RS}, \quad (3.53e)$$

$$-\nu_{i,x} \leq v_{i,x} \leq \nu_{i,x}, \quad \forall i \in \mathcal{A}, \quad (3.53f)$$

$$-\nu_{i,y} \leq v_{i,y} \leq \nu_{i,y}, \quad \forall i \in \mathcal{A}, \quad (3.53g)$$

$$\nu_{i,x}, \nu_{i,y} \geq 0, \quad \forall i \in \mathcal{A}. \quad (3.53h)$$

Proposition 3.4 establishes the equivalency between constraints  $(N_0)$ ,  $(N_1)$ ,  $(S_1)$  and  $(S_0)$ , which are expressed in terms of random relative velocity variables  $\tilde{v}_{ij,x}$  and  $\tilde{v}_{ij,y}$ , and a tractable integer-linear reformulation of these constraints which uses additional real variables. Using this reformulation, we propose the following robust counterpart formulation of the ACRP.

**Model 3.7** (Robust counterpart nonconvex formulation of the 2D ACRP).

$$\min \sum_{i \in \mathcal{A}} (1-w)(1-q_i)^2 + w\theta_i^2, \quad (3.54a)$$

s.t.

Motion Equations (3.2),

Robust Separation Constraints and Variables (3.53),

Speed and Heading Control Constraints (3.12), (3.13),

$$v_{ij,x}, v_{ij,y} \in \mathcal{B}_{ij}, \quad \forall (i,j) \in \mathcal{P}, \quad (3.54b)$$

$$z_{ij} \in \{0, 1\}, \quad \forall (i,j) \in \mathcal{P}, \quad (3.54c)$$

$$\underline{q}_i \leq q_i \leq \bar{q}_i, \quad \forall i \in \mathcal{A}, \quad (3.54d)$$

$$\underline{\theta}_i \leq \theta_i \leq \bar{\theta}_i, \quad \forall i \in \mathcal{A}. \quad (3.54e)$$

The formulation presented in Model 3.7 is nonconvex due to trigonometric functions and non-linear components. To solve the robust optimization problem represented by Model 3.7, we use the complex number formulation of the ACRP summarized in Model 3.2 and adapt it for the robust ACRP. Specifically, the complex number reformulation of the robust is constructed by substituting the original motion equations (3.2) with the reformulated

---

aircraft velocities constraints (3.55):

$$v_{i,x} = \delta_{i,x} \hat{v}_i \cos(\hat{\theta}_i) - \delta_{i,y} \hat{v}_i \sin(\hat{\theta}_i), \quad (3.55a)$$

$$v_{i,y} = \delta_{i,y} \hat{v}_i \cos(\hat{\theta}_i) - \delta_{i,x} \hat{v}_i \sin(\hat{\theta}_i). \quad (3.55b)$$

The speed and heading control constraints and variables bounds are replaced with (3.22), (3.23) and (3.21), respectively; and the original objective function (3.17a) is replaced with (3.24). Observe that the robust separation constraints (3.53) remain unchanged since aircraft velocity variables  $v_{i,x}$  and  $v_{i,y}$  are linked to variables  $\delta_{i,x}$  and  $\delta_{i,y}$  via constraints (3.55). We refer to the resulting formulation as the robust complex number formulation and use Algorithm 4 to solve this problem to optimality.

## 3.6 Numerical experiments

We first introduce the experimental framework used to test the proposed mixed-integer formulations and algorithms in Section 3.6.1. We then explore the behaviour of the proposed 2D objective function in Section 3.6.2. Numerical results for the 2D ACRP are presented in Section 3.6.3, and results for the 2D+FL problem are presented in Section 3.6.4.

### 3.6.1 Experiments design

We test the performance of the proposed mixed-integer formulations and algorithms using four benchmarking problems from the literature: the Circle Problem (CP), the Flow Problem (FP), the Grid Problem (GP) and the Random Circle Problem (RCP). The four types of benchmarking instances are illustrated in Figure 3.5. The CP consists of a set of aircraft uniformly positioned on the circle heading towards its centre. Aircraft speeds are assumed to be identical, hence the problem is highly symmetric (see Figure 3.5a). The CP is notoriously difficult due to the geometry of aircraft initial configuration and has been widely used for benchmarking CD&R algorithms in the literature [Durand and Alliot, 2009, Rey et al., 2015, Cafieri and Omhni, 2017, Cafieri and Rey, 2017, Rey and Hijazi, 2017]. To break the symmetry of CP benchmarking instances, Vanaret et al. [2012] introduced the RCP which builds on the same framework, but aircraft initial speeds and headings are randomly deviated within specified ranges to create randomised instances with less structure (see Figure 3.5b). CP and RCP instances are named CP-N and RCP-N, respectively, where N is the total number of aircraft. Lehouillier et al. [2017] formally introduced two additional structured problems which aim to represent more realistic air traffic configurations: the FP and the GP. The FP consists of two streams of aircraft separated by an angle  $\alpha$  and anchored on the circumference of a circle. In each stream, aircraft are separated by at least 5 NM from each other (see Figure 3.5c). The GP consists of two FP instances separated by 15 NM diagonally (see Figure 3.5d). In our experiments, on each stream of aircraft in FP and GP instances, consecutive aircraft are initially separated by 15 NM. FP and GP instances are named FP-N and GP-N, respectively, where N denotes the number of aircraft per stream.

In all experiments, we use a circle of radius 200 NM. For CP, FP and GP instances, all aircraft have an initial speed of 500 NM/h. For RCP instances, aircraft initial speeds are randomly chosen in the range 486-594 NM/h and their initial headings are deviated from a radial trajectory (i.e. towards the centre of the circle) by adding a randomly chosen angle

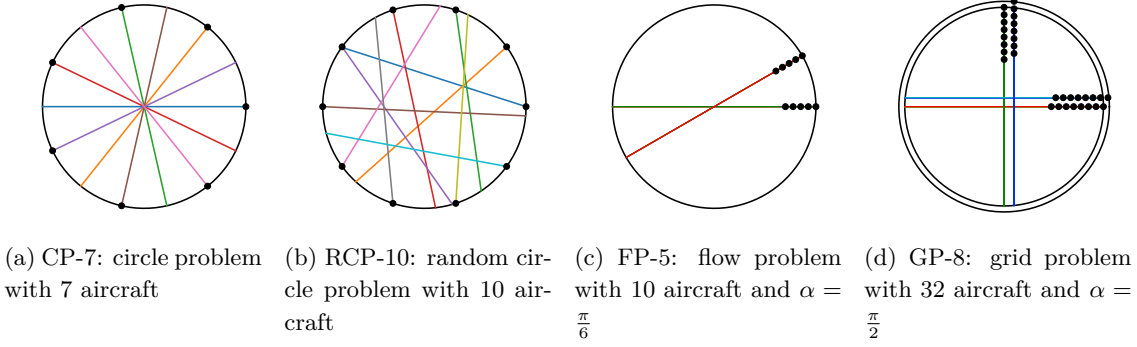


Figure 3.5: Example of 2D benchmarking instances for the Circle Problem (CP), Random Circle Problem (RCP), Flow Problem (FP) and Grid Problem (GP).

between  $-\frac{\pi}{6}$  and  $+\frac{\pi}{6}$ . For FP and GP instances, we use  $\alpha = \frac{\pi}{6}$  and  $\alpha = \frac{\pi}{2}$ , respectively, as proposed by [Lehouillier et al. \[2017\]](#).

We consider a subliminal speed control range of  $[-6\%, +3\%]$  [[Bonini et al., 2009](#)]. We consider two heading control ranges, we first assume that aircraft can modify their heading within the range  $[-30^\circ, +30^\circ]$  as commonly used in the literature [[Cafieri and Omhenni, 2017](#), [Rey and Hijazi, 2017](#)], and we also consider a reduced heading control range of  $[-15^\circ, +15^\circ]$ . For conflict resolution problems with altitude control, we randomly assign each aircraft  $i \in \mathcal{A}$  to a FL  $\hat{\rho}_i \in \mathcal{Z}$ , and we assume that only adjacent FLs are available for aircraft, i.e.  $\mathcal{Z}_i = \{\hat{\rho}_i - 1, \hat{\rho}_i, \hat{\rho}_i + 1\}$ .

The proposed approach is referred to as Disjunctive and is compared to two benchmarks from the literature: the method proposed by [Rey and Hijazi \[2017\]](#) named Disjunctive-2017 and an implementation based on the so-called shadow separation constraints named Shadow. The method Disjunctive corresponds to Algorithm 4 for 2D ACRP instances and to Algorithm 5 for 2D+FL ACRP instances. The method Disjunctive-2017 is based on the same disjunctive linear separation conditions as Disjunctive but uses the algorithm proposed by [Rey and Hijazi \[2017\]](#). This algorithm has 3 steps: i) solve the MIQP relaxation (Model 3.4), ii) solve the MIQCP relaxation (Model 3.5) without any mixed-integer cuts (3.37), and, if a feasible solution has not been obtained, iii) solve Model 3.2 as NLP for fixed  $\mathbf{z}$ . Compared to Disjunctive, the method Disjunctive-2017 is identical in the first step only. In step ii) of Disjunctive-2017, the MIQCP relaxation is solved without any constraint generation, thus all convex quadratic cuts (3.34) are added for all aircraft simultaneously and no mixed-integer cuts are generated. In addition, if the NLP fails to find a feasible solution in step iii), then the method fails to yield a feasible solution whereas the method Disjunctive is guaranteed to converge to a global optimal solution if one exists. In the implementation of the method Shadow, the control variables and the algorithm are identical to that of the method Disjunctive, and the only difference between both methods is the set of separation constraints used, i.e. Constraints (3.10) and (3.11) are replaced with the shadow separation conditions and the number of binary variables required to express these On/Off constraints. Specifically, for the 2D ACRP, Disjunctive only requires a single binary variable per aircraft pair ( $z_{ij}$ ) whereas Shadow requires four binary variables per aircraft pair [[Pallottino et al., 2002](#), [Alonso-Ayuso et al., 2011, 2016](#)]. All three methods are implemented using the same pre-processing procedure (Algorithm 3) to eliminate conflict-free aircraft pairs.

All 2D ACRPs are solved by implementing Algorithm 4 with an optimality gap  $\epsilon = 0.01$

---

and a time limit of 10 minutes. All 2D+FL problems are solved by implementing Algorithm 5 which calls Algorithm 4 using the same optimality gap, and we allocate a time limit of 10 minutes per FL. All models are implemented using PYTHON on a personal computer with 16 GB of RAM and an Intel i7 processor at 2.9GHz. The MIQPs and MIQCPs are solved with CPLEX v12.10 [CPLEX, 2009] API for PYTHON using default options.

We next conduct a sensitivity analysis on the preference weight  $w$  in the 2D objective function to explore its impact on aircraft trajectories in Section 3.6.2. We present results on 2D problems in Section 3.6.3 and results on 2D+FL problems in Section 3.6.4.

### 3.6.2 Sensitivity analysis on the preference weight $w$

To quantify the impact of the preference weight  $w$  in the proposed 2D objective function (3.24), we conduct numerical experiments on one instance of each of the four types of benchmarking instances for varying values of  $w$ . For this experiment, we focus on the typical heading control range  $[-30^\circ, +30^\circ]$ . We report the total speed deviation  $\Sigma_q = \sum_{i \in \mathcal{A}} (1 - q_i)^2$ , and the total heading deviation  $\Sigma_\theta = \sum_{i \in \mathcal{A}} \theta_i^2$ . Our goal is to show that by varying the preference weight  $w \in ]0, 1[$ , the decision-maker can control the desired level of trade-off between total speed deviation and total heading deviation. Recall that in objective function (3.24),  $w$  is the coefficient of  $\delta_{i,y}^2 = (q_i \sin(\theta_i))^2$  which is minimal for  $\theta_i = 0$ ; while  $(1 - w)$  is the coefficient of  $(1 - \delta_{i,x})^2 = (1 - q_i \cos(\theta_i))^2$  which is minimal for  $q_i = 1$  and  $\theta_i = 0$ . Hence, one can expect that increasing (resp. decreasing)  $w$  will tend to penalise heading (resp. speed) deviations more than speed (resp. heading) deviations.

This behaviour is confirmed in our numerical experiments. Specifically, we solve the 2D instances CP-8, FP-10, GP-10 and one RCP-30 instance for  $w = 0.1, \dots, 0.9$  in steps of size 0.1, i.e. for a total of 9 values of  $w$  per instance. All instances are solved to optimality using Algorithm 4 with no MIQCP iterations, i.e. the MIQP returned a global optimal solution for all tests. The change in the total speed deviation  $\Sigma_q$  and in the total heading deviation  $\Sigma_\theta$  are reported in Figure 3.6. For all four instances tested, we find that increasing  $w$  monotonically decreases the total heading deviation and monotonically increases the total speed deviation. Further, we observe that in all cases both the total speed and heading deviations are of similar order of magnitudes.

This sensitivity analysis shows that using the proposed 2D objective function, the decision-maker can control which manoeuvre is prioritised by scaling up or down the preference weight  $w$  accordingly. Higher values of  $w$  will minimise the total heading deviation while lower values of  $w$  will minimise the total speed deviation. We use  $w = 0.5$  in the numerical experiments presented in the remaining of this study.

### 3.6.3 Results on 2D instances

Results for RCP instances are reported in Tables 3.1 and 3.2 for four instance sizes with 10, 20, 30 and 40 aircraft per group. For each instance size, 100 RCP instances are randomly generated. We compare the performance of the proposed formulations for both the standard and reduced heading control ranges.

Each row in the results tables represents a group of instances (RCP). The header of the results tables is presented from left to right. The left-most column, Instance, identifies

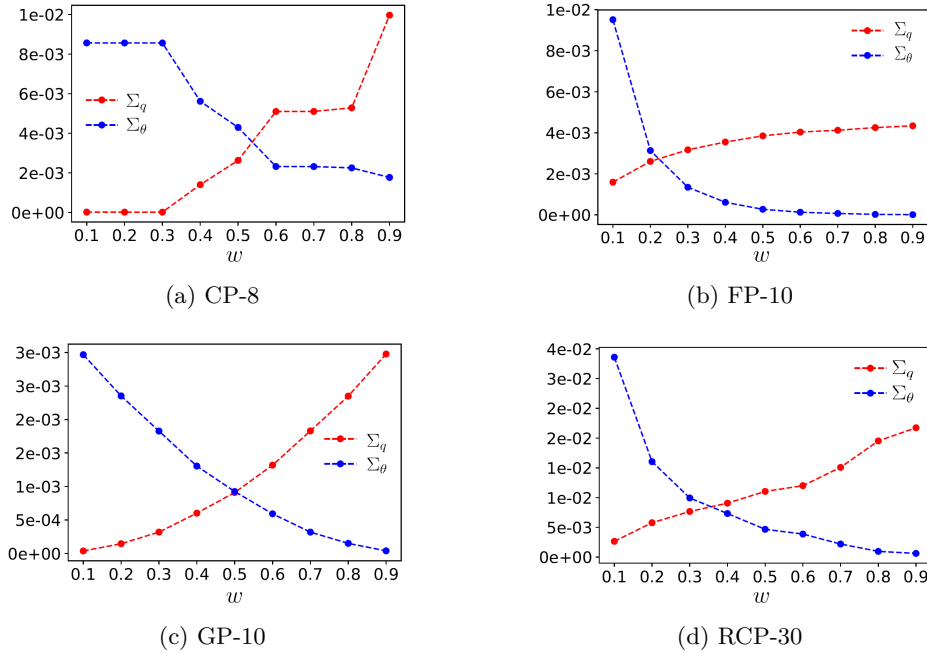


Figure 3.6: Sensitivity analysis on the preference weight  $w$  of the 2D objective function (3.24). For all figures,  $\Sigma_q$  represents the total speed deviation defined as  $\sum_{i \in \mathcal{A}} (1 - q_i)^2$  (in red) and  $\Sigma_\theta$  represents the total heading deviation defined as  $\sum_{i \in \mathcal{A}} \theta_i^2$  (in blue).

the instance;  $|\mathcal{A}|$  is the number of aircraft and  $n_c$  is the number of the conflicts. The next three columns summarize the performance of the pre-processing algorithm:  $|\mathcal{P}_F|/|\mathcal{P}|$  is the proportion of conflict-free aircraft pairs;  $|\mathcal{P}_1|/|\mathcal{P}|$  is the proportion of non-separable aircraft pairs; and Time is the runtime of Algorithm 3 in seconds. The next six columns summarize the performance of Disjunctive: LB and UB are the lower and upper bound; Gap is the optimality gap in percent calculated using LB and UB; Time is the total runtime in seconds;  $n_i$  is the number of iterations of the **while** loop in Algorithm 4, and  $n_t$  represents the proportion of instances that could not be solved within the time limit (10 minutes), i.e. the number of time-outs. The next five columns summarize the performance of Disjunctive-2017:  $\Delta$ UB is the upper bound of Disjunctive-2017 minus that of Disjunctive; Gap is the optimality gap, Time is the total runtime,  $n_t$  is the proportion of time-outs, and Gain is the performance gain in runtime for instances solved within the time limit calculated as the runtime of Disjunctive-2017 minus that of Disjunctive in percentage: a positive value indicates that Disjunctive is faster. The right-most five columns summarize the performance of Shadow relative to Disjunctive, similarly to the five previous columns. For RCP instances, we also report the number of times each method failed to find a feasible solution in columns named  $n_f$ . This value is always zero for CP, FP and GP instances hence we do not report it in their corresponding results tables.

The experiments performed on RCP instances reveal that the pre-processing procedure (Algorithm 3) can eliminate approximately 8% of aircraft pairs when using a reduced heading control range (Table 3.2), whereas no aircraft pairs are conflict-free using a standard heading control range (Table 3.1). The implementation of Disjunctive on RCP instances (Tables 3.1 and 3.2) reveals that while all 10- and 20-aircraft RCP instances can be solved via the MIQP iteration, 30- and 40-aircraft RCP instances may require additional MIQCP iterations. Since no MIQCP iterations are performed ( $n_i = 0$ ) for RCP-10 and RCP-20, the results obtained

using *Disjunctive-2017* are identical to those obtained using *Disjunctive*. Using the standard heading control range (Table 3.1), on average, RCP-10 instances can be solved in 0.05 s and 0.14 s using *Disjunctive* and *Shadow*, respectively. RCP-20 instances require 0.26 s and 2.52 s, on average, using *Disjunctive* and *Shadow*, respectively, under the standard heading control range. The performance of the proposed formulations on RCP-10 and RCP-20 using a reduced heading control range (Table 3.2) is of similar order of magnitude due to the relatively low average number of conflicts per instance, i.e. 3.1 and 13.1, respectively. Those values lead to a performance gain of *Disjunctive* over *Shadow* of 71.8% and 62.6% for standard and reduced ranges, respectively, for RCP-10 instances; and of 89.6% and 81.3% for RCP-20 instances.

RCP-30 and RCP-40 instances have on average 32.9 and 59.3 conflicts, respectively, and present considerable computing challenges, notably the latter. Overall, we observe that reducing the heading control range tends to improve the performance while retaining comparable optimal solutions, as indicated by the similar UB values obtained. *Disjunctive* requires an average of 0.4 MIQCP iterations for RCP-30 instances using a standard heading control range (Table 3.1) compared to 1.4 when using a reduced heading control range (Table 3.2). These figures increase to 0.8 and 1.6 in RCP-40 instances. For RCP-30 instances, the average optimality gaps are 0.17%, 4.60% and 2.36% using *Disjunctive*, *Disjunctive-2017* and *Shadow*, respectively using the standard heading control range (Table 3.1). These figures are reduced to 0.01%, 3.13% and 0.89% using the reduced heading control range (Table 3.2). For RCP-40 instances, the average optimality gaps values are significantly greater: 15%, 51.2% and 29.8% for a standard heading control (Table 3.1); and 13.1%, 52.0% and 25.8% for a reduced heading control range (Table 3.2). This shows that *Shadow* tends to double the optimality gap compared to *Disjunctive*, while *Disjunctive-2017* tends to triple this figure compared to *Disjunctive*. Using a standard heading control range (Table 3.1), *Disjunctive* is able to solve all but 3% of the RCP-30 instances whereas *Disjunctive-2017* and *Shadow* time out on 26% and 20% of these instances, respectively. The performance gain of *Disjunctive* compared to *Disjunctive-2017* and *Shadow* in terms of runtime are 21.0% and 41.7%, respectively. For RCP-40 instances, *Disjunctive* can solve 28% of the problems (72% of time-outs), while *Disjunctive-2017* and *Shadow* time out on 73% and 100% of the instances, respectively. We also observe that *Disjunctive-2017* and *Shadow* may occasionally fail to find a UB as competitive as that found by the *Disjunctive* on RCP-40 instances. The runtime performance gains of *Disjunctive* on RCP-30 instances are 21.0% and 41.7% compared to *Disjunctive-2017* and *Shadow*, respectively, using standard heading control range; and increase to 57.2% and 60.5% using the reduced heading control range. For RCP-40 instances, the gains of *Disjunctive* compared to *Disjunctive-2017* are 3.73% and 24.6% for standard and reduced heading control ranges, respectively; while *Shadow* is unable to solve any RCP-40 instances to optimality (100% of time-outs). The method *Disjunctive-2017* fails to find a feasible solution for 19% and 48% of the RCP-40 instances using the standard and reduced heading control ranges, respectively.

### 3.6.4 Results on 2D+FL instances

Results on 2D+FL instances are reported using similar tables as in Section 3.6.3. Instead of reporting the number of aircraft and the number of conflicts, we report the average number of aircraft and conflicts per FL, i.e.  $\frac{|A|}{|Z|}$ ,  $\frac{n_c}{|Z|}$  respectively. In addition, we include a section corresponding to the FL assignment formulation. In all our numerical experiments on 2D+FL instances, only a single pass through the **while** loop of Algorithm 5 is required. Hence, we only report Obj which is the objective function value of MILP 3.6; and Time which is



Instance	$ \mathcal{A} $	$n_c$	Pre-processing			Disjunctive						
			$\frac{ \mathcal{P}_F }{ \mathcal{P} }$	$\frac{ \mathcal{P}_I }{ \mathcal{P} }$	Time	LB	UB	Gap	Time	$n_i$	$n_t$	$n_f$
RCP-10	10	3.10	0	0	0.12	2.2E-4	2.2E-4	0.00	0.05	0.0	0	0
RCP-20	20	13.1	0	0	0.23	1.7E-3	1.7E-3	0.00	0.26	0.0	0	0
RCP-30	30	32.9	0	0	0.45	7.1E-3	7.1E-3	0.17	135	0.4	3	0
RCP-40	40	59.3	0	0	0.60	1.8E-2	2.4E-2	15.0	516	0.8	72	0

Instance	Disjunctive-2017						Shadow					
	$\Delta$ UB	Gap	Time	$n_t$	$n_f$	Gain	$\Delta$ UB	Gap	Time	$n_t$	$n_f$	Gain
RCP-10	0.00	0.00	0.05	0	0	0.00	0.00	0.02	0.14	0	0	71.8
RCP-20	0.00	0.00	0.26	0	0	0.00	0.00	0.01	2.52	0	0	89.6
RCP-30	0.00	4.60	171	26	0	21.0	0.00	2.36	231	20	0	41.7
RCP-40	0.01	51.2	536	77	19	3.73	0.02	29.8	600	100	0	-

Table 3.1: Results for 2D RCP instances with a speed control range of  $[-6\%, +3\%]$  and a heading control range of  $[-30^\circ, +30^\circ]$ . Times (Time) are reported in seconds. The proportions of conflict-free ( $\frac{|\mathcal{P}_F|}{|\mathcal{P}|}$ ) and non-separable ( $\frac{|\mathcal{P}_I|}{|\mathcal{P}|}$ ) pairs, optimality gaps (Gap), time-outs ( $n_t$ ), infeasible solutions ( $n_f$ ) and the performance gain (Gain) are reported in %.

the corresponding computing runtime in seconds. For numerical experiments on 2D+FL problems, we focus on RCP instances named RCP-N-Z where N is the number of aircraft and Z is the number of FLs. We consider three numbers of aircraft: 50, 100 and 150; and two numbers of FLs: 3 and 5. In terms of conflict density, for the same number of aircraft, instances with 3 FLs have a greater number of conflicts compared to instances with 5 FLs and are more computationally challenging. The results are reported in Table 3.3 for the standard heading control range, and in Table 3.4 for the reduced heading control range.

Using the standard heading control range ( $[-30^\circ, +30^\circ]$ , see Table 3.3), the numerical experiments reveal that among all six groups of 2D+FL instances, only some RCP-150-3 and RCP-150-5 instances may require aircraft to change FLs. That is, for all other 2D+FL instances, all conflicts can be resolved using 2D trajectory control only and thus aircraft do not require performing any FL change. This can be explained by observing that for RCP-150-3 instances, the average number of aircraft per FL is 50 which corresponds to a denser aircraft configuration compared to the 2D RCP-40 instances which all admit feasible solutions. In comparison, only a single RCP-150-5 instance required a FL change for a single aircraft. For RCP-150-5, even though the average density per FL is 30, some instances may have denser FL requiring FL separation. Among all 100 RCP-150-3 instances, 35% of the instances requires a FL change. The maximum number of non-separable pairs ( $|\mathcal{P}_I|$ ) is 2 and the maximum objective value of Model 3.6 is 1, indicating that only a single aircraft deviated from its initial FL.

Comparing the methods, we find that Disjunctive slightly outperforms the two benchmarks in terms of solution quality by occasionally finding better UBs than Disjunctive-2017 or Shadow. For instances with 3 FLs, the optimality gaps are relatively small, i.e. less than 1.0% for RCP-50-3 and RCP-100-3 using Disjunctive and around 10% for RCP-150-3. However, using Disjunctive-2017 and Shadow, the optimality gaps are considerably larger. Especially

Instance	$ \mathcal{A} $	$n_c$	Pre-processing			Disjunctive						
			$\frac{ \mathcal{P}_F }{ \mathcal{P} }$	$\frac{ \mathcal{P}_I }{ \mathcal{P} }$	Time	LB	UB	Gap	Time	$n_i$	$n_t$	$n_f$
RCP-10	10	3.10	8.2	0	0.33	2.2E-4	2.2E-4	0.03	0.04	0.0	0	0
RCP-20	20	13.1	7.7	0	0.51	1.7E-3	1.7E-3	0.01	0.24	0.0	0	0
RCP-30	30	32.9	7.7	0	0.62	7.2E-3	7.2E-3	0.01	66.3	1.4	3	0
RCP-40	40	59.3	7.9	0	0.75	1.8E-2	2.2E-2	13.1	389	1.6	59	0

Instance	Disjunctive-2017						Shadow					
	$\Delta$ UB	Gap	Time	$n_t$	$n_f$	Gain	$\Delta$ UB	Gap	Time	$n_t$	$n_f$	Gain
RCP-10	0.00	0.03	0.04	0	0	0.00	0.00	0.03	0.11	0	0	62.6
RCP-20	0.00	0.01	0.24	0	0	0.00	0.00	0.01	1.39	0	0	81.3
RCP-30	0.00	3.13	155	22	0	57.2	0.00	0.89	167	10	0	60.5
RCP-40	0.01	52.0	516	78	48	24.6	0.02	25.8	600	100	0	-

Table 3.2: Results for 2D RCP instances with a speed control range of  $[-6\%, +3\%]$  and a heading control range of  $[-15^\circ, +15^\circ]$ . Times (Time) are reported in seconds. The proportions of conflict-free ( $\frac{|\mathcal{P}_F|}{|\mathcal{P}|}$ ) and non-separable ( $\frac{|\mathcal{P}_I|}{|\mathcal{P}|}$ ) pairs, optimality gaps (Gap), time-outs ( $n_t$ ), infeasible solutions ( $n_f$ ) and the performance gain (Gain) are reported in %.

for the former which yields an average gap of 15.7% for RCP-50-3 instances, and of 42.0% and 61.6% for RCP-100-3 and RCP-150-3 instances, respectively. Using *Shadow*, those values are less than 1.0% for RCP-50-3 instances, 53.2% for RCP-100-3 instances and 56.6% for RCP-150-3 instances. From a computational standpoint, the average runtime of Model 3.6 on RCP-150-3 instances is 4.56 s. For RCP-50-3 instances, *Disjunctive*, *Disjunctive-2017* and *Shadow* can solve all instances in an average time of 0.88 s, 2.81 s and 10.2 s, respectively. For RCP-100-3 instances, which corresponds to an average number of 33.3 aircraft per FL, *Disjunctive* solves all instances in an average time of 3.3 minutes, whereas using *Disjunctive-2017* there is a total of 64 % instances that time out, and for the remaining instances that can be solved the runtime is 411 s. *Shadow* times out on 46% of the instances and requires an average runtime of 9.5 minutes for the instances solved. The denser RCP-150-3 instances, with an average of 50 aircraft per FL, present substantial computational challenges. *Disjunctive* is able to solve only 40% of these instances (60% of time-outs) and an average of 2.8 MIQCP iterations are required. In comparison, *Disjunctive-2017* is unable to solve most of these instances (95% of time-outs) and fails to find a feasible solution for 5% of them. The method *Shadow* is unable to solve any of these instances within the available time limit. The performance gain using 3 FLs shows that a gain of 87.8% and 51.8% can be obtained using *Disjunctive* over *Disjunctive-2017* and 91.3% and 61.9% using *Disjunctive* over *Shadow* for RCP-50-3 and RCP-100-3 instances, respectively.

Increasing the number of FLs from 3 to 5 reduces the density of aircraft per FL which translates into better computational performance for all three methods. The UB value obtained by those instances using *Disjunctive-2017* is relatively close to the UB obtained with *Disjunctive* yielding deviations of 0.08, 0.09 and 0.10 in RCP-50-5, RCP-100-5 and RCP-150-5 instances, respectively (see Table 3.3). *Shadow* only deviates from the UB found by *Disjunctive* by 0.08 for RCP-150-5 instances. All RCP-50-5 and RCP-100-5 instances are

solved to optimality within the time limit using all three methods. The optimality gap is relatively small for *Disjunctive* and *Shadow*, and does not exceed 0.2% in those instances. Using *Disjunctive-2017*, it reaches 13.4% for RCP-50-5 instances and 45.5% for RCP-100-5 instances. For RCP-150-5 instances, using *Disjunctive* solves all problems with an optimality gap of 0.05%. However, *Disjunctive-2017* fails to solve all RCP-150-5 instances and yields an average optimality gap of 45.1%. The method *Shadow* also fails to solve all RCP-150-5 instances and yields an average optimality gap of 1.56%. In terms of runtime, we observe that *Disjunctive* solves all instances with 5 FLs in less than 142 s (see Table 3.3). The performance of the methods on 2D+FL instances with 5 FLs reveal that gains of 75.6% and 42.7% can be obtained using *Disjunctive* over *Disjunctive-2017*, and 81.8% and 98.8% using *Disjunctive* over *Shadow* for RCP-50-5 and RCP-100-5 instances, respectively.

Reducing the heading control range (see Table 3.4), yields an overall similar performance. The main differences relative to the results obtained using the standard heading control range are observed in the pre-processing procedure which, as in the 2D RCP instances, eliminates on average 8% of the aircraft pairs. We also find that reducing the heading control range increases the number of non-separable pairs to 1.45% for RCP-150-3 instances and to 0.03% for RCP-150-5 instances. This increase in the proportion of non-separable aircraft pairs is reflected in the solution of the flight assignment formulation (MILP 3.6) which has an average value of 1.62 on RCP-150-3 instances and 0.02 on RCP-150-5 instances, respectively. For RCP-150-3, 67% of the instances required a FL change, the maximum number of non-separable pairs ( $|\mathcal{P}_1|$ ) is 6 and the maximum objective value of Model 3.6 is 3, indicating that three aircraft deviated from their initial FL. The methods find the same UBs for RCP-50-5 and RCP-100-5 instances, whereas for RCP-150-5 instances we observe an average deviation of 0.05 using *Disjunctive-2017* and of 0.02 using *Shadow*. The optimality gap is below 1% for all instances using *Disjunctive*, while it ranges from 13.3% in RCP-50-3 instances to 40.1% in RCP-150-3 instances using *Disjunctive-2017*, and from 0.01% to 42.1% using *Shadow*. In terms of runtime, all three methods are able to solve all RCP-50-3 instances in on average 0.88 s, 2.37 s and 11.3 s using *Disjunctive*, *Disjunctive-2017* and *Shadow*, respectively. For RCP-100-3 instances, we observe that all instances are solved by *Disjunctive* in 168 s, while 26% and 39% of them time out using *Disjunctive-2017* and *Shadow*, respectively. In addition, *Disjunctive-2017* fails to find a feasible solution in 10% of the instances. For those instances that can be solved, the average runtime is 474 s for *Disjunctive-2017* and 453 s for *Shadow*. For RCP-150-3 instances, *Disjunctive*, *Disjunctive-2017* and *Shadow* time out on 46%, 74% and 100% of the instances, respectively; and *Disjunctive-2017* fails to find a feasible solution in 14% of the instances. For the remaining instances, the runtime is 420 s using *Disjunctive* and 587 s using *Disjunctive-2017*. The performance gains of *Disjunctive* compared to both benchmarks are above 62% for RCP-50-3 and RCP-100-3 instances, and near 40% compared to *Disjunctive-2017* for RCP-150-3 instances. All RCP-50-5 and RCP-100-5 instances are solved to optimality by all methods. The runtime for those instances is less than 1 s for RCP-50-5 instances and 35.1 s for RCP-100-5 instances using *Disjunctive*, while *Disjunctive-2017* and *Shadow* yield larger runtimes but of the same order of magnitude. These figures do not carry over to RCP-150-5 instances. Using *Disjunctive*, all RCP-150-5 instances can be solved with an average runtime of 152 s. However, using *Disjunctive-2017* all instances time out and this method fails to find a feasible solution in 15% of the cases. Further, the average UB deviation is 0.12 using *Disjunctive-2017* and the average optimality gap is 42.3%. Using *Shadow*, all RCP-150-5 instances also time out, the average UB deviation is 0.09 and the average optimality gap is 12.3%.

Instance	$\frac{ A }{ Z }$	$\frac{n_c}{ Z }$	Pre-processing			FL assignment			Disjunctive					
			$\frac{ P_F }{ P }$	$\frac{ P_I }{ P }$	Time	Obj.	Time	LB	UB	Gap	Time	$n_i$	$n_t$	$n_f$
RCP-50-3	15	3.15	0	0.00	0.42	0.00	0.00	1.2E-2	1.3E-2	0.02	0.88	0.0	0	0
RCP-100-3	33	35.6	0	0.00	0.86	0.00	0.00	1.4E-1	1.4E-1	0.12	198	1.2	0	0
RCP-150-3	50	64.1	0	0.55	1.02	0.62	4.56	4.5E-2	5.2E-2	10.2	520	2.8	60	0
RCP-50-5	10	3.11	0	0.00	0.62	0.00	0.00	9.4E-3	9.4E-3	0.00	0.26	0.0	0	0
RCP-100-5	20	14.1	0	0.00	1.25	0.00	0.00	8.3E-3	8.3E-2	0.00	2.51	0.0	0	0
RCP-150-5	30	34.2	0	0.01	1.92	0.01	0.11	3.3E-1	3.3E-1	0.05	142	2.3	0	0

Instance	Disjunctive-2017							Shadow					
	$\Delta UB$	Gap	Time	$n_t$	$n_f$	Gain	$\Delta UB$	Gap	Time	$n_t$	$n_f$	Gain	
RCP-50-3	0.02	15.7	2.81	0	0	87.8	0.00	0.01	10.2	0	0	91.3	
RCP-100-3	0.02	42.0	411	64	0	51.8	0.05	53.2	572	46	0	61.9	
RCP-150-3	0.10	61.6	600	95	5	-	0.03	56.6	600	100	0	-	
RCP-50-5	0.08	13.4	2.17	0	0	75.6	0.00	0.00	1.43	0	0	81.8	
RCP-100-5	0.09	45.5	67.2	0	0	42.7	0.00	0.13	126	0	0	98.0	
RCP-150-5	0.10	45.1	600	100	0	-	0.08	1.56	600	100	0	-	

Table 3.3: Results for 2D+FL RCP instances with a speed control range of  $[-6\%, +3\%]$  and a heading control range of  $[-30^\circ, +30^\circ]$ . Times (Time) are reported in seconds. The proportions of conflict-free ( $\frac{|P_F|}{|P|}$ ) and non-separable ( $\frac{|P_I|}{|P|}$ ) pairs, optimality gaps (Gap), time-outs ( $n_t$ ), infeasible solutions ( $n_f$ ) and the performance gain (Gain) are reported in %.

### 3.6.5 Results on the robust ACRP

In this section, we analyze the performance of the robust complex number formulation for a varying level of robustness ( $\Gamma$ ) for a fixed maximum uncertainty of  $\bar{\epsilon} = 5\%$ . We present the results for RCP instances in Table 3.5 for 3 instance sizes with 10, 20 and 30 aircraft per group. For each instance group, 100 RCP instances are randomly generated and we report the average performance. For all instances, we compare the performance of the proposed formulations for varying level of robustness where  $\Gamma = 0$  corresponds to the deterministic case and  $\Gamma = 4$  corresponds to the most robust configuration.

Each row in the results tables represents a group of instances (RCP). The header of the results tables is presented from left to right:  $\Gamma$  is the level of robustness;  $UB$  is the objective function value obtained after solving Model 3.7; Gap is the optimality gap in percent obtained after solving Model 3.7, it is calculated directly by CPLEX; Time is the total runtime in seconds,  $n_i$  is the number of MIQCP iterations where  $n_i = 0$  means that an optimal solution was found or the time limit was reached during the initial MIQP solve,  $n_t$  represents the proportion of instances that could not be solved within the time limit (10 minutes), i.e. the number of time-outs and  $n_\emptyset$  indicates the number/proportion of infeasible instances.

The results of the experiments on RCP instances are summarized in Table 3.5. Increasing the level of robustness in RCP instances also tends to increase the objective value (UB), and the effect is on average magnified on instances with a large number of aircraft. For RCP-10 instances, the average objective value for  $\Gamma = 4$  is two orders of magnitude greater than that obtained using a deterministic configuration. For RCP-20 instances, the average objective values obtained are one order of magnitude greater than those obtained for RCP-10. We observe that RCP-10 instances can be solved within less than a second using for any level of

Instance	$\frac{ A }{ Z }$	$\frac{n_c}{ Z }$	Pre-processing			FL assignment			Disjunctive					
			$\frac{ P_F }{ P }$	$\frac{ P_I }{ P }$	Time	Obj.	Time	LB	UB	Gap	Time	$n_i$	$n_t$	$n_f$
RCP-50-3	15	3.15	7.0	0.00	0.52	0.00	0.00	1.2E-2	1.2E-2	0.02	0.88	0.0	0	0
RCP-100-3	33	35.6	7.2	0.00	0.62	0.00	0.00	1.4E-1	1.4E-1	0.15	168	2.2	0	0
RCP-150-3	50	64.1	8.1	1.45	1.22	1.62	5.62	4.5E-2	5.2E-2	8.60	420	4.8	46	0
RCP-50-5	10	3.11	8.6	0.00	0.52	0.00	0.00	9.4E-3	9.4E-3	0.00	0.26	0.0	0	0
RCP-100-5	20	14.1	7.8	0.00	1.22	0.00	0.00	8.3E-3	8.3E-2	0.00	35.1	0.0	0	0
RCP-150-5	30	34.2	8.0	0.03	1.86	0.02	0.21	3.3E-1	3.3E-1	0.51	152	4.3	0	0

Instance	Disjunctive-2017							Shadow					
	$\Delta$ UB	Gap	Time	$n_t$	$n_f$	Gain	$\Delta$ UB	Gap	Time	$n_t$	$n_f$	Gain	
RCP-50-3	0.00	13.3	2.37	0	0	62.9	0.00	0.01	11.3	0	0	91.3	
RCP-100-3	0.00	37.4	474	26	10	64.6	0.08	43.5	453	39	0	62.4	
RCP-150-3	0.05	40.1	587	74	14	39.7	0.02	42.1	600	100	0	-	
RCP-50-5	0.00	11.6	1.22	0	0	78.6	0.00	0.00	1.43	0	0	81.8	
RCP-100-5	0.00	36.3	54.1	0	0	35.1	0.00	0.13	186	0	0	81.2	
RCP-150-5	0.12	42.3	600	100	15	-	0.09	12.3	600	100	0	-	

Table 3.4: Results for 2D+FL RCP instances with a speed control range of  $[-6\%, +3\%]$  and a heading control range of  $[-15^\circ, +15^\circ]$ . Times (Time) are reported in seconds. The proportions of conflict-free ( $\frac{|P_F|}{|P|}$ ) and non-separable ( $\frac{|P_I|}{|P|}$ ) pairs, optimality gaps (Gap), time-outs ( $n_t$ ), infeasible solutions ( $n_f$ ) and the performance gain (Gain) are reported in %.

robustness. RCP-20 instances require less than a minute for  $\Gamma \leq 2$ , on average, but using  $\Gamma$  equal to 3 and to 4, requires around 2 and 3 minutes, respectively. We also find that 34%, 46% and 67% of RCP-20 instances cannot be solved when  $\Gamma$  is equal to 2, 3 and 4, respectively. The results for RCP-30 instances reveal that all problems with  $\Gamma \geq 2$  are infeasible, while only 40% of these instances can be solved with  $\Gamma = 1$  and all 100 RCP-30 instances are feasible in the deterministic case.

The impact of the maximum uncertainty is next examined. For this experiment, we compare the performance of the proposed formulation for a maximum level of robustness, i.e.  $\Gamma = 4$  under varying maximum uncertainty  $\bar{\epsilon}$ : 0% (for the deterministic counterpart), 2.5%, 5%, 7.5% and 10%. The performance is reported similarly as in the previous section.

The results obtained using RCP instances (Table 3.6) reveal that while all problems are feasible under deterministic conditions, the proportion of infeasible problems increase with the maximum uncertainty and the number of aircraft. Notably, all RCP-30-1 instances are found to be infeasible for  $\bar{\epsilon} \geq 2.5\%$  (recall that a maximal level of robustness is used). Examining the change in the objective value, for RCP-10 instances we observe that on average the total deviation of aircraft increases by three orders of magnitude. In terms of runtime, RCP-10 instances, are solved within 1 s using any value of  $\bar{\epsilon}$ . RCP-20 instances, require on average 0.24 s in the deterministic case. Using  $\bar{\epsilon} = 5\%$ , 93 instances are infeasible, 2 timed-out and among the remaining 5 instances solved to optimality the average runtime is 200 s.

$\Gamma$	Instance	UB	Gap	Time	$n_i$	$n_t$	$n_\emptyset$
0	RCP-10	$2.2e^{-4}$	0.00	0.05	0.0	0	0
	RCP-20	$1.7e^{-3}$	0.00	0.26	0.0	0	0
	RCP-30	$7.1e^{-3}$	0.17	135	0.4	3	0
1	RCP-10	$1.0e^{-2}$	0.00	0.24	0.0	0	0
	RCP-20	$9.8e^{-2}$	0.00	10.2	0.0	0	0
	RCP-30	$7.5e^{-1}$	0.17	287	0.4	10	60
2	RCP-10	$2.7e^{-2}$	0.00	0.34	0.0	0	0
	RCP-20	$2.8e^{-2}$	0.00	25.1	0.0	8	34
	RCP-30	-	-	-	-	-	100
3	RCP-10	$4.2e^{-2}$	0.00	0.54	0.0	0	0
	RCP-20	$7.4e^{-1}$	0.00	134	0.0	10	46
	RCP-30	-	-	-	-	-	100
4	RCP-10	$5.3e^{-2}$	0.00	0.59	0.0	0	0
	RCP-20	$8.8e^{-1}$	0.00	201	0.0	15	67
	RCP-30	-	-	-	-	-	100

Table 3.5: Summary of results for RCP instances with a speed control range of  $[-6\%, +3\%]$  and a heading control range of  $[-30^\circ, +30^\circ]$  using a maximal uncertainty of  $\bar{\epsilon} = 5\%$ .

### 3.7 Conclusion

This chapter presented mixed-integer formulations and exact solution methods for aircraft conflict resolution problems (ACRP). We first considered the 2D ACRP with continuous speed and heading control manoeuvres and proposed compact disjunctive separation conditions. The proposed disjunctive separation conditions are linear with regards to aircraft relative velocity variables and only require a single binary variable per pair of aircraft. We introduced a simple pre-processing algorithm to identify aircraft pairs which are conflict-free or non-separable for any combination of controls, which may help in reducing the size of ACRPs by omitting conflict-free pairs. We built on and extended the complex number formulation for the ACRP introduced by [Rey and Hijazi \[2017\]](#) by augmenting its objective function with a preference weight to balance the trade-off between speed and heading deviations. The resulting formulation is a nonconvex mixed-integer program. This 2D formulation is extended to the context of altitude control by flight level (FL) change and we proposed a lexicographic optimization to solve the 2D+FL ACRP which aims to minimise the number of FL changes in priority and resolve outstanding conflicts by 2D trajectory control. The deterministic formulation was also extended to a stochastic variant wherein aircraft velocities are subject to uncertainty and a robust optimization approach was developed.

We presented exact solution algorithms to solve the nonconvex 2D and 2D+FL ACRPs. The proposed algorithms refine the convex relaxations introduced by [Rey and Hijazi \[2017\]](#). For the 2D problem, the nonconvex formulations are first relaxed to mixed-integer quadratic programs (MIQP) which solution may violate the speed control constraint. If such violations occur, convex quadratic constraints are added together with a constraint generation algorithm that iteratively refines an outer piecewise linear approximation of the speed control constraint by solving a sequence of mixed-integer quadratically constrained programs (MIQCP). The 2D+FL lexicographic optimization problem is solved by decomposing the nonconvex problem into a flight assignment problem and a series of FL-based 2D problems. The proposed flight

$\bar{\epsilon}$ (%)	Instance	UB	Gap (%)	Time (s)	$n_i$	$n_t$	$n_\emptyset$
0.0	RCP-10	$2.2e^{-4}$	0.00	0.05	0.0	0	0
	RCP-20	$1.7e^{-3}$	0.01	0.24	0.0	0	0
	RCP-30	$7.2e^{-3}$	0.01	66.3	1.4	3	0
2.5	RCP-10	$1.6e^{-2}$	0.00	0.47	0.0	0	1
	RCP-20	$8.9e^{-1}$	0.00	193	0.0	12	4
	RCP-30	-	-	-	-	-	100
5.0	RCP-10	$5.3e^{-2}$	0.00	0.60	0.0	0	1
	RCP-20	$8.8e^{-1}$	0.00	200	0.0	2	93
	RCP-30	-	-	-	-	-	100
7.5	RCP-10	$1.3e^{-1}$	0.00	0.76	0.0	0	9
	RCP-20	-	-	-	-	-	100
	RCP-30	-	-	-	-	-	100
10.0	RCP-10	$3.0e^{-1}$	0.00	0.89	0.0	0	26
	RCP-20	-	-	-	-	-	100
	RCP-30	-	-	-	-	-	100

Table 3.6: Summary of results for RCP instances with a speed control range of  $[-6\%, +3\%]$  and a heading control range of  $[-30^\circ, +30^\circ]$  using a level of robustness of  $\Gamma = 4$ .

assignment formulation is based on a reformulation of the FL separation constraint which requires an exponential number of constraints and is embedded into an iterative approach to generate altitude separation constraints as needed.

The performance of the proposed mixed-integer formulations and algorithms was tested on a total of 2072 benchmarking instances. These instances include four types of ACRPs with up to 60 aircraft per instance for 2D problems and 150 aircraft per instance for 2D+FL problems. The performance of the proposed solution algorithms highlights the scalability of the approach compared to existing methods in the literature. Further, we find that the combination of the pre-processing algorithm with the MIQP convex relaxation is sufficient to solve FP and GP instances with up to 60 aircraft, and most RCP instances with up to 30 aircraft. We also find that the number of MIQCP iterations remains low on average when solving larger problems. For 2D+FL lexicographic optimization problems, we find that the pre-processing procedure generates enough altitude separation constraints to solve dense instances with an average of 50 aircraft per FL. Last, the results of the sensitivity analyses conducted highlight that while the robust ACRP can be solved without significantly increasing the level of computational resources required, the impact of on system costs (total deviation) increase rapidly with the level of robustness and/or uncertainty.

## Chapter 4

# Modeling approaches for autonomous intersection management

This chapter is based on [Levin and Rey \[2017\]](#), [Rey and Levin \[2019\]](#) and [Rey et al. \[2021\]](#).

### 4.1 Introduction

Traffic intersections are major bottlenecks of urban transport networks. Signalized traffic intersections are typically controlled so as to improve throughput, minimize vehicle delays or reduce emissions. The emergence of automated mobility in urban networks leads to new operational challenges that have received a growing attention over the past few years. Several new technologies have been proposed for AVs to improve traffic flow [Chen and Englund \[2016\]](#). The overarching theme consists of replacing signalized traffic intersection with signal-free intersections that are controlled by an intersection manager. This has led to the development of autonomous intersection management (AIM) wherein the control of signal-free traffic intersections is assumed to be in the hands of an autonomous agent governed by decision-making algorithms that aim to regulate traffic.

Of particular interest is the future paradigm wherein legacy (or human-operated) vehicles have been fully replaced with autonomous vehicles (AV). In this futuristic context, several studies have shown that the management of urban networks may benefit from forecasted technological advancements, such as improved trajectory control, automatic collision avoidance or platooning. However, the intermediate traffic state wherein legacy and autonomous vehicles co-exist has not been nearly as much examined by researchers. In such an intermediate traffic context, urban networks will need to adapt to make the most out of AV technology and allow the transition towards fully autonomous traffic, if this is ever to happen. The first AVs evolving in urban traffic networks are likely to have to abide by the existing infrastructure and legislation. However, this picture may change rapidly. We conjecture that with the increase in AV demand, urban traffic networks will adapt and that AV-specific infrastructure will be available to improve network operations at traffic intersections.

The emergence of automated mobility also offers the opportunity to rethink traffic in-



tersections in the light of users' preferences. Auction-based mechanisms have emerged as promising alternatives to traditional traffic intersection control approaches [Schepperle and Böhm, 2007, Vasirani and Ossowski, 2012, Carlino et al., 2013]. In auction-based mechanisms, users are assumed to be able to declare their preferences, e.g. value of time, to an AIM so as to obtain services commensurate to their need. The intersection can be viewed as a server which role is to process users' service requests. In this context, the intersection manager acts as a controller which decides users' service sequence and users' payments.

The chapter summarizes three approaches that address different facets of AIM. First, a mixed-integer programming formulation is presented to maximize throughput in a fully autonomous mobility context where only AVs share the infrastructure. Second, this formulation is embedded in a stochastic network traffic control with hybrid mobility wherein legacy vehicles (LV) and AVs co-exist. It is shown that in this hybrid mobility context, a max-pressure traffic control policy can stabilize the network. While the design criteria of these approaches have merits they are oblivious to users' preferences. In a last section, a mechanism design perspective is considered wherein users have the possibility to bid for priority service at traffic intersections.

## 4.2 Literature review

We first discuss the literature on autonomous intersection management before discussing studies on network traffic control. We then review efforts that have considered a traffic intersection auction context.

### 4.2.1 Autonomous intersection management

Due to the real-time computing needs of auction-based traffic intersection mechanisms, such approaches are typically conceived in the context of autonomous intersection management (AIM), as proposed in the seminal work of Dresner and Stone [2004, 2008]. In this paradigm, traffic control is assumed to be signal-free and users are assumed to be able reserve space-time trajectories through intersections. Several works have built on and extended the AIM protocol to richer configurations. Fajardo et al. [2011] and Li et al. [2013] proposed AIM protocols based on a First-Come-First-Served (FCFS) policies where vehicles are prioritized based on their arrival time at the intersection. De La Fortelle and Qian [2015] and Alché and De La Fortelle [2016] developed microscopic vehicle trajectory optimization formulations to coordinate vehicles through intersections. In these formulations, the intersection manager decides vehicles' service time and speed while time is discretized. Zhang et al. [2016, 2017] and Malikopoulos et al. [2018] proposed decentralized approaches for traffic control based on FCFS conditions and considered energy consumption as well as vehicle separation and throughput in their formulations. Mirheli et al. [2019] developed a consensus-based approach for cooperative trajectory planning at signal-free intersections and shows that near-optimal solution can be achieved in competitive time. Wu et al. [2019] designed a multi-agent Markov decision process for cooperative trajectory planning in AIM. Levin and Rey [2017] proposed a conflict point model that obviates the need to discretize the intersection space using tiles. This conflict point model was then adapted by Rey and Levin [2019] to accommodate both legacy and automated vehicles with signalized and signal-free traffic phases; and more recently by Chen et al. [2020] to account for pedestrians movements.

---

### 4.2.2 Network traffic control

The literature on network traffic control spans across several fields such as optimization and control theory, transportation engineering and computer science. The seminal work of [Tassiulas and Ephremides \[1992\]](#) pioneered the research on stability conditions in network traffic control with an application to data packets routing in communication networks. The authors notably introduced the concept of back-pressure algorithm as method for decentralized control of a network of routers. In the context of urban transport networks, there is an extensive body of research on network traffic signal control. [Varaiya \[2013\]](#) proposed a network traffic control policy based on max-pressure, a variant of the back-pressure algorithm wherein the control policy chooses the signal phase that maximizes the pressure at each intersection. Stability was proven assuming each turning movement has a distinct queue. [Wongpiromsarn et al. \[2012\]](#) proposed a pressure-based policy which maximizes throughput for unbounded queues. However, practical limitations such as link length require a careful choice of the pressure function to avoid queue spillback. Building on this effort, [Xiao et al. \[2014\]](#) proposed a pressure-releasing policy that accounts for finite queue capacities. Nonetheless, to more canonically apply the pressure-based routing they assumed that each turning movement has a separate queue, which is often not realistic. [Le et al. \[2015\]](#) proposed a fixed-time policy wherein all phases are assigned a non-zero activation time and proved stability for fixed turning proportions and unbounded queues. Recently, [Valls et al. \[2016\]](#) propose a convex optimization approach to traffic signal control that decouples the stability of the system from the choice of traffic control policy.

Starting from the seminal work of [Smith \[1979\]](#), several efforts have also attempted to model the impact of traffic signal optimization on route choice [[Gregoire et al., 2014](#), [Zaidi et al., 2016](#)]. [Le et al. \[2017\]](#) used utility functions in max-pressure control to influence routing. However, while [Tassiulas and Ephremides \[1992\]](#)'s policy is provably throughput-optimal, max-pressure route choice makes no guarantees on the efficiency of the travel times. We assume fixed route choice in our network traffic control model to focus on green and blue phases signal control, and leave route choice for later studies.

### 4.2.3 Traffic intersection auctions

Traffic intersection auctions have received a growing attention over the past few years, notably with the advent of connected and automated vehicle technology. [Schepperle and Böhm \[2007\]](#) proposed a subsidy-based mechanism for slot allocation which aim to balance vehicle waiting time. First-In First-Out (FIFO) constraints are accounted for but incentive-compatibility is not guaranteed. The “effect of starvation” and its impact on fairness are discussed. [Vasirani and Ossowski \[2012\]](#) developed a policy based on combinatorial auctions for the allocation of reservations at traffic intersections. The auction winner pays a price that is exactly the bid that was submitted, which is not incentive-compatible. [Carlino et al. \[2013\]](#) proposed several types of auctions where users can bid on phases or reservations (one vehicle at a time). All drivers can participate in the auction but the candidates only include drivers at the front of their lane. Winners split the cost of the second-highest bid with proportional payment, which yields a static incentive-compatible mechanism and the strategyproofness of the dynamic case is not guaranteed. [Lloret-Batlle and Jayakrishnan \[2016\]](#) proposed envy-minimizing mechanisms for traffic intersections with traditional phasing. The authors developed a revenue-neutral, pareto-efficient approach that minimizes envy and delays by considering monetary transfers across users. A formal proof of incentive-compatibility is not provided although sim-

Paper	Mechanism	Auction	IC	F	E	C
Schepperle and Böhm [2007]	subsidy-based	static	no	some	no	some
Vasirani and Ossowski [2012]	combinatorial auction	static	no	no	yes	yes
Carlino et al. [2013]	marginal cost	static	yes	yes	yes	some
Lloret-Batlle and Jayakrishnan [2016]	envy-minimizing	static	no	yes	yes	yes
Sayin et al. [2018]	marginal cost	static	yes	no	yes	yes
Censi et al. [2019]	karma system	static	no	yes	yes	some
Lin and Jabari [2020]	transferable utility	static	no	some	yes	yes
Rey et al. [2021]	expected marginal cost	dynamic	yes	no	yes	some

Table 4.1: Summary of the state-of-the-art on traffic intersection auctions. The column **Mechanism** describes the mechanism; **Auction** indicates the type of auction; **IC** indicates whether the mechanism is incentive-compatible; **F** indicates if fairness is considered in the mechanism; **E** indicates if the mechanism is efficient, i.e. social welfare is optimized; **C** indicates if intersection capacity is considered or optimized.

ulations show that arbitrage is unlikely. Sayin et al. [2018] proposed an auction protocol and mechanism in which all vehicles within range of the intersection manager communicate their value of time, and the intersection manager assigns reservations to those vehicles via a static auction. The proposed mechanism is incentive-compatible in the static sense, i.e. assuming all participants in the auction are known at the time payments are determined. Censi et al. [2019] introduced a credit-based auction mechanism, i.e. a karma system in which agents pay other agents karma for priority. Agents with a low priority today have an incentive to lose bids so they can achieve higher priority tomorrow by acquiring more karma. The authors attempt to calculate a Nash equilibria of user bids, and they show that a centralized strategy can be more unfair than some of the Nash equilibria identified. Recently, Lin and Jabari [2020] proposed a mechanism for pricing intersection priority based on transferable utility games. In each game, players have the possibility to trade time among themselves and “winners pay losers to gain priority”. The authors provide empirical evidence that their approach is robust against adversarial user behavior, but no formal proof of incentive-compatibility is presented.

A synthesis of the state-of-the-art on traffic intersection auctions is presented in Table 4.1. The column **Mechanism** describes the mechanism of the corresponding paper. The column **Auction** indicates the type of auction: static or dynamic. A static auction is an auction in which the set of participants is assumed known, whereas a dynamic auction allows participants to enter and leave the auction dynamically. Observe that, in a static traffic intersection auction, losers may need participate in multiple auction rounds until being serviced during which new users may arrive and further delay these losers. The column **IC** indicates whether the mechanism is incentive-compatible or not. The column **F** indicates if fairness is considered in the mechanism, either to some extent (some) or via constraints within the mechanism (yes). The column **E** indicates if the mechanism is efficient, i.e. if social welfare is optimized. The column **C** indicates if intersection capacity is considered in the mechanism, either to some extent (some) or explicitly optimized (yes). Table 4.1 highlights that, to the best of our knowledge, existing mechanisms for traffic intersection auctions are restricted to static auctions. This synthesis also emphasizes that fairness considerations and intersection capacity are seldom optimized jointly, thus underlining existing research gaps in the design of traffic intersection auction mechanisms.

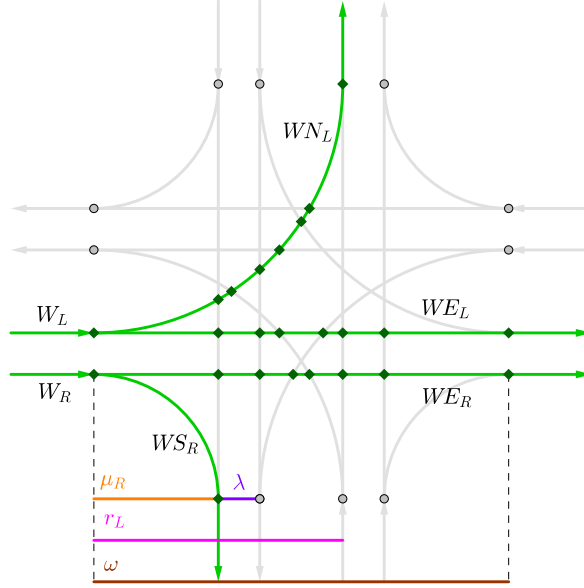


Figure 4.1: 4-approach, 2-lane intersection. Four vehicle paths are available  $WS_R$ ,  $WE_R$ ,  $WE_L$ ,  $WN_L$  for the West approach. All conflicts points on each trajectory are represented with a rhombus.

### 4.3 Conflict point formulation

We now develop an MILP to determine the optimal reservations for a single intersection. We replace the tiles in early work on reservations [Dresner and Stone 2004] with *conflict points*, which are the intersections between different turning movement paths. Conflict points are an alternative method of ensuring collision avoidance that are more amenable to mathematical programming formulations. Figure 4.1 illustrates the conflict points for a standard four-approach intersection. Details on the computation of geometrical parameters are provided in Levin and Rey [2017].

Two vehicles are separated if they do not occupy the same conflict point at the same time. This is similar to the separation requirement for tiles [Dresner and Stone, 2004, 2006], but conflict points are only placed where potential conflicts occur. Conflict points have previously been used to model reservations by Zhu and Ukkusuri [2015], although their formulation was intended for use within dynamic traffic assignment rather than as a deployable solution to reservation-based intersection control. Conflict points have also been used for aircraft separation models Rey et al. [2012, 2015, 2016c], but aircraft require different separation constraints.

#### 4.3.1 Notation

Consider a single intersection with set of vehicles  $\mathcal{V}$ . The intersection has sets of entry and exit lanes,  $\Gamma^-$  and  $\Gamma^+$ , respectively. Let  $\mathcal{C}$  be the set of all conflict points. Each vehicle requests a turning movement traveling from an entry lane to an exit lane. We assume all specified turning movements are legal and physically feasible. Each vehicle requires a path  $\rho_i$  through the intersection, which is an ordered set of conflict points. For each vehicle  $i$ , we

must determine  $t_i(c)$ , which is the time that  $i$  passes through point  $c$ , for all  $c \in \rho_i$ . The first point in  $\rho_i$  is the entrance to the intersection, and the last point is the exit point for the intersection. Then  $t_i(\gamma_i^-)$  is the time at which  $i$  enters the intersection.

We assume that vehicles travel at an uniform speed through the intersection, but that the speed is a decision variable. The reason for this assumption is that varying speeds can greatly complicate determining how long vehicles occupy conflict points because vehicles may occupy multiple conflict points simultaneously. Although the uniform speed assumption may be relaxed in future work, we use it here to focus on the exposition of the conflict avoidance formulation.

Let  $\underline{U}_i$  and  $\overline{U}_i$  be the minimum and maximum speed for which  $i$  can travel through the intersection, respectively. We require a minimum speed to ensure that vehicles do not spend arbitrarily long times in the intersection. For each vehicle  $i$ , and conflict point  $c \in \rho_i$ ,  $i$  takes some time  $\tau_i(c)$  to completely pass through  $c$ .  $\tau_i(c)$  is a function of the vehicle speed during the turning movement. Thus, when  $i$  moves through the intersection,  $c$  is unusable during the time interval  $[t_i(c), t_i(c) + \tau_i(c)]$ . An appropriate buffer may be added to  $\tau_i(c)$  to ensure sufficient separation. Observe that  $t_i(\gamma_i^+) + \tau_i(\gamma_i^+)$  is the time at which  $i$  exits the intersection.

### 4.3.2 Vehicle ordering

Assume that vehicles do not change lanes while under the control of the intersection manager. Therefore, FIFO holds on all lanes. Let  $e_i$  denote the earliest time that  $i$  could reach the intersection. Then

$$t_i(\gamma_i^-) \geq e_i, \quad \forall i \in \mathcal{V}. \quad (4.1)$$

Assume without loss of generality that  $e_i < e_j$ , so  $i$  comes before  $j$  in the lane queue. Then  $t_i(\gamma_i^-) < t_j(\gamma_j^-)$  must hold as well. Because  $i$  occupies  $\gamma_i^-$  for  $\tau_i(\gamma_i^-)$ :

$$t_i(\gamma_i^-) + \tau_i(\gamma_i^-) \leq t_j(\gamma_j^-), \quad \forall i, j \in \mathcal{V} : \gamma_i^- = \gamma_j^-, e_i < e_j. \quad (4.2)$$

### 4.3.3 Travel time between conflict points

For some vehicle  $i$  and any two conflict points  $c_1, c_2 \in \rho_i$  with  $c_2$  after  $c_1$  in  $\rho_i$ , let  $d_i(c_1, c_2)$  be the distance  $i$  travels between  $c_1$  and  $c_2$ . Note that  $d_i(c_1, c_2)$  may not be the Euclidean distance between  $c_1$  and  $c_2$  because turning vehicles may travel along an elliptical path (e.g. for left turns). The minimum and maximum vehicle speeds place constraints on the travel time through the intersection:

$$\frac{d_i(\gamma_i^-, \gamma_i^+)}{\overline{U}_i} \leq t_i(\gamma_i^+) - t_i(\gamma_i^-) \leq \frac{d_i(\gamma_i^-, \gamma_i^+)}{\underline{U}_i}, \quad \forall i \in \mathcal{V}. \quad (4.3)$$

Furthermore, we require that vehicles maintain the same speed throughout the intersection. This is to ensure that vehicles move through conflict points properly. If several conflict points are close together, with distance smaller than vehicle length, speed changes could result in the vehicle blocking a conflict point but not triggering the corresponding constraint.

Therefore:

$$\frac{t_i(c) - t_i(\gamma_i^-)}{d_i(\gamma_i^-, c)} = \frac{t_i(\gamma_i^+) - t_i(\gamma_i^-)}{d(\gamma_i^-, \gamma_i^+)}, \quad \forall i \in \mathcal{V}, \forall c \in \rho_i. \quad (4.4)$$

#### 4.3.4 Time spent at conflict points

For each vehicle  $i$  traversing the intersection, we can determine the time spent at each conflict point  $c \in \rho_i$ ,  $\tau_i(c)$ . based on geometry of  $i$  and turning movement shape (straight or circular). Specifically, let  $D_i(c)$  be the spacing required by  $i$  at  $c$ . This spacing should depend on the vehicle speed in case an emergency braking is required to avoid a collision for unforeseen circumstances (such as a vehicle stall or a pedestrian jaywalking). The spacing for a given speed may be derived from the speed-density relationship, and can be used to specify the desired flow-density relationship for reservation travel. Equation (4.4) gives the speed of vehicle  $i$ ,  $u_i$ , as:

$$u_i = \frac{d_i(\gamma_i^-, \gamma_i^+)}{t_i(\gamma_i^+) - t_i(\gamma_i^-)}. \quad (4.5)$$

Using the inverse speed of the vehicle, the time vehicle  $i$  spends at conflict point  $c$  is:

$$\tau_i(c) = \frac{D_i(c)}{u_i}, \quad \forall i \in \mathcal{V}, \forall c \in \rho_i. \quad (4.6)$$

For simplicity in the formulation, assume that the fundamental diagram is triangular [Newell \[1993\]](#), [Yperman et al. \[2005\]](#). The fundamental diagram is defined as:

$$q(k) = \min \left\{ \bar{U}_i k, w(K - k) \right\}, \quad (4.7)$$

where  $k$  is the density,  $K$  is the jam density,  $w$  is the congested wave speed, and  $q(k)$  is the flow-density relationship. We assume that  $K$  and  $w$  are constants.

Speeds are constrained by spacing only in the congested region, and from equation (4.7) the speed satisfies:

$$u_i = \frac{w}{k} (K - k). \quad (4.8)$$

Let  $L_i(c)$  be the distance the vehicle travels around  $c$ . If the turning movement is a straight line,  $L_i(c) = l_i$ , where  $l_i$  is the length of  $i$ . For left and right turns, the vehicle will travel for a greater distance while occupying  $c$ . For circular turns (e.g. left and right turns in a symmetric intersection),  $L_i(c) = 4R \sin^{-1} \left( \frac{l_i}{2R} \right)$  where  $R$  is the turn radius.

Using the fact that spacing is the inverse of density, i.e.  $D_i(c) = \frac{1}{k}$  and  $K = \frac{1}{L_i(c)}$ , equation (4.8) can be rewritten as

$$D_i(c) = L_i(c) \frac{u_i + w}{w}. \quad (4.9)$$

From equation (4.9), equation (4.6) can then be rewritten as

$$\begin{aligned}
\tau_i(c) &= L_i(c) \frac{u_i + w}{u_i w}, \\
&= \frac{L_i(c)}{w} + \frac{L_i(c)}{u_i}, \\
&= \frac{L_i(c)}{w} + \frac{L_i(c) \left( t_i(\gamma_i^+) - t_i(\gamma_i^-) \right)}{d_i(\gamma_i^-, \gamma_i^+)},
\end{aligned} \tag{4.10}$$

which is linear in the decision variables of conflict point arrival times.

### 4.3.5 Separation at conflict points

For any two vehicles  $i, j \in \mathcal{V}$ , let  $\delta_{ij}(c) \in \{0, 1\}$  denote whether  $i$  and  $j$  overlap at  $c$ .  $\delta_{ij}(c)$  is sensitive to the order of vehicles, so each pair of vehicles has two such  $\delta$  variables per shared conflict point.  $\delta_{ij}(c) = 1$  if  $j$  arrives at  $c$  after  $i$  has left. Thus,  $\delta_{ij}(c) = 1$  if  $t_j(c) \geq t_i(c) + \tau_i(c)$ . This is written as

$$t_i(c) + \tau_i(c) - t_j(c) \leq (1 - \delta_{ij}(c))M_{ij}, \quad \forall i, j \in \mathcal{V}, \forall c \in \rho_i \cap \rho_j, \tag{4.11}$$

where  $M_{ij}$  is a large positive constant. If  $t_j(c) \geq t_i(c) + \tau_i(c)$  then  $\delta_{ij}(c) = 1$ ; otherwise  $\delta_{ij}(c) = 0$ . Note that it suffices to have  $M_{ij} \geq \max\{t_i(c) + \tau_i(c) - t_j(c)\}$  for this separation constraint to be valid.

Separation is guaranteed for any two pair of vehicles  $i, j \in \mathcal{V}$  and any conflict point  $c$  if  $j$  enters  $c$  after  $i$  leaves, or if  $i$  enters  $c$  after  $j$  leaves:

$$\delta_{ij}(c) + \delta_{ji}(c) = 1, \quad \forall i, j \in \mathcal{V} : i < j, \forall c \in \rho_i \cap \rho_j. \tag{4.12}$$

Constraint (4.12) does not depend on vehicle ordering, and therefore occurs once per pair of vehicles per shared conflict point.

Note that the number of binary  $\delta_{ij}(c)$  variables may be reduced for vehicles with  $\gamma_i^- = \gamma_j^-$ . For such pairs vehicles, the ordering at the entry point determines the ordering at any conflict points that they share. Constraints (4.11) and (4.12) may therefore be replaced with

$$t_i(c) + \tau_i(c) \leq t_j(c), \quad \forall i, j \in \mathcal{V} : \gamma_i^- = \gamma_j^-, e_i < e_j, \forall c \in \rho_i \cap \rho_j, \tag{4.13}$$

for pairs of vehicles  $i, j \in \mathcal{V}$  with  $\gamma_i^- = \gamma_j^-$ .

### 4.3.6 Objective function

The objective function represents how the intersection manager aims to operate traffic. An intuitive objective consists in maximizing intersection throughput. While this approach may maximize social welfare, such an intersection control policy may also lead to imbalanced vehicle delays, thus potentially impacting acceptability. Alternatively, the intersection manager may seek to minimize the maximum delay experienced or maximize fairness. These alternative objective functions aim to incorporate some level of fairness in the allocation of reservation assignments at the vehicular scale. More elaborate fair objective functions can

---

also be designed to balance reservation assignments across routes or traffic approaches (e.g. North, South, East, West).

Because the constraints guarantee complete separation of vehicles through the intersection, the objective function decides in which order vehicles should move. Denote by  $f_i > 0$  the weight of each vehicle, which indicates its priority for movement. The weights can model a variety of situations. For instance,  $f_i = 1$  for all vehicles is maximum throughput.  $f_i = e_i$  weights vehicles by earliest arrival time, which is similar to a fairness based objective. Note that  $f_i = e_i$  is not the same as the FCFS policy used in previous work on reservations, because FCFS results in absolute priority for the vehicle with the earliest arrival time. The objective  $f_i = e_i$  may delay early-arriving vehicles to find the combination with the greatest fairness.

The proposed objective function values movements by the time vehicles exit the intersection,  $t_i(\gamma_i^+) + \tau_i(\gamma_i^+)$ :

$$\min \sum_{i \in \mathcal{V}} f_i \left( t_i \left( \gamma_i^+ \right) + \tau_i \left( \gamma_i^+ \right) \right). \quad (4.14)$$

Exit times are chosen because vehicle speeds are an implicit decision variable. Both vehicle delays and slower turn speeds will increase the objective value. Note that the objective function (4.14) will never be equal to 0 if  $f_i > 0$ . To obtain a measure of delay for vehicle  $i$ , we can subtract the constant term  $\frac{d_i(\gamma_i^-, \gamma_i^+)}{U_i} + e_i$ , which is the earliest time that vehicle  $i$  could exit the intersection.

### 4.3.7 Mixed integer linear program

Combining the above constraints and objective function results in the MILP represented by (4.15) through (4.24), henceforth referred to as the *Conflict-Point Intersection Control*



(CPIC) formulation.

$$\min \sum_{i \in \mathcal{V}} f_i \left( t_i \left( \gamma_i^+ \right) + \tau_i \left( \gamma_i^+ \right) \right), \quad (4.15)$$

$$\text{s.t. } t_i \left( \gamma_i^- \right) \geq e_i, \quad \forall i \in \mathcal{V}, \quad (4.16)$$

$$t_i \left( \gamma_i^- \right) + \tau_i \left( \gamma_i^- \right) \leq t_j \left( \gamma_j^- \right), \quad \forall i, j \in \mathcal{V} : \gamma_i^- = \gamma_j^-, e_i < e_j, \quad (4.17)$$

$$\tau_i(c) = \frac{L_i}{w} + \frac{L_i \left( t_i(\gamma_i^+) - t_i(\gamma_i^-) \right)}{d_i \left( \gamma_i^-, \gamma_i^+ \right)}, \quad \forall i \in \mathcal{V}, \forall c \in \rho_i, \quad (4.18)$$

$$\frac{d_i \left( \gamma_i^-, \gamma_i^+ \right)}{\bar{U}_i} \leq t_i \left( \gamma_i^+ \right) - t_i \left( \gamma_i^- \right) \leq \frac{d_i \left( \gamma_i^-, \gamma_i^+ \right)}{\underline{U}_i}, \quad \forall i \in \mathcal{V}, \quad (4.19)$$

$$\frac{t_i(c) - t_i \left( \gamma_i^- \right)}{d_i \left( \gamma_i^-, c \right)} = \frac{t_i \left( \gamma_i^+ \right) - t_i \left( \gamma_i^- \right)}{d \left( \gamma_i^-, \gamma_i^+ \right)}, \quad \forall i \in \mathcal{V}, \forall c \in \rho_i, \quad (4.20)$$

$$t_i(c) + \tau_i(c) \leq t_j(c), \quad \forall i, j \in \mathcal{V} : \gamma_i^- = \gamma_j^-, e_i < e_j, \forall c \in \rho_i \cap \rho_j, \quad (4.21)$$

$$t_i(c) + \tau_i(c) - t_j(c) \leq (1 - \delta_{ij}(c)) M_{ij}, \quad \forall i, j \in \mathcal{V} : \gamma_i^- \neq \gamma_j^-, \forall c \in \rho_i \cap \rho_j, \quad (4.22)$$

$$\delta_{ij}(c) + \delta_{ji}(c) = 1, \quad \forall i, j \in \mathcal{V} : \gamma_i^- \neq \gamma_j^-, i < j, \forall c \in \rho_i \cap \rho_j, \quad (4.23)$$

$$\delta_{ij}(c) \in \{0, 1\}, \quad \forall i, j \in \mathcal{V} : \gamma_i^- \neq \gamma_j^-, \forall c \in \rho_i \cap \rho_j, \quad (4.24)$$

The decision variables are the arrival times at each conflict point for each vehicle  $i$ ,  $t_i(c)$ , and vehicle ordering at conflict points, determined by  $\delta_{ij}(c)$  variables. The times spent at conflict points,  $\tau_i(c)$ , is defined by the speed at which vehicle  $i$  moves through the intersection, which is implicitly defined by vehicle-specific arrival times at conflict points. It is not known whether Model CPIC is NP-hard. However, similar collision avoidance models for aircraft [Chiang et al., 1997] can be generalized as a motion planning problem in the presence of obstacles, which is NP-hard in most cases [Reif and Sharir, 1994].

Model CPIC is always feasible because there is no upper bound on vehicle entry and exit times. If necessary, vehicles could be given arbitrarily long delays to ensure that collision avoidance constraints are satisfied, this is summarized in the following proposition.

**Proposition 4.1.** *Model CPIC has at least one feasible solution.*

Vehicles' waiting time at the intersection entry points is unbounded, hence the values  $M_{ij}$  may also be unbounded. Specifically, if  $\delta_{ij} = 0$ , then Constraint (4.22) becomes  $t_i(c) + \tau_i(c) - t_j(c) \leq M_{ij}$ , which should hold for any values of the decision variables involved therein. Since  $\max\{t_i(c) + \tau_i(c) - t_j(c)\} \rightarrow \infty$ , the big-M values should be chosen carefully to maintain feasibility and avoid cutting out the optimal solution. The proof of Proposition 4.1 provides a method to construct a feasible solution to Model CPIC. We can use this solution to tighten the big-M values  $M_{ij}$ . Let  $i = |\mathcal{V}| - 1$  denote the last vehicle in the sorted set  $\mathcal{V}$  and let  $\bar{t} = t_{|\mathcal{V}|-1} \left( \gamma_{|\mathcal{V}|-1}^+ \right) + \tau_{|\mathcal{V}|-1} \left( \gamma_{|\mathcal{V}|-1}^+ \right)$  denote the latest instant reserved by the intersection. Then  $M = \bar{t} - e_0$  is a valid big-M value for any pair of vehicles.

---

## 4.4 Hybrid network traffic control policy

We now embed the intersection-based conflict point formulation into a stochastic network model wherein LVs and AVs co-exist.

### 4.4.1 Stochastic network model

Consider now a traffic network  $\mathcal{G} = (\mathcal{N}, \mathcal{A})$  with a set of intersections  $\mathcal{N}$  connected by a set of links  $\mathcal{A}$ . The set of links is partitioned into three subsets: internal links that connect two intersections, denoted  $\mathcal{A}_o \subset \mathcal{A}$ , source links at which vehicles enter the network, denoted  $\mathcal{A}_r \subset \mathcal{A}$ , and sink links at which vehicles exit, denoted  $\mathcal{A}_s \subset \mathcal{A}$ . We model congestion at network intersections using point-queues of infinite size and we are interested in the evolution of queue lengths over the entire network.

We consider two classes of vehicles and lanes: autonomous vehicles (AVs), denoted  $a$ , and legacy (or human-driven) vehicles (LVs), denoted  $l$ . We assume that each link of the network consists of a set of lanes which are either restricted to AVs (AV-lanes) or available to both AVs and LVs (LV-lanes). We use  $\mathcal{A}_a$  and  $\mathcal{A}_l$  to denote AV-lanes and LV-lanes, respectively, and we assume that vehicle movement between different classes of lanes are forbidden. Although AVs can use LV-lanes, we do not model any type of interaction at the traffic-flow level among AVs and LVs on LV-lanes: if an AV uses an LV-lane we assume that it behaves as an LV. We assume that each class of lanes is served by a color-coded traffic phase. Specifically, we assume that LV-lanes are served exclusively by traditional green signal phases. In turn, we assume that AV-lanes are served exclusively by signal-free blue phases. Blue phases differ from green phases in that they directly control vehicles' trajectory within the intersection. The proposed hybrid network control policy presented hereafter chooses which phase (green or blue) should be activated at each intersection of the network and each time period over a discretized time horizon.

Let  $x_i(t) \in \mathfrak{R}_+$  be the number of vehicles on link  $i \in \mathcal{A}$  seeking to enter the intersection at time  $t$ . Although we discretize vehicles in our numerical experiments, integer queue lengths are not necessary for the analytical results presented hereafter. Let  $\mathbf{x}(t)$  be the array of all queue lengths at time  $t$ .  $\mathbf{x}(t)$  is the state of the network, and the state space is  $\mathcal{X} = \{x_i(t) \geq 0 : i \in \mathcal{A}\}$ . We consider discretized time and we assume fixed phase time of length  $\Delta t$ . Further, we assume that the state of the network  $\mathbf{x}(t)$  is known at each time period  $t = 0, \Delta t, 2\Delta t, \dots$ . The goal is to design a throughput-optimal network traffic control policy that optimally selects a traffic signal control at each time period  $[t, t + \Delta t[$ .

The proposed stochastic network traffic control formulation is based on the concept of vehicle *movements* which are formally defined below.

**Definition 4.1.** *A movement  $(i, j) \in \mathcal{A}^2$  is a vehicle trajectory from lane  $i$  to lane  $j$  across a common intersection  $n \in \mathcal{N}$  in the network. We denote  $\mathcal{M}$  the set of all movements in the network.*

AVs communicate their position with IMs to make use of the AIM protocol, hence we assume that movement-specific queues are known for AVs. In contrast, LV-queues can be detected through loop detectors and flow sensors currently in use for traffic signals but their destination is assumed unknown. Specifically, let  $\mathcal{A}_a \subset \mathcal{A}$  be the set of AV-restricted lanes. For these lanes, we assume known movements queues, i.e. if  $j \in \mathcal{A}_a$ , then  $x_i(t) = \sum_{j \in \mathcal{A}_a} x_{ij}(t)$

and  $x_{ij}(t)$  is known. In contrast, for other lanes  $i \in \mathcal{A}_l = \mathcal{A} \setminus \mathcal{A}_a$ , only  $x_i(t)$  is known since route choice for LVs is assumed unknown.

For each lane  $i \in \mathcal{A}$  we assume that lane capacity  $C_i$  is known and determined assuming a triangular fundamental diagram relationship, that is  $C_i = \frac{\bar{U}_i w K}{\bar{U}_i + w}$ , where  $\bar{U}_i$  is the free-flow speed on lane  $i$ ,  $K$  is the jam density and  $w$  is the congestion wave speed. We also assume that the maximum, unconditional movement service rate  $\bar{s}_{ij}$  is known for each movement and determined based on lost time, specifically: unconditional movement service rates for all movements  $(i, j)$  are calculated as  $\bar{s}_{ij} = \min\{C_i, C_j\} \frac{\Delta t - L}{\Delta t}$ .

Let  $P_{ij}(t)$  be a random variable denoting the turning proportion from lane  $i$  to  $j$  at  $t$  with known mean  $p_{ij}$ . Let  $D_i(t)$  be a random variable denoting the external incoming traffic onto lane  $i$  at time  $t$  with known mean  $d_i$ . We denote  $\mathbf{p}$ ,  $\mathbf{d}$  and  $\bar{\mathbf{s}}$  the vectors of mean turning proportions, mean demands and unconditional movement service rates, respectively. These upper bounds on movement service rates represent the maximum number of vehicles that can be moved from lane  $i$  to lane  $j$  during time period  $t$  when no conflicting movement with  $(i, j)$  is activated. From these rates, we can determine maximum, unconditional lane service rates  $\bar{s}_i = \sum_{j \in \mathcal{A}: (i,j) \in \mathcal{M}} \bar{s}_{ij}$ . For convenience if  $i$  and  $j$  do not correspond to a possible movement in the network, e.g. they belong to different intersections or they are both entry or exit lanes of the same intersection, we assume that  $p_{ij} = 0$ . Hence, we can define the sets of AV-movements  $\mathcal{M}_a \equiv \{(i, j) \in \mathcal{A}_a^2 : p_{ij} \neq 0\}$  and LV-movements  $\mathcal{M}_l \equiv \{(i, j) \in \mathcal{A}_l^2 : p_{ij} \neq 0\}$ .

Traffic at network intersections is coordinated by phases which are determined by the selection of the *activation matrix*. In addition, we also introduce the concept of *service matrix* which is used in the proposed traffic control formulations.

**Definition 4.2.** An activation matrix  $\beta(t)$  is a  $|\mathcal{A}| \times |\mathcal{A}|$ -matrix wherein all entries take value 0 (inactive) or 1 (active), i.e.  $\beta_{ij}(t) \in \{0, 1\}$  at time  $t$ .

**Definition 4.3.** A service matrix  $\alpha(t)$  is a  $|\mathcal{A}| \times |\mathcal{A}|$ -matrix wherein all entries take a value between 0 (not serviced) and 1 (fully serviced), i.e.  $\alpha_{ij}(t) \in [0, 1]$  at time  $t$ .

The entries of an activation matrix characterize the *activeness* of the corresponding phase:  $\beta_{ij}(t) = 1$  means that movement  $(i, j)$  is active during phase  $t$  whereas  $\beta_{ij}(t) = 0$  means that movement  $(i, j)$  is inactive. The entries of the service matrix characterize the *service level* of active movements during phase  $t$ :  $\alpha_{ij}(t) = 1$  corresponds to a maximal service level for of movement  $(i, j)$ , whereas  $\alpha_{ij}(t) = 0$  means that movement  $(i, j)$  cannot serve any vehicles during time period  $t$ . Fractional service level values model situations where conflicting movements, i.e. posing a safety risk, simultaneously have non-zero activation values. The activation and service matrices are linked through the movement-based constraints  $\alpha_{ij}(t) \leq \beta_{ij}(t)$ . These linking constraints ensure that an intersection can only service vehicles on movement  $(i, j)$  if this movement is active  $\beta_{ij}(t) = 1$ . Further, for priority movements, we set  $\alpha_{ij}(t) = \beta_{ij}(t)$  which implies that an activated priority movements has full service level. In the proposed traffic control policy, the selection of the activation matrix requires the solution of two mathematical optimization problems. Green phases serve LVs using a MILP that captures FIFO blocking effects on lanes. Blue phases serve AVs using an a modified version of the MILP presented in Section 4.3 wherein the objective function is adapted for network traffic control. These MILPs are referred to as GREEN and BLUE. Further details on these MILPs can be found in [Rey and Levin \[2019\]](#).

Let  $Y_i(t)$  be a random variable with mean  $y_i(t)$  denoting the number of vehicles serviced in lane  $i$  at time  $t$ . The vector  $\mathbf{Y}(t)$  is endogenous to the service matrix  $\alpha(t)$  selected by the

control policy. Note that the mean  $y_i(t)$  of  $Y_i(t)$  is unknown and  $y_i(t)$  is modeled as a control variable in the proposed traffic phase optimization formulation. The proposed stochastic network traffic control model is summarized by the lane-queue evolution equation (4.25):

$$x_j(t + \Delta t) = x_j(t) - Y_j(t) + \sum_{i \in \mathcal{A}} P_{ij}(t) Y_i(t) + D_j(t), \quad \forall j \in \mathcal{A}. \quad (4.25)$$

Note, if  $j \notin \mathcal{A}_r$  then  $D_j(t) = 0$ . Conversely, if  $j \in \mathcal{A}_r$  then  $j$  has no predecessor links, thus  $\sum_{i \in \mathcal{A}} P_{ij}(t) = 0$ . Although AV-lanes and other lanes have identical queue evolution equations, the information available is more accurate for AV-lanes, this used in the calculation of network control policies.

In related works, the lane service rate vector  $\mathbf{y}(t)$  is commonly calculated as the minimum between the supply (lane or movement service capacity) and the demand (lane or movement queue length) [Varaiya, 2013, Le et al., 2015]. However, this modeling approach does not always capture the interdependency between the activation of possibly conflicting movements with lane or movement service capacity and queue length. Precisely, previous efforts have assumed that a set of activation matrices is provided for each intersection and that lane or movement service capacities can be pre-processed accordingly. This overlooks the impact of queue length on intersection capacity. In contrast our proposed integrated approach aims to accurately estimate the expected number of serviced vehicles at each time period by leveraging the available lane-queue length information (note that this information is also assumed available in the aforementioned papers).

#### 4.4.2 Stability analysis

We present a network traffic control policy for intersection control combining green (LV-lane restricted) and blue phases (AV-lane restricted) and prove that it maximizes throughput. The proposed network traffic control policy works by repeatedly solving GREEN and BLUE at each time period  $t$  based on the network state  $\mathbf{x}(t)$  and combining local (intersection-level), optimal activation matrices into a network-wide activation matrix  $\boldsymbol{\alpha}(t)$ .

Let  $\mathbb{M}$  be the set of service matrices. A *policy* is a function  $\pi : \mathcal{X} \rightarrow \mathbb{M}$  that chooses a service matrix  $\boldsymbol{\alpha}(t) \in \mathbb{M}$  for every state  $\mathbf{x}(t)$ . We use the concept of *strong stability* to characterize the proposed stochastic queuing process [Leonardi et al., 2001, Wongpiromsarn et al., 2012, Zaidi et al., 2016]:

**Definition 4.4.** *Let  $\bar{t}$  be a time period index. A stochastic queue evolution process is strongly stable under policy  $\pi$  if and only if there exists  $K < \infty$  such that the network state  $\mathbf{x}(t)$  verifies:*

$$\limsup_{\bar{t} \rightarrow \infty} \mathbb{E} \left[ \frac{1}{\bar{t}} \sum_{t=1}^{\bar{t}} |\mathbf{x}(t)| \right] < K. \quad (4.26)$$

For brevity, we hereby referred to *strong stability* as *stability*.

To show that the proposed policy is stabilizing and define the stability region of the system, we introduce artificial decision variables in GREEN and BLUE MILPs. Let  $\mathcal{A}_{ro} = \mathcal{A}_r \cup \mathcal{A}_o$  and let  $\gamma_i(t) \in [0, 1]$  be a decision variable representing the lane service supply on

lane  $i \in \mathcal{A}_{ro}$ . Recall that  $\bar{s}_i = \sum_{j \in \mathcal{A}: (i,j) \in \mathcal{M}} \bar{s}_{ij}$  is the unconditional lane supply. Consider the lane supply constraints of the form:

$$y_i(t) \leq \gamma_i(t) \bar{s}_i, \quad \forall i \in \mathcal{A}_{ro}, \quad (4.27a)$$

$$\gamma_i(t) \leq \beta_{ij}(t), \quad \forall (i,j) \in \mathcal{M}. \quad (4.27b)$$

Constraints (4.27) require that  $\gamma_i(t)$  be at least  $y_i(t)/\bar{s}_i$  and impose that  $\gamma_i(t)$  be null if any movement emanating from lane  $i$  is inactive. Observe that since  $\beta_{ij}(t) = 0 \Rightarrow y_{ij}(t) = 0$ , FIFO conditions on lanes imply that if there exists a lane  $j$  such that  $p_{ij} > 0$  and  $\beta_{ij}(t) = 0$ , then  $y_i(t) = 0$ , which is consistent with Constraints (4.27). Observe that Constraints (4.27a) are equivalent to the class-specific lane supply constraints:

$$x_i(t)p_i(t) \leq \gamma_i(t)\bar{s}_i, \quad \forall i \in \mathcal{A}_l, \quad (4.28a)$$

$$\sum_{v \in \mathcal{V}_i^n(t)} z_v \leq \gamma_i(t)\bar{s}_i, \quad \forall i \in \mathcal{A}_a. \quad (4.28b)$$

Lane supply constraints (4.27) can be incorporated in GREEN by adding (4.28a) and (4.27b) to GREEN. Similarly, (4.27) can be incorporated in BLUE by adding (4.28b) and (4.27b) to BLUE. Observe that variables  $\gamma_i(t) \in [0, 1]$  do not influence the optimal solutions of GREEN or BLUE MILPs since  $y_i(t) \leq \bar{s}_i$  for all  $i \in \mathcal{A}_{ro}$ . We henceforth refer to the MILPs with lane supply constraints (4.28) and variables  $\gamma_i(t)$ ,  $\forall i \in \mathcal{A}_{ro}$  as the extended GREEN and BLUE MILPs.

Let  $\Omega_{\text{GREEN}}$  and  $\Omega_{\text{BLUE}}$  denote the feasible regions of the extended GREEN and BLUE MILPs, respectively. Further, let  $\Omega = \Omega_{\text{GREEN}} \cup \Omega_{\text{BLUE}}$  be the feasible region verifying the constraint sets of both GREEN and BLUE, and let  $\text{CONV}(\Omega)$  be the convex hull of  $\Omega$ . We say that an infinite sequence of controls  $\{\gamma(t)\}$  for  $t = 1, 2, \dots$  is admissible if  $\{\gamma(t) \in \Omega\}$  for all time periods  $t$ .

**Definition 4.5.** For any lane  $i \in \mathcal{A}_{ro}$ , let  $\hat{\gamma}_i = \lim_{\bar{t} \rightarrow \infty} \frac{1}{\bar{t}} \sum_{t=1}^{\bar{t}} \gamma_i(t)$  be the average lane supply of  $i$  for sequence  $\{\gamma(t) \in \Omega\}$ . An admissible control sequence  $\{\gamma(t) \in \Omega\}$  accommodates flow  $\mathbf{f}$  if

$$f_i < \hat{\gamma}_i \bar{s}_i, \quad \forall i \in \mathcal{A}_{ro}. \quad (4.29)$$

Definition 4.5 states that a flow  $\mathbf{f}$  can be accommodated if, on average,  $\hat{\gamma}_i \bar{s}_i$  vehicles can be served on lane  $i \in \mathcal{A}_{ro}$ . From Proposition 2 of Varaiya [2013],  $\hat{\gamma} \in \text{CONV}(\Omega)$  if and only if  $\{\gamma(t) \in \Omega\}$  is an admissible control sequence. We define the stability region of the system accordingly.

**Definition 4.6.** Let  $\mathcal{D}$  be the set of demand rate vectors such that there exists an admissible control sequence  $\{\gamma(t) \in \Omega\}$  with  $\hat{\gamma} \in \text{CONV}(\Omega)$  defined as:

$$\mathcal{D} \equiv \left\{ \mathbf{d} \in \mathbb{R}_+^{|\mathcal{A}_r|} : (f_i \leq \hat{\gamma}_i \bar{s}_i, i \in \mathcal{A}_{ro}, \{\gamma(t) \in \Omega\}) \right\}. \quad (4.30)$$

Let  $\mathcal{R}$  be the interior of  $\mathcal{D}$  i.e.  $\mathcal{R} = \{\mathbf{d} \in \mathcal{D}^\circ\}$ .  $\mathcal{R}$  is the stability region of the system.

The proposed hybrid max-pressure policy selects at each time period  $t$  the service matrix  $\alpha(t)$  which maximizes the network-wide pressure among all possible green and blue phases.

---

**Definition 4.7.** *The network traffic control policy  $\pi^*(\mathbf{x}(t))$ , defined as:*

$$\pi^*(\mathbf{x}(t)) = \arg \max \left\{ \sum_{n \in \mathcal{N}} (Z_G^n(\mathbf{x}(t)) \vee Z_B^n(\mathbf{x}(t))) : \boldsymbol{\alpha}(t) \in \mathbb{M} \right\}. \quad (4.31)$$

*is hereby referred to as the hybrid max-pressure network control policy for legacy and autonomous vehicles.*

At each intersection  $n \in \mathcal{N}$ , the hybrid max-pressure policy selects the phase (green or blue) and service matrix  $(\boldsymbol{\alpha}(t))$  maximizing the local pressure. The service matrix is either  $\boldsymbol{\alpha}_G^n(t)$  if the green phase is activated or  $\boldsymbol{\alpha}_B^n(t)$  if the blue phase is activated. The implementation of the hybrid max-pressure policy requires the resolution of the two MILPs GREEN and BLUE at each intersection of the network. We next show that policy  $\pi^*$  is stabilizing for any demand rate in the stability region of the system  $\mathcal{R}$ .

**Theorem 4.1.** *The stochastic queue evolution process (4.25) is stable under the hybrid max-pressure policy  $\pi^*$  (4.31) for any demand rates vector  $\mathbf{d} \in \mathcal{R}$ .*

A natural extension of Theorem 4.2 is that the pure network traffic control policies wherein only GREEN or BLUE is used to coordinate traffic are also stable.

**Corollary 4.1.** *The pure pressure-based network traffic control policy consisting of policy  $\pi^*$  (4.31) with only green (respectively, blue) phases coordinated by GREEN (respectively, BLUE) are stable for any demand rates in the stability region  $\mathcal{R}$ .*

Theorem 4.2 proves that the proposed hybrid network control policy  $\pi^*$  (4.31) stabilizes any demand vector in the stability region of the system  $\mathcal{R}$  (4.30). According to Tassiulas and Ephremides [1992], and as discussed in Wongpiromsarn et al. [2012] and Varaiya [2013] for the case of signalized traffic control, this is equivalent to throughput optimality since it shows that the stability region of policy (4.31) is a superset of the stability region of any policy.

Corollary 4.1 establishes that pure policies based on GREEN or BLUE traffic control models also maximize throughput. We note that stability of the pure green network traffic control case is an extension of the work of Varaiya [2013]. Varaiya [2013] proposed a network traffic control policy for a single class of vehicles and assumed that each movement had a dedicated queue. In addition, it was assumed that movement capacities are exogenous to the traffic signal control policy. We have both relaxed this framework by only requiring knowledge of lane-queues, and extended the formulation to two classes of lanes. Further, we introduced pressure-based formulations for both green and blue phases which account for FIFO blocking effects on upstream lanes, and account for the loss of capacity due to conflicting movements being simultaneously activated.

We are now ready to present our decentralized network traffic control algorithm used to implement the proposed hybrid max-pressure network control policy. The pseudo-code of the proposed policy is summarized in Algorithm 6. At each time period  $t$ , we calculate the optimal GREEN and BLUE phases at each intersection of the network  $n \in \mathcal{N}$  based on the current state of the network  $\mathbf{x}(t)$ . The phase with the highest local pressure is selected for each intersection.

**Algorithm 6:** Hybrid max-pressure network control policy**Input:**  $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ ,  $\mathbf{d}$ ,  $\mathbf{p}$ ,  $\bar{s}$ ,  $t$ ,  $\mathbf{x}(t)$ **Output:**  $\alpha(t)$ 


---

```

1 for  $n \in \mathcal{N}$  do
2    $Z_G^n(\mathbf{x}(t)) \leftarrow$  Solve GREEN
3    $Z_B^n(\mathbf{x}(t)) \leftarrow$  Solve BLUE
4    $\alpha^n(t) \leftarrow \arg \max_{\alpha_G^n(t), \alpha_B^n(t)} \{Z_G^n(\mathbf{x}(t)), Z_B^n(\mathbf{x}(t))\}$ 
5  $\alpha(t) \leftarrow [\alpha^n(t)]_{n \in \mathcal{N}}$ 

```

---

**4.4.3 Numerical experiments**

We implement the proposed hybrid network control policy on artificial datasets to test computational performance and analyze the algorithm's behavior. We use a synthesized grid network of size  $5 \times 5$ , wherein each of the 25 nodes corresponds to a controlled intersection and each edge represents a bidirectional link between adjacent nodes. All intersections have the same topology as that depicted in Figure 4.2a, i.e. each node has four incoming and four outgoing links, each of which has one LV-lane and one AV-lane. Each incoming lane allows three movements: through, left turn and right turn.

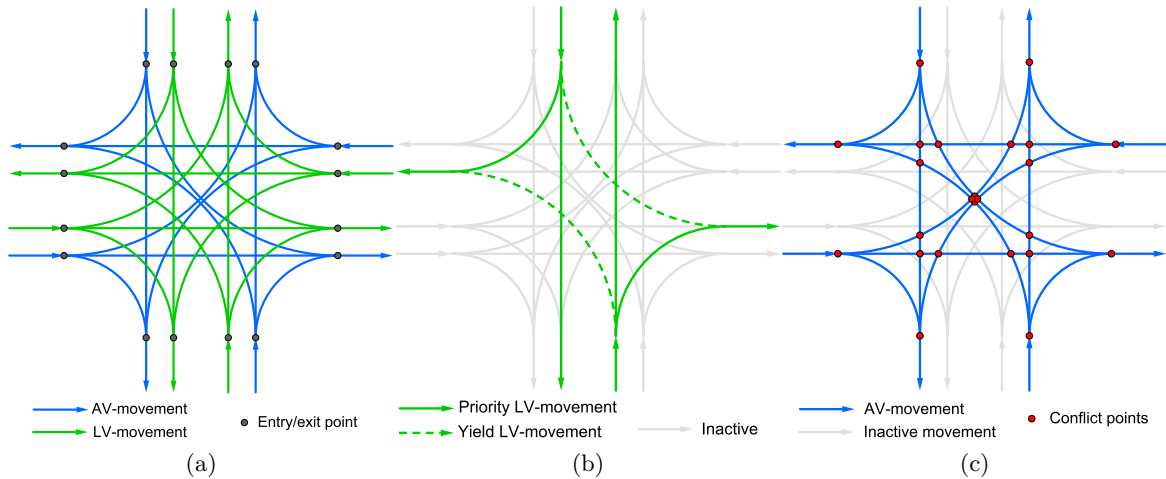


Figure 4.2: Figure 4.2a shows all possible LV- and AV-movements on a typical intersection. Figure 4.2b depicts a possible green phase involving priority and yield LV-movements and Figure 4.2c illustrates the conflict-point formulation during blue phase.

We assume that vehicles' routes in the network are fixed. In each instance generated, we randomly and uniformly assign an origin and a destination to each vehicle, a route among these nodes and a departure time within the considered time horizon. Origins and destinations are chosen among nodes at the edge of the grid. The level of travel demand is determined by the *departure rate* of vehicles into the network and the impact of travel demand onto network performance is assessed through a sensitivity analysis.

The time period is set to  $\Delta t = 10$  s. We assume that green phases have a *lost time* of 2 s to account for vehicle start-up delays and signal clearance intervals and we conduct a sensitivity analysis on this input parameter. In turn, blue phases are assumed to have zero lost time.

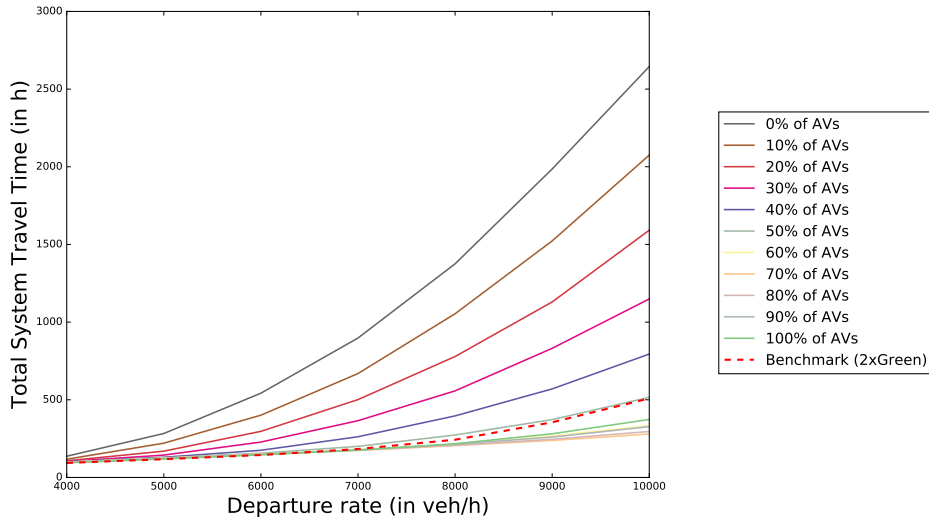


Figure 4.3: Total system travel time for a varying proportion of AVs and vehicle departure rate. The results illustrate the trend of the mean total system travel time over 40 simulations on a  $5 \times 5$  grid network with each link having one AV and one LV lane. The 2×Green experiment (benchmark) corresponds to the scenario where each link has the capacity of two LV-lanes.

We use point-queues for all links in the network and we assume that vehicles take three time periods to travel from an intersection to the next intersection on their route. Vehicles travel through 3 links, each with a 10 s free flow time, between each intersection. Hence, in this configuration, it takes 30 s for a vehicle to travel between two adjacent intersections at free flow. We set the time horizon to 30 minutes and we execute Algorithm 6 periodically until all vehicles have exited the network. Vehicles’ speed limit through intersections is assumed to be uniform and equal to 44 ft/s and the wave propagation speed is taken as 11 ft/s. We assume that all vehicles have a length of 17.6 ft and that lanes have a width of 12 ft. We use the triangular fundamental diagram to determine lane capacity which results in 1,440 veh/h or 5 vehicles per time period.

We explore the sensitivity of the proposed policy with regards to the proportion of AVs by varying the proportion of AVs from 0% to 100% in increments of 10%. In all our numerical experiments we assume that AVs always choose AV-lanes. To benchmark the performance of the proposed hybrid network control policy (summarized in Algorithm 6), we also simulate network traffic under a traditional traffic signal configuration wherein AV-lanes and blue phases are nonexistent. Under this configuration all AV-lanes are treated as LV-lanes, and we model this by using single-lane links with twice the lane capacity of the LV-lanes in the network configuration with both LV- and AV-lanes. Thus, in this benchmark there are no AV-lanes and since we assume that AVs behave as LVs and LV-lanes, all vehicles are treated as LVs. This benchmark is hereby referred to as 2×Green. Each experiment is simulated 40 times and average performance is reported.

The simulation framework is implemented in Java on a Windows machine with 8 Gb of RAM and a CPU at 3 GHz. All MILPs are solved with CPLEX 12.8 (Java API) with a time limit of 60 s and default options.



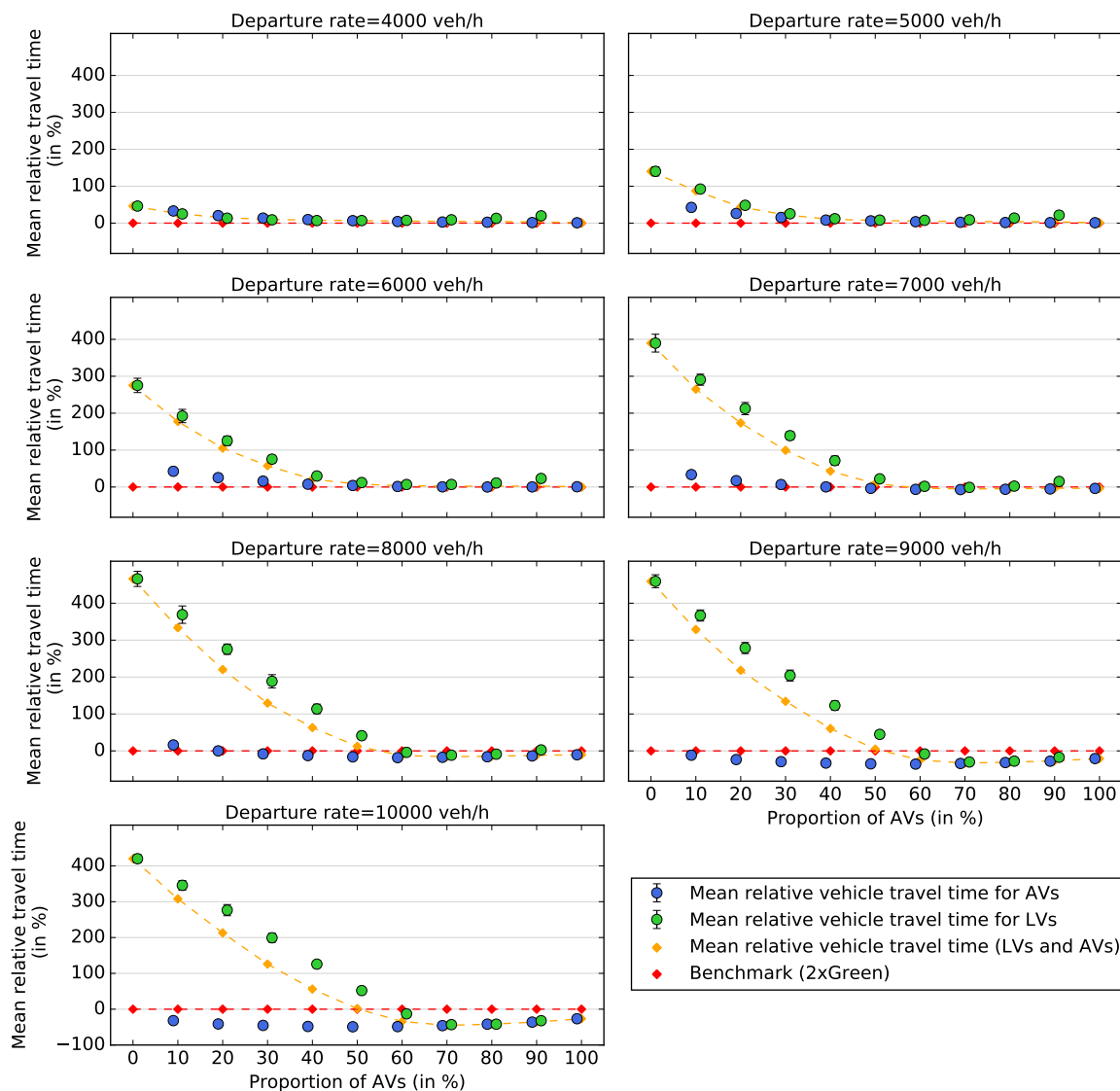


Figure 4.4: Average vehicle travel time based on AV proportion and vehicle departure rate. The results depict the mean and standard deviation over 40 simulations on a  $5 \times 5$  grid network with each link having one AV and one LV lane. The  $2 \times \text{Green}$  experiment (benchmark) corresponds to the scenario where each link has the capacity of two LV-lanes.

---

The evolution of the total system travel time (TSTT) for a varying departure rate is depicted in Figure 4.3. For this experiment, the green phase lost time is set to 2 s and we vary the departure rate from 4,000 veh/h (lowest demand) to 10,000 (highest demand). As expected, we observe that TSTT increases super-linearly with the departure rate, i.e. travel demand. We find that the market penetration of AVs has a significant impact on TSTT. If less than 50% of AVs are present in the system, we find that the use of dedicated AV lanes and the blue phase is not beneficial for the network in terms of TSTT, even at high demands, as noted by the performance of the benchmark which outperforms the hybrid network control policy for these levels of AV market penetration. Recall that the benchmark represents the TSTT when the pure, green network traffic control policy is used with the equivalent of two LV-lanes. At high demands (i.e. 8,000 veh/h and beyond), we find that a market penetration of more than 50% of AVs outperforms the benchmark. At the highest departure rate tested (i.e. 10,000 veh/h), for a market penetration of 70%, the average TSTT is reduced by 1/2 compared to the benchmark. For AV market penetration rates of 80% and higher, we find that the reduction of the average TSTT is lesser than that achieved at 70% of AV market penetration. This can be explained by the fact that network capacity is not fully used at 100% of AVs compared to 70% since in the former configuration LV-lanes remain empty throughout the experiment.

To further investigate the behavior of the proposed hybrid network control policy, we examine average, vehicle-class travel times relative to the benchmark configuration. Figure 4.4 shows the average vehicle travel time for AVs (blue series), LVs (green series) and overall (orange series) in the network based on AV proportion and departure rate. The benchmark is shown as a dashed flat line in red. Three main trends can be identified: first, we find that increasing the departure rate mainly impacts the mean relative travel time of LVs with regards to the benchmark. Second, the proportion of AVs minimizing the mean vehicle travel time in the network is relatively robust to the travel demand and is found to be around 70% of AVs. This suggests that at lower or higher market penetration rates, network capacity is not saturated and AV- or LV-lanes are under-utilized, respectively. This insight can help in identifying the optimal AV market penetration rate to deploy AV lanes and blue phases. Third, for high levels of travel demand, a sufficiently high proportion of AVs improves on the benchmark, i.e. the average vehicle travel time (over both LVs and AVs) obtained using the hybrid network control policy is lower than the average vehicle travel time obtained with the pure, green network control policy.

For a departure rate of 4,000 veh/h, LVs and AVs' average travel time remain similar and the hybrid network control policy performs similarly to the benchmark. Increasing the departure rate from 5,000 to 7,000 veh/h, we observe congestion effects impacting LVs' average travel time, while AVs' average travel time remain only slightly penalized. Further increasing the departure rate to 8,000 veh/h and beyond yields a new pattern: for a proportion of AVs greater or equal to 60%, the aggregate average vehicle travel time improves on the benchmark although LVs' average travel time remain more penalized than that of AVs. At a departure rate of 10,000 veh/h, we find that both LVs and AVs' average travel time are lower than the average travel time in the benchmark configuration and considerable travel time reduction are achieved from 60% of AVs and beyond. Specifically, at this departure rate and with a market penetration of AVs of 70%, we find that the average vehicle travel time is reduced by approximately 1/2 compared to the benchmark. We also observe that the mean relative vehicle (LVs and AVs) travel time exhibits a convex-shaped profile with a minima at 70% of AV market penetration.

## 4.5 Online mechanism design for traffic intersection auctions

We now turn our attention to market-based AIM mechanisms and present two online payment mechanisms for traffic intersection auctions. For this study, we restrict ourselves to a simplified intersection model and focus on the game-theoretical aspects of dynamic user pricing.

### 4.5.1 Dynamic traffic intersection auction framework

We consider a traffic intersection with a finite number of access lanes. We consider discrete time and we assume that at each time period, users arrive on access lanes with a known probability. Upon arriving at the front of their lane, users are requested to declare their delay cost which will be used as their bid in a combinatorial auction. We assume that users seek to minimize a generalized cost function which is a linear combination of their expected waiting time and their payment to the intersection manager. We assume that users are serviced sequentially by a single server, i.e. the traffic intersection and that service is nonpreemptive, i.e. cannot be interrupted once started. This does not prevent several users to simultaneously traverse the intersection. Instead, the proposed traffic intersection auction only aims to determine the sequence in which users are to be serviced and the payments they should be charged based on their declared preferences.

The goal of the intersection manager is to maximize social welfare which is defined as the total generalized user cost. To achieve optimal social welfare, we seek to determine incentive-compatible user payments to ensure the truthful declaration of their delay costs. To ensure truthful user behavior, traffic intersection lanes are serviced in order of decreasing declared user delay costs. The expected waiting time of users is a function of the declared delay costs and the state of the system. In turn, the payment of users is to be determined by the intersection manager so as to maximize social welfare and ensure truthfulness.

To present the proposed online mechanism for determining incentive-compatible payments, we consider the case of a user  $i$  arriving at the front of its lane and declaring a delay cost of  $v_i$ . Let  $Q$  be the number of access lanes of the intersection. We denote  $\bar{q}_i$  (resp.  $\underline{q}_i$ ) the number of lanes which are occupied by users with greater (resp. lower) delay cost than  $i$ . Further, we denote  $q_\emptyset$  the number of empty lanes. Summing  $\bar{q}_i$ ,  $\underline{q}_i$  and  $q_\emptyset$  corresponds to the  $Q - 1$  lanes of the intersection which are not occupied by user  $i$ . Hence, the following equation holds for any user  $i$  at any time period:

$$Q = \underline{q}_i + \bar{q}_i + q_\emptyset + 1. \quad (4.32)$$

Accordingly, we can define the pricing queue of the proposed dynamic traffic intersection auction.

**Definition 4.8** (Pricing queue). *In a traffic intersection with  $Q$  access lanes, the pricing queue is a dynamic set of at most  $Q$  users which represents the users at the front of their lane at a given time period.*

The pricing queue consists of the set of users which have no users in front of them in their access lane to the intersection. In the proposed traffic intersection auction, only users in the pricing queue are participants. This implies that users who are behind other users

---

are not eligible to participate until they reach the front of their lane. This auction design aims to obviate blocking effects induced by non-overtaking conditions in access lanes of the intersection. Under such typical, non-overtaking conditions users waiting behind other users in their lane cannot bypass users in front of them, thus preventing them to be serviced before reaching the front of their lane. In the proposed approach, only users in the pricing queue participate in the auction. While this dynamic auction framework restricts the number of participants to at most  $Q$  users, every user queueing in the intersection will eventually reach the front of its lane and participate in a single instance of the auction. This is in contrast to approaches in the literature where auction “losers” may participate in several auctions. We refer readers to [Carlino et al. \[2013\]](#) who presented pricing mechanisms for traffic intersection auctions in which blocked users are allowed to “vote” for their lane.

The objective of the traffic intersection auction is to determine incentive-compatible user payments upon users joining the pricing queue. Let  $W_i(v_i)$  be the expected waiting time of user  $i$  declaring a delay cost of  $v_i$ . Let  $v_i^*$  be the true delay cost of user  $i$ , the expected waiting time cost of user  $i$  declaring a delay cost of  $v_i$  is  $v_i^*W_i(v_i)$ . Let  $P_i(v_i)$  be the payment of user  $i$  declaring  $v_i$ , which is to be determined by the intersection manager. We denote  $C_i(v_i)$  the generalized cost for user  $i$  and define the user objective function as:

**Definition 4.9** (User objective function). *The objective function of user  $i$  is:*

$$\min_{v_i} C_i(v_i) = \min_{v_i} (v_i^*W_i(v_i) + P_i(v_i)). \quad (4.33)$$

Following the approach of [Dolan \[1978\]](#), we will show that setting the payment  $P_i(v_i)$  equal to the expected marginal delay cost user  $i$  imposes to other users is incentive-compatible, i.e. the user objective is minimal for  $v_i = v_i^*$ .

We assume that the intersection manager has knowledge of the probability distribution of users’ valuation (delay costs), and of the arrival probability of users on access lanes of the intersection. These assumptions are reasonable since one can assume that the intersection manager can observe user behavior (arrival rate and valuations) and learn these probabilities over time. Let  $F(v_i)$  be the cumulative distribution function representing the probability that a user declares a delay cost  $v \leq v_i$ . Further, let  $\underline{v}$  and  $\bar{v}$  be lower and upper bounds on users’ delay costs. For conciseness, we hereby use the term *lower-bidding user* (resp. *higher-bidding user*) to refer to a user with a declared delay cost lower (resp. higher) than that of user  $i$ .

An overview of the proposed dynamic auction process is summarized in Figure 1. Users are assumed to arrive dynamically and submit their bids (delay cost) to the intersection manager. The intersection manager then determines a user payment  $P_i(v_i)$  based on the user’s bid  $v_i$  using either the proposed queue- or lane-based model. The user is then charged  $P_i(v_i)$  and moves through the intersection.

#### 4.5.2 Online incentive-compatible mechanisms

We propose two online mechanisms to determine the expected waiting time and payment of users. The proposed mechanisms differ in the way the expected waiting time of users is determined. We first consider a queue-based model which only tracks the number of users bidding higher or lower than the reference user (Section 4.5.2). We next extend this model to a lane-based approach which tracks users’ arrival lane (Section 4.5.2). A generic

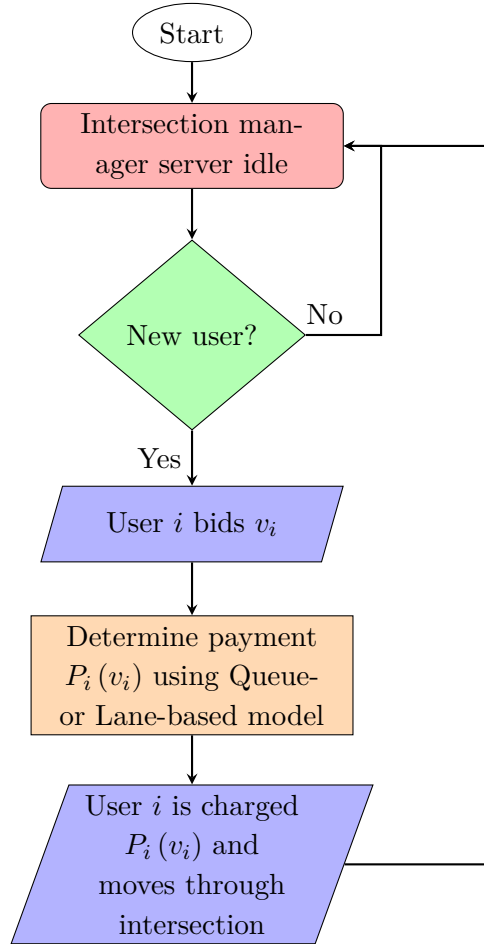


Figure 4.5: Online traffic intersection auction process

payment mechanism is proposed to determine incentive-compatible payments for both queue- and lane-based models which result in two online mechanisms (Section 4.5.3). The queue-based mechanism assumes that the arrival probability of users is uniform across lanes of the intersection and provides a relatively low-dimensional mathematical framework. This uniformness assumption is relaxed in the lane-based mechanism which can accommodate non-uniform lane arrival probabilities at the expense of an extended state space.

### Queue-based model

We use a MC to determine the expected waiting time of user  $i$  under different system states. The state of the queuing system w.r.t. user  $i$  is represented by the number of lanes occupied by lower bidding vehicles  $\underline{q}_i$  the number of empty lanes  $q_\emptyset$ . We denote  $\mathbf{q}_i$  the vector  $\mathbf{q}_i = (\underline{q}_i, q_\emptyset)$  characterizing the state of the system w.r.t. user  $i$ . User  $i$  is serviced when all lanes are either occupied by a lower-bidding user or empty. Observe that once a lane is occupied by a lower-bidding user w.r.t. user  $i$ , it remains in this state until  $i$  is serviced. Hence, from the perspective of user  $i$ , the state space of the MC can be characterized by the inequality  $\underline{q}_i + q_\emptyset \leq Q - 1$ . This leads to the following characterization of the state space.

**Definition 4.10** (State space in the queue-based model). *In the queue-based model, the state*

space of the MC for user  $i$  declaring a delay cost of  $v_i$  in a pricing queue of size  $Q$  is:

$$\mathcal{S}_i^q = \left\{ \mathbf{q}_i = (\underline{q}_i, q_\emptyset) \in \mathbb{N}^2 : \underline{q}_i + q_\emptyset \leq Q - 1 \right\}. \quad (4.34)$$

For each state  $\mathbf{q}_i \in \mathcal{S}_i^q$ , we are interested in determining the cost-to-go from this state through the MC. If  $\underline{q}_i + q_\emptyset = Q - 1$ , then (4.32) yields  $\bar{q}_i = 0$  which is equivalent to require that the pricing queue does not contain any user with declared delay costs higher than that of  $i$ . Hence, if  $\underline{q}_i + q_\emptyset = Q - 1$ , then the MC converges and user  $i$  is serviced. Hence, to determine the state transition probabilities from state  $\mathbf{q}_i$  to state  $\mathbf{q}'_i$ , we assume that  $\underline{q}_i + q_\emptyset < Q - 1$ .

To calculate the expected waiting time of user  $i$  in the queue-based model, we enumerate system states with varying number of lower-bidding and empty lanes w.r.t. user  $i$ . We wish to define a function  $f_q$  which gives the transition between the state at time step  $t$  to the state at time  $t + 1$ . The transition function of the MC,  $f_q$ , depends on the realization of random variables. For this problem, there are two random variables affecting the transition: user delay costs, a continuous random variable; and whether a new vehicle arrives on each lane, a boolean value. Hence, the uncertain data of the problem can be represented by a vector  $\xi \in \mathcal{R}^Q \times 2^Q$  where  $\mathcal{R}^Q$  represents user delays costs and  $2^Q$  represents vehicle arrivals. Since the state space of the queue-based model is  $\mathcal{S}_i^q$ , the transition function maps the current state  $\mathbf{q}_i \in \mathcal{S}_i^q$  and the uncertain data  $\xi \in \mathcal{R}^Q \times 2^Q$  to the next state, i.e.  $f_q : \mathcal{S}_i^q \times \mathcal{R}^Q \times 2^Q \rightarrow \mathcal{S}_i^q$ . We denote  $g_q(\mathbf{q}_i)$  the one-step-cost of the MC representing the service time of the system in state  $\mathbf{q}_i$  and we assume that the one-step-cost is deterministic. This assumption is plausible since users' behavior is assumed to be deterministic, i.e. users cannot change their declared delay cost over time.

Let  $W_q(v_i, \mathbf{q}_i)$  be the expected waiting time of user  $i$  declaring  $v_i$  for state  $\mathbf{q}_i$ , which represents the cost-to-go of the MC representing the state evolution in the queue-based model. Let  $\mathcal{T}_i^q = \{\mathbf{q}_i \in \mathcal{S}_i^q : q_\emptyset + \underline{q}_i = Q - 1\}$  be the set of terminal states in the queue-based mechanism. The expected waiting time of user  $i$  declaring  $v_i$  for state  $q$  can be determined as:

$$W_q(v_i, \mathbf{q}_i) = \begin{cases} 0 & \text{if } \mathbf{q}_i \in \mathcal{T}_i^q, \\ \mathbb{E}[W_q(v_i, f_q(\mathbf{q}_i, \xi))] + g_q(\mathbf{q}_i) & \text{otherwise.} \end{cases} \quad (4.35)$$

Note that the term  $\mathbb{E}[W_q(v_i, f_q(\mathbf{q}_i, \xi))]$  represents the expected value of the expected waiting time  $W_q(v_i, f_q(\mathbf{q}_i, \xi))$  after the transition function  $f_q$ . If  $\mathbf{q}_i$  is not a terminal state, i.e.  $\underline{q}_i + q_\emptyset < Q - 1$ , the expected waiting time of user  $i$  can be calculated using the probability  $\Pr[\mathbf{q}'_i | \mathbf{q}_i]$  of transitioning from state  $\mathbf{q}_i$  to state  $\mathbf{q}'_i$ . The expected waiting time of user  $i$  declaring  $v_i$  is:

$$\mathbb{E}[W_q(v_i, f_q(\mathbf{q}_i, \xi))] = \sum_{\mathbf{q}'_i \in \mathcal{S}_i^q} \Pr[\mathbf{q}'_i | \mathbf{q}_i] W_q(v_i, \mathbf{q}'_i). \quad (4.36)$$

Combining Eqs. (4.35) and (4.36) yields the following system of linear equations:

$$W_q(v_i, \mathbf{q}_i) = 0, \quad \forall \mathbf{q}_i \in \mathcal{T}_i^q, \quad (4.37a)$$

$$W_q(v_i, \mathbf{q}_i) (1 - \Pr[\mathbf{q}_i | \mathbf{q}_i]) - \sum_{\mathbf{q}'_i \in \mathcal{S}_i^q : \mathbf{q}'_i \neq \mathbf{q}_i} W_q(v_i, \mathbf{q}'_i) \Pr[\mathbf{q}'_i | \mathbf{q}_i] = g_q(\mathbf{q}_i), \quad \forall \mathbf{q}_i \in \mathcal{S}_i^q \setminus \mathcal{T}_i^q. \quad (4.37b)$$

Let  $p$  be the probability that a user arrives on a lane of the intersection at the next time period. Recall that  $F(v_i)$  is the cumulative distribution function of users bids. On each lane, three events may occur with the following probabilities: i) a lane can become or remain empty with probability  $(1 - p)$ ; ii) a lower-bidding user can arrive with probability  $pF(v_i)$ ; and iii) a higher-bidding user can arrive with probability  $p(1 - F(v_i))$ . Observe that there are  $\binom{q_0+1}{q'_0}$  ways to choose  $q'_0$  empty lanes among  $q_0 + 1$  lanes. There are  $\underline{q}'_i - \underline{q}_i$  lanes which become occupied by lower-bidding users and there are  $\binom{q_0+1-q'_0}{\underline{q}'_i - \underline{q}_i}$  ways to choose  $\underline{q}'_i - \underline{q}_i$  of the remaining  $q_0 + 1 - q'_0$  non-empty lanes to become occupied by a lower-bidding user. Accordingly, the transition probability from state  $(\underline{q}_i, q_0)$  to state  $(\underline{q}'_i, q'_0)$  is:

$$\Pr[\underline{q}'_i | \underline{q}_i] = \binom{q_0 + 1}{q'_0} (1-p)^{q'_0} \binom{q_0 + 1 - q'_0}{\underline{q}'_i - \underline{q}_i} (pF(v_i))^{\underline{q}'_i - \underline{q}_i} (p(1 - F(v_i)))^{q_0 + 1 - q'_0 - \underline{q}'_i + \underline{q}_i}. \quad (4.38)$$

The expected waiting time defined in Eq. (4.35) is a recursive function which can be calculated by solving a series of systems of linear equations where each equation corresponds to a possible state  $\mathbf{q}_i \in \mathcal{S}_i^q$ . Since all states such that  $\underline{q}_i + q_0 = Q - 1$  are terminal, their corresponding expected waiting time is null and these variables can be eliminated from the system of equations. Hence, we need only to consider the set of pairs of nonnegative integers  $(\underline{q}_i, q_0)$  such that  $\underline{q}_i + q_0 \leq Q - 2$ . Observe that each variable  $\underline{q}_i$  or  $q_0$  can take  $Q - 2 + 1$  values, thus the number of equations in the system is  $\sum_{k=0}^{Q-1} k = \binom{Q}{2}$ .

Further, observe that  $\underline{q}_i$  can never decrease while  $q_0$  can increase or decrease. Hence, for any state  $\mathbf{q}_i = (\underline{q}_i, q_0)$ , we can calculate  $W_q(v_i, \mathbf{q}_i)$  in a dynamic programming fashion starting with the maximum number of lanes, i.e.  $k = Q - 1$ , and iterating downwards until  $k = \underline{q}_i$ . At each iteration  $k$ , we solve the system of  $Q - k$  linear equations with  $Q - k$  variables corresponding to states  $\mathbf{q}_i = (k, l)$  for  $0 \leq l \leq Q - 1 - k$ .

### Lane-based model

We now consider an alternative approach where lane arrival probabilities can be non-uniform across lanes of the intersection. We abuse notation and denote  $p_j$  the probability of a user arriving on lane  $j$  at the next time period. Let  $\mathcal{L}_i = \{\sigma_\emptyset, \sigma_{\underline{v}_i}, \sigma_{\bar{v}_i}\}$  be the set of possible lane-based states with regards to user  $i$  bidding  $v_i$ ; where  $\sigma_\emptyset$  represents an empty lane,  $\sigma_{\underline{v}_i}$  the arrival of a user bidding lower than  $v_i$  and  $\sigma_{\bar{v}_i}$  the arrival of a user bidding higher than  $v_i$ . Let  $z_{ij} \in \mathcal{L}_i$  be the state of lane  $j$  w.r.t. user  $i$ . We denote  $\mathbf{z}_i = (z_{i,1}, \dots, z_{i,Q-1})$  the state w.r.t. user  $i$  in the lane-based model.

**Definition 4.11** (State space for lane-based model). *In the lane-based model, the state space of the MC for user  $i$  declaring a delay cost of  $v_i$  in a pricing queue of size  $Q$  is*

$$\mathcal{S}_i^z = \left\{ \mathbf{z}_i = (z_{i,j_1}, \dots, z_{i,j_{Q-1}}) \in \mathcal{L}_i^{Q-1} \right\}. \quad (4.39)$$

In the lane-based model, the number of possible states w.r.t. user  $i$  in an intersection with  $Q$  lanes is  $|\mathcal{L}_i|^{Q-1}$ , hence a four-lane intersection has  $3^3 = 27$  possible states.

Let  $f_z : \mathcal{S}_i^z \times \mathcal{R}^Q \times 2^Q \rightarrow \mathcal{S}_i^z$  be the MC transition function of the lane-based model and let  $g_z(\mathbf{z}_i)$  be the one-step cost. Let  $\mathcal{T}_i^z \subset \mathcal{S}_i^z$  be the set of terminal states, i.e.  $\mathcal{T}_i^z = \{\mathbf{z}_i \in$

$\mathcal{S}_i^z : z_{ij} = \sigma_\emptyset \vee z_{ij} = \sigma_{v_i}, j = 1, \dots, Q - 1$ . The expected waiting time  $W_z(v_i, \mathbf{z}_i)$  of user  $i$  declaring  $v_i$  for state  $\mathbf{z}_i$  in the lane-based model is thus:

$$W_z(v_i, \mathbf{z}_i) = \begin{cases} 0 & \text{if } \mathbf{z}_i \in \mathcal{T}_i^z, \\ \mathbb{E}[W_z(v_i, f_z(\mathbf{z}_i, \xi))] + g_z(\mathbf{z}_i) & \text{otherwise,} \end{cases} \quad (4.40)$$

with:

$$\mathbb{E}[W_z(v_i, f_z(\mathbf{z}_i, \xi))] = \sum_{\mathbf{z}'_i \in \mathcal{S}_i^z} \Pr[\mathbf{z}'_i | \mathbf{z}_i] W_z(v_i, \mathbf{z}'_i). \quad (4.41)$$

This leads to the following system of linear equations:

$$W_z(v_i, \mathbf{z}_i) = 0 \quad \forall \mathbf{z}_i \in \mathcal{T}_i^z, \quad (4.42a)$$

$$W_z(v_i, \mathbf{z}_i)(1 - \Pr[\mathbf{z}_i | \mathbf{z}_i]) - \sum_{\mathbf{z}'_i \in \mathcal{S}_i^z : \mathbf{z}'_i \neq \mathbf{z}_i} W_z(v_i, \mathbf{z}'_i) \Pr[\mathbf{z}'_i | \mathbf{z}_i] = g_z(\mathbf{z}_i) \quad \forall \mathbf{z}_i \in \mathcal{S}_i^z \setminus \mathcal{T}_i^z. \quad (4.42b)$$

Unlike the systems of equations (4.37), the system (4.42) does not admit an intuitive decomposition which could be used to solve the system of equations via a recursive algorithm. Thus, in our numerical experiments, (4.42) is solved in a single step using traditional linear algebra codes.

To determine transition probabilities in the lane-based mechanism, it is necessary to track which user will be serviced at the next time period. For any non-terminal state, there is at least one higher-bidder in the pricing queue, i.e.  $\bar{q}_i \geq 1$ , and the highest-bidder in the queue will be serviced next. Let  $\mathcal{V}_{>i}$  be the set of higher-bidders, i.e.  $\mathcal{V}_{>i} = \{j_1, j_2, \dots, j_{\bar{q}_i}\}$  with  $v_i < v_{j_1} < v_{j_2} < \dots < v_{j_{\bar{q}_i}}$ . If there is more than a single user bidding higher than user  $i$ , i.e.  $\bar{q}_i > 1$ , it is unknown which of these higher-bidders will be the highest-bidder. Since users' delay costs are assumed to all follow the same probability distribution, the probability that a higher-bidder is the highest-bidder is uniform, i.e.  $1/\bar{q}_i$ .

Let  $z_{ij}(k)$  be the state of lane  $j$  with regards to user  $i$  assuming lane  $k$  is the moving lane, i.e. the lane occupied with the highest-bidder in the pricing queue. Let  $\Pr[z'_{ij} | z_{ij}(k)]$  be the transition probability of lane  $j$  with regards to user  $i$  from state  $z_{ij}(k)$  to state  $z'_{ij}$ . Since the arrival process of a lane is assumed to be independent of that of other lanes, the transition probability from state  $\mathbf{z}_i$  to  $\mathbf{z}'_i$  denoted  $\Pr[\mathbf{z}'_i | \mathbf{z}_i]$  is

$$\Pr[\mathbf{z}'_i | \mathbf{z}_i] = \Pr[(z'_{i,1}, \dots, z'_{i,Q-1}) | (z_{i,1}, \dots, z_{i,Q-1})] = \frac{1}{\bar{q}_i} \sum_{k \in \mathcal{V}_{>i}} \prod_{j=1}^{Q-1} \Pr[z'_{ij} | z_{ij}(k)]. \quad (4.43)$$

In the lane-based mechanism, transition probabilities are function of current lane state. If lane  $j$  is empty or if lane  $j$  is occupied by the highest bidder in the pricing queue, then we say that this lane is open. Otherwise, lane  $j$  is either occupied by a lower-bidding user or a higher-bidding user who will not be serviced next and we say that his lane is closed. Let  $\mathcal{V}_{-i}$  be the set of users in the pricing queue other than user  $i$ . Let  $\mathcal{O}_{i,j} \subseteq \mathcal{L}_i$  and  $\mathcal{C}_{i,j} \subseteq \mathcal{L}_i$  be the



set of open and closed states for lane  $j$  with regards to user  $i$ , respectively:

$$\mathcal{O}_{i,j} = \begin{cases} \{\sigma_\emptyset, \sigma_{\bar{v}_i}\} & \text{if } j \in \arg \max\{v_k : k \in \mathcal{V}_{-i}\}, \\ \{\sigma_\emptyset\} & \text{otherwise,} \end{cases} \quad (4.44)$$

$$\mathcal{C}_{i,j} = \begin{cases} \{\sigma_{v_i}\} & \text{if } j \in \arg \max\{v_k : k \in \mathcal{V}_{-i}\}, \\ \{\sigma_{v_i}, \sigma_{\bar{v}_i}\} & \text{otherwise.} \end{cases} \quad (4.45)$$

For any lane  $j$  in state  $\sigma \in \mathcal{O}_{i,j}$ , we have the following transition probabilities:

$$\Pr[\sigma_\emptyset | \sigma] = 1 - p_j \quad \forall j \in \{1, \dots, Q-1\}, \forall \sigma \in \mathcal{O}_{i,j}, \quad (4.46a)$$

$$\Pr[\sigma_{v_i} | \sigma] = p_j F(v_i) \quad \forall j \in \{1, \dots, Q-1\}, \forall \sigma \in \mathcal{O}_{i,j}, \quad (4.46b)$$

$$\Pr[\sigma_{\bar{v}_i} | \sigma] = p_j(1 - F(v_i)) \quad \forall j \in \{1, \dots, Q-1\}, \forall \sigma \in \mathcal{O}_{i,j}. \quad (4.46c)$$

Otherwise if lane  $j$  is in state  $\sigma \in \mathcal{C}_{i,j}$ , this lane remains in its current state with probability one, since the corresponding user cannot be serviced at the next time period.

$$\Pr[\sigma' | \sigma] = \begin{cases} 1 & \text{if } \sigma' = \sigma, \\ 0 & \text{otherwise,} \end{cases} \quad \forall j \in \{1, \dots, Q-1\}, \forall \sigma \in \mathcal{C}_{i,j}. \quad (4.47)$$

Using these lane-based transition probabilities, we can determine intersection-based transition probabilities via Eq. (4.43).

The queue-based model can be viewed as a special case of the lane-based model. Let  $\mathbf{1}_{z_{ij}=\sigma}$  be the indicator function taking value 1 if  $z_{ij} = \sigma$  and 0 otherwise. We first define the mapping between the state spaces of both queue- and lane-based models.

**Definition 4.12.** Let  $h_i : \mathcal{S}_i^z \rightarrow \mathcal{S}_i^q$  be a function mapping the state space of the lane-based model to the state space of the queue-based model:  $h_i : \mathbf{z}_i \mapsto \mathbf{q}_i = h_i(\mathbf{z}_i)$  with  $q_i = \sum_{j=1}^{Q-1} \mathbf{1}_{z_{ij}=\sigma_{v_i}}$  and  $q_\emptyset = \sum_{j=1}^{Q-1} \mathbf{1}_{z_{ij}=\sigma_\emptyset}$ .

The proposition below highlights the correspondence between queue- and lane-based models.

**Proposition 4.2.** If lane arrival probabilities are uniform, i.e.  $p = p_j$  for all  $j = 1, \dots, Q-1$ , and if the one-step-costs of the queue- and lane-based models are such that  $g_z(\mathbf{z}_i) = g_q(\mathbf{q}_i)$  for all states  $\mathbf{z}_i \in \mathcal{S}_i^z$  and  $\mathbf{q}_i \in \mathcal{S}_i^q$  such that  $\mathbf{q}_i = h_i(\mathbf{z}_i)$ , then  $W_z(v_i, \mathbf{z}_i) = W_q(v_i, \mathbf{q}_i)$  for all such states.

Proposition 4.2 establishes a correspondence between the proposed queue- and lane-based models for determining the expected waiting times of users and shows that under uniform lane arrival probabilities and one-step-costs, both models are equivalent.

We next present a payment mechanism that can be used with both of these models to determine incentive-compatible payments.

---

### 4.5.3 Payment mechanism

The proposed payment mechanism can be used with both queue- and lane-based models for determining the expected waiting time of users. Hence, in this section, we simply denote  $W_i(v_i)$  the expected waiting time of user  $i$  bidding  $v_i$ . This expected waiting time can be replaced by  $W_q(v_i, \mathbf{q}_i)$  for the queue-based model, or by  $W_z(v_i, \mathbf{z}_i)$  for the lane-based model. To determine incentive-compatible user payments, we calculate the expected marginal cost user  $i$  imposes on other users. For this, we first examine the period of time over which an extra user in the lane of user  $i$  is expected to have an impact on other users.

The busy period w.r.t. user  $i$  is the expected time required to clear the queues if  $i$  declares a minimal delay cost, i.e.  $v_i = \underline{v}$ . Observe that if  $v_i = \underline{v}$ , then  $\underline{q}_i = 0$ , since no user can bid lower than  $\underline{v}$  i.e.  $F(\underline{v}) = 0$ . We measure the impact of user  $i$  onto other users throughout the remaining busy period, i.e. the duration between the start of service of user  $i$  if  $i$  declares  $v_i$ , and the start of service of  $i$  if  $i$  had declared a minimal delay cost. Hence, the remaining busy period w.r.t. user  $i$  bidding  $v_i$  denoted  $\Delta W_i(v_i)$  can be defined as:

$$\Delta W_i(v_i) \equiv W_i(\underline{v}) - W_i(v_i). \quad (4.48)$$

To determine the marginal impact of user  $i$  on other users, we can split the remaining busy period w.r.t. user  $i$  into two components: the expected portion of  $\Delta W_i(v_i)$  which affects users who arrived *before* user  $i$ ; and the expected portion of  $\Delta W_i(v_i)$  which affects users who are expected to arrive *after* user  $i$ . We denote  $B_i(v_i)$  and  $A_i(v_i)$  these quantities, respectively, and intuitively we define

$$\Delta W_i(v_i) \equiv B_i(v_i) + A_i(v_i). \quad (4.49)$$

Calculating  $B_i(v_i)$  and  $A_i(v_i)$  correctly is not trivial, especially the latter since the exact number of users arriving after  $i$  that will be delayed by user  $i$  is unknown. We next propose a method to determine  $B_i(v_i)$ , then use Eq. (4.49) together with Eq. (4.48) to obtain  $A_i(v_i)$ .

To determine the marginal delay caused by user  $i$  onto users already in the pricing queue, we need only to consider those  $\underline{q}_i$  users which bid lower than  $i$  since others users are not delayed by user  $i$ . Let  $\mathcal{V}_{<i}$  be the set of all users bidding lower than  $i$  ordered by increasing declared delay costs, i.e.  $\mathcal{V}_{<i} = \{j_1, j_2, \dots, j_{\underline{q}_i}\}$  with  $v_{j_1} < v_{j_2} < \dots < v_{j_{\underline{q}_i}} < v_i$ . To determine the expected portion of the remaining busy period w.r.t. user  $i$  which affects user  $j \in \mathcal{V}_{<i}$ , we take the perspective of an extra user bidding  $v_j$  in the lane of user  $i$ . The expected period of time during which this extra user affects user  $j$  is the difference between the waiting time of the extra user if the extra user is bypassed by user  $j$  and that if the extra user bypasses user  $j$  in the pricing queue. We abuse notation and denote  $W_i(v_j : z_{ij} = \sigma_{\bar{v}_i})$  and  $W_i(v_j : z_{ij} = \sigma_{\underline{v}_i})$  these expected waiting times, respectively.

We propose the following formula for  $B_i(v_i)$ :

$$B_i(v_i) = \sum_{k=1}^{\underline{q}_i} W_i(v_{j_k} : z_{ij_k} = \sigma_{\bar{v}_i}) - W_i(v_{j_k} : z_{ij_k} = \sigma_{\underline{v}_i}). \quad (4.50)$$

The definitional relationship linking the time periods  $B_i(v_i)$  and  $\Delta W_i(v_i)$ , i.e. Eq. (4.49), implicitly assumes that  $B_i(v_i) \leq \Delta W_i(v_i)$ . However, it is not obvious that the right-hand side of Eq. (4.50) is never greater than  $\Delta W_i(v_i)$ . Since this is necessary for the correctness of the proposed formula for the time period  $B_i(v_i)$ , i.e. Eq. (4.50), we prove this relationship in Proposition 4.3.

**Proposition 4.3.** *For any user  $i$  bidding  $v_i$ ,*

$$\sum_{k=1}^{q_i} W_i(v_{j_k} : z_{ij_k} = \sigma_{\bar{v}_i}) - W_i(v_{j_k} : z_{ij_k} = \sigma_{\underline{v}_i}) \leq \Delta W_i(v_i). \quad (4.51)$$

Proposition 4.3 together with Eq. (4.49) ensure that the expected portion of the remaining busy period corresponding to future arrivals is nonnegative, i.e.  $\Delta W_i(v_i) - B_i(v_i) = A_i(v_i) \geq 0$ . Using Eq. (4.50), the expected marginal delay cost that user  $i$  imposes on users who arrived before her can be determined as:

$$MB_i(v_i) = \sum_{k=1}^{q_i} (W_i(v_{j_k} : z_{ij_k} = \sigma_{\bar{v}_i}) - W_i(v_{j_k} : z_{ij_k} = \sigma_{\underline{v}_i}))v_{j_k}. \quad (4.52)$$

To determine the expected marginal delay cost that user  $i$  imposes on future arrivals, denoted  $MA_i(v_i)$ , we use the result of Dolan [1978] who observed that this marginal delay cost can be calculated by integrating  $\frac{dA_i(v)}{dv}v$  over the bid range  $[\underline{v}, v_i]$ . Further, the author observed that the state is constant for any bid comprised between two consecutive bids of users in the pricing queue. Thus, for any segment corresponding to a pair of consecutive bids in the sequence  $\{\underline{v}, v_{j_1}, \dots, v_{j_{q_i-1}}, v_{j_{q_i}}, v_i\}$ , the expected marginal delay cost imposed by user  $i$  onto users bidding in this segment can be calculated by integrating  $\frac{dA_i(v)}{dv}v$  over the corresponding domain. Specifically, let  $v_{j_0} = \underline{v}$  and let  $v_{j_{q_i+1}} = v_i$ . Let  $D(k) = (v_{j_k}, v_{j_{k+1}})$  be the domain corresponding to the bid segment  $(v_{j_m}, v_{j_n})$  for  $m < n$  such that  $j_m, j_n \in \mathcal{V}_{<i}$ . Accordingly, the expected marginal delay cost imposed by user  $i$  on future arrivals can be determined as:

$$MA_i(v_i) = \sum_{k=0}^{q_i} \int_{D(k)} \frac{dA_i(v)}{dv}v dv = \sum_{k=0}^{q_i} \int_{D(k)} \frac{-dW_i(v)}{dv}v dv. \quad (4.53)$$

The expected marginal delay cost imposed by user  $i$  declaring  $v_i$  on other users is thus:

$$MC_i(v_i) = MB_i(v_i) + MA_i(v_i). \quad (4.54)$$

We next give the main result.

**Theorem 4.2.** *If the user objective function is  $\min_{v_i} v_i^* W_i(v_i) + P_i(v_i)$  and users are served in order of decreasing declared delay costs, the payment  $P_i(v_i) = MC_i(v_i)$  is incentive-compatible.*

Theorem 4.2 proves that the expected marginal delay cost as defined by Eq. (4.54) represents an incentive-compatible user payment in the dynamic sense, i.e. by taking into account the expected impact of users on future arrivals. This provides a basis to implement the proposed queue- and lane-based mechanisms in an online fashion. Upon reaching the front of their lane, users are asked to declare their delay cost  $v_i$  and receive a corresponding payment  $P_i(v_i)$  which ensures strategyproof user behavior in the long run.

The proposed queue- and lane-based traffic intersection mechanisms are computationally efficient. In terms of computational resources, both mechanisms require the solution of Eqs.

(4.52) and (4.53) to determine user payments. Eq. (4.52) requires the solution of multiple systems of linear equations ((4.37) or (4.42) depending on the mechanism selected) which can be accomplished using standard linear algebra codes. To compute Eq. (4.53), standard numerical integration and differentiation techniques can be used in combination with codes for systems of linear equations. As will be shown in our numerical experiments, both mechanisms have low average computation time per user and can thus be expected to be executed in real-time to determine incentive-compatible payments.

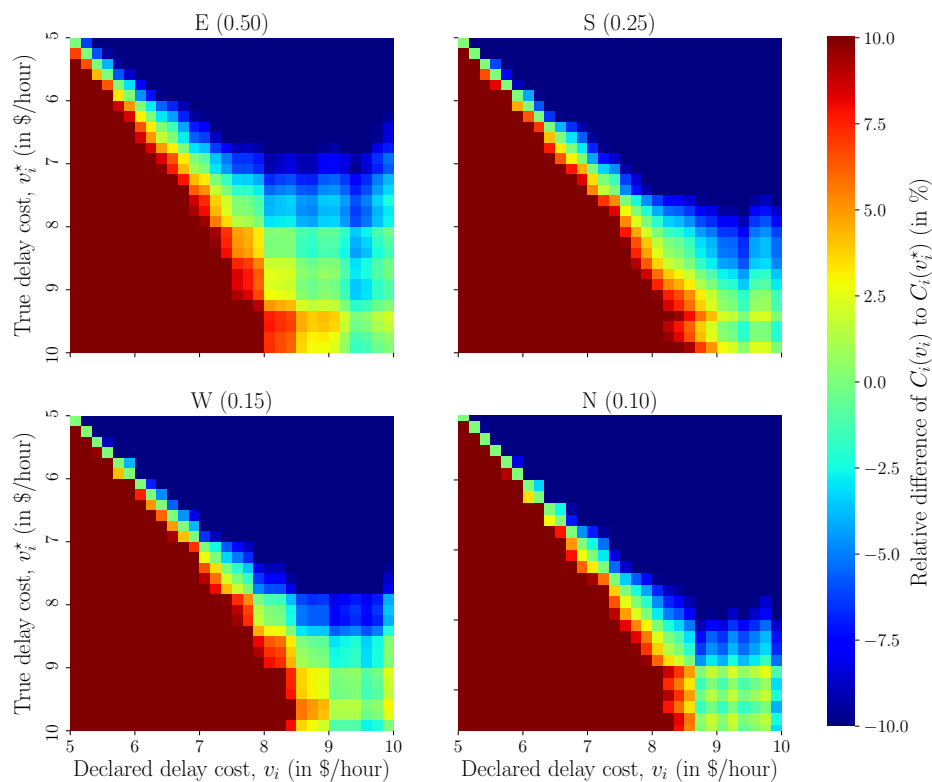
#### 4.5.4 Numerical experiments

We implement the proposed online mechanisms by simulating the arrival and departure of users at an intersection using a discrete time process. At every time period, random trials are conducted for each empty lane to simulate the stochastic arrival process of users. For all our experiments, we use a unit one-step cost, i.e.  $g_q(\mathbf{q}_i) = g_z(\mathbf{z}_i) = 1$  for all states  $\mathbf{q}_i \in \mathcal{S}_i^q$  and  $\mathbf{z}_i \in \mathcal{S}_i^z$ . We set the time period duration to 1 s, i.e. representing a uniform user service time of 1 s. This value is chosen based on the geometric configuration of typical 4-lane intersection and average vehicle speeds [Levin and Rey, 2017]. Users' delay costs are randomly drawn from a uniform distribution  $\mathcal{U}(5, 10)$ , i.e.  $\underline{v} = \$5$  /hour and  $\bar{v} = \$10$  /hour. These values are chosen based on a study on value of time for automated vehicles [Wong et al., 2018]. This delay cost range corresponds to user bids between 0.14 ¢ and 0.28 ¢ for a time period duration of 1 s which aims to represent users' willingness to pay for service priority at traffic intersections.

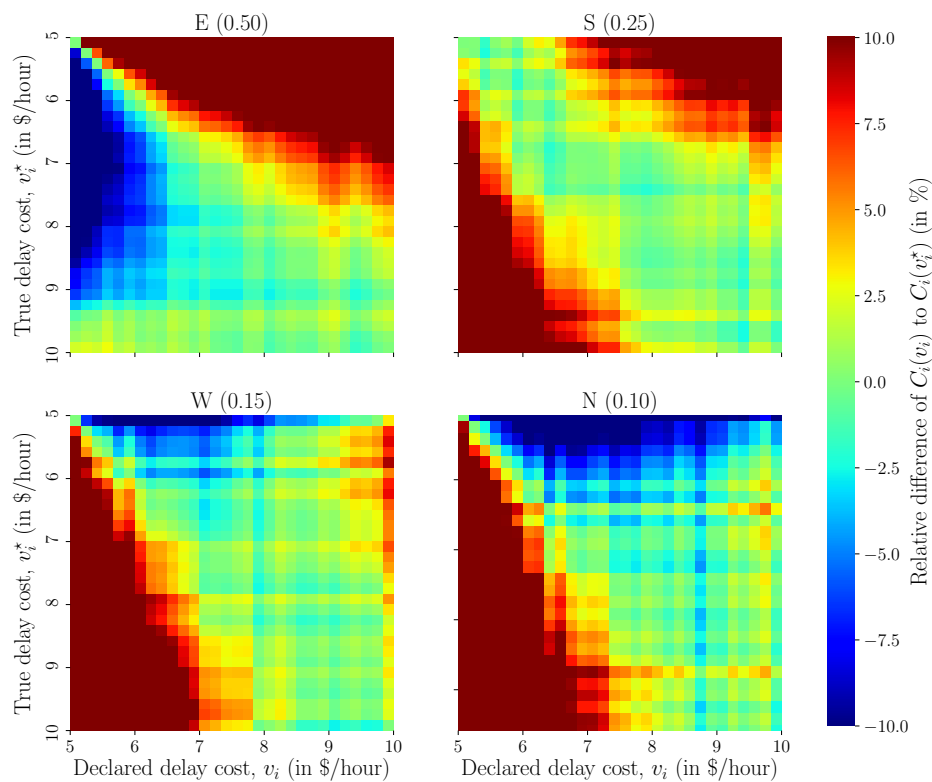
To quantify the value of using a dynamic auction compared to a static auction, we implement the static mechanism outlined by Dolan [1978] for priority queueing systems. This static mechanism is a Vickrey-Clarke-Groves auction which is known to be incentive-compatible in the static sense, i.e. all participating users are assumed to be known when determining payments [Vickrey, 1961, Clarke, 1971, Groves and Loeb, 1975]. The static mechanism utilizes the same priority queueing model than that of the proposed online mechanism, i.e. users in the pricing queue are sorted by decreasing declared delay costs, but differ in the determination of waiting time and user payments. In the static mechanism, the waiting time of user  $i$  is only function of the position of user  $i$  in the pricing queue, e.g. if user  $i$  is third in the pricing queue, her waiting time is the sum of the two one-step costs for transitioning from the current state to the state where user  $i$  served. The payment of user  $i$  is sum of user valuations over lower-bidders in the pricing queue. This payment is incentive-compatible in the static sense [Dolan, 1978].

In each experiment, we simulate the service of one million users and report the average user behavior by segmenting users' delay cost. Specifically, we segment the delay cost range  $[5, 10]$  into 30 uniform bins and group all serviced users throughout the simulation into these bins. We then report average waiting times, payments and generalized user costs based on the average quantities obtained for each of these 30 bins. The simulations are implemented in Python and the linear systems of equations (4.37) and (4.42) are solved using Numpy's linear algebra solver on a Windows 10 machine with 8 Gb of RAM and a CPU of 2.7 Ghz. We compare the behavior of the proposed traffic intersection auctions mechanisms on a four-lane intersection under non-uniform lane arrival probabilities on a four-lane intersection. We consider non-uniform lane arrival probabilities  $p_j$  for all lanes  $j$  of a four-lane intersection.

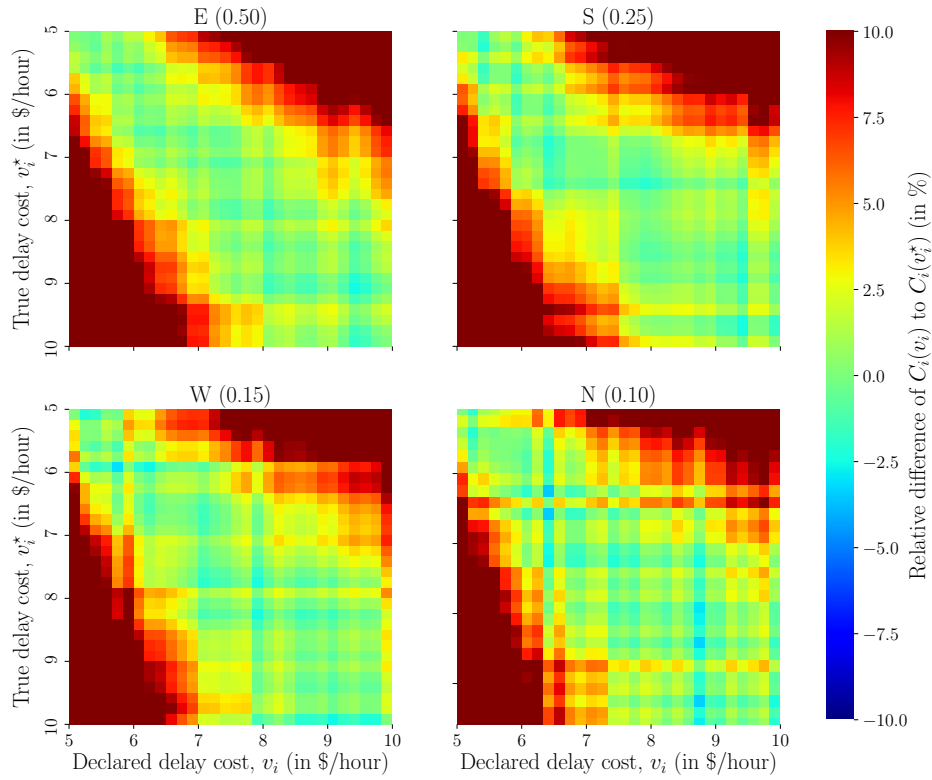
We use cardinal directions, i.e. East (E), South (S), West (W) and North (N) to refer to all  $Q = 4$  lanes of the intersection and assign lane arrival probabilities of 0.50, 0.25,



(a) Static mechanism.



(b) Queue-based mechanism.



(c) Lane-based mechanism.

Figure 4.5: Relative user generalized cost  $(C_i(v_i) - C_i(v_i^*)) / C_i(v_i^*)$ . Results of a simulation with 1 million users with lane-based arrival probabilities of 0.50, 0.25, 0.15 and 0.10. Figure 4.6a illustrates the static mechanism; Figure 4.6b illustrates the queue-based mechanism and Figure 4.5c illustrates the lane-based mechanism.

0.15 and 0.10, respectively. For the queue-based mechanism, we use the average lane arrival probability, i.e.  $p = \sum_{j=1}^Q p_j = 0.25$ . We report lane-based heatmaps of the relative difference in terms of relative user generalized cost  $(C_i(v_i) - C_i(v_i^*)) / C_i(v_i^*)$  for the static, queue- and lane-based mechanisms in Figure 4.5. Figure 4.6a shows the outcome of the simulation using the static mechanism. We find that the static mechanism does not promote truthful user behavior for all four lanes of the intersection, i.e. regardless of the lane arrival probability the mechanism is not incentive-compatible, and on all lanes users may reduce their generalized cost in the long run by over-reporting their delay cost. The outcome of the same simulation obtained using the queue-based mechanism is summarized in Figure 4.6b. The outcome of these simulation highlight that using the average lane arrival probability in the queue-based mechanism fails to yields incentive-compatible outcome for the East, West and North lanes which have a lane arrival probability different that  $p = 0.25$ . Only users on the South lane with  $p_S = 0.25 = p$  receive incentive-compatible payments. In turn, Figure 4.5c illustrates that the lane-based mechanism is incentive-compatible for each lane of the intersection.

Additional numerical results can be found in [Rey et al. \[2021\]](#).

## 4.6 Conclusion

This chapter first presented a method for optimizing reservation-based intersection controls in AIM before embedding this conflict point formulation in a stochastic network traffic model. A pressure-based network traffic control policy for coordinating legacy vehicles (LV) and autonomous vehicles (AV) was proposed under the assumption that LVs and AVs share the urban network infrastructure. This policy combines green (for LVs) and blue (for AVs) phases to coordinate traffic at network intersections. We characterized the stability region of the proposed queuing system and showed that the proposed decentralized hybrid network control policy is stable, i.e. that it maximizes network throughput, under conventional travel demand conditions.

The last part of this chapter focus on market-based auction mechanisms for AIM. Assuming that users can declare their delay cost privately to the intersection manager, we investigated the computation of incentive-compatible payments for users to minimize users' generalized cost which is defined as a linear combination of expected waiting time and user payment. We introduced two Markov chain models to determine users' expected waiting time, and presented a payment mechanism that can be implemented with both models. We showed that the proposed online traffic intersection mechanisms are incentive-compatible in the dynamic sense and thus maximize social welfare.

# Chapter 5

## Perspectives

The goal of this thesis was to provide an overview of some contributions in optimization and game theory, and their application to decision-making problems encountered in transportation systems. These contributions have been organized into three main chapters that differ by the operations research (OR) methodologies used therein, as well as by the decision-making problems selected in each chapter. Chapter 2 focused on contributions in the field of bilevel optimization and its applications to transportation network design problems. Chapter 3 turned to mixed-integer nonlinear programming and its applications to aircraft conflict resolution problems. Chapter 4 embraced a broader modeling perspective by summarizing different optimization and game-theoretical approaches to autonomous intersection management. In each of these three chapters, novel approaches that leveraged state-of-the-art advances in OR techniques were conceived, developed and implemented. The contributions of this thesis are thus both theoretical and numerical, and they lie both in OR as well as in the field of transportation systems science.

Next, an attempt is made to identify research perspectives that could build on these works as well as to provide guidelines for extending the OR methodologies used therein. The rest of this section is organized in two parts: methodological perspectives are examined in Section 5.1, and perspectives in transportation are discussed in Section 5.2.

### 5.1 Methodological perspectives

This section attempts to synthesize the state-of-the-art in the fields of bilevel optimization (Section 5.1.1) and mixed-integer programming (Section 5.1.2) and discusses how methodological developments could be steered to achieve greater efficiency. This section also extends the discussion to machine learning and data-driven optimization approaches (Section 5.1.3), and their potential to boost optimization and game-theoretical methods.

#### 5.1.1 Bilevel optimization

Bilevel optimization is a rapidly growing field of research. Chapter 2 discussed its applications to transportation network design problems, which provide a natural framework to develop bilevel optimization approaches given the existing hierarchical interactions among multiple players such as transportation authorities and users. Of the many challenges faced



by researchers in this field, one of most stringent is the computational scalability and the quality of solution methods for bilevel optimization problems. It can roughly be claimed that bilevel optimization problems whose follower problem(s) enjoy convexity have a substantial advantage over problems that do not enjoy this property. This is largely due to the fact that bilevel optimization problems with convex follower problem can be reformulated into single-level optimization problems by making use of the follower's first-order optimality conditions, such as Karush Kuhn Tucker (KKT) conditions or strong duality. It should be noted that the single-level reformulation of bilevel optimization problems is not trivial and, often, not equivalent. It is non-trivial because after embedding the follower's optimality conditions in a single-level formulation, the latter is a nonconvex optimization problem, e.g. a mathematical program with complementarity constraints (MPCC) that requires further care to be solved efficiently to optimality. In general, the single-level MPCC reformulation is not equivalent to the bilevel optimization problem in the sense that their solution sets may not always coincide. However, it is well-known that, under certain constraint qualification conditions, an optimal (global) solution of the single-level MPCC reformulation corresponds to an optimal (global) solution of the bilevel optimization problem [Dempe and Dutta, 2012]. Nevertheless, bilevel optimization problems with a convex follower problem are globally much more tractable than their counterparts. From a practical standpoint, however, this limitation may tend to incentivize modelers to simplify their representation of the follower problem so as to achieve convexity. This simplification is not always desirable as it prevents, notably, the treatment of bilevel optimization problems with integer follower problems even though substantial progress has been achieved for this class of bilevel optimization problems [Fischetti et al., 2017, Liu et al., 2021]. Nevertheless, these approaches remain quite challenging to implement and may require certain assumptions on the structure of the bilevel optimization problems that can be handled. Even in the case of linear bilevel optimization, computational tractability and scalability remains a concern [Kleinert et al., 2020, 2021]. Further, by large, most efforts in the community have been dedicated to optimistic bilevel optimization problems and few methods are available for solving pessimistic bilevel optimization problems [Aussel and Svensson, 2019, Zeng, 2020].

Transportation systems often aim to improve social welfare using specific performance metrics, e.g. congestion, pollution. Hence, such systems shall be designed to anticipate the reaction of the users of the system whose goals are unlikely to be aligned with system-level objectives. This was the context of the transportation network design problems discussed in Chapter 2 wherein users' collective reaction is modeled as a convex optimization problem that characterizes a Nash equilibrium. Yet, as outlined therein, solving large scale transportation network design problems to optimality is extremely challenging. This is partly due to the fact that traditional branch-and-bound algorithms that branch on complementary slackness conditions cannot scale easily due to the large number of nodes and links—let alone paths—in large-scale transportation networks. There is thus a need to develop more scalable exact algorithms that can handle the large dimensions of realistic city-scale transportation networks. A possible avenue of research is to explore the potential of column generation procedures to iteratively obtain new path variables and embed this approach in a traditional complementary slackness based branch-and-bound algorithm. One caveat is that iteratively generating columns may imply that some follower constraints must also be generated iteratively, thus leading to a row-and-column generation framework wherein newly generated columns also create new follower constraints for which complementary slackness conditions must hold.

Efforts are also needed to go beyond the classical bilevel network design problem under Nash equilibrium. While this model has been extensively studied, it relies on several

---

fundamental modeling assumptions such as link travel cost convexity. The incorporation of multiple players, as opposed to a system-level approach also holds great potential for development. This is reinforced by emerging trends in on-demand mobility and logistics where several service providers compete for transportation demand [Perboli et al., 2021]. Bilevel optimization offers the possibility to consider games with multiple leaders and/or followers and provide new insights for market design and pricing in transportation systems.

A complementary avenue of research is to attempt to refine the leader’s model of the follower by incorporating uncertainty. Stochastic optimization—and stochastic programming especially—have greatly contributed to extending deterministic formulations to account for the uncertainty of model parameters. Much fewer efforts have been devoted to the study of stochastic bilevel optimization problems [Patriksson, 2008, Fampa et al., 2008]. A colleague once told me: “solving a bilevel optimization problem is working for the leader”. This is a general statement about bilevel optimization that recognizes the fact that the follower problem is only the leader’s representation of the follower’s best response. It is thus natural to question whether the leader has perfect knowledge of the follower’s strategy. In the event that the answer to this question is not *yes*, stochastic bilevel optimization could provide elements of answer to design richer formulations that account for the leader’s uncertainty regarding the follower’s response. For example, a transportation network user may adopt a mixed strategy based on travel time probability distributions to choose her route. Assuming that the leader has knowledge of such distributions, it could therefore be appropriate for the leader to anticipate this mixed strategy. This highlights that stochastic bilevel optimization holds substantial potential to enrich existing deterministic approaches by introducing uncertainty at either or both levels.

### 5.1.2 Mixed-integer programming and solvers

Mixed-integer programming (MIP) has become a standard approach to solve challenging decision-making problems. Recent advances in off-the-shelf MIP solver capabilities and competition among MIP solvers has led to substantial computational improvements over the past decades [Achterberg and Wunderling, 2013]. This means that the size of MIP problems that can be solved in reasonable time (and memory) today is much greater than that it was in the 2000s [Lodi, 2010]. This trend has supported the democratization of MIP solvers in the OR community and beyond. This has led to a paradigm shift: a mixed-integer linear programming (MILP) problem can now be viewed as tractable, even though no polynomial-time exact algorithm is available to solve it. Not all MILPs enjoy this treatment. There remains families of MILPs, such as vehicle routing problems, that are much too challenging to solve optimally using off-the-shelf solvers. However, some MILPs have become relatively easy to solve using off-the-shelf MIP solvers. One of the immediate consequences is that it has now become reasonable to devise solution methods wherein a MILP is solved repeatedly under the assumption that a (near-)optimal solution can be found rather quickly. This has also boosted the development of matheuristics wherein mathematical programming problems are solved in the course of a broader heuristic algorithm [Fischetti and Fischetti, 2018]. Other classes of MIP problems have also benefited from this progress, e.g. mixed-integer quadratic programming (MIQP) and mixed-integer quadratically-constrained programming (MIQCP). Chapter 3 illustrates this by showing that a challenging nonconvex mixed-integer nonlinear programming (MINLP) problem can be solved to optimality in reasonable time by an algorithm which repeatedly solves MIQPs and MIQCPs. This suggests that the use of certain MIP formulations within broader algorithmic frameworks holds great potential.

Transportation systems are rich in decision-making problems, many of which can be viewed as variants of classical OR problems. There are therefore strong incentives to develop the practice of using MIP in solution approaches to decision-making problems encountered in transportation systems, even when real-time capabilities are at stake. Chapter 4 advocates that vehicle scheduling within an autonomous intersection management context can be solved efficiently using MILP up to certain levels of demand. This off-the-shelf approach fits well operational problems that can be solved repeatedly using a rolling horizon framework. For instance, in air traffic control, it is difficult to solve conflict resolution problems with a large lookahead since there is often too much uncertainty associated to aircraft’s future positions beyond a 30-minute horizon. Hence, from a practical standpoint, it is meaningful to adopt a rolling horizon framework with a balanced lookahead and to repeatedly solve a small-to-medium scale MILP that can provide competitive solutions. The main drawback of this approach is that this may be too myopic in that future events are ignored even though some knowledge of possible scenarios may be available. There are thus opportunities to enrich MIP-based approaches by incorporating discounted lookahead information or even attempting to learn data features over time. This could be achieved by using surrogate objective functions that aim to anticipate the impact of future events.

MIP provides a flexible modeling framework for OR practitioners. Cutting-edge solution methods for challenging OR problems often go beyond the implementation of direct, off-the-shelf MIP solvers. Modern MIP solvers now offer additional features such as the automatic and customizable application of Benders’ decomposition to MILPs [Bonami et al., 2020]. MIP modeling environments also provide new tools to implement branch-and-cut-and-price algorithms wherein column generation tasks are made easier to handle for the non-expert, even though the learning of such modeling environments itself may present some challenges [Maher et al., 2017]. This trend supports the development of MIP and also advocates for the use of MIP solvers as a default go-to modeling and solution approach. One notable side-effect is the computational opportunities that arise from the ubiquity of MIP solvers: under the hypothesis that their computational performance will continue to improve, there may be a time when formulating and solving a MIP outperforms dedicated combinatorial algorithms. An immediate benefit is the facilitation of complex solution methodologies such as branch-and-cut-and-price algorithms. In the latter, a pricing algorithm is required to price columns as part of the column generation process. In the context of vehicle routing problems, the pricing subproblem is often a constrained shortest path problem that is NP-hard. Historically, such subproblems have been solved extensively with customized label correcting algorithms that may be difficult to implement efficiently. The uprising of MIP solvers raises the question: *When, if ever, will MIP solvers become competitive with label-correcting algorithms for pricing subproblems that arise when solving vehicle routing problems via branch-and-price algorithms?* This also applies to the context of bilevel optimization where MIP solvers may be used to solve follower problems (e.g. Nash equilibrium problems) faster than dedicated algorithms, thus potentially helping in rapidly obtaining competitive bounds. More broadly, this perspective may someday shift the practice of algorithm benchmarking in certain families of OR problems and this may provide opportunities to develop innovative, MIP-powered solution approaches.

### 5.1.3 Machine learning

Machine learning (ML) has been quite successful in solving challenging optimization problems [Bengio et al., 2021]. Most ML-powered approaches to optimization have been targeted

---

towards large-scale problem instances and are therefore heuristic by design. More recently, the OR and ML communities started to explore the potential of ML-powered approaches to enhance mathematical programming-based solution algorithms for challenging optimization problems. This may take the form of data-driven heuristics that are triggered within the course of branch-and-cut-and-bound algorithm in an attempt to obtain competitive bounds to prune the tree. One drawback lies in the vast amount of computational resources required for efficient model training in order to obtain a well-trained predictive model. To overcome this challenge, carefully selected samples should be obtained to balance training time and the quality of the predictive model. The convex approximation of congestion effects developed in Chapter 2 to obviate the need to solve repeated follower problems can be viewed as an attempt to integrate predictive modeling—in this case nonlinear regression—within a branch-and-price framework. Other applications of ML are possible: for example, it may be used to learn the likelihood that a variable be part of the optimal solution [Tahir et al., 2021]. Another instance is the use of ML to learn optimal branching rules [Lodi and Zarpellon, 2017]. A compromise between avoiding resource-intensive model training and reaping the benefits of ML-powered approaches may be found in the field of reinforcement learning (RL) which is closely related to Markov decision processes (MDP). RL consists of a training-free approach to ML: in RL, optimal decision rules are learnt *online*, i.e. over repeated learning experiences. Hence, casting a branch-and-bound tree search as a MDP opens the door to a suite of RL-based algorithms that could help in identifying optimal tree search decisions such as branching, node selection rules or the decision of when triggering certain heuristics.

So far the interplay between ML and OR has been examined from the perspective of *ML-powered approaches to enhance solution algorithms for OR problems*. The reciprocal, i.e. *the use of OR techniques to enhance solution approaches for ML*, may be more challenging since ML usually targets large scale problem instances. It is unlikely that tools from mathematical programming or combinatorial optimization could enhance the solving of large-scale ML problems from a computational efficiency standpoint. Nevertheless, recent works on feature selection for predictive modeling, a classical problem in the ML community, suggest that bilevel optimization may bring novel insights in regards to solution quality. Feature selection can be cast as a bilevel optimization problem where the leader represents the modeler and the follower represents the training problem [Agor and Özalpın, 2019]. If the target ML model admits a convex optimization formulation, such as support vector machines, one may reformulate the latter using its first-optimality conditions to obtain a single-level mathematical program with complementary constraints that is amenable to classical solution approaches of the bilevel optimization literature. While scaling up may remain a challenge, such OR-driven approaches may assist in discovering global optimal solutions of certain ML problems. Hence, this stream of research provides an exciting sandbox to test the potential of OR techniques to assist solving ML problems.

## 5.2 Perspectives in transportation systems

This section adopts a different stance and attempts to discuss perspectives and trends in transportation systems. The realm of stochastic, dynamic and online transportation problems is first discussed (Section 5.2.1). Future transportation systems are then considered both from the perspective of technological advances and business models (Section 5.2.2). This section concludes by discussing pressing transportation challenges that are inherently tied to sustainability (Section 5.2.3).

### 5.2.1 Stochastic, dynamic and online transportation problems

Transportation systems have evolved considerably over the past few years. The advent of mobile communication technologies and the scaling up of computing capabilities has opened the door to new frameworks for on-demand mobility and logistics. This has been first leveraged by transportation service providers, e.g. Uber, Lyft, Didi; who have developed competitive solutions for urban mobility. Over time, public transportation authorities have also embraced this new paradigm, and several cities worldwide have welcome new mobility solutions that attempt to complement classical services. This shift from traditional fixed schedule mobility systems towards demand-responsive transportation is reflected in the scientific literature where a fast-growing number of studies are devoted to the latter [Vazifeh et al., 2018].

On-demand transportation presents specific challenges: anticipating future demand and adjusting transportation solutions over time call for stochastic and dynamic solution approaches [Soeffker et al., 2021]. Often, dynamic transportation models aim to capture either time-varying traffic conditions or time-dependent pricing policies. It should be emphasized that this is different to multi-period formulations: in the latter, the time scale is larger and at each time period a “copy” of the problem is solved—this is the case of the multi-period DNDP studied in Chapter 2. In contrast, dynamic routing problems target a much finer time resolution, e.g. time-dependent shortest path problems aim to find least-cost paths in time-dependent graphs where link weights vary based on the arrival time at the head of the link. There exists also a distinction between dynamic optimization problems where the data is time-dependent and an *a priori* solution is sought, and online optimization problems where new data is available over time. In the latter, the solution of the problem is a policy, that is, a decision rule. Online optimization problems are highly relevant to modern transportation given the emergence of on-demand mobility and logistics systems where user requests are not known in advance, such as in same-day delivery systems [Voccia et al., 2019]. In several situations, the modeler may have access to historical data that allows her to obtain probability distributions of, for example, user demands. In this context, the decision-making problem can be cast as a stochastic programming problem where an *a priori* solution is sought by optimizing the expected value—or other non-risk-neutral metrics—of the objective function under such probability distribution. If the problem is both stochastic and online, i.e. new data is obtained on-the-fly, parameters’ probability distributions can be learnt or refined over time. This provides room to explore related solution concepts such as RL, which is widely popular in the artificial intelligence community yet remains in its infancy among OR and transportation practitioners [Powell, 2019]. RL, much like approximate dynamic programming, can be viewed as a heuristic to solve large scale MDPs which in turn can be used to model several stochastic and online decision-making problems encountered in transportation. There exists thus ample opportunities to develop innovative solution approaches to modern transportation problems, notably in the realm of on-demand mobility and logistics.

### 5.2.2 Future transportation systems

Recent technological developments in automated and autonomous transportation systems bring interesting perspectives. While, to date, autonomous cars are not patrolling our streets, unmanned aerial vehicles—or drones—are increasingly being tracked for their potential to change urban and rural logistics. In the longer term, urban air mobility may emerge as a novel mode of transportation [Al Haddad et al., 2020]. The development of autonomous transportation systems offers a rich playground for the OR community. An example is provided in

---

Chapter 4 which explores how autonomous intersection management could be designed and implemented on top of—or alongside—the existing transportation infrastructure. Unmanned air mobility and logistics brings exciting decision-making problems to the table such as finding optimal location of vertiports [Shin et al., 2022] or the operations of urban airways to route drone and air taxi traffic [Rajendran and Srinivas, 2020]. Future air traffic management systems are also expected to benefit from automation. This is largely discussed in Chapter 3 at the tactical decision-making level in the context of air traffic control. At a broader scale, automation holds great potential to sustainably improve air traffic flow management [Bertsimas et al., 2011]. Several recent efforts have also investigated the potential of market mechanisms to account for users’ preferences in the decision process [Castelli et al., 2011, Dal Sasso et al., 2019]. On the road, autonomous mobility provides numerous opportunities to rethink how we move. Shared autonomous vehicles are increasingly being considered as opposed to private autonomous vehicle ownership. In a shared autonomous vehicle context, these vehicles are as a service and thus are not owned by consumers [Krueger et al., 2016]. Automated mobility also has implications regarding users’ value of time which could be used to dampen congestion effects if travelers are busy working in their vehicles during their commute [Monteil, 2014, Rashidi et al., 2020].

The future of transportation is steered towards new business models. As the number of mobility or logistics service providers increase, transportation systems are becoming marketplaces where transportation may be priced “sur mesure”. Customized user pricing is at the heart of the mobility-as-a-service (MaaS) and the logistics-as-a-service (LaaS) paradigms [Polydoropoulou et al., 2020]. Both MaaS and LaaS are receiving an increasing attention from both practitioners and researchers due to their potential to shift the way people and goods are moved thanks to path-based pricing and integrated payments. The endgoal of MaaS and LaaS is to provide users the possibility of making seamless payments for their transportation needs. This would allow users to make multimodal trips, possibly via competing service providers, by making a single payment that captures precisely users’ willingness-to-pay for such services. Pricing in MaaS and LaaS systems may require the coordination of multiple non-cooperative service providers as well as consumers competing for resources. This has motivated researchers to study transportation systems through the lens of two-sided markets wherein buyers and sellers interact through a virtual marketplace. In recent efforts, we have investigated the potential of single-leader multi-follower games for two-sided platform pricing in MaaS systems [Xi et al., 2022], and other efforts have been dedicated to study collaboration across MaaS platforms [Pantelidis et al., 2020]. This stream of research highlights the pivotal role of business models in shaping future MaaS and LaaS transportation systems.

It is worth emphasizing that, business implications aside, future transportation systems that rely on path-based pricing face considerable computational challenges. Path-based pricing is often synonymous with non-additive pricing [Han and Lo, 2004]. An example of non-additive, path-based pricing consists of determining optimal user subsidies to encourage the use of less-congested or low-emission routes in a congested road network. Analogously, one could seek optimal user subsidies in a multimodal mobility network where the links of the network are operated by multiple, non-cooperative service providers. These concepts are promising yet they remain computationally challenging to implement. This stems from the inherent complexity of path finding with non-additive costs in medium to large transportation networks. Therefore, innovative methodologies are required to develop non-additive path-based pricing models at scale.

### 5.2.3 Sustainable transportation

Climate change, ongoing conflicts and, more recently, the covid-19 pandemic have also contributed to rethinking transportation systems. Remote working initiatives and e-commerce deliveries have become part of our daily routine. While the “pandemic effect” could be ephemeral, it has triggered several research efforts toward studying and developing more resilient transportation systems capable of absorbing non-recurrent as well as recurrent perturbations such as pollution peaks and adverse weather effects [Budd and Ison, 2020]. The electrification of transportation systems also poses a greater threat to energy networks. Several countries have adopted policies that strongly encourage the adoption of electric vehicles and/or aim to forbid the use of fuel vehicles in a not-so-far future. One may thus expect that cities with a high rate of electrified transportation fleets will face considerable challenges in the context of energy demand surges [Muratori, 2018]. This motivates research efforts focused on developing integrated modeling frameworks that are able to capture the interplay between transportation and energy networks.

Advances in optimization and transportation systems modeling are also expected to lift the potential of humanitarian logistics. Transportation is often central to crisis-induced logistical challenges such as post-disaster network recovery, as outlined in Chapter 2. This also concerns more operational decision-making problems such as humanitarian aid and food rescue logistics. Such resource allocation and/or routing problems share similarities with their profit-driven counterparts but may differ in the choice of the objective function(s) or in the properties of the solutions sought wherein fairness and equity play a central role [Eisenhandler and Tzur, 2019]. At the same time, climate change puts sustainability at the heart of existing and future transportation systems [Zhao et al., 2020]. Although sustainability-driven transportation systems have existed for several years in the literature, there is now more than ever a pressing need to engineer these solutions and develop incentives for practitioners to implement them. From a modeling standpoint, this provides a narrative to support the development of incentive mechanisms for promoting sustainable transportation such as active transportation modes [Kaspi et al., 2014]. In this regard, game theory and mechanism design offer several methodological solutions to develop next generation transportation systems. Pricing problems can directly benefit from game-theoretical enhancements [Brotcorne et al., 2008]. Pricing is central to several transportation systems such as fare structure in public transportation, travel demand balancing, but also last-mile delivery systems. Designing sustainability-driven incentive mechanisms by adjusting service prices is a promising research direction. One of the main challenge is to develop both implementable and scalable pricing mechanisms. Eliciting user preferences requires incentive-compatible mechanisms. Yet, standard truthful mechanisms such as Vickrey-Clarke-Groves auctions may require private data that users themselves may not be accurately aware of. For example, one may not know her value of time with enough precision. Another limitation of such mechanisms arise in the context of combinatorial auctions: it might not be practical for users to report their preferences for all combinations of a mobility bundle since the number of such combinations may be very large. There is thus a need to develop incentive mechanisms that are practical, and that also target goals that are beneficial for the greater good.

# Bibliography

- Tobias Achterberg and Roland Wunderling. Mixed integer programming: Analyzing 12 years of progress. In *Facets of combinatorial optimization*, pages 449–481. Springer, 2013.
- Joseph Agor and Osman Y Özaltın. Feature selection for classification models via bilevel optimization. *Computers & Operations Research*, 106:156–168, 2019.
- Christelle Al Haddad, Emmanouil Chaniotakis, Anna Straubinger, Kay Plötner, and Constantinos Antoniou. Factors affecting the adoption and use of urban air mobility. *Transportation research part A: policy and practice*, 132:696–712, 2020.
- Antonio Alonso-Ayuso, Laureano F Escudero, and F Javier Martín-Campo. Collision avoidance in air traffic management: A mixed-integer linear optimization approach. *IEEE Transactions on Intelligent Transportation Systems*, 12(1):47–57, 2011.
- Antonio Alonso-Ayuso, Laureano F Escudero, and F Javier Martín-Campo. Exact and approximate solving of the aircraft collision resolution problem via turn changes. *Transportation Science*, 50(1):263–274, 2014.
- Antonio Alonso-Ayuso, Laureano F Escudero, and F Javier Martín-Campo. An exact multi-objective mixed integer nonlinear optimization approach for aircraft conflict resolution. *TOP*, 24(2):381–408, 2016.
- Florent Althé and Arnaud De La Fortelle. Analysis of optimal solutions to robot coordination problems to improve autonomous intersection management policies. In *Intelligent Vehicles Symposium (IV), 2016 IEEE*, pages 86–91. IEEE, 2016.
- Didier Aussel and Anton Svensson. Is pessimistic bilevel programming a special case of a mathematical program with complementarity constraints? *Journal of Optimization Theory and Applications*, 181(2):504–520, 2019.
- Saeed Asadi Bagloee, Majid Sarvi, and Michael Patriksson. A hybrid branch-and-bound and Benders decomposition algorithm for the network design problem. *Computer-Aided Civil and Infrastructure Engineering*, 32(4):319–343, 2017.
- Hillel Bar-Gera. Traffic assignment by paired alternative segments. *Transportation Research Part B: Methodological*, 44(8):1022–1046, 2010.
- Mikel Barbara, David Rey, and Ali Akbarnezhad. Optimizing location of new public schools in town planning considering supply and demand. *Journal of Urban Planning and Development*, 147(4):04021057, 2021.
- Jonathan F Bard. *Practical bilevel optimization: algorithms and applications*, volume 30. Springer Science & Business Media, 2013.



- Cynthia Barnhart, Ellis L Johnson, George L Nemhauser, Martin WP Savelsbergh, and Pamela H Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329, 1998.
- Nicolas Barnier and Cyril Allignol. 4d-trajectory deconfliction through departure time adjustment. In *ATM 2009, 8th USA/Europe Air Traffic Management Research and Development Seminar*, 2009.
- Martin Beckmann, Charles B McGuire, and Christopher B Winsten. Studies in the economics of transportation. Technical report, 1956.
- Shlomo Bekhor and Michael Sorani. Are there inefficient links in a real transportation network? In *6th Symposium of the European Association for Research in Transportation (hEART), Haifa, Israel*, 2017.
- Yoshua Bengio, Andrea Lodi, and Antoine Prouvost. Machine learning for combinatorial optimization: a methodological tour d’horizon. *European Journal of Operational Research*, 290(2):405–421, 2021.
- Dimitris Bertsimas and Melvyn Sim. The price of robustness. *Operations Research*, 52(1):35–53, 2004.
- Dimitris Bertsimas, Guglielmo Lulli, and Amedeo Odoni. An integer optimization approach to large-scale air traffic flow management. *Operations research*, 59(1):211–227, 2011.
- Karl Bilimoria. A geometric optimization approach to aircraft conflict resolution. In *18th Applied aerodynamics conference*, page 4265, 2000.
- Karl Bilimoria, Banavar Sridhar, and Gano Chatterji. Effects of conflict resolution maneuvers and traffic density on free flight. In *Guidance, navigation, and control conference*, page 3767, 1996.
- Pierre Bonami, Domenico Salvagnin, and Andrea Tramontani. Implementing automatic Benders decomposition in a modern MIP solver. In *International conference on integer programming and combinatorial optimization*, pages 78–90. Springer, 2020.
- Deidre Bonini, Carole Dupré, and Géraud Granger. How erasmus can support an increase in capacity in 2020. In *Proceedings of the 7th International Conference on Computing, Communications and Control Technologies: CCCT*, page 6. Citeseer, 2009.
- Dietrich Braess. Über ein Paradoxon aus der Verkehrsplanung. *Unternehmensforschung*, 12(1):258–268, 1968.
- Luce Brotcorne, Martine Labbé, Patrice Marcotte, and Gilles Savard. Joint design and pricing on a network. *Operations research*, 56(5):1104–1115, 2008.
- Lucy Budd and Stephen Ison. Responsible transport: A post-covid agenda for transport policy and practice. *Transportation Research Interdisciplinary Perspectives*, 6:100151, 2020.
- Sonia Cafieri. Aircraft conflict avoidance: a mixed-integer nonlinear optimization approach. In *GOW 2012, Global Optimization Workshop 2012*, pages pp–43, 2012.
- Sonia Cafieri and Riadh Omheni. Mixed-integer nonlinear programming for aircraft conflict avoidance by sequentially applying velocity and heading angle changes. *European Journal of Operational Research*, 260(1):283–290, 2017.

- 
- Sonia Cafieri and David Rey. Maximizing the number of conflict-free aircraft using mixed-integer nonlinear programming. *Computers & Operations Research*, 80:147–158, 2017.
- Dustin Carlino, Stephen D Boyles, and Peter Stone. Auction-based autonomous intersection management. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pages 529–534. IEEE, 2013.
- Lorenzo Castelli, Raffaele Pesenti, and Andrea Ranieri. The design of a market mechanism to allocate air traffic flow management slots. *Transportation research part C: Emerging technologies*, 19(5):931–943, 2011.
- Andrea Censi, Saverio Bolognani, Julian G Zilly, Shima Sadat Mousavi, and Emilio Frazzoli. Today me, tomorrow thee: Efficient resource allocation in competitive settings using karma games. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 686–693. IEEE, 2019.
- Martina Cerulli, Claudia d’Ambrosio, Leo Liberti, and Mercedes Pelegrín. Detecting and solving aircraft conflicts using bilevel programming. *Journal of Global Optimization*, 81(2):529–557, 2021.
- Shantanu Chakraborty, David Rey, Michael W Levin, and S Travis Waller. Freeway network design with exclusive lanes for automated vehicles under endogenous mobility demand. *Transportation Research Part C: Emerging Technologies*, 133:103440, 2021.
- Yu-Yuan Chang, Omar B Sawaya, and Athanasios K Ziliaskopoulos. A tabu search based approach for work zone scheduling. In *Proceeding of the Transportation Research Board 80th Annual Meeting, Washington, DC*, 2001.
- Lei Chen and Cristofer Englund. Cooperative intersection management: a survey. *IEEE Transactions on Intelligent Transportation Systems*, 17(2):570–586, 2016.
- Nan Chen, Lauren Gardner, and David Rey. Bilevel optimization model for the development of real-time strategies to minimize epidemic spreading risk in air traffic networks. *Transportation Research Record*, 2569(1):62–69, 2016.
- Nathan Chen, David Rey, and Lauren Gardner. Multiscale network model for evaluating global outbreak control strategies. *Transportation Research Record*, 2626(1):42–50, 2017.
- Rongsheng Chen, Jeffrey Hu, Michael W Levin, and David Rey. Stability-based analysis of autonomous intersection management with pedestrians. *Transportation Research Part C: Emerging Technologies*, 114:463–483, 2020.
- Yi-Jen Chiang, James T Klosowski, Changkil Lee, and Joseph SB Mitchell. Geometric algorithms for conflict detection/resolution in air traffic management. In *Decision and Control, 1997., Proceedings of the 36th IEEE Conference on*, volume 2, pages 1835–1840. IEEE, 1997.
- James C Chu and Yin-Jay Chen. Optimal threshold-based network-level transportation infrastructure life-cycle management with heterogeneous maintenance actions. *Transportation Research Part B: Methodological*, 46(9):1123–1143, 2012.
- Edward H Clarke. Multipart pricing of public goods. *Public Choice*, pages 17–33, 1971.

- Carleton Coffrin, Hassan Hijazi, and Pascal Van Hentenryck. The qc relaxation: A theoretical and computational study on optimal power flow. *IEEE Transactions on Power Systems*, 31(4):3008–3018, 2015.
- IBM ILOG CPLEX. V12. 1: Users manual for CPLEX. *International Business Machines Corporation*, 46(53):157, 2009.
- IBM ILOG CPLEX. v12.6: User’s Manual for CPLEX, 2014.
- Stella Dafermos. Traffic equilibrium and variational inequalities. *Transportation Science*, 14(1):42–54, 1980.
- Veronica Dal Sasso, Franklin Djeumou Fomeni, Guglielmo Lulli, and Konstantinos G Zografos. Planning efficient 4d trajectories in air traffic flow management. *European Journal of Operational Research*, 276(2):676–687, 2019.
- Arnaud De La Fortelle and Xiangjun Qian. Autonomous driving at intersections: combining theoretical analysis with practical considerations. In *ITS World Congress*, 2015.
- Stephan Dempe and Joydeep Dutta. Is bilevel programming a special case of a mathematical program with complementarity constraints? *Mathematical programming*, 131:37–48, 2012.
- Fernando HC Dias and David Rey. Robust aircraft conflict resolution under trajectory prediction uncertainty. *Operations Research Letters*, 50(5):503–508, 2022.
- Fernando HC Dias, Hassan Hijazi, and David Rey. Disjunctive linear separation conditions and mixed-integer formulations for aircraft conflict resolution. *European Journal of Operational Research*, 296(2):520–538, 2022.
- Robert J Dolan. Incentive mechanisms for priority queuing problems. *The Bell Journal of Economics*, pages 421–436, 1978.
- Xiaotong Dong, David Rey, and S Travis Waller. Dial-a-ride problem with users’ accept/reject decisions based on service utilities. *Transportation Research Record*, 2674(10):55–67, 2020.
- Xiaotong Dong, Joseph YJ Chow, S Travis Waller, and David Rey. A chance-constrained dial-a-ride problem with utility-maximising demand and multiple pricing structures. *Transportation Research Part E: Logistics and Transportation Review*, 158:102601, 2022.
- Kurt Dresner and Peter Stone. Multiagent traffic management: A reservation-based intersection control mechanism. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 530–537. IEEE Computer Society, 2004.
- Kurt Dresner and Peter Stone. Traffic intersections of the future. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, page 1593. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.
- Kurt Dresner and Peter Stone. A multiagent approach to autonomous intersection management. *Journal of Artificial Intelligence Research*, 31:591–656, 2008.
- Nicolas Durand and Jean-Marc Alliot. Ant colony optimization for air traffic conflict resolution. In *ATM Seminar 2009, 8th USA/Europe Air Traffic Management Research and Development Seminar*, 2009.

- 
- Nicolas Durand, Jean-Marc Alliot, and Olivier Chansou. Optimal resolution of en route conflicts. In *Proceedings of the 1st USA/Europe Seminar*, 1997.
- Ohad Eisenhandler and Michal Tzur. The humanitarian pickup and distribution problem. *Operations Research*, 67(1):10–32, 2019.
- David Fajardo, Tsz-Chiu Au, S Waller, Peter Stone, and David Yang. Automated intersection control: Performance of future innovation versus current traffic signal control. *Transportation Research Record: Journal of the Transportation Research Board*, (2259):223–232, 2011.
- M Fampa, LA Barroso, D Candal, and Luidi Simonetti. Bilevel optimization applied to strategic pricing in competitive electricity markets. *Computational Optimization and Applications*, 39(2):121–142, 2008.
- Hamid Farvaresh and Mohammad Mehdi Sepehri. A single-level mixed integer linear formulation for a bi-level discrete network design problem. *Transportation Research Part E: Logistics and Transportation Review*, 47(5):623–640, 2011.
- Hamid Farvaresh and Mohammad Mehdi Sepehri. A branch and bound algorithm for bi-level discrete network design problem. *Networks and Spatial Economics*, 13(1):67–106, 2013.
- Martina Fischetti and Matteo Fischetti. Matheuristics. In *Handbook of heuristics*, pages 121–153. Springer, 2018.
- Matteo Fischetti, Ivana Ljubić, Michele Monaci, and Markus Sinnl. A new general-purpose algorithm for mixed-integer bilevel linear programs. *Operations Research*, 65(6):1615–1637, 2017.
- Pirmin Fontaine and Stefan Minner. Benders decomposition for discrete–continuous linear bilevel problems with application to traffic network design. *Transportation Research Part B: Methodological*, 70:163–172, 2014.
- Pirmin Fontaine and Stefan Minner. A dynamic discrete network design problem for maintenance planning in traffic networks. *Annals of Operations Research*, 253(2):757–772, 2017.
- Emilio Frazzoli, Z-H Mao, J-H Oh, and Eric Feron. Resolution of conflicts involving many aircraft via semidefinite programming. *Journal of Guidance, Control, and Dynamics*, 24(1):79–86, 2001.
- Ziyou Gao, Jianjun Wu, and Huijun Sun. Solution algorithm for the bi-level discrete network design problem. *Transportation Research Part B: Methodological*, 39(6):479–495, 2005.
- Lauren M Gardner, David Rey, Anita E Heywood, Renin Toms, James Wood, S Travis Waller, and C Raina MacIntyre. A scenario-based evaluation of the middle east respiratory syndrome coronavirus and the hajj. *Risk Analysis*, 34(8):1391–1400, 2014.
- Jean Gregoire, Emilio Frazzoli, Arnaud de La Fortelle, and Tichakorn Wongpiromsarn. Back-pressure traffic signal control with unknown routing rates. *IFAC Proceedings Volumes*, 47(3):11332–11337, 2014.
- Theodore Groves and Martin Loeb. Incentives and public inputs. *Journal of Public economics*, 4(3):211–226, 1975.
- Erik Halvorson, Vincent Conitzer, and Ronald Parr. Multi-step multi-sensor hider-seeker games. In *Proceedings of the 21st international joint conference on Artificial intelligence (IJCAI)*, pages 159–166. Morgan Kaufmann Publishers Inc., 2009.

- Ahmed WA Hammad, Ali Akbarnezhad, and David Rey. A multi-objective mixed integer nonlinear programming model for construction site layout planning to minimise noise pollution and transport costs. *Automation in Construction*, 61:73–85, 2016a.
- Ahmed WA Hammad, Ali Akbarnezhad, David Rey, and S Travis Waller. A computational method for estimating travel frequencies in site layout planning. *Journal of Construction Engineering and Management*, 142(5):04015102, 2016b.
- Ahmed WA Hammad, Ali Akbarnezhad, and David Rey. Bilevel mixed-integer linear programming model for solving the single airport location problem. *Journal of Computing in Civil Engineering*, 31(5):06017001, 2017a.
- Ahmed WA Hammad, Ali Akbarnezhad, and David Rey. Sustainable urban facility location: Minimising noise pollution and network congestion. *Transportation Research Part E: Logistics and Transportation Review*, 107:38–59, 2017b.
- Ahmed WA Hammad, David Rey, and Ali Akbarnezhad. A cutting plane algorithm for the site layout planning problem with travel barriers. *Computers & Operations Research*, 82: 36–51, 2017c.
- Deren Han and Hong K Lo. Solving non-additive traffic assignment problems: a descent method for co-coercive variational inequalities. *European Journal of Operational Research*, 159(3):529–544, 2004.
- Elise Henry, Angelo Furno, Nour-Eddin El Faouzi, and David Rey. Locating park-and-ride facilities for resilient on-demand urban mobility. *Transportation Research Part E: Logistics and Transportation Review*, 158:102557, 2022.
- Hassan Hijazi, Pierre Bonami, and Adam Ouorou. A Note on Linear On/Off Constraints. *NICTA Technical Report*, April 2014. URL [http://www.optimization-online.org/DB\\_FILE/2014/04/4309.pdf](http://www.optimization-online.org/DB_FILE/2014/04/4309.pdf).
- Hassan Hijazi, Carleton Coffrin, and Pascal Van Hentenryck. Convex quadratic relaxations for mixed-integer nonlinear programs in power systems. *Mathematical Programming Computation*, 9(3):321–367, 2017.
- Taha Hossein Rashidi, David Rey, Sisi Jian, and Travis Waller. A clustering algorithm for bi-criteria stop location design with elastic demand. *Computer-Aided Civil and Infrastructure Engineering*, 31(2):117–131, 2016.
- Jianghai Hu, Maria Prandini, and Shankar Sastry. Optimal coordinated maneuvers for three-dimensional aircraft conflict resolution. *Journal of Guidance, Control, and Dynamics*, 25(5):888–900, 2002.
- Annex ICAO. The convention on international civil aviation. *Montreal, Canada*, 1, 2010.
- Manish Jain, Erim Kardeş, Christopher Kiekintveld, Milind Tambe, and Fernando Ordóñez. Security games with arbitrary schedules: a branch and price approach. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, pages 792–797. AAAI Press, 2010.
- Sisi Jian, David Rey, and Vinayak V Dixit. Dynamic optimal vehicle relocation in carshare systems. *Transportation Research Record*, 2567(1):1–9, 2016.

- 
- Sisi Jian, David Rey, and Vinayak V Dixit. An integrated supply-demand approach to solving optimal relocations in station-based carsharing systems. *Networks and Spatial Economics*, 19(2):611–632, 2019.
- Mor Kaspi, Tal Raviv, and Michal Tzur. Parking reservation policies in one-way vehicle sharing systems. *Transportation Research Part B: Methodological*, 62:35–50, 2014.
- Thomas Kleinert, Martine Labbé, Fränk Plein, and Martin Schmidt. There’s no free lunch: on the hardness of choosing a correct big-M in bilevel optimization. *Operations Research*, 68(6):1716–1721, 2020.
- Thomas Kleinert, Martine Labbé, Ivana Ljubić, and Martin Schmidt. A survey on mixed-integer programming techniques in bilevel optimization. *EURO Journal on Computational Optimization*, 9:100007, 2021.
- Rico Krueger, Taha H Rashidi, and John M Rose. Preferences for shared autonomous vehicles. *Transportation research part C: emerging technologies*, 69:343–355, 2016.
- Tung Le, Péter Kovács, Neil Walton, Hai L Vu, Lachlan LH Andrew, and Serge SP Hoogendoorn. Decentralized signal control for urban road networks. *Transportation Research Part C: Emerging Technologies*, 58:431–450, 2015.
- Tung Le, Hai L Vu, Neil Walton, Serge P Hoogendoorn, Péter Kovács, and Rudesindo N Queija. Utility optimization framework for a distributed traffic control of urban road networks. *Transportation Research Part B: Methodological*, 105:539–558, 2017.
- Larry J Leblanc. An algorithm for the discrete network design problem. *Transportation Science*, 9(3):183–199, 1975.
- Hsin-Yun Lee. Optimizing schedule for improving the traffic impact of work zone on roads. *Automation in Construction*, 18(8):1034–1044, 2009.
- Thibault Lehouillier, Jérémy Omer, François Soumis, and Guy Desautniers. Two decomposition algorithms for solving a minimum weight maximum clique model for the air conflict resolution problem. *European Journal of Operational Research*, 256(3):696–712, 2017.
- Emilio Leonardi, Marco Mellia, Fabio Neri, and M Ajmone Marsan. Bounds on average delays and queue size averages and variances in input-queued cell-based switches. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 1095–1103. IEEE, 2001.
- Michael W Levin and David Rey. Conflict-point formulation of intersection control for autonomous vehicles. *Transportation Research Part C: Emerging Technologies*, 85:528–547, 2017.
- Zhixia Li, Madhav Chitturi, Dongxi Zheng, Andrea Bill, and David Noyce. Modeling reservation-based autonomous intersection control in Vissim. *Transportation Research Record: Journal of the Transportation Research Board*, (2381):81–90, 2013.
- DianChao Lin and Saif Eddin Jabari. Pay for intersection priority: A free market mechanism for connected vehicles. *arXiv preprint arXiv:2001.01813*, 2020.
- Shaonan Liu, Mingzheng Wang, Nan Kong, and Xiangpei Hu. An enhanced branch-and-bound algorithm for bilevel integer linear programming. *European Journal of Operational Research*, 291(2):661–679, 2021.

- Roger Lloret-Batlle and R Jayakrishnan. Envy-minimizing pareto efficient intersection control with brokered utility exchanges under user heterogeneity. *Transportation Research Part B: Methodological*, 94:22–42, 2016.
- Andrea Lodi. Mixed integer programming computation. In *50 years of integer programming 1958-2008*, pages 619–645. Springer, 2010.
- Andrea Lodi and Giulia Zarpellon. On learning and branching: a survey. *Top*, 25(2):207–236, 2017.
- Paramet Luatthep, Agachai Sumalee, William HK Lam, Zhi-Chun Li, and Hong K Lo. Global optimization method for mixed transportation network design problem: a mixed-integer linear programming approach. *Transportation Research Part B: Methodological*, 45(5): 808–827, 2011.
- Marco E Lübbecke and Jacques Desrosiers. Selected topics in column generation. *Operations Research*, 53(6):1007–1023, 2005.
- Thomas L Magnanti and Richard T Wong. Network design and transportation planning: Models and algorithms. *Transportation Science*, 18(1):1–55, 1984.
- Stephen J Maher, Tobias Fischer, Tristan Gally, Gerald Gamrath, Ambros Gleixner, Robert Lion Gottwald, Gregor Hendel, Thorsten Koch, Marco Lübbecke, Matthias Miltenberger, et al. The SCIP optimization suite 4.0. 2017.
- Andreas A Malikopoulos, Christos G Cassandras, and Yue J Zhang. A decentralized energy-optimal control framework for connected automated vehicles at signal-free intersections. *Automatica*, 93:244–256, 2018.
- Garth P McCormick. Computability of global solutions to factorable nonconvex programs: Part I—convex underestimating problems. *Mathematical Programming*, 10(1):147–175, 1976.
- Amir Mirheli, Mehrdad Tajalli, Leila Hajibabai, and Ali Hajbabaie. A consensus-based distributed trajectory control in a signal-free intersection. *Transportation Research Part C: Emerging Technologies*, 100:161–176, 2019.
- Julien Monteil. *Investigating the effects of cooperative vehicles on highway traffic flow homogenization: analytical and simulation studies*. PhD thesis, Université de Lyon, 2014.
- Matteo Muratori. Impact of uncoordinated plug-in electric vehicle charging on residential power demand. *Nature Energy*, 3(3):193–201, 2018.
- Divya J Nair, Hanna Grzybowska, David Rey, and Vinayak V Dixit. Food rescue and delivery: Heuristic algorithm for periodic unpaired pickup and delivery vehicle routing problem. *Transportation Research Record*, 2548(1):81–89, 2016.
- Divya J Nair, David Rey, and Vinayak V Dixit. Fair allocation and cost-effective routing models for food rescue and redistribution. *IIEE Transactions*, 49(12):1172–1188, 2017.
- Ali Najmi, David Rey, and Taha H Rashidi. Novel dynamic formulations for real-time ride-sharing systems. *Transportation Research Part E: Logistics and Transportation Review*, 108:122–140, 2017.

- 
- Gordon F Newell. A simplified theory of kinematic waves in highway traffic, part I: General theory. *Transportation Research Part B: Methodological*, 27(4):281–287, 1993.
- ManWo Ng, Dung Ying Lin, and S Travis Waller. Optimal long-term infrastructure maintenance planning accounting for traffic dynamics. *Computer-Aided Civil and Infrastructure Engineering*, 24(7):459–469, 2009.
- Jérémy Omer. A space-discretized mixed-integer linear model for air-conflict resolution with speed and heading maneuvers. *Computers & Operations Research*, 58:75–86, 2015.
- Jérémy Omer and Jean-Loup Farges. Hybridization of nonlinear and mixed-integer linear programming for aircraft separation with trajectory recovery. *IEEE Transactions on Intelligent Transportation Systems*, 14(3):1218–1230, 2013.
- Lucia Pallottino, Eric M Feron, and Antonio Bicchi. Conflict resolution problems for air traffic management systems solved with mixed integer programming. *IEEE transactions on intelligent transportation systems*, 3(1):3–11, 2002.
- Theodoros P Pantelidis, Joseph YJ Chow, and Saeid Rasulkhani. A many-to-many assignment game and stable outcome algorithm to evaluate collaborative mobility-as-a-service platforms. *Transportation Research Part B: Methodological*, 140:79–100, 2020.
- Michael Patriksson. On the applicability and solution of bilevel optimization models in transportation science: A study on the existence, stability and computation of optimal solutions to stochastic mathematical programs with equilibrium constraints. *Transportation Research Part B: Methodological*, 42(10):843–860, 2008.
- Mercedes Pelegrín and Claudia d’Ambrosio. Aircraft deconfliction via mathematical programming: Review and insights. *Transportation Science*, 56(1):118–140, 2022.
- Guido Perboli, Luce Brotcorne, Maria Elena Bruni, and Mariangela Rosano. A new model for last-mile delivery and satellite depots management: The impact of the on-demand economy. *Transportation Research Part E: Logistics and Transportation Review*, 145:102184, 2021.
- Amalia Polydoropoulou, Ioanna Pagoni, Athena Tsirimpa, Athena Roumboutsos, Maria Karmargianni, and Ioannis Tsouros. Prototype business models for mobility-as-a-service. *Transportation Research Part A: Policy and Practice*, 131:149–162, 2020.
- Warren B Powell. A unified framework for stochastic optimization. *European Journal of Operational Research*, 275(3):795–821, 2019.
- Suchithra Rajendran and Sharan Srinivas. Air taxi service for urban mobility: A critical review of recent developments, future challenges, and opportunities. *Transportation research part E: logistics and transportation review*, 143:102090, 2020.
- Taha Hossein Rashidi, Travis Waller, and Kay Axhausen. Reduced value of time for autonomous vehicle users: Myth or reality? *Transport Policy*, 95:30–36, 2020.
- John Reif and Micha Sharir. Motion planning in the presence of moving obstacles. *Journal of the ACM (JACM)*, 41(4):764–790, 1994.
- David Rey. Computational benchmarking of exact methods for the bilevel discrete network design problem. *Transportation Research Procedia*, 47:11–18, 2020.



- David Rey and Hillel Bar-Gera. Long-term scheduling for road network disaster recovery. *International Journal of Disaster Risk Reduction*, 42:101353, 2020.
- David Rey and Hassan Hijazi. Complex number formulation and convex relaxations for aircraft conflict resolution. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 88–93. IEEE, 2017.
- David Rey and Michael W Levin. Blue phase: Optimal network traffic control for legacy and autonomous vehicles. *Transportation Research Part B: Methodological*, 130:105–129, 2019.
- David Rey, Christophe Rapine, Rémy Fondacci, and Nour-Eddin El Faouzi. Minimization of potential air conflicts through speed regulation. *Transportation research record: journal of the transportation research board*, 2300:59–67, 2012.
- David Rey, Christophe Rapine, Vinayak V Dixit, and Steven Travis Waller. Equity-oriented aircraft collision avoidance model. *IEEE Transactions on Intelligent Transportation Systems*, 16(1):172–183, 2015.
- David Rey, Vinayak V Dixit, Jean-Luc Ygnace, and S Travis Waller. An endogenous lottery-based incentive mechanism to promote off-peak usage in congested transit systems. *Transport Policy*, 46:46–55, 2016a.
- David Rey, Lauren Gardner, and S Travis Waller. Finding outbreak trees in networks with limited information. *Networks and Spatial Economics*, 16(2):687–721, 2016b.
- David Rey, Christophe Rapine, Rémy Fondacci, and Nour-Eddin El Faouzi. Subliminal speed control in air traffic management: Optimization and simulation. *Transportation Science*, 50(1):240–262, 2016c.
- David Rey, Khaled Almi’ani, and Divya J Nair. Exact and heuristic algorithms for finding envy-free allocations in food rescue pickup and delivery logistics. *Transportation Research Part E: Logistics and Transportation Review*, 112:19–46, 2018.
- David Rey, Hillel Bar-Gera, Vinayak V Dixit, and S Travis Waller. A branch-and-price algorithm for the bilevel network maintenance scheduling problem. *Transportation Science*, 53(5):1455–1478, 2019.
- David Rey, Michael W Levin, and Vinayak V Dixit. Online incentive-compatible mechanisms for traffic intersection auctions. *European Journal of Operational Research*, 293(1):229–247, 2021.
- Muhammed O Sayin, Chung-Wei Lin, Shinichi Shiraishi, Jiajun Shen, and Tamer Başar. Information-driven autonomous intersection control via incentive compatible mechanisms. *IEEE Transactions on Intelligent Transportation Systems*, 20(3):912–924, 2018.
- Heiko Schepperle and Klemens Böhm. Agent-based traffic control using auctions. In *International Workshop on Cooperative Information Agents*, pages 119–133. Springer, 2007.
- Babak Shahbazi, Ali Akbarnezhad, David Rey, Alireza Ahmadian Fard Fini, and Martin Loosemore. Optimization of job allocation in construction organizations to maximize workers’ career development opportunities. *Journal of Construction Engineering and Management*, 145(6):04019036, 2019.
- Hyelim Shin, Taesik Lee, and Hyun-Rok Lee. Skyport location problem for urban air mobility system. *Computers & Operations Research*, 138:105611, 2022.

- 
- Michael J Smith. Traffic control and route-choice; a simple example. *Transportation Research Part B: Methodological*, 13(4):289–294, 1979.
- Wayne E Smith. Various optimizers for single-stage production. *Naval Research Logistics Quarterly*, 3(1-2):59–66, 1956.
- Ninja Soeffker, Marlin W Ulmer, and Dirk C Mattfeld. Stochastic dynamic vehicle routing in the light of prescriptive analytics: A review. *European Journal of Operational Research*, 2021.
- Changle Song, Julien Monteil, Jean-Luc Ygnace, and David Rey. Incentives for ridesharing: A case study of welfare and traffic congestion. *Journal of Advanced Transportation*, 2021, 2021.
- Adil Tahir, Frédéric Quesnel, Guy Desaulniers, Issmail El Hallaoui, and Yassine Yaakoubi. An improved integral column generation algorithm using machine learning for aircrew pairing. *Transportation Science*, 55(6):1411–1429, 2021.
- Leandros Tassioulas and Anthony Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *Automatic Control, IEEE Transactions on*, 37(12):1936–1948, 1992.
- Víctor Valls, Julien Monteil, and Mélanie Bourouche. A convex optimisation approach to traffic signal control. In *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*, pages 1508–1515. IEEE, 2016.
- Charlie Vanaret, David Gianazza, Nicolas Durand, and Jean-Baptiste Gotteland. Benchmarking conflict resolution algorithms. In *ICRAT 2012 Fifth International Conference on Research in Air Transportation*, 2012.
- Pravin Varaiya. Max pressure control of a network of signalized intersections. *Transportation Research Part C: Emerging Technologies*, 36:177–195, 2013.
- Matteo Vasirani and Sascha Ossowski. A market-inspired approach for intersection management in urban road traffic networks. *Journal of Artificial Intelligence Research*, 43: 621–659, 2012.
- Mohammad M Vazifeh, Paolo Santi, Giovanni Resta, Steven H Strogatz, and Carlo Ratti. Addressing the minimum fleet problem in on-demand urban mobility. *Nature*, 557(7706): 534–538, 2018.
- Adan Vela, Senay Solak, William Singhose, and John-Paul Clarke. A mixed integer program for flight-level assignment and speed control for conflict resolution. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 5219–5226. IEEE, 2009a.
- Adan E Vela, Erwan Salaün, Senay Solak, and Eric Feron. A two-stage stochastic optimization model for air traffic conflict resolution under wind uncertainty. In *Digital Avionics Systems Conference, 2009. DASC’09. IEEE/AIAA 28th*, pages 2–E. IEEE, 2009b.
- Adan E Vela, Senay Solak, Eric Feron, Karen Feigh, William Singhose, and John-Paul Clarke. A fuel optimal and reduced controller workload optimization model for conflict resolution. In *Digital Avionics Systems Conference, 2009. DASC’09. IEEE/AIAA 28th*, pages 3–C. IEEE, 2009c.

- William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of Finance*, 16(1):8–37, 1961.
- Stacy A Voccia, Ann Melissa Campbell, and Barrett W Thomas. The same-day delivery problem for online purchases. *Transportation Science*, 53(1):167–184, 2019.
- Andreas Wächter and Lorenz T Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1):25–57, 2006.
- David ZW Wang, Haoxiang Liu, and Wai Y Szeto. A novel discrete network design problem formulation and its global optimization solution algorithm. *Transportation Research Part E: Logistics and Transportation Review*, 79:213–230, 2015.
- Shuaian Wang, Qiang Meng, and Hai Yang. Global optimization methods for the discrete network design problem. *Transportation Research Part B: Methodological*, 50:42–60, 2013.
- John Glen Wardrop. Some theoretical aspects of road traffic research. In *Inst Civil Engineers Proc London/UK/*, 1952.
- Tat Wang Wong, Neeraj Saxena, and Vinayak V Dixit. A study of route choice behavior of drivers in autonomous vehicles. In *Transportation Research Board 97th Annual Meeting*. Transportation Research Board, 2018.
- Tichakorn Wongpiromsarn, Tawit Uthaicharoenpong, Yu Wang, Emilio Frazzoli, and Danwei Wang. Distributed traffic signal control for maximum network throughput. In *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, pages 588–595. IEEE, 2012.
- Yuanyuan Wu, Haipeng Chen, and Feng Zhu. Dcl-aim: Decentralized coordination learning of autonomous intersection management for connected and automated vehicles. *Transportation Research Part C: Emerging Technologies*, 103:246–260, 2019.
- Haoning Xi, Didier Aussel, Wei Liu, S Travis Waller, and David Rey. Single-leader multi-follower games for the regulation of two-sided mobility-as-a-service markets. *European Journal of Operational Research*, 2022.
- Nan Xiao, Emilio Frazzoli, Yitong Li, Yu Wang, and Danwei Wang. Pressure releasing policy in traffic signal control with finite queue capacities. In *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, pages 6492–6497. IEEE, 2014.
- Zhitao Xiong, David Rey, Vinayak V Dixit, and S Travis Waller. An algorithmic framework for the scheduling of construction projects based on ant colony optimization and expert knowledge. In *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, pages 2446–2452. IEEE, 2014.
- Isaak Yperman, Steven Logghe, and Ben Immers. The link transmission model: An efficient implementation of the kinematic wave theory in traffic networks. In *Proceedings of the 10th EWGT Meeting, Poznan, Poland*, 2005.
- Ali A Zaidi, Balázs Kulcsár, and Henk Wymersch. Back-pressure traffic signal control with fixed and adaptive routing for urban vehicular networks. *IEEE Transactions on Intelligent Transportation Systems*, 17(8):2134–2143, 2016.

- 
- Bo Zeng. A practical scheme to compute the pessimistic bilevel optimization problem. *INFORMS Journal on Computing*, 32(4):1128–1142, 2020.
- Xiang Zhang, David Rey, and S Travis Waller. Multitype recharge facility location for electric vehicles. *Computer-Aided Civil and Infrastructure Engineering*, 33(11):943–965, 2018.
- Xiang Zhang, David Rey, S Travis Waller, and Nathan Chen. Range-constrained traffic assignment with multi-modal recharge for electric vehicles. *Networks and Spatial Economics*, 19(2):633–668, 2019a.
- Xiang Zhang, S Travis Waller, David Rey, and Melissa Duell. Integrating uncertainty considerations into multi-objective transportation network design projects accounting for environment disruption. *Transportation Letters*, 11(7):351–361, 2019b.
- Yue Zhang, Andreas A Malikopoulos, and Christos G Cassandras. Decentralized optimal control for connected automated vehicles at intersections including left and right turns. In *Decision and Control (CDC), 2017 IEEE 56th Annual Conference on*, pages 4428–4433. IEEE, 2017.
- Yue J Zhang, Andreas A Malikopoulos, and Christos G Cassandras. Optimal control and coordination of connected and automated vehicles at urban traffic intersections. In *American Control Conference (ACC), 2016*, pages 6227–6232. IEEE, 2016.
- Xianbo Zhao, Yongjian Ke, Jian Zuo, Wei Xiong, and Peng Wu. Evaluation of sustainable transport research in 2000–2019. *Journal of Cleaner Production*, 256:120404, 2020.
- Hong Zheng, Chengdong Cai, Eric Nava, V Dixit, Yi-Chang Chiu, Essam Radwan, and Dane Ismart. Optimization of renewal-based project scheduling in an urban network. In *91st Annual Meeting of the Transportation Research Board, Washington DC*, pages 1–20, 2012.
- Feng Zhu and Satish V Ukkusuri. A linear programming formulation for autonomous intersection control within a dynamic traffic assignment and connected vehicle environment. *Transportation Research Part C: Emerging Technologies*, 55:363–378, 2015.
- Aleksa Zlojutro, David Rey, and Lauren Gardner. A decision-support framework to optimize border control for global outbreak mitigation. *Scientific Reports*, 9(1):1–14, 2019.