Université de Montréal

Data-driven optimization of bus schedules under uncertainties

par

Léa Ricard

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Thèse présentée en vue de l'obtention du grade de
Philosophiæ Doctor (Ph.D.)
en Informatique

5 juin 2023

Université de Montréal

Faculté des arts et des sciences

Cette thèse intitulée

Data-driven optimization of bus schedules under uncertainties

présentée par

Léa Ricard

a été évaluée par un jury composé des personnes suivantes :

Margarida Carvalho
(président-rapporteur)

Andrea Lodi
(directeur de recherche)

Guy Desaulniers
(codirecteur)

Louis-Martin Rousseau
(codirecteur)

Michel Gendreau
(membre du jury)

Michel Bierlaire
(examinateur externe)

Marine Carrasco
(représentant du doyen de la FESP)

Dedication

To Mathilde

Résumé

Plusieurs sous-problèmes d'optimisation se posent lors de la planification des transports publics. Le problème d'itinéraires de véhicule (PIV) est l'un d'entre eux et consiste à minimiser les coûts opérationnels tout en assignant exactement un autobus par trajet planifié de sorte que le nombre d'autobus entreposé par dépôt ne dépasse pas la capacité maximale disponible. Bien que les transports publics soient sujets à plusieurs sources d'incertitude (à la fois endogènes et exogènes) pouvant engendrer des variations des temps de trajet et de la consommation d'énergie, le PIV et ses variantes sont la plupart du temps résolus de façon déterministe pour des raisons de résolubilité. Toutefois, cette hypothèse peut compromettre le respect de l'horaire établi lorsque les temps des trajets considérés sont fixes (c.-à-d. déterministes) et peut produire des solutions impliquant des politiques de gestion des batteries inadéquates lorsque la consommation d'énergie est aussi considérée comme fixe. Dans cette thèse, nous proposons une méthodologie pour mesurer la fiabilité (ou le respect de l'horaire établi) d'un service de transport public ainsi que des modèles mathématiques stochastiques et orientés données et des algorithmes de *branch-and-price* pour deux variantes de ce problème, à savoir le problème d'itinéraires de véhicule avec dépôts multiples (PIVDM) et le problème d'itinéraires de véhicule électrique (PIV-E).

Afin d'évaluer la fiabilité, c.-à-d. la tolérance aux délais, de certains itinéraires de véhicule, nous prédisons d'abord la distribution des temps de trajet des autobus. Pour ce faire, nous comparons plusieurs modèles probabilistes selon leur capacité à prédire correctement la fonction de densité des temps de trajet des autobus sur le long terme. Ensuite, nous estimons à l'aide d'une simulation de Monte-Carlo la fiabilité des horaires d'autobus en générant des temps de trajet aléatoires à chaque itération. Nous intégrons alors le modèle probabiliste le plus approprié, celui qui est capable de prédire avec précision à la fois la véritable fonction de densité conditionnelle des temps de trajet et les retards secondaires espérés, dans nos modèles d'optimisation basés sur les données.

Deuxièmement, nous introduisons un modèle pour PIVDM fiable avec des temps de trajet stochastiques. Ce problème d'optimisation bi-objectif vise à minimiser les coûts opérationnels et les pénalités associées aux retards. Un algorithme heuristique basé sur la génération de colonnes avec des sous-problèmes stochastiques est proposé pour résoudre ce problème. Cet

algorithme calcule de manière dynamique les retards secondaires espérés à mesure que de nouvelles colonnes sont générées.

Troisièmement, nous proposons un nouveau programme stochastique à deux étapes avec recours pour le PIVDM électrique avec des temps de trajet et des consommations d'énergie stochastiques. La politique de recours est conçue pour rétablir la faisabilité énergétique lorsque les itinéraires de véhicule produits a priori se révèlent non réalisables. Toutefois, cette flexibilité vient au prix de potentiels retards induits. Une adaptation d'un algorithme de branch-and-price est développé pour évaluer la pertinence de cette approche pour deux types d'autobus électriques à batterie disponibles sur le marché.

Enfin, nous présentons un premier modèle stochastique pour le PIV-E avec dégradation de la batterie. Le modèle sous contrainte en probabilité proposé tient compte de l'incertitude de la consommation d'énergie, permettant ainsi un contrôle efficace de la dégradation de la batterie grâce au contrôle effectif de l'état de charge (EdC) moyen et l'écart de EdC. Ce modèle, combiné à l'algorithme de *branch-and-price*, sert d'outil pour balancer les coûts opérationnels et la dégradation de la batterie.

Mots-clés : Apprentissage statistique, géneration de colonnes, optimisation stochastique, programmation en nombres entiers, problème d'itinéraires de véhicule, transport public

Abstract

The vehicle scheduling problem (VSP) is one of the sub-problems of public transport planning. It aims to minimize operational costs while assigning exactly one bus per timetabled trip and respecting the capacity of each depot. Even thought public transport planning is subject to various endogenous and exogenous causes of uncertainty, notably affecting travel time and energy consumption, the VSP and its variants are usually solved deterministically to address tractability issues. However, considering deterministic travel time in the VSP can compromise schedule adherence, whereas considering deterministic energy consumption in the electric VSP (E-VSP) may result in solutions with inadequate battery management. In this thesis, we propose a methodology for measuring the reliability (or schedule adherence) of public transport, along with stochastic and data-driven mathematical models and branch-and-price algorithms for two variations of this problem, namely the multi-depot vehicle scheduling problem (MDVSP) and the E-VSP.

To assess the reliability of vehicle schedules in terms of their tolerance to delays, we first predict the distribution of bus travel times. We compare numerous probabilistic models for the long-term prediction of bus travel time density. Using a Monte Carlo simulation, we then estimate the reliability of bus schedules by generating random travel times at each iteration. Subsequently, we integrate the most suitable probabilistic model, capable of accurately predicting both the true conditional density function of the travel time and the expected secondary delays, into the data-driven optimization models.

Second, we introduce a model for the reliable MDVSP with stochastic travel time minimizing both the operational costs and penalties associated with delays. To effectively tackle this problem, we propose a heuristic column generation-based algorithm, which incorporates stochastic pricing problems. This algorithm dynamically computes the expected secondary delays as new columns are generated.

Third, we propose a new two-stage stochastic program with recourse for the electric MDVSP with stochastic travel time and energy consumption. The recourse policy aims to restore energy feasibility when a priori vehicle schedules are unfeasible, which may lead to delays. An adapted algorithm based on column generation is developed to assess the relevance of this approach for two types of commercially available battery electric buses.

Finally, we present the first stochastic model for the E-VSP with battery degradation. The proposed chance-constraint model incorporates energy consumption uncertainty, allowing for effective control of battery degradation by regulating the average state-of-charge (SOC) and SoC deviation in each discharging and charging cycle. This model, in combination with a tailored branch-and-price algorithm, serves as a tool to strike a balance between operational costs and battery degradation.

Keywords: Column generation, integer programming, public transport, statistical learning, stochastic programming, vehicle scheduling problem

Contents

List of Tables

List of Figures

List of acronyms and abbreviations

AEC	Accumulated energy consumption
APTS	Advanced public transportation systems
A-S-I	Avoid-Shift-Improve
BEB	Battery electric bus
CDF	Cumulative distribution function
DoD	Depth-of-discharge
DTW	Departure time window
EoL	End of life
E-VSP	Electric vehicle scheduling problem
MDEVSP	Multi-depot electric vehicle scheduling problem
MDVSP	Multi-depot vehicle scheduling problem

MILP	Mixed integer linear programming
MIP	Mixed-integer programming
MSE	Mean squared error
NLL	Negative log-likelihood
PDF	Probability density function
PDTT	Long term prediction of the density of the travel time
PMF	Probability mass function
R-MDVSP-STT	Reliable multi-depot vehicle scheduling problem with stochastic travel time
SDVSP	Single-depot vehicle scheduling problem
SoC	State-of-charge
TT	Travel time
VSP	Vehicle scheduling problem

Acknowledgements

First, I would like to thank my supervisors, Andrea Lodi, Guy Desaulniers, and Louis-Martin Rousseau, for their intellectual rigor and for the numerous academic opportunities they have provided me over the years. Thank you for introducing me to academia and sharing your passion for operations research.

To all the team in the optimization algorithms at GIRO Inc., particularly Charles Fleurent, Véronique Bouffard, Loïc Bodart, and Sofiane Soualah, thank you for your valuable insights and willingness to collaborate on academic research projects.

This thesis would probably not have been completed in this timeframe without the help of François Lessard. I am grateful for your hard work and commitment. Collaborating with you has always been a pleasure.

I would also like to thank Khalid, Mehdi, Pierre, and all the staff at CERC DS4DM, GERAD, CIRRELT, and Hanalog. Your assistance has been more than helpful, ranging from bringing a smile to my face when I was having a rough day to helping me solve urgent bugs. To all the fantastic colleagues and friends I met at André-Aisenstadt—Jiaqi, Federico, Matteo, Defeng, Didier, Youssouf, Alexandre, and Claudia—thank you. My journey as a Ph.D. student would not have been the same without you. I will always cherish our conversations, countless barbecues, and laughter.

To my long-time friends—Maude, Samuel, Anne-Laurie, and Catherine—I am grateful for your unwavering friendship and support.

Mom, Dad, and Camille, thank you for your endless support, good advice, and efforts in distracting me from work when I needed it most.

Last but certainly not least, I want to express my deepest gratitude to my beloved Mathilde. Thank you for continuously pushing me beyond my comfort zone and being a comforting figure, all at once. Your support, especially during the final moments, was truly invaluable.

Chapter 1

Introduction

"Essentially, all models are wrong, but some are useful."
- George Box

Public transport planning is complex and subject to various endogenous and exogenous causes of uncertainty. This thesis proposes efficient data-driven methods to consider the inherent stochasticity in these problems and help filling part of the gap between simplified mathematical modeling of public transit systems and the multifaceted reality. Our end goal is to build *useful* models to find operational plans and schedules that are more desirable (i.e., optimal) in real-world settings.

In this section, we first present the crucial role that public transport and its electrification could and should play in the shift from our car-centric cities to greener and more equitable ones. Then, we give an overview of the principal sub-problems of the public transport planning process and lay out several frameworks used in the literature to deal with uncertainties in these types of problems. Subsequently, we describe the research objectives and contributions of the main chapters before concluding with an overview of the thesis content.

1.1. Motivation

Movement and mobility are integral to the human condition, shaping our societies and economies. Throughout history, the movement of people through immigration has played a pivotal role in this respect, from the earliest waves of globalization marked by colonization and the exploitation of exotic resources (Cox, 2013). In today's globalized epoch, the movement of people and goods has reached unprecedented levels. This surge in travel has significantly exacerbated the climate crisis we face today. Indeed, most of these travels heavily rely on motorized modes of transportation, including private cars, public transportation, and airplanes, which collectively contribute to a large share of the greenhouse gases (GHGs)

emitted globally. In Canada, the transport of people accounted for 12% of the nation's total emissions in 2020, with an overwhelming 92% of these emissions stemming from passenger cars and light trucks (Environment and Climate Change Canada, 2022). Indeed, automobiles have one of the largest ratio of GHGs per passenger kilometer among all modes of transportation and are the most widespread mean of transport in Western countries. This behavioral pattern, commonly referred to as the *automobile dependence*, is exacerbated by what John Adams [1] has called 'hypermobility'. In this vicious circle, the construction of new routes does not lead to a decrease in traffic congestion but rather to more urban sprawl, longer trips, and more energy consumption. Besides its impact on climate change, automobile dependence also negatively impacts human health (e.g., road accidents, asthma, and other pulmonary illnesses) and living conditions (e.g., noise and visual pollution and severance of communities). These consequences have, in general, a more significant impact on certain groups in our society, such as children, black people, older people, and people with disabilities, even though these groups generally have less access to cars for their daily travel needs (Ducan, 2011; Gössling, 2016). Therefore, car-centric transport systems are inequitable systems that benefit a few but place a heavy economic, social, health, and environmental burden on the community.

From a collective point of view, it is therefore desirable to promote sustainable mobility, which is commonly defined as the combination of active means of transport (e.g., walking, biking, trolleys, wheelchairs, etc.) and public transportation (e.g., buses, subways, trains, shared taxis, etc.). A widely accepted approach to structuring and fostering the transition from car-centric to greener and more equitable cities is the *Avoid-Shift-Improve* (A-S-I) framework. Figure 1.1 schematizes the levels of the A-S-I framework and provides examples of measures that can be taken at each level. The first level is to avoid or reduce the need to travel by increasing the density of existing neighborhoods, working towards degrowth policies, or managing travel demand through telecommuting or parking pricing strategies. Once as much travel is avoided, the second level consists of measures to promote the shift of the remaining trips to energy-efficient modes (i.e., active transport and transit). This shift can be achieved primarily through infrastructure development, improving the quality of public transit services, new integrated mobility services, and various economic and regulatory incentives. The last level consists in improving the energy efficiency of the remaining motorized travel (public and individual) through fuel improvements, among others.

This thesis focuses on integrating specific measures from the A-S-I framework into public transport optimization problems, encompassing network design, frequency setting, as well as vehicle and crew scheduling. Specifically, we investigate how bus service quality can be factored into the planning stage and how to improve battery management of battery electric

1. John Adams is an emeritus Professor of Geography at the University College London and a specialist in risk compensation.

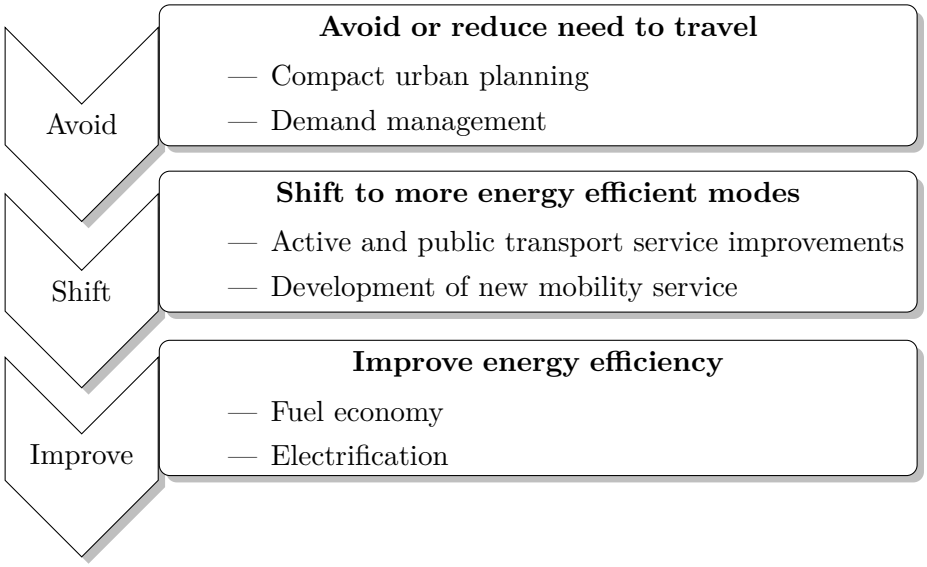


Figure 1.1 – The Avoid-Shift-Improve framework

buses (BEBs) - two measures that fall under the *shift* and the *improve* levels of the A-S-I framework, respectively.

First, the travel mode choice of individuals is highly impacted by the perceived quality of transit service, characterized by several attributes such as the level of information provided to the passengers, comfort, accessibility, and timeliness. Under identical road and public transport network infrastructure, an individual's preference for public transport over private car usage can be influenced, for example, by the consistency of travel time in public transport or the availability of sufficient seating capacity on trains, enabling the individual to comfortably engage in activities such as reading the daily newsletter. Enhancing the quality of public transport services is crucial in facilitating a modal shift from private car usage to public transport, as it plays a pivotal role in influencing individuals' travel mode choices (see e.g., Vij et al., 2017).

Second, many regions worldwide have implemented zero-emission zones within city centers to reduce local air pollution, compelling public agencies to convert their fleets to BEBs. By 2025, it is estimated that 47% of worldwide public buses will be electric (Abdelaty et al., 2021). However, while BEBs contribute to reducing local air pollution in the *Global North*, the production of batteries, the extraction of their raw materials, and their disposal pose significant environmental and social challenges. These processes contribute to pollution, destruction of natural habitats, and exploitation of populations in the *Global South* (Amnesty international, 2016; Weigl, 2021). In light of the global inequalities associated with electric batteries, it becomes clear that battery management should be addressed rationally within sustainable mobility.

We detail the main public transport planning optimization problems in what follows and highlight the adverse consequences of oversimplifying assumptions in one of these problems: vehicle scheduling. In particular, these simplifications compromise the quality of public transport services and hinder the implementation of effective battery management policies.

1.2. Background

The overall problem in transit planning involves determining how to provide passengers with the highest level of service while respecting budgetary restrictions. We define public transport's service level as both the transport service's availability (e.g., spatial, temporal, and capacity availability) and the quality (e.g., safety, reliability, average travel time, and comfort). This problem is excessively complex to solve as a whole and, instead, is usually divided into sub-problems that are solved sequentially or in a (partially) integrated manner. These problems arise during the strategic, tactical, or operational planning phases or operations (real-time control strategies) (Desaulniers et al., 2005). Figure 1.2 illustrates the public transport planning process and some sub-problems' interactions. At the strategic planning level, bus routes and networks are designed to meet passenger demand. The tactical planning level concerns the frequency setting of bus routes and the formalization of the timetables. The strategic and tactical sub-problems rely heavily on passenger route choice models to determine the demand on each route for a given network and bus route frequencies. The operational planning step involves scheduling vehicles and crews at minimum cost while respecting trip schedules, workforce and vehicle capacity at each depot, and local work regulations. The construction of crew schedules is commonly divided into two problems: duty scheduling and rostering, which involves establishing anonymous driver workdays and personalized driver schedules that are valid for several weeks, respectively. If the solution to the vehicle scheduling problem is too costly, a modified version of the frequency setting and the timetabling problems with a reduced minimum service level is solved again. Similarly, suppose the solution to the crew scheduling problem exceeds budgetary restrictions. In that case, the vehicle scheduling problem is solved again to find an alternative solution with more useful *relief points*, i.e., points along a vehicle block where driver exchange can occur. In some cases, when no alternative vehicle schedule can reduce the cost of the crew scheduling problem, backtracking to the frequency setting and timetabling problems becomes inevitable. Finally, making certain adjustments to the operational plan may be necessary during its execution. One of the most studied control strategies is vehicle holding. This strategy consists of holding buses at specific stops to improve on-time performance at subsequent stops.

This thesis addresses two variants of the vehicle scheduling problem (VSP), namely the multi-depot vehicle scheduling problem (MDVSP) and the electric vehicle scheduling problem (E-VSP). In the MDVSP, the number of depots is greater or equal to two, and each depot

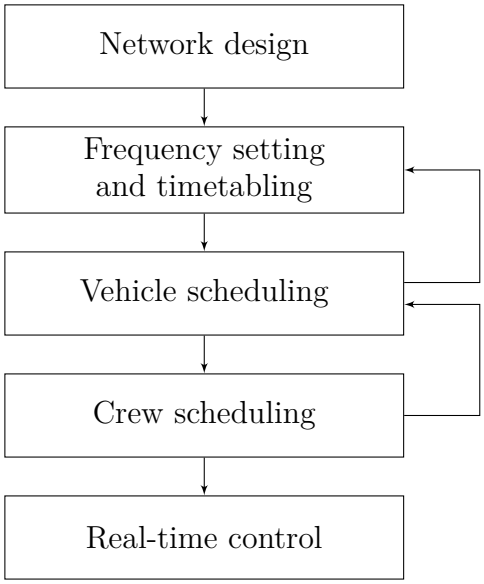


Figure 1.2 – Public transport planning sub-problems

is limited in capacity. The E-VSP considers the specific operational constraints associated with BEBs, particularly the limited driving range and longer refueling time. The MDVSP and the E-VSP has been proven to be NP-hard (Bertossi et al., 1987; Sassi and Oulamara, 2017).

Usually, the MDVSP and E-VSP assume deterministic travel time and energy consumption, which has two main consequences. On the one hand, considering travel time as deterministic may compromise schedule adherence, as travel times inevitably deviate from the planned duration (Kramkowski et al., 2009; Amberg et al., 2019). On the other hand, E-VSP with deterministic energy consumption may lead to solutions with sub-optimal true costs (including recourse costs and the cost of ownership of BEBs). First, to ensure the BEBs do not run out of energy, it is customary to incorporate a safety threshold for battery level when solving the deterministic version of the E-VSP. A large safety threshold can lead to the use of unnecessary buses, while a small safety threshold may lead to many costly recourse actions. Second, considering deterministic energy consumption in the E-VSP may result in underestimating the battery wear cost. This discrepancy arises due to the accelerated aging of lithium-ion batteries, which are commonly used in modern BEBs (Pelletier et al., 2016; Zhang et al., 2021), caused by significant variations in the state of charge (SoC) (Lam and Bauer, 2012). Typically, batteries are replaced when their remaining capacity reaches approximately 70-80% of the original capacity (Zhang et al., 2021). However, this replacement incurs substantial maintenance expenses since battery cost constitutes a significant portion of the total cost of a new BEB. Therefore, effective control of battery aging becomes crucial to ensure a reasonable cost of ownership over the long term.

Several interrelated frameworks are proposed in the literature to deal with uncertainty in mixed integer linear programs (MILPs), such as stochastic travel time and energy consumption in the MDVSP and the E-VSP. Here, we focus on frameworks that have been used in transport planning: (1) stochastic programming (see, e.g., Naumann et al., 2011; Shen et al., 2016), (2) robust optimization (see, e.g., Bie et al., 2021; Jiang et al., 2021), (3) chance-constrained programming (see, e.g., Bie et al., 2021), (4) simulation-based optimization (see, e.g., Osorio and Bierlaire, 2013), and (5) dynamic programming (see, e.g., He et al., 2018). A short review of these frameworks is conducted next to establish a clear foundation for the directions pursued in this thesis. We refer interested readers to the Chapter 2 of Powell (2022) for a comprehensive discussion on these frameworks and their similarities.

Stochastic programming. Stochastic programming is a collection of optimization methods dealing with a random objective function and potentially random constraints, where the random information follows a known distribution. This framework can deal with single-stage and multistage problems, i.e., sequential decision-making problems where information is revealed incrementally. Single-stage problems can usually be addressed with slightly modified deterministic approaches (Hannah, 2015). On the other hand, multistage problems have a structure of their own, leading us to focus on one of the most studied multistage stochastic programs in the literature, namely 2-stage problems. For more details on single-stage and multistage problems, we suggest readers refer to Hannah (2015).

A standard form of a 2-stage MILP is

$$\min_{x \in \mathcal{X}} \quad c^T x + \mathbb{E}[Q(x,\omega)] \tag{1.2.1}$$

$$\text{s.t.} \quad Ax \geq b \tag{1.2.2}$$

$$x_i \in \mathbb{Z}, \qquad \forall i \in I_1 \tag{1.2.3}$$

where

$$Q(x,\omega) := \min_{y \in \mathcal{Y}(\omega)} \quad f^T y \tag{1.2.4}$$

$$\text{s.t.} \quad Dy = e - Tx \tag{1.2.5}$$

$$y_i \in \mathbb{Z}, \qquad \forall i \in I_2. \tag{1.2.6}$$

In the previous program, the variables x and y represent the first- and second-stage decisions, respectively. Also, I_1 and I_2 are index sets of the variables required to be integer. The random information ω (here parameters (f, e, T) are the actual realization of ω) is revealed after the first stage decision and before making the second stage decision.

The stochastic programming community typically represents all potential random information outcomes through a *scenario set* Ω (Powell, 2022), where each outcome ω is assigned a corresponding probability $p(\omega)$. By adopting this terminology, the MILP (1.2.1) - (1.2.6) can be expressed in terms of the scenario set as follows:

$$\min_{x \in X} c^T x + \sum_{\omega \in \Omega} p(\omega) \min_{y \in \mathcal{Y}(\omega)} f^T y. \tag{1.2.7}$$

Robust optimization. In some cases, it is not possible or practical (for computational reasons) to consider the whole probability distributions of the random information in a MILP. Sometimes, one primarily wants to ensure that a MILP solution works in the worst case scenario. Robust optimization is of interest in these situations. This framework captures random parameters by incorporating minimal information about their nature into uncertainty sets. Designing uncertainty sets that are meaningful and relevant poses a significant challenge and remains a key focus of research in the field of robust optimization. Some of the most common uncertainty sets in the literature are the box, the polyhedral, and the ellipsoidal uncertainty sets. We refer readers interested in this field of research to Ben-Tal et al. (2009).

A standard form of a robust optimization problem is

$$\begin{aligned}
\min_{x \in \mathcal{X}} \quad & c^T x & (1.2.8)\\
\text{s.t.} \quad & a_i^T x = b_i, \qquad \forall a_i \in U_{a_i}, \forall b_i \in U_{b_i}, i = 1, \dots, m & (1.2.9)\\
& x_i \in \mathbb{Z}, \qquad \forall i \in I, & (1.2.10)
\end{aligned}$$

where U_{a_i} and U_{b_i} are uncertainty sets and I is an index set of the variables required to be integer.

Chance-constrained programming. This mathematical approach models problems involving one or several probabilistic constraints. A standard form of such a problem is

$$\begin{aligned}
\min_{x \in \mathcal{X}} \quad & c^T x & (1.2.11)\\
\text{s.t.} \quad & \mathbb{P}[Ax \leq b] \geq 1 - \epsilon & (1.2.12)\\
& x_i \in \mathbb{Z}, \qquad \forall i \in I. & (1.2.13)
\end{aligned}$$

In some cases, it is possible to assume that there is no correlation between the rows of matrix A and then the decomposition of constraint (1.2.12) into m constraints is given by

$$\mathbb{P}[a_i^T x \leq b_i] \geq 1 - \epsilon, \qquad i = 1, \dots, m. \tag{1.2.14}$$

This latter formulation is called *joint* chance-constraint, whereas (1.2.11) - (1.2.13) is called *individual* chance-constraint.

Simulation-based optimization. This framework regroups techniques where a stochastic simulation, e.g., a Monte Carlo simulation, is optimized to find good decisions or strategies. These decisions or strategies are the input of the stochastic simulation, and one wants to find appropriate input such that the output of the stochastic simulation minimizes some objective function. Stochastic optimization is particularly relevant to problems where the distribution of random information is unknown or too complex to be taken into account in a stochastic optimization framework. There exists a wide range of stochastic simulations; some are expressed in algebraic form and others are black box input-output models, some have deterministic and others stochastic outputs (Amaran et al., 2016). Each type of simulation can be addressed by classes of simulation optimization algorithms such as direct-search methods (Kolda et al., 2003), stochastic gradient methods (Robbins and Monro, 1951; Kiefer and Wolfowitz, 1952), and response surface methodology or metamodel methods (Søndergaard, 2003; Barton and Meckesheimer, 2006).

Dynamic programming. Finally, dynamic programming approaches can be used to solve dynamic and stochastic problems. The input of these problems is partially or all uncertain, but some stochastic knowledge on the uncertain parameters can be exploited. Information is revealed along the way and the problem's solution can be adjusted online (Pillac et al., 2013). The dynamic dial-a-ride problem is an excellent example of a problem where demand, here customer requests, is revealed dynamically (see e.g., Bongiovanni et al., 2022). Unfortunately, this approach suffers from the curse of dimensionality, i.e., the state space grows rapidly with the problem size. Therefore, approximate dynamic programming approaches, e.g., scenario-based approaches or metaheuristics, are usually employed to address real-world cases.

1.3. Objectives and contributions

Public transport planning involves several challenging MILPs that are typically hard to solve. Simplifying assumptions are frequently employed in solving these complex problems to address tractability issues. Examples of such assumptions include treating all parameters as deterministic or assuming fixed demand. However, by doing so, even optimal solutions to these simplified problems can turn out to be sub-optimal with respect to the real world. Some assumptions are more erroneous than others, leading to solutions that further deviate from the absolute optimal solution. This thesis aims to identify some of the most erroneous assumptions and propose methods to overcome them.

The overall objective of this thesis is to leverage the power of decomposition methods, specifically column generation, and readily available APTS (Advanced Public Transportation Systems) data to relax commonly made erroneous deterministic assumptions in public

transport planning problems. We focus on the (electric) VSP, one of the most studied public transport planning sub-problems, and investigate the impact of relaxing two common assumptions in this planning problem: deterministic travel time and energy consumption.

In the first step, we consider the MDVSP with stochastic travel time, aiming to improve the reliability of selected vehicle schedules. In the second step, we address the E-VSP with stochastic energy consumption to enhance battery management. Reliable bus service and efficient battery management are vital for attracting and retaining public transport ridership, reducing the cost of electric bus ownership, and promoting a rational use of batteries. Subsequently, we outline the research objectives and contributions of these two steps.

MDVSP with stochastic travel time for more reliable service.

In order to improve the reliability (or delay-tolerance) of their service, transport agencies and authorities need decision-support tools to (1) *measure* service reliability with meaningful indicators, (2) *predict* the effect of their decisions on these indicators, (3) *optimize* over a large number of possible decisions service reliability indicators. The first two works presented in this thesis aim at providing such a toolkit; chapter 3 focuses on tasks (1) and (2) and chapter 4 on task (3).

Our first work (see Chapter 3) aims at finding good approximations of the conditional probability distribution of bus travel time. Schedulers can then use this information to *measure* and *predict* the reliability of a vehicle schedule. We compare probabilistic models for the long-term prediction of the probability density functions (PDFs) of the travel time of bus trips. A dataset of more than 41,000 trips collected over a 2-month period on board of buses from the city of Montréal is used to train and test these models. Our objective is to find the model that generates the most accurate estimations of both the PDFs of the travel time and the expected secondary delays. A Monte Carlo simulation is provided to translate travel time PDFs predictions into expected secondary delay predictions. The best probabilistic model over our dataset is used to build data-driven mathematical programs in our second and third contributions.

Our second contribution (see Chapter 4) presents a data-driven approach to the MDVSP considering travel time uncertainty. We use the PDFs of bus travel time, derived from the work in Chapter 3, to *optimize* over many possible vehicle schedules. The objective is to select schedules that exhibit cost efficiency and demonstrate tolerance towards potential delays. We introduce a stochastic integer program for the reliable MDVSP with stochastic travel time (R-MDVSP-STT). The expected cost of the R-MDVSP-STT is a weighted sum of the operational costs and a penalty for delays. Furthermore, we propose a tailored heuristic branch-and-price algorithm. The column generation algorithm produces new variables (i.e., columns) as needed by solving stochastic pricing problems. Three reliability metrics to evaluate the solutions of the R-MDVSP-STT are defined, namely the expected secondary delay per passenger, the probability that a passenger boards a delayed timetabled trip, and

the average number of timetabled trips needed to recover from secondary delays, and a Monte Carlo simulation is provided to approximate these metrics. We carry out numerical experiments on three real-life instances with up to 2,195 trips and 2-3 depots from the city of Montréal.

E-VSP with stochastic energy consumption for better battery management.

The electrification of public transit fleets raises questions about battery management and, more specifically, how to model energy feasibility constraints in a way that maximizes the utilization of BEBs. In an attempt to partially answer these questions, our third work (see Chapter 5) presents a two-stage stochastic model with recourse for the multiple depot electric vehicle scheduling problem (MDEVSP) with stochastic travel time and energy consumption. Vehicles can be partially recharged, and we consider a non-linear charging function. Our model takes advantage of the full information on the current SoC available in operation by extending planned charge time when energy consumption deviations are observed. We propose a column generation-based heuristic featuring stochastic pricing problems to solve a real-life instance from the city of Montréal, Canada. An analysis of the relevance of our approach for different commercially available BEBs is also provided.

Our fourth contribution (see Chapter 6) addresses the tradeoff between operational costs and battery degradation in the E-VSP. To that end, we introduce a novel chance-constraint model for the E-VSP with battery degradation and stochastic energy consumption. This model controls factors that accelerate battery aging, namely high average SoC and SoC deviation, in two ways. First, a parameter called the maximum allowed accumulated energy consumption (AEC) controls both the average SoC and SoC deviation. Specifically, this parameter acts as the chargers' cutoff SoC (i.e., the threshold at which chargers stop transferring energy to batteries). SoC deviation is further reduced by the proposed chance constraint, which limits the probability of overusing the battery (i.e., reaching an AEC greater than or equal to a given maximum recommended AEC). Our model also considers nonlinear charging profiles, partial en-route charging, and charging stations with limited capacity. Computational experiments of our branch-and-price algorithm are carried out on randomly generated instances. We compare the tradeoff between operational costs and battery degradation of the solutions obtained by our approach to those of a baseline deterministic approach.

1.4. Thesis outline

The remainder of this document is organized as follows. Chapter 2 reviews the literature on statistical learning for probabilistic predictions, the deterministic (electric) VSP, the (electric) VSP under uncertainties, and the electric VSP with battery degradation. Chapters 3, 4, 5, and 6 form the main body of this thesis. In Chapter 3, statistical learning approaches are explored to predict bus travel time uncertainty based on real-world historical data. A

data-driven stochastic programming framework for the MDVSP with stochastic travel time is then provided in Chapter 4. Chapter 5 introduces a two-stage stochastic program with recourse for the MDEVSP with stochastic travel time and energy consumption. A chance-constraint model for the E-VSP with battery degradation and stochastic energy consumption is presented in Chapter 6. Finally, we discuss the contribution of this thesis from a unified perspective and provide some conclusions in Chapter 7.2.

Chapter 2

Related works

The VSP and its variants have been studied extensively in the literature over the past decades, and several surveys have been proposed, particularly those of Desaulniers and Hickman (2007) and Bunte and Kliewer (2010). This section reviews the main models and solution approaches for this optimization problem. Moreover, since we study in this thesis the VSP from a *data-driven* and a *stochastic* perspective, we also include a discussion on statistical learning approaches and the VSP under uncertainties. Indeed, data-driven approaches rely on data analysis and, in our case, statistical learning.

The remainder of this section is organized as follows. In section 2.1, we provide some background on machine learning from a probabilistic point of view and review probabilistic models proposed in the literature. The VSP's main modeling approaches and algorithms are introduced in section 2.2, before going through works on the VSP under uncertainties in section 2.3. Section 2.4 presents how the models and algorithms for the VSP have been adapted to the E-VSP. Finally, contributions for the E-VSP under uncertainties and the E-VSP with battery degradation are reviewed in sections 2.5 and 2.6, respectively.

2.1. Statistical learning for probabilistic predictions

Machine learning models are usually grouped into three main categories: supervised (or predictive), unsupervised (or descriptive), and reinforcement learning. Supervised learning aims at modeling the mapping between an input vector $\mathbf{x}$ and an output y given a training set $\mathcal{B} = \{(\mathbf{x}_i,y_i)\}_{i=1}^N$ of N points. In contrast, unsupervised learning aims at learning patterns given a dataset $\mathcal{B} = \{\mathbf{x}_i\}_{i=1}^N$ without labeled outputs. Examples of some common unsupervised learning tasks are density estimation, clustering, and dimensionality reduction. Supervised learning is the most common type of machine learning in practice (Murphy, 2012) and takes two forms: *classification* and *regression*. Classification is a predictive task where the output or response variable y_i is categorical, i.e., $y_i \in \{1, \ldots, C\}$, where C is the number

of classes. When the output is real-valued, the prediction is referred to as regression. Reinforcement learning is a third and distinct machine learning type that is rooted in dynamic programming methods. In short, reinforcement learning algorithms learn how to behave in a system by receiving rewards or punishments for each action. This section focuses on regression, but we refer interested readers to the textbook of Murphy (2012) on supervised and unsupervised learning from a probabilistic perspective and the textbook of Sutton and Barto (2018) on reinforcement learning.

Let $z_i = (\mathbf{x}_i, y_i)$ for $i = 1, \ldots, N$ be independent and identically distributed draws with an unknown distribution $p(Z)$. The standard learning framework consists in (1) defining a family $\mathcal{F}$ of functions, often a parametric one, (2) choosing a method to evaluate the quality of each function $f \in \mathcal{F}$, i.e., a *loss function*, and (3) optimizing over all functions considered the loss. Examples of loss functions are the classification error $\mathcal{L}(f(\mathbf{x}),y) = I_{f(\mathbf{x})\neq y}$, for classification tasks, and the quadratic error $\mathcal{L}(f(\mathbf{x}),y) = (f(\mathbf{x}) - y)^2$, for regression tasks. Ideally, one would like to find a function $\hat{f}$ that minimizes the expected risk $\mathbb{E}[\mathcal{L}(f(\mathbf{x}),y)]$. However, because $P(Z)$ is unknown, the *empirical risk* $\hat{R}$ is instead minimized; i.e.,

$$\hat{f}(\mathcal{B}) = \arg\min_{f\in\mathcal{F}} \hat{R}(f,\mathcal{B}) \tag{2.1.1}$$

$$= \arg\min_{f\in\mathcal{F}} \frac{1}{|\mathcal{B}|} \sum_{(\mathbf{x}_i,y_i)\in\mathcal{B}} \mathcal{L}(f(\mathbf{x}_i),y_i). \tag{2.1.2}$$

Different models are to be compared based on their generalization error, i.e., the expected loss averaged over unseen or future data. To avoid biased generalization error estimation, one must split the available dataset into a *training set* for parameters selection, a *validation set* for hyper-parameters selection, and a *test set* for the final evaluation of the models. The model $\hat{f}$ selected is the one that minimizes the empirical risk $\hat{R}$ over the test set.

Supervised learning can also be thought of in a probabilistic manner as a method for finding the most probable output $\hat{y}$ given an input $\mathbf{x}$, by using

$$\hat{y} = \hat{f}(\mathbf{x}) = \arg\max_{f\in\mathcal{F}} p(y|\mathbf{x},\mathcal{F},\mathcal{B}). \tag{2.1.3}$$

Yet, if most supervised learning models can be interpreted in a probabilistic way, the focus through the loss function is usually put on outputting the best conditional expectation of the response variable, $\mathbb{E}(y|\mathbf{x})$. Other aspects of the conditional distribution functions, such as the skewness, quantiles of interest, and variance, must be deliberately optimized. Indeed, in some cases, the expected value is only one of the most important aspects for a decision-maker. In meteorology, for example, a decision-maker may prefer to know the probability of rain versus no rain rather than the expected precipitation. In what follows, we review recent works on various applications that have proposed models specifically designed to capture

all aspects of the distribution and output conditional probability density functions (PDFs). Performing distributional regression with non-Gaussian and heteroscedastic variables is a non-trivial task, although it is common in many applications. For example, the conditional probability distribution of bus travel time is typically skewed to the right and its variance varies between peak and off-peak periods. The works of Carney et al. (2005), Rigby and Stasinopoulos (2005), Schlosser et al. (2019), Duan et al. (2020), and Dutordoir et al. (2018) all propose approaches for addressing these challenges.

Carney et al. (2005) used an ensemble of mixture density networks (MDNs) to predict the conditional PDF of surf height. MDNs are a type of mixture of experts (Jordan and Jacobs, 1993), i.e., an ensemble of submodels (or "experts"), each specialized in a region of the input space where each expert and the gating function, deciding which expert to use for each input, are neural networks. The response variable of MDNs is represented by Gaussian mixture models (GMMs), which can model probability distributions of various shapes and natures. Several MDNs are combined in Carney et al. (2005) using an ensemble approach to increase the stability of a single MDN. Two ensemble approaches are compared, namely bagging (Breiman, 1996) and boosting (Freund and Schapire, 1996). It was found that boosting provided slightly better results over the authors' datasets.

Generalized additive models for location, scale, and shape (GAMLSS, Rigby and Stasinopoulos, 2005) are an extension of generalized linear models (Nelder and Wedderburn, 1972) and generalized additive models (Hastie and Tibshirani, 1986), where each parameter of a probability distribution, namely the location, the scale, and the shape, are smooth linear or nonlinear functions of the input vector. The heteroscedasticity is modeled by allowing each observation to have a specific set of distributional parameters.

Schlosser et al. (2019) proposed a framework for distributional regression forests, combining tree-based methods (i.e., regression tree and random forest) and parametric distributional models. Unlike GAMLSS, distributional regression forest models can map smooth and non-smooth relationships between the input vector and the output and handle variable selection and complex interactions. The numerical results show that the distributional regression forest performs equivalent to or better than GAMLSS without requiring prior meteorological knowledge for variable selection.

Recently, Duan et al. (2020) introduced the natural gradient boosting (NGBoost) algorithm, a supervised learning method using boosting to predict probability distributions conditional on covariates. Their method assumes that each conditional probability distribution follows a specific parametric form with two or more degrees of freedom and simultaneously boosts multiple parameters from the base learners. This multiparameter boosting approach introduces poor training dynamics corrected by a generalized natural gradient descent approach. NGBoost has performed comparably on many datasets to state-of-the-art

probabilistic regression models, but it is considered more flexible, scalable, and easier to use than the latter.

All previous works propose *frequentist* models (Koller and Friedman, 2009). However, one should not forget that Bayesian methods are probabilistic in nature and can therefore also be used to predict conditional PDFs. Gaussian processes (GPs) are a non-parametric Bayesian method where a prior over functions is first defined before computing a posterior over functions once data is revealed. The function's values at a finite and random set of points are interpreted as a multivariate Gaussian with each element of the covariance matrix given by a positively defined kernel function (Rasmussen and Williams, 2005; Murphy, 2012). Recently, Dutordoir et al. (2018) introduced a GP-based model with an input augmented with latent variables that can estimate non-Gaussian conditional densities.

2.2. Vehicle scheduling problem

Vehicle scheduling is a crucial step in the service planning of public transportation and it has been studied intensively in the last decades. Once the timetables of trips are fixed for a given period (typically one season), transit agencies have to assign buses and drivers to each of these trips. One timetable for all weekdays and two timetables for Saturdays and Sundays are generally provided, thus three VSPs are usually solved every period.

The VSP can be formalized as follows. Given a set of n timetabled trips $\mathcal{V} = \{1, 2, \ldots, n\}$, the start time d_i^0 and the end time d_i^1 for each trip $i \in \mathcal{V}$, the deadhead travel time κ_{ij} between the end of trip i and the beginning of trip j for every pair of trips i and $j \in \mathcal{V}$, a set of depots $\mathcal{D}$, and the capacity (number of buses available) b_d of each depot $d \in \mathcal{D}$, assign exactly one bus to each timetabled trip while minimizing the total cost and respecting vehicle availability. The total cost is typically composed of a cost for the acquisition of the buses and operational costs (per distance traveled and/or time spent outside the depot). For small- to medium-scale transit agencies with only one depot, this problem corresponds to the single-depot vehicle scheduling problem (SDVSP). Whereas for medium- to large-scale transit agencies with two or more depots, this problem corresponds to the MDVSP.

A vehicle schedule s is a sequence of trips planned over the horizon of one day. The first trip of a vehicle schedule is called a pull-out trip and corresponds to a trip from a depot $d \in \mathcal{D}$ to the start location of the first timetabled trip of the schedule, whereas the last trip is called a pull-in trip and corresponds to a trip from the last timetabled trip of the schedule to a depot $d \in \mathcal{D}$. A common constraint is to consider that a vehicle schedule is feasible if the pull-out and the pull-in trips are associated to the same depot (i.e., if it starts and ends the day at the same depot). In between the pull-out and the pull-in trips, the vehicle performs connections between pairs of trips. The connection of two consecutive trips i and $j \in s$ is possible if $d_j^0 \geq d_i^1 + \kappa_{ij} + \tau$, where τ is the minimum layover time between two

trips (to account for the break time for drivers, among other things). We denote by $\mathcal{S}$ and $\mathcal{S}^d \subset \mathcal{S}$ the set of all feasible vehicle schedules and the set of all feasible vehicle schedules starting and ending at a depot d, respectively.

This section is organized as follows. The main models for the SDVSP and the MDVSP and some extensions are introduced in Section 2.2.1. Exact and heuristic solution approaches are reviewed in Section 2.2.2.

2.2.1. Models

The SDVSP can be modeled as a minimal decomposition problem, an assignment problem, a transport problem, or a network flow problem. In this section, the latter model is presented as it is the basis for two common models for the MDVSP, namely the multi-commodity (connection-based and time-space networks) and the set partitioning models. The single commodity models with subtour breaking constraints and assignment variables can also be used for the MDVSP, but they are not presented in this thesis as they provide weaker linear programming bounds than their counterparts (Mesquita and Paixão, 1999; Ribeiro and Soumis, 1994). At the end of this section, some extensions of these models are listed. For a complete overview on models for the SDVSP and the MDVSP, we refer interested readers to the survey of Bunte and Kliewer (2010).

Network flow model. This model, first presented in Raff (1983), is defined on networks where each trip $i \in \mathcal{V}$ is represented by a node and the depot is represented by nodes n_0^d and n_1^d for the beginning and the end of the day, respectively. Pull-out and pull-in arcs connect n_0^d to all trips $i \in \mathcal{V}$ and all trips $i \in \mathcal{V}$ to n_1^d, respectively. Furthermore, all compatible trips are connected. An example of this so-called connection-based network $G = (\mathcal{V} \cup \{n_0^d\} \cup \{n_1^d\}, A)$ with node set $\mathcal{V} \cup \{n_0^d\} \cup \{n_1^d\}$ and arc set A is illustrated in Figure 2.1a. This network is acyclic because of the compatibility requirements imposed on the arcs.

The network flow model uses the following additional notation. Let X_{ij} and c_{ij} be the flow and the cost of arc $(i,j) \in A$, respectively. It can be formulated as the following integer program:

$$\min \quad \sum_{(i,j)\in A} c_{ij} X_{ij} \tag{2.2.1}$$

$$\text{s.t.} \quad \sum_{i:(i,j)\in A} X_{ij} = 1, \qquad \forall j \in \mathcal{V} \tag{2.2.2}$$

$$\sum_{i:(i,j)\in A} X_{ij} - \sum_{i:(j,i)\in A} X_{ji} = 0, \qquad \forall j \in \mathcal{V} \cup \{n_0^d, n_1^d\} \tag{2.2.3}$$

$$X_{ij} \in \{0,1\}, \qquad \forall (i,j) \in A. \tag{2.2.4}$$

The objective function (2.2.1) minimizes the total cost, while constraints (2.2.2) ensure that every timetabled trip is included in the solution. Constraints (2.2.3) and (2.2.4) ensure flow conservation and binary flow values, respectively. An additional constraint can be added to (2.2.1) - (2.2.4) to limit vehicle capacity at the depot

$$\sum_{j \in \mathcal{V}} X_{n_0^d j} \leq b, \tag{2.2.5}$$

where b is the number of buses available at the depot. If we consider nodes as transshipment nodes, then the problem can be seen as finding the minimum cost flow in G.

Multi-commodity model. This model is a generalization of the network flow model with more than one depot, where one connection-based network G^d is built per depot $d \in \mathcal{D}$. As proposed by Ribeiro and Soumis (1994) and Raff (1983), it can be expressed as the following integer program:

$$\min \quad \sum_{d \in \mathcal{D}} \sum_{(i,j) \in A^d} c_{ij} X_{ij}^d \tag{2.2.6}$$

$$\text{s.t.} \quad \sum_{d \in \mathcal{D}} \sum_{i:(i,j) \in A^d} X_{ij}^d = 1, \qquad \forall j \in \mathcal{V} \tag{2.2.7}$$

$$\sum_{i:(i,j) \in A^d} X_{ij}^d - \sum_{i:(j,i) \in A^d} X_{ji}^d = 0, \qquad \forall d \in \mathcal{D}, j \in \mathcal{V} \cup \{n_0^d, n_1^d\} \tag{2.2.8}$$

$$\sum_{j \in \mathcal{V}} X_{n_0^d j}^d \leq b_d, \qquad \forall d \in \mathcal{D} \tag{2.2.9}$$

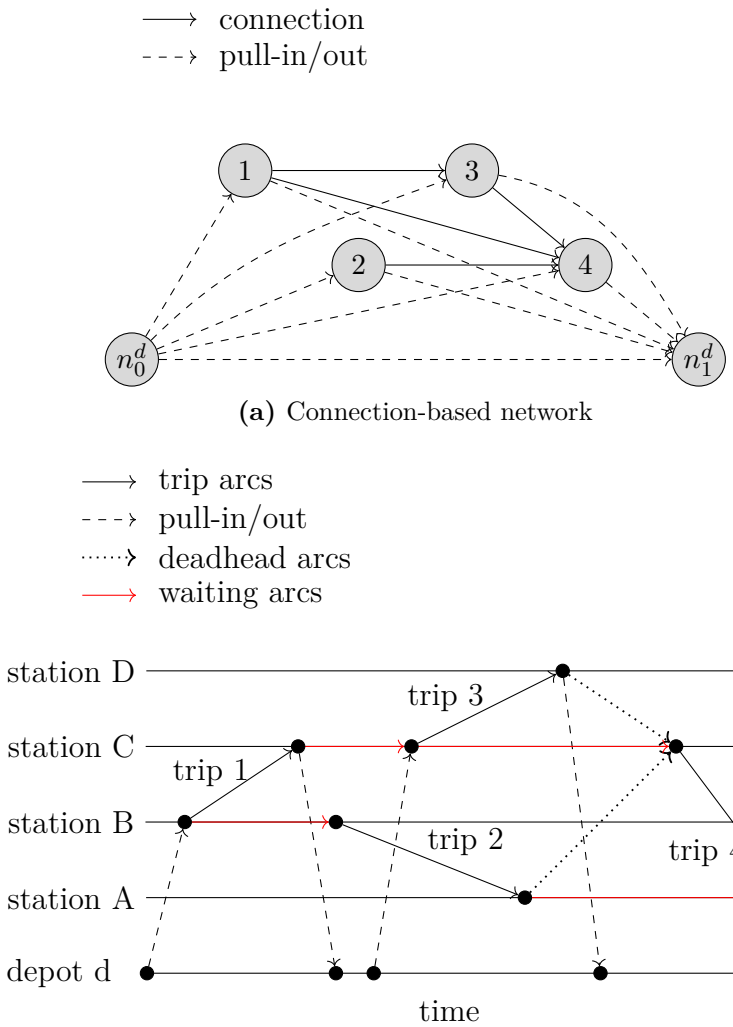
$$X_{ij}^d \in \{0,1\}, \qquad \forall d \in \mathcal{D}, (i,j) \in A^d, \tag{2.2.10}$$

where A^d is the arc set of G^d and X_{ij}^d are binary variables of the flow on arc $(i,j) \in A$ in network G^d (either 0 or 1).

The objective function (2.2.6) and constraints (2.2.7), (2.2.8), (2.2.9), and (2.2.10) are generalizations of the objective function (2.2.1) and the equations (2.2.2), (2.2.3), (2.2.4), and (2.2.5), respectively.

The main drawback of using this model with connection-based networks is that the number of arcs of the model grows quadratically with the number of timetabled trips (Bunte and Kliewer, 2010). To reduce the number of arcs, Kliewer et al. (2006) introduced a multi-commodity model based on time-space networks. In this type of network, potential compatible deadhead trips are aggregated using a two-stage aggregation procedure and the number of arcs is further reduced by replacing several arcs starting or ending at the same location (either a terminus or a depot) by waiting arcs. An example of this this network is shown in Figure 2.1b.

Set partitioning model. Ribeiro and Soumis (1994) reformulated the multi-commodity model as a set partitioning problem, where each feasible vehicle schedule is a variable and



(a) Connection-based network

(b) Time-space network

Figure 2.1 – Comparison of (a) connection-based and (b) time-space networks for an example with $|\mathcal{V}| = 4$

the number of constraints is small. Let y_s be a binary variable equal to 1 if vehicle schedule s is selected and a_{is} be a binary parameter equal to 1 if vehicle schedule s covers trip $i \in \mathcal{V}$. The MDVSP can be expressed as this set partitioning problem

$$\min \quad \sum_{s \in \mathcal{S}} c_s y_s \tag{2.2.11}$$

$$\text{s.t.} \quad \sum_{s \in \mathcal{S}} a_{is} y_s = 1, \qquad \forall i \in \mathcal{V} \tag{2.2.12}$$

$$\sum_{s \in \mathcal{S}^d} y_s \leq b_d, \qquad \forall d \in \mathcal{D} \tag{2.2.13}$$

$$y_s \in \{0,1\}, \qquad \forall s \in \mathcal{S}, \tag{2.2.14}$$

where c_s is the cost of vehicle schedule s.

This formulation typically contains a large number of variables (i.e., vehicle schedules) and it is therefore not computationally efficient to enumerate all possible vehicle schedules. Instead, useful vehicle schedules can be generated by a column generation-based algorithm (see section 2.2.2 for an overview on column generation algorithms introduced for the MDVSP).

The set partitioning model is often preferred to the multi-commodity model (with connection-based or time-space network) because it is more easily adaptable. Indeed, the definition of a feasible vehicle schedule can easily be modified to include path constraints, e.g., related to fuel or electricity consumption. These modifications are then only reflected in the pricing problems of (2.2.11) - (2.2.14), i.e., the problems solved at each iteration of the column generation algorithm to generate new vehicle schedules.

Extensions. There is a wide range of MDVSP extensions including, among others, the MDVSP with multiple vehicle types, with trip shifting, with trip-depot compatibility constraints, and with departure-duration restrictions. We refer interested readers to (Bunte and Kliewer, 2010; Desaulniers and Hickman, 2007)

2.2.2. Algorithms

While there are polynomial time algorithms for solving the SDVSP, the MDVSP (i.e., when $|\mathcal{D}| \geq 2$) has been proven to be NP-hard (Bertossi et al., 1987). It is therefore interesting to review in this section exact and heuristic algorithms for the most complicated version of the problem, which is also the one closer to the reality of medium- and large-scale transit agencies, namely the MDVSP. We will pay particular attention to algorithms that can be used for real-world, large-scale cases.

A first exact solution approach for the MDVSP is presented in Carpaneto et al. (1989). since then, several others were proposed, including in Ribeiro and Soumis (1994), Forbes et al. (1994), Bianco et al. (1994), Löbel (1998), Mesquita and Paixão (1999), Hadjar et al. (2006), Kliewer et al. (2006), and Oukil et al. (2007). In Ribeiro and Soumis (1994) a so-called branch-and-price algorithm (Barnhart et al., 1998), i.e., a column generation algorithm (with columns being feasible vehicle schedules) embedded in a branch-and-bound tree, is proposed to solve the MDVSP formulated as a set partitioning problem. At each iteration, the linear relaxation of (2.2.11) - (2.2.14), called the master problem (MP), is solved with a restricted number of vehicle schedules. New columns to add to the restricted MP (RMP) are identified by solving a set of shortest path problems on connection-based networks with modified arc costs based on dual information obtained from the resolution of the RMP. If vehicle schedules with negative reduced cost are found, some of them are added to the RMP. Otherwise, the algorithm stops and we have a guarantee that the solution to the current

RMP is also optimal for the MP. We refer interested readers to Desaulniers et al. (2005) and Lübbecke and Desrosiers (2005) for more details on column generation. Later on, Hadjar et al. (2006) enhanced this algorithm by introducing variable fixing and cutting planes. A variable X_{ij}^d, for $i,j \in \mathcal{V}$ and $d \in \mathcal{D}$, is fixed to zero (i.e., the arc (i,j) in the corresponding network is removed) when its reduced cost is greater than or equal to a threshold value that depends on the current best feasible solution and its associated dual variables. A first feasible solution is found by performing a depth-first search in the branch-and-bound tree, and then the algorithm backtracks in the tree. Furthermore, Hadjar et al. (2006) introduced cutting planes to reduce the number of *odd cycles* in the MDVSP and a heuristic procedure to lift these cuts. Löbel (1998) proposed to solve directly the multi-commodity model using a column generation algorithm, where arc flow variables are generated instead of vehicle schedules. To speed up the resolution, variables are first generated based on a Lagrangian pricing strategy. When the objective function does not improve sufficiently in a few iterations, the algorithm switches to the standard variable selection criterion, i.e., negative reduced cost. When no additional arc flow variables are generated, and if there are still some fractional variables, a rounding strategy is used. In practice, at the end of the process most of the variables were already integral and most instances were solved to optimality. Kliewer et al. (2006) solved large-scale instances of the multi-commodity problem with an underlying time-space network using the general-purpose CPLEX mixed integer programming solver. More recently, Oukil et al. (2007) proposed dual variable stabilization strategies to speed up the convergence of a standard column generation algorithm for the MDVSP.

Several heuristic algorithms for the MDVSP have been proposed in the literature, among others, Raff (1983), Lamatsch (1992), Dell'Amico et al. (1993), Gintner et al. (2005), Laurent and Hao (2009), Pepin et al. (2009), Otsuki and Aihara (2016), Kulkarni et al. (2018), and Moreno et al. (2019). Gintner et al. (2005) devised a two-phase heuristic approach to solve large-scale MDVSP with time-space networks. In the first phase, one SDVSP problem per depot is solved and *stable chains*, i.e., sequences of trips included in the solution of each SDVSP are identified. In the second phase, all the stable chains are fixed to one by treating each chain as a trip. The resulting problem is solved exactly. Laurent and Hao (2009) presented an iterated local search algorithm for the MDVSP. An initial solution is found by transforming the MDVSP into a SDVSP with a fictitious depot and solving the resulting problem using the auction algorithm proposed by Freling et al. (2001). Then, random neighboring solutions are found by shifting random sequences of consecutive trips covered by the same vehicle, referred to by the authors as *block chains*, to another vehicle. In Pepin et al. (2009), five heuristics and matheuristics based on multi-commodity and set partitioning models are compared. Out of them, the column generation heuristic provided the best solutions, but additional time savings were achieved with a large neighborhood search method. Additionally, Otsuki and Aihara (2016) conducted a comparison between a local

search metaheuristic that employed pruning and deepening techniques in a variable depth search and a second-best local search. Their experimental results on MDVSP instances demonstrated that the former approach yields superior results. Recently, Kulkarni et al. (2018) introduced a variation of the multi-commodity model with time-space networks, the so-called inventory formulation.

2.3. Vehicle scheduling problem under uncertainties

The VSP is typically solved deterministically, that is by disregarding the stochasticity inherent to transportation networks. However many parameters and constraints are actually subject to variability. For example, driver absenteeism may make one or several vehicle schedules infeasible on a given day or the capacity of a depot may be reduced in case of equipment failure. Another source of uncertainty in the VSP is the duration of trips, whether they are deadhead, pull-out, pull-in, or timetabled trips. We focus in this section on SDVSP and MDVSP models explicitly considering travel time uncertainty.

Cost-efficient solutions to the deterministic version of the VSP are typically compact (i.e., do not include much buffer time). Indeed, buffer times induce direct and/or indirect costs that are not desirable in the deterministic version of this problem. When the cost of a vehicle schedule includes a component for the time spent outside the depot, buffer times induce a direct cost, whereas buffer times are indirectly associated with an opportunity cost for the immobilized buses. This compactness explains why cost-efficient bus schedules are especially prone to delays; when a disruption occurs (e.g., road accident), travel times may deviate from the planned duration and bus schedules may be delayed if there is not enough buffer time to absorb these deviations. Two types of delays are distinguished in the literature: primary (or exogenous) and secondary (or endogenous, propagated) delays (Naumann et al., 2011; Kramkowski et al., 2009; Amberg et al., 2011, 2019). The former type of delay is a direct consequence of a disruption in the system. Secondary delays occur when buffer times are insufficient to recover primary delays. Figure 2.2 illustrates an example of delay propagation between two trips i and j. The primary delay of trip i is given by the difference between its actual travel time and its planned travel time. The primary delay of trip i causes a secondary delay of trip j equal to the difference between its actual start time and its planned start time.

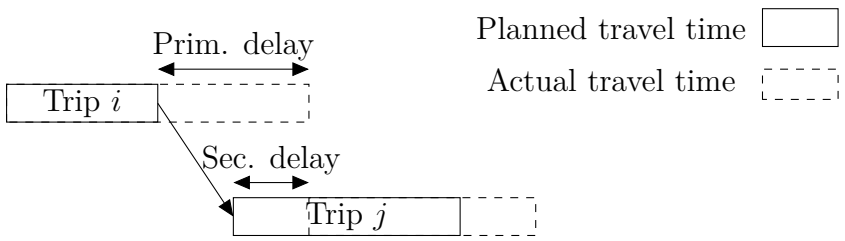


Figure 2.2 – Delay propagation

Primary delays are considered unavoidable in daily operations (Kramkowski et al., 2009; Amberg et al., 2019), while secondary delays are endogenous to resource planning. Thus, at the planning level, reliable approaches for the VSP should try to mitigate the propagation of delays. This approach has been explored by two main families of methods in the literature: (1) online (or real-time) and (2) offline methods. This section mainly discusses the latter type, but the works of Huisman et al. (2004) and He et al. (2018) on solution approaches for the dynamic VSP are worth mentioning. They present algorithms to dynamically schedule a bus fleet according to the state of the network. Kramkowski et al. (2009), Naumann et al. (2011), Shen et al. (2016), van Kooten Niekerk (2018), Amberg et al. (2011), and Amberg et al. (2019) proposed offline models and solution approaches for the VSP under uncertain travel time. Table 2.1 summarizes the main characteristics of these works, namely the type of model, the solution method (Sol. method), if delays are fully propagated or not (Full delay proba.), the number of trips ($|\mathcal{V}|$), the number of depots ($|\mathcal{D}|$), and the number of vehicle types ($|\mathcal{T}|$) of the largest instance tested. The value "NA" in this table indicates that the solution method is not explicitly specified in Naumann et al. (2011). Note that they all deal with the single depot case, but those that consider two or more vehicle types are also NP-hard. Some variants of SDVSP with multiple vehicle types can be solved similarly to the MDVSP by constructing a network by vehicle type.

Table 2.1 – Summary of the literature on the VSP under uncertain travel time. NA - unspecified. *Model*: MCF - multi-commodity model, NF - network flow model. *Algorithm*: MH - metaheuristic, MIP - mixed integer programming method, H - heuristic, CG - column generation, LR - Lagrangian relaxation

				Experiments		
Authors	Model	Sol. method	Full delay propa.	$\|\mathcal{V}\|$	$\|\mathcal{D}\|$	$\|\mathcal{T}\|$
Kramkowski et al. (2009)	MCF	MH	•	1296	1	3
Naumann et al. (2011)	NF	NA		426	1	1
Shen et al. (2016)	NF	H	•	450	1	1
van Kooten Niekerk (2018)	NF	MIP		3496	1	1
Amberg et al. (2011)	MCF	CG, LR	•	580	1	1
Amberg et al. (2019)	MCF	CG, LR	•	661	1	2

Kramkowski et al. (2009) presented an offline metaheuristic to increase the reliability (or delay-tolerance) of the solutions of the VSP with multiple vehicle types. From an initial solution computed as in Kliewer et al. (2006), a simulated annealing for noisy environments seeks valid neighboring solutions. Two variants of neighborhood generators are compared: a random based neighborhood operator and a selective neighborhood operator. The former randomly selects a timetabled trip from a vehicle schedule and randomly assigns it to another one such that every vehicle schedule remains feasible. If the timetabled trip cannot

be assigned to any vehicle schedule, a new vehicle schedule is created. The selective neighborhood operator selects the timetabled trip with the highest expected secondary delay and assigns it, if possible, to the vehicle schedule providing the longest buffer time before the timetabled trip. Experimental results on three real-life instances suggest that the random-based neighborhood operator provides better tradeoffs between total cost and reliability on small instances while the selective neighborhood operator is preferable on larger ones. In addition, both methods provided more reliable solutions than the initial solutions at a slightly higher cost.

A stochastic programming framework for the VSP with stochastic travel time was proposed by Naumann et al. (2011). They expanded the network of Kliewer et al. (2006) (i.e., a time-space network) with extra waiting and deadhead arcs to be able to consider a penalty for delays between two timetabled trips in the arc cost. However, even with this expansion, delay propagation is restricted to pairs of consecutive trips and the cascade effect of delays is neglected. In experiments on a real-life instance and 100 delay scenarios, several solutions using the same number of vehicles and with lower penalty cost than the deterministic approach were found.

Shen et al. (2016) introduced two models for the VSP with stochastic travel time. Both models are formulated as a network flow problem with stochastic trip compatibility, i.e., an arc (i,j) is built if the compatibility probability of trips i and j is greater or equal to zero. If so, the arc is associated with a compatibility probability. A penalty for the expected infeasible time of an arc (i,j) is imposed in the first model, for each arc in the network. The infeasible time of an arc (i,j) is equivalent to the secondary delay of trip j if we assume trip i always starts on time (i.e., only delays between pairwise consecutive trips are considered). Full delay propagation is only considered in the second enhanced model by redefining the departure time of a trip as a stochastic variable. Shen et al. (2016) provided an exact method to compute the probability density function of the secondary delay of a trip based on the probability density functions of the departure and arrival times of previous timetabled trips. This second model cannot be solved by existing solution approaches for the vehicle scheduling problem (e.g., mixed integer programming method), so an hybrid solution approach is proposed instead. An initial solution is computed by a matching-based heuristic and then this solution is refined using an iterative greedy local search method. Experimental results showed that both models provide more reliable solutions than the deterministic model while using the same number of vehicles. The model considering full delay propagation achieved higher punctuality than the first model with a little increase in costs.

van Kooten Niekerk (2018) introduced the stochastic departure time dependent VSP. The model allows negative buffer times and the cost of the arcs between pairs of trips is modified to include a cost for secondary delays. Different cost calculations with delay propagation between a sequence of maximum two trips were compared to assess the potential of

an approach incorporating full delay propagation. After carrying out computational experiments, the authors concluded that accounting for the propagation of delays over longer trip sequences in their model promised little benefit. Solutions 2 to 3% more reliable than the baseline approach of imposing minimum buffer times were achieved.

The work of Amberg et al. (2019) is an extension of Amberg et al. (2011), and they both address the sequential, partially integrated, and integrated vehicle and crew scheduling problems. The SDVSP and the MDVSP are modeled as multi-commodity problems with underlying time-space networks in both works. In their first short proceeding paper, mandatory buffer times between trips covered by the same vehicle schedule are imposed. Furthermore, novel decomposition schemes of flows (i.e., bundle of equal-cost solutions) taking into account secondary delays between timetabled trips and pieces of work, for the sequential and partially integrated problems, respectively, are proposed. This method finds the most delay-tolerant schedules included in a flow, but the reliability improvements are limited by the flow considered. This limitation is lifted in Amberg et al. (2019), where the solution approach of Amberg et al. (2011) is used in the initialization phase. The solution approach of Amberg et al. (2019) combines Lagrangian relaxation and column generation and takes into account delay propagation in vehicle and crew duties resulting from expected primary delays (or equivalently expected travel time deviations). Delays are propagated dynamically when building vehicle and crew duties and a penalty for delay propagation is added to the total cost of each duty type. Experiments on real-life instances from Germany showed that the integrated scheduling scheme provides the best trade-offs between reliability and total cost, compared to the sequential and partially-integrated schemes.

2.4. Electric vehicle scheduling problem

The batteries of BEBs are mainly characterized by their capacity and their charging technologies. Nowadays, the main charging technologies used in public transport are conductive charging, inductive charging, and battery swapping and these differ primarily in location and power (see Häll et al., 2019; Li, 2016; Pelletier et al., 2016, for details on charging technologies). First, level 1 to level 2 conductive chargers are used primarily for charging at the depot, whereas opportunity charging (or en-route charging) at end points are often done using level 3 conductive chargers (also known as fast chargers). Second, inductive chargers are mostly used for opportunity charging at bus stops or even charging during driving. Third, battery can be swapped in about 10 minutes at swapping stations. Since opportunity charging at end points modifies the VSP the most, we will only consider this type of charging in what follows.

The E-VSP is an extension of the VSP with some additional limitations: shorter driving range, longer refueling time, and special charging infrastructure that are usually limited in

number (Li et al., 2021). Moreover, the SoC of the battery of a BEB cannot fall below zero and cannot exceed the battery capacity at any time. Many variations of the E-VSP are addressed in the literature. First, some variations assume that vehicles are always charged to full capacity (or to a given percentage of full capacity, typically 80% or 90%) and others assume partial charging. Second, a maximum capacity for the charging infrastructure can also be taken into account. This capacity can be limited by the number of plugs at each charging station or by the station's electrical grid capacity. Third, nonlinear charging profiles of BEBs (Olsen and Kliewer, 2020) can be considered.

E-VSP extensions of network flow, multi-commodity and set partitioning models, as well as column generation algorithms, have been extensively studied in the literature in recent years (Perumal et al., 2022). In this section, particular attention is paid to how the models for the SDVSP and the MDVSP have been adapted to include charging stations and plans, a maximum driving range and non-linear charging profiles, and how these changes reflect in the solution approach. Thus, this review is not intended to be extensive, but rather insightful on the particular challenges of the E-VSP and ways in which these challenges have been addressed in the literature so far. For a detailed overview on models and solution approaches for the E-VSP, see Perumal et al. (2022).

2.4.1. Charging stations and plans

Let $\mathcal{H}$ be a set of charging stations located at a depot, at a terminal, or elsewhere in the bus network. In the E-VSP, a vehicle schedule is defined as a sequence of timetabled trips and charging periods, each associated with a start time, a duration, and a charging station in $\mathcal{H}$. A charge plan consists of one or several charging periods. In what follows, we discuss three modeling approaches for charging stations and plans that we have identified in the literature.

The first approach consists in creating one charging node for each pair of charging station in $h_k \in \mathcal{H}$ and timetabled trip $i \in \mathcal{V}$, that we denote $h_{k,i}$. The charging node $h_{k,i}$ represents visiting charging station h_k after completing timetabled trip i. In Adler and Mirchandani (2017), one charging node $h_{k,d}$ for each pair of charging station $h_k \in \mathcal{H}$ and depot $d \in \mathcal{D}$ is also built. The corresponding network in presented in Figure 2.3. This approach can easily account for partial charging (see for example the model of Wen et al. (2016)), but it is difficult to limit the capacity of each charging station.

The second approach is to discretize the planning horizon into ρ-minute time intervals to build time-expanded charging stations. Let R be a set of time intervals each of duration ρ and $\mathcal{H}^E$ be a set of time-expanded charging nodes, where each node is associated with a charging station in $\mathcal{H}$ and a time interval in $\mathcal{R}$. For example, we denote by $h_1^E r_5$ the node associated with charging station $h_1 \in \mathcal{H}$ and time interval $r_5 \in \mathcal{R}$. This approach illustrated

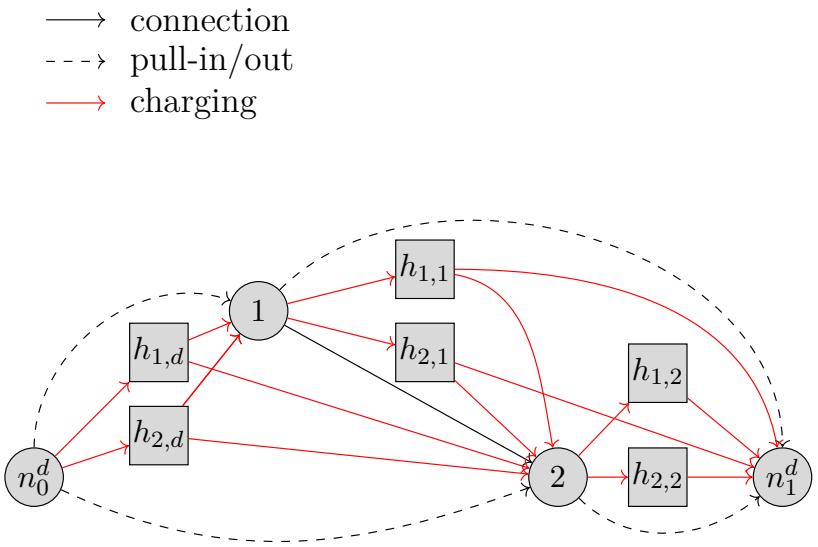


Figure 2.3 – Connection-based network with charging nodes associated with each trip and depot

in Figure 2.4 has been used in Tang et al. (2019), Li (2014), Wu et al. (2022), and Li et al. (2021) with the following assumptions. On the one hand, it is assumed that a vehicle can only start a charge period every ρ minutes. On the other hand, it is assumed that after ρ minutes of charge, BEBs are always fully charged. While with this strategy the capacity of each charging station can be limited by adding a capacity constraint for each time-expanded charging station node, partial charging cannot be easily accommodated since the duration of each node is fixed. This model is thus most appropriate in the context of fast charging and battery swapping, when it is reasonable to assume a fix refueling time, say 10 minutes (Adler and Mirchandani, 2017; Li, 2014), to fully recharge the battery or swap it for a full one.

The approach of Tang et al. (2019), Li (2014), and Li et al. (2021) can be expanded using the idea of Janovec and Kohání (2019) of connecting consecutive time-expanded charging station nodes in order to model charging periods of different duration. We call the corresponding network the connection-based network with interconnected time-expanded charging station nodes. This type of network easily accommodate partial charging and charging stations with limited capacity.

In the third approach, no additional nodes for charging stations are added to the connection-based networks. Instead, charging time and cost are included in the connection arcs (i.e., arcs between timetabled trips). In van Kooten Niekerk et al. (2017), it is assumed that, whenever possible, vehicles recharge as much energy as possible between trips. In contrast, Perumal et al. (2021) assumed that a vehicle is recharged between two timetabled trips when there is enough time to fully recharge the battery.

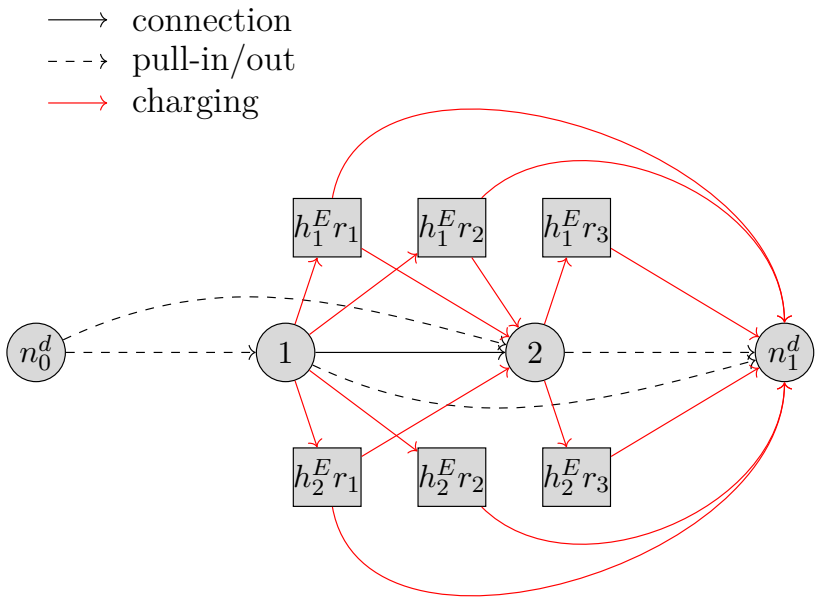


Figure 2.4 – Connection-based network with time-expanded charging station nodes

To the best of our knowledge, the adaptation of time-space networks to the E-VSP has only be addressed by Li et al. (2021, 2019) and (see Section 2.5 for more details). As mentioned in Section 2.2.1, the set partitioning formulation with underlying connection-based networks is indeed more easily adaptable to the electric version of the VSP and it is therefore not surprising to find more work in the literature using the latter type of networks.

2.4.2. Driving range

Li (2014) added the following three driving range constraints to (2.2.1)-(2.2.4):

$$g_j = \sum_{i:(i,j)\in A} (g_i + \iota_{ij}) X_{ij}, \qquad \forall j \in \mathcal{V} \tag{2.4.1}$$

$$g_j = 0, \qquad \forall j \in \mathcal{H}^E \cup \{n_0^d\} \tag{2.4.2}$$

$$(g_i + \iota_{ij}) X_{ij} \leq W, \qquad \forall j \in \mathcal{H}^E \cup \{n_1^d\}, (i,j) \in A, \tag{2.4.3}$$

where g_i is the accumulated energy consumption (or distance traveled) at node i, ι_{ij} is the energy (or distance) consumed from i to j, including deadhead travel between nodes i and j and the energy (or distance) for trip j, if applicable, and W is the battery capacity (or the maximum distance that can be traveled before battery renewal). Constraints (2.4.1) compute the accumulated energy consumption (or distance traveled) since the last battery renewal, constraints (2.4.2) enforce that the accumulated energy consumption (or distance traveled)

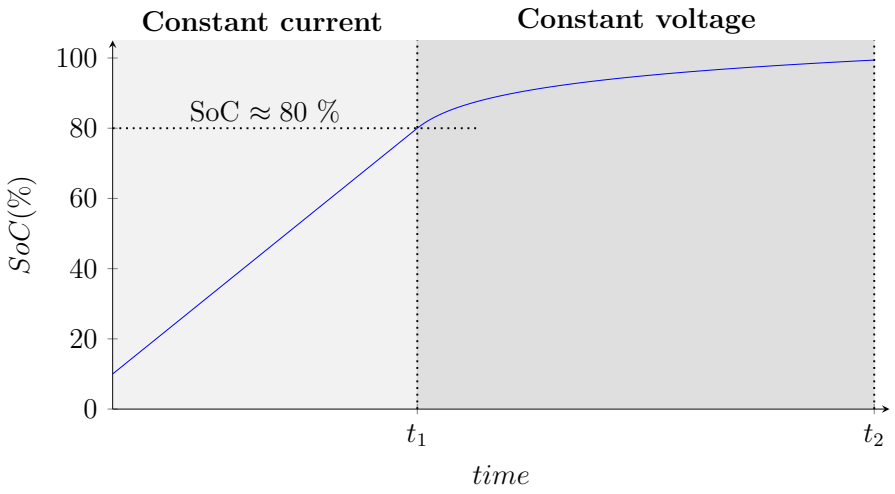


Figure 2.5 – Example of a charging curve

is set to zero after leaving the depot or a charging station, and constraints (2.4.3) limit the accumulated energy consumption (or distance traveled) to a maximum of W. Constraints (2.4.1)-(2.4.3) concern the case where only full charging is considered. See Wen et al. (2016), Janovec and Kohání (2019), and Li et al. (2020) for examples of network flow and multi-commodity models with battery capacity constraints considering partial charging.

When the E-VSP is (re)formulated as a set partitioning problem and solved by a column generation algorithm, all constraints related to the driving range are shifted to the sub-problems, i.e., resource-constrained shortest path problems (Li, 2014) or weight constrained shortest path problems with replenishment (Adler and Mirchandani, 2017).

2.4.3. Non-linear charging profiles

In order to avoid excessively high voltage that could damage BEBs lithium-ion batteries and reduce their life cycle, one can charge batteries using the *constant current/constant voltage* charging process illustrated in Figure 2.5. In the CC phase, the charging current is kept constant until the voltage reaches a threshold and then in the CV phase the voltage is kept constant while the current decreases (Montoya et al., 2017). These voltage and current regime changes make the charging functions nonlinear.

Furthermore, the charging functions of BEBs depend on many factors, including current, voltage, self-recovery and temperature (Wang et al., 2013). To capture the complex interaction between all these factors, the SoC of BEBs must be described using differential equations. However, these equations can hardly be included in the E-VSP. Thus, approximations of the charging functions, in most cases linear ones, are studied in the literature (Olsen and Kliewer, 2020). Montoya et al. (2017) and Olsen and Kliewer (2020) showed that linear approximations of the charging functions can lead to infeasible solutions in the electric

vehicle routing problem and in the E-VSP, respectively. Instead, both works propose to use a piecewise linear approximation. To the best of our knowledge, non-linear approximations have only been implemented by Liu and Ceder (2020), Olsen and Kliewer (2020), and Zhang et al. (2021) in the E-VSP literature.

2.5. Electric vehicle scheduling problem under uncertainties

The E-VSP takes in input the travel time and the energy consumption of a set of timetabled, deadhead, pull-out, and pull-in trips. In real-life settings these values vary from day-to-day due to, among other factors, weather conditions, traffic conditions, and passenger demand. Yet, most works on the E-VSP consider deterministic travel time and energy consumption. To the best of our knowledge, there are only three works in the literature on the E-VSP with stochastic travel time: Tang et al. (2019), Bie et al. (2021), and Jiang et al. (2021). Furthermore, the works of Bie et al. (2021) and Li et al. (2021) consider stochastic energy consumption. Tables 2.2 and 2.3 summarize the main characteristics and the constraints, out of the ones discussed in Section 2.4, of the literature on the E-VSP under uncertainties. In Table 2.2, column $|\mathcal{H}|$ is the number of charging stations in the largest instance tested. The value "NA" in this table indicates that the number of chargers used in the numerical experiments is not explicitly specified in Tang et al. (2019). The constraints considered in Table 2.3 are: charging stations with limited capacity (# chargers), limited driving range, non-linear recharging functions (non-linear recharge), partial recharge, stochastic charge time (stochas. charge), stochastic energy consumption (stochas. ene. cons.), and stochastic travel time (stochas. travel time).

Table 2.2 – Summary of the literature on the E-VSP under uncertainties. NA - unspecified. *Model*: SP- set partitioning, RR - range reliability-based stochastic program, NLP - non-linear optimization model *Algorithm*: CG - column generation, NSGA-II - nondominated sorting genetic algorithm with the elitist strategy, GD- gradient descent

			Experiments			
Authors	Model	Algo.	$\|\mathcal{V}\|$	$\|\mathcal{D}\|$	$\|\mathcal{T}\|$	$\|\mathcal{H}\|$
Tang et al. (2019)	SP	CG	96	1	1	NA
Bie et al. (2021)	NLP	NSGA-II	220	1	1	2
Jiang et al. (2021)	SP	CG	466	3	1	3
Li et al. (2021)	RR	GD	128	2	2	6

To the best of our knowledge, Tang et al. (2019) were the first to address the E-VSP with uncertain travel time through a static and a dynamic vehicle scheduling model. They

Table 2.3 – Summary of the constraints of the E-VSP under uncertainties

Authors	# chargers	Driving range	Non-linear recharge	Partial recharge	Stochas. charge	Stochas. ene. cons.	Stochas. travel time
Tang et al. (2019)	•	•					•
Bie et al. (2021)		•		•		•	•
Jiang et al. (2021)		•		•	•		•
Li et al. (2021)		•	•			•	

assume that the energy consumption of a trip depends on the driving speed and therefore on the travel time of the trip. In the static model, negative buffer times between two trips are allowed with a penalty and a robust strategy using a buffer-distance parameter is used to offset the potential impact of travel time stochasticity. Travel time scenarios are used in the dynamic model to dynamically reschedule a bus fleet at the beginning of each period of the day based on current and predicted future traffic conditions. Both models are reformulated as set partitioning and a branch-and-price solution approach is provided. Experimental results on small instances from the city of Beijing showed that their method is able to address the tradeoff between total cost and en-route breakdown risks. The dynamic model takes more computing time than the static model but outputs better solutions in terms of operational costs.

Bie et al. (2021) proposed a model for the E-VSP with stochastic travel time and energy consumption and possible recharge during buffer times. They derived an equation to compute the energy consumption of a trip based on the SoC at departure, the trip travel time, and the temperature. The energy consumption is assumed to follow a normal distribution and the variance of this distribution is estimated based on a data sample. Their multi-objective model minimizes the expected secondary delay and the expected energy consumption of the selected vehicle schedules as well as the procurement cost. Chance-constraints and robust constraints are included for time feasibility and the energy feasibility, respectively. They solved the E-VSP using a nondominated sorting genetic algorithm with the elitist strategy. Numerical tests were conducted on a real-life instance of 220 trips. The authors concluded that with judicious bus scheduling, buffer times can stop delay propagation and reduce battery SoC variability, or in other words, improve schedule robustness.

In Jiang et al. (2021), a robust E-MDVSP model with uncertain travel time, partial recharging, and time-of-use electricity tariff considering the cardinality constrained set is formulated. Travel time deviations affect both the recharging time, as delays can make original charging plans infeasible, and the energy consumption. It is assumed that the energy consumption is proportional to the travel time deviation. A solution to the E-MDVSP is deemed feasible when, in addition to the usual requirements of the MDVSP, as long as the travel times vary in the cardinality constrained set, (i) the departure time of each timetabled

trip is always respected and (ii) the vehicle SoC always lies between a given safety range. E-MDVSP instances of up to 466 trips, 3 bus routes, and 3 depots are solved to optimality using a branch-and-price algorithm.

Li et al. (2021) addressed the multi-depot vehicle location-routing-scheduling problem with multi-vehicle types under range uncertainty (MM-LRSP). They formulated the problem as a two-stage stochastic problem, where in the first stage routing, scheduling and location of charging infrastructure under different random range scenarios are generated and, in the second stage, ad hoc service to handle incomplete trips (due to energy shortage) are scheduled. The objective is to minimize the cost of the regular service, including costs incurred by the transport agency (i.e., operational costs, demand-loss, infrastructure investment, and emissions) and the passengers, and the cost for the ad hoc trips. The MM-LRSP is modeled by time-space-energy and passenger time-space networks. The former is an adaptation of the time-space network where each node is associated with a location, a time, and a SoC. A gradient descent algorithm is used to solve a range reliability-based stochastic reformulation of the original problem. This reformulation separates the two-stage stochastic problem into two independent problems, with the second problem being a MILP solved for each scenario. The results showed that accounting for range uncertainty in the MM-LRSP saved up to 20% of the total cost compared to the deterministic method for a real-world bus network of Hong Kong city.

2.6. Electric vehicle scheduling problem with battery degradation

BEBs' batteries and their replacement represent a significant portion of the cost of ownership of BEBs; this cost may even exceed the cost of charging in some cases (Pelletier et al., 2018). Packs need to be replaced periodically due to the gradual fading of lithium-ion battery capacity during charging and discharging cycles, what is often referred to as cycle aging, and during storage time, what is often referred to as calendar aging. A battery is often considered to have reached its end of life when its capacity has dropped to 70-80% of its initial capacity (Lam and Bauer, 2012; Zhang et al., 2019). Cluzel and Douglas (2012) reported that the typical lifespan of battery electric vehicles ranges between five to ten years, depending on the operating and storage conditions. After that period, the battery packs of the BEBs must be replaced, which results in high maintenance costs. From a price perspective, it is thus important to consider battery aging mechanisms when scheduling BEBs. This approach is also relevant from an environmental point of view, as the production of lithium-ion batteries represents about 20% of the CO_2 equivalent emitted during the construction of BEBs (Nordelöf et al., 2019). The use of battery packs must therefore be done in a rational way.

At the planning stage, battery degradation can be considered in the E-VSP. To the best of our knowledge, it has only been addressed by Zhang et al. (2021) and Zhou et al. (2022).

A first model for the E-VSP with battery degradation and non-linear charging profile is presented in Zhang et al. (2021). This model considers a bus network with one terminal. BEBs are only allowed to charge at the terminal, where it is assumed that a limited number of chargers are available, and buses are always fully charged. Furthermore, trips are assumed to be a loops starting and ending at the terminal (roundtrips). The authors defined the total operational costs of a vehicle schedule as the sum of charging fees, the cost of battery degradation, and a fixed cost to acquire the buses. The capacity fading model of Lam and Bauer (2012) is borrowed to approximate the cost of battery degradation. A branch-and-price algorithm tailored to the problem is proposed to solve exactly real-world instances of up to 160 trips. In addition, a case study in a Chinese city showed that this approach could achieve significant cost savings (10 to 27%) due to longer battery life.

The model of Zhou et al. (2022) extends the one of Zhang et al. (2021) by considering partial charging. The authors addressed the electric bus charging scheduling problem (EB-CSP), which involves assigning BEBs to trips and scheduling the charge periods for a single-terminal transit network with roundtrips. A mixed-integer nonlinear nonconvex program and a MILP approximation are introduced. The MILP uses linear functions to approximate the charging and battery degradation functions.

Zeng et al. (2022) also proposed a model for the EB-CSP (under predetermined bus-to-trip assignments) with battery degradation using peak-to-average power ratio, time-of-use electricity price, and battery wear cost.

Chapter 3

Article 1 - Predicting the probability distribution of bus travel time to measure the reliability of public transport services

Prologue

This work was presented at the 6th International Workshop and Symposium on Research and Applications on the Use of Passive Data from Public Transport (TRANSIT DATA) and was published in Transportation Research Part C: Emerging Technologies:

> Ricard, L., Desaulniers, G., Lodi, A., Rousseau, L.M., 2022. Predicting the probability distribution of bus travel time to measure the reliability of public transport services. Transportation Research Part C: Emerging Technologies 138, 103619.

The research direction of this article, namely the prediction of bus travel time, was initially proposed to me by Guy Desaulniers, Andrea Lodi, and Louis-Martin Rousseau. I suggested refining this direction towards probabilistic prediction models in order to be able to integrate these models into stochastic optimization models later on. I was responsible for handling the data, implementing the prediction models, running the tests, analyzing the results, and writing the article. Guy Desaulniers, Andrea Lodi, and Louis-Martin Rousseau revised and edited the article.

3.1. Introduction

In order to increase the ridership and attract new users, public transport agencies put increasing emphasis on improving the quality of the service they provide and particularly its regularity, also referred to as reliability (Ma et al., 2014). Studies show that a majority

of passengers put more value on a reduction of the travel time (TT) variability than on a reduction of TT itself (Bates et al., 2001). Reliability can be addressed at different levels, either during the strategic planning, the tactical planning or the operational planning stages and during operations. At the strategic planning level, adding reserved lanes for buses can increase the reliability of the service, while during operations, bus holding is a popular solution to alleviate risks of bus bunching. The latter consists of holding a bus at key locations along a bus trip if it is running ahead of time. However, service reliability is rarely taken into account at the tactical and operational planning levels, when the detailed planning of the service is computed (van Oort, 2011). The network design, the frequencies and/or timetables of buses, the vehicle schedules and the crew schedules are built during these stages, among other things (Desaulniers and Hickman, 2007). This work aims at providing tools to measure, and eventually improve, the reliability of one of the output of the service planning phase, namely vehicle schedules. These schedules are defined as a sequence of timetabled trips and waiting times starting and ending at the same depot, such that each travel is either a timetabled trip or a deadhead trip (e.g., between a depot and a terminal or between two terminals). A deadhead trip between two terminals enables the connection of two timetabled trips ending and starting at different terminals.

To assess the reliability of a vehicle schedule, Kramkowski et al. (2009) introduced the concept of delay tolerance, a term reused in the works of Amberg et al. (2019); van Kooten Niekerk (2018), among others. This concept is based on primary and secondary delays that we distinguish below. On the one hand, a primary delay (or exogenous delay) is a deviation from the planned duration of a timetabled trip caused by a disruption (e.g., bus bunching) or variability during operation. This type of delay cannot be avoided by scheduling decisions. Indeed, day-to-day disruptions and delays are considered unavoidable on the day of operation (Amberg et al., 2019; Kramkowski et al., 2009) due to the randomness of incidents and the variation in demand and capacity factors. Bus sharing the road with other road-based vehicles (e.g., cars, bikes and trucks) are likely to have even higher degrees of variability, because they are subject to the same - morning and evening peaks - traffic patterns (Comi et al., 2017). On the other hand, a secondary delay (or endogenous delay) occurs when the primary delays of previous trips using the same resource (e.g., vehicle or crew) cannot be absorbed during idle time and thus propagate to the next trip. If a trip starts on time, its secondary delay is null. Otherwise, it is equal to the delay at the departure. Scheduling decisions, that is, the allocation of timetabled trips to resources, can influence the expected secondary delays of timetabled trips. Thus, the delay tolerance of a vehicle schedule is measured by the average expected secondary delay of its timetabled trips.

Secondary delays are stochastic, meaning that the departure of a timetabled trip may be late on a given day and on time on the next day even if the two trips belong to the same vehicle schedule because the secondary delay of a trip depends on the TTs of the

previous trips covered in the schedule, which are also stochastic. TT variability is explained by yearly, monthly, day-to-day and hourly variability as well as vehicle-to-vehicle variability (Kumar et al., 2014; Büchel and Corman, 2018; Kieu et al., 2015). The following equations show the dependence between the secondary delay and the TT. Consider a vehicle schedule $s = \{v_1, v_2, ..., v_{m_s}\}$ with m_s trips planned on a given day. For notational conciseness, we denote the trip v_i by i directly in the following. The secondary delay R_i of a trip i is the difference between its actual departure time D_i and its planned departure time d_i(assuming that $D_i \geq d_i$), computed as

$$R_i = D_i - d_i. \tag{3.1.1}$$

The random variable D_i is a convolution of the previous trip's actual departure time (D_{i-1}), actual TT (T_{i-1}) and the minimum in-between time between trips $i-1$ and i ($l_{i-1,i}$):

$$D_i = max\{D_{i-1} + T_{i-1} + l_{i-1,i}, d_i\}, \quad i = 2, \ldots, m_s \tag{3.1.2}$$

$$D_1 = d_1. \tag{3.1.3}$$

Precisely, $l_{i-1,i}$ accounts for the deadhead travel between terminals, if the trip $i-1$ ends at a different terminal than the departure terminal of trip i, and the minimum break time for drivers. The duration of deadhead travels is stochastic, but for simplicity a fixed value for each pair of terminals is used in the following. This value is given by the operator.

In order to compute $\mathbb{E}(R_i)$, the expected secondary delay of trip i, we claim that the expected TT of trips $1, \ldots, i-1$ provide insufficient information. To illustrate this, let's have a look at a simple case. Consider trip 2 scheduled to start at $d_2 = 8{:}40$AM and preceded by trip 1 that has started at $D_1 = 8{:}00$AM. Let also $l_{1,2} = 5$ minutes. If the probability that the actual TT of trip 1 is equal to 34 minutes is $P(T_1 = 34 \text{ minutes}) = 0.75$ and the probability that it is equal to 38 minutes is $P(T_1 = 38 \text{ minutes}) = 0.25$, then the expected TT of trip 1 is $\mathbb{E}(T_1) = 35$ minutes. Thus, if we only consider $\mathbb{E}(T_1)$, we get that the expected secondary delay of trip 2 is $\mathbb{E}(R_2) = 0$ minute. However, considering the probability distribution of T_1, we get that $R_2 = 0$ minute with a probability of 0.75 and $R_2 = 3$ minutes with a probability of 0.25 and therefore $\mathbb{E}(R_2) = 0.75$ minutes. This example confirms that the correct way to compute the expected secondary delay takes into account the probability distributions of the TT. Moreover, and at a more fundamental level, since the planned duration of a trip i is usually set to a value close to $\mathbb{E}(T_i)$, by computing the expected secondary delays using the expected TTs, any potential delay propagation is ignored. To take into consideration the fact that some trips are more uncertain than others, we must compute the expected secondary delays based on the complete probability distributions of the TT. In reality, these

distributions are much more complex than the one presented in the above example, justifying the need to explore models for the prediction of the probability distributions of the TT.

There are two types of TT prediction: short-term and long-term. Both types can predict either a segment or a complete trip TT. The former is usually performed less than one hour before a trip and uses online information as well as external factors (e.g., weather). This type of prediction can be integrated to the operator's operations control system and provides online information to the users about the estimated arrival time of a bus. On the other hand, the long-term TT prediction can be performed a few days before the trip and helps for transit planning. In this work, we are interested in computing long-term TT predictions.

We frame the long-term prediction of the density of the TT (PDTT) as a supervised learning problem which aims at predicting, for each trip i in a set of unseen trips (test set), an estimate of the complete conditional probability density function (PDF) of its TT, $\hat{p}(T_i|\boldsymbol{x}_i)$, given x_i the set of characteristics of trip i. We assume that the conditional PDF of the TT does not depend on scheduling decisions, i.e., the TT uncertainty is exogenous to the resource allocation. The end-goal of this problem is to accurately estimate the expected secondary delays of trips in a test set. To this end, we perform simulations using the predicted probability distributions of the TT to approximate the true expected secondary delays. The model that generates the most accurate approximations of the expected secondary delays is selected. This information can then be used by the operator's schedulers to evaluate and compare vehicle schedules in terms of their reliability or by a computer to optimize over a large number of possible vehicle schedules. Probabilistic models are compared to a Random Forests (RF) model, which provided the most promising results among the three regression models studied in the work of Moreira et al. (2012). We introduce two types of probabilistic models, namely the similarity-based density estimation models and the smoothed logistic regression model for probabilistic classification, and present experimental results on a large-scale dataset of more than 41,000 trips and 50 bus routes. Our contribution is threefold:

— The state-of-the-art for long-term prediction of public bus TT is almost nonexistent (Moreira-Matias et al., 2015). This work tries to fill this gap and, in addition, it is to our knowledge the first work to propose probabilistic models for the long-term prediction of public bus TT.

— We propose a novel method to approximate the expected secondary delays based on the probability distributions of the TT.

— To the best of our knowledge, it is the first study in the field of public transport that empirically studies such a large number of bus route's TTs simultaneously. We hope this can make our results relevant to other bus networks.

The remainder of this paper is organized as follows. In Section 3.2, we review the literature on TT analysis. The dataset used for the PDTT is presented in Section 3.3. We

overview the main bus route characteristics and portray a preliminary analysis of the features. Section 3.4 describes the methodology that can be applied for the PDTT. A Monte Carlo simulation to compute the expected secondary delays based on the results of the models for the PDTT is introduced in Section 3.5. In Section 3.6, data preparation as well as features and parameters selection are presented, before the evaluation metrics are specified and the performance of all models for the PDTT is compared. Thereafter, in Section 3.7, a preview of an optimization model that uses the approximations of the expected secondary delays in an attempt to improve the reliability of bus schedules is presented. Section 3.8 summarizes our findings.

3.2. Related works

The introduction of automatic vehicle location (AVL) data has given rise to a flourishing number of studies in the field of public transport on speed, arrival time and TT analysis. Because TT and arrival time measures are closely related, studies on both measures are treated without distinction. Indeed, the arrival time A_i of a trip i is given by

$$A_i = D_i + T_i. \tag{3.2.1}$$

In this section, three topics are covered: long-term TT prediction, TT variability analysis and TT distribution modeling. The latter fits the TT distribution of trips that occurred during a given period in order to analyze the shape and nature of the PDF of the TT, without trying to predict future events, as in the PDTT. We extend the field of the first topic to all road-based transport, but the subsequent topics are restricted to the public transport field. Approaches proposed for the PDTT are inspired by lessons learned through the review of the literature on these topics.

Compared with the literature on short-term TT prediction, studies on long-term TT prediction are rare and to the best of our knowledge, only the works of Chen et al. (2020), Moreira et al. (2012) and Klunder et al. (2007) proposed or reviewed long-term TT prediction methods. In a survey on improving the planning of public transit using AVL data, Moreira-Matias et al. (2015) suggested that the long-term TT prediction should be valid for an horizon of at least the entire forecasting period. They divided models found in the literature for short-term TT prediction in four categories: (i) machine learning and regression, (ii) state-based and time-series, (iii) traffic theory-based and (iv) historical databased, and suggested that some regression algorithms applied for short-term TT prediction could be adapted for long-term TT prediction. Gradient boosting is an example of a model that was successfully applied to the short-term TT prediction and then adapted by Chen et al. (2018) to the long-term TT prediction of trips on a freeway segment in Taiwan. The features were ranked in order of relative importance: time of the day, day of the week, national holiday, day of

long / consecutive holiday, big event / activity, electronic tool collection fee promotion and narrowing of roadway. Three regression models, namely a Projection Pursuit Regression, a Support Vector Machine and a Random Forest, were compared by Moreira et al. (2012) for the long-term TT prediction of trips of one public bus route in Porto. With a basic pre-processing work, the Random Forest had better results, but was slightly outperformed by a Projection Pursuit Regression when the authors added an instance selection step. Klunder et al. (2007) trained a k Nearest Neighbors algorithm (kNN) with only time-based variables for the long-term TT prediction on a motorway network in the Netherlands.

The reasons for TT variability can be external or internal (Yetiskul and Senbil, 2012) and related to demand or capacity (Mazloumi et al., 2010) (see Figure 3.1). In an early study, Abkowitz and Engelstein (1983) suggested that shorter routes may have reduced TT variablity. Also, they reported that a running time deviation at the beginning of a route tends to propagate downstream. Hence, control actions to correct early deviations on a route could reduce TT variability. Strathman and Hopper (1993) reported that the afternoon peak period has higher TT variability, in particular because of the higher passenger demand. In a study on TT variability in the city of Ankara, Yetiskul and Senbil (2012) found major differences in regional TT variability and suggested that bus-stop spacing should depend on the neighborhood density. Comi et al. (2017) performed a time series decomposition of the TT and compared it to the temporal traffic patterns. The two are reported to have similarities. Also, the seasonality of the time series decomposition was most significant for the hour of the day.

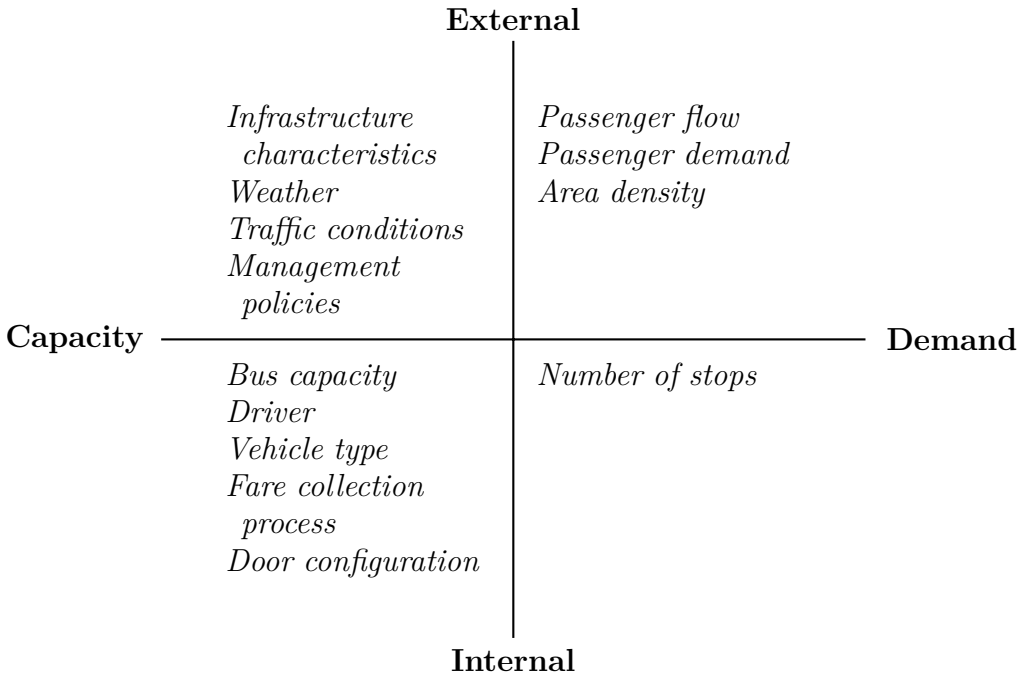


Figure 3.1 – Reasons of TT variability

TT distribution modeling has been studied mostly with the objective of quantifying the reliability of a transit service. Most works on TT distribution modeling in public transit occurred after the introduction of AVL systems. We focus our review on the work of Mazloumi et al. (2010), Ma et al. (2016) and Büchel and Corman (2018), as they are, in our view, the most comprehensive studies, from which useful lessons can be learned for the PDTT. Mazloumi et al. (2010) assessed the shape and nature of the TT distribution over the course of the day and for different levels of temporal aggregation on segments of a bus route. The authors measured the level of temporal aggregation by the length of the departure time windows (DTW), which are time slots for which trips departing during the slot are aggregated for subsequent analyzes (e.g., 15 minutes, 30 minutes or 1 hour). The study concluded that for shorter DTWs, the TT distribution follows a Normal distribution. For longer DTWs, this result holds for peak periods, but not for off-peak periods. For the latter, the Log-Normal distribution fits better. Also, the contribution of a set of features to the TT was assessed through a linear regression analysis. The land use (industrial vs. residential) and the length of the segment were those affecting the most the TT variance. Ma et al. (2016) studied intensively the influence of temporal and spatial aggregation on the TT distribution, with the objective of providing common grounds for modeling and evaluating the performance. To this end, several settings of temporal and spatial aggregations were assessed and an evaluation approach based on a statistical hypothesis test was proposed. The TT of a trip starting at stop i and ending at stop j was decomposed in dwelling times DT_k at each stop k of the trip and running times $RT_{(k,k+1)}$ between each pair of stops

$$T_{(i,j)} = \sum_{k=i}^{j-1} DT_k + RT_{(k,k+1)}. \tag{3.2.2}$$

Results concerning the normality of the TT distribution were in line with the ones of Mazloumi et al. (2010). Also, the analysis suggested that spatial aggregation tends to decrease the multimodality of TT distribution. A multimodal distribution is defined as a probability distribution with several modes. A Gaussian mixture model (GMM) was proposed to address the multimodality of the link level TT distribution. Büchel and Corman (2018) found that the Log-Normal distribution was, out of four unimodal statistical distributions, the best fit for the TT distribution modeling.

3.3. Data

Before introducing the data, it is essential to distinguish terms that are used in the following and that should not be confused, namely bus routes, bus lines, timetabled trips and trips. First, we define a bus route as an ordered sequence of road segments and bus stops, where the first and the last stops are called terminals. Second, a bus line usually has two associated routes, each one going in opposite directions (e.g., North-South or East-West

axis). Third, timetabled trips are generated during service planning, which is performed for typical days in the planning horizon. For example, service planning for the next two months can be reduced to planning for a typical weekday, Saturday and Sunday. A timetabled trip is associated with a given route, time and typical day and is valid for the planning horizon. It is therefore not associated with a given date. Fourth, trips are a unique event associated with a timetabled trip and a given date. Buses record trip data as they travel, so each data point in the dataset is associated with a trip.

The dataset used for this study was collected during a 2-month period from 08/28/2017 to 10/29/2017 by in-car Advanced Public Transport Systems (APTS) installed in buses running in the city of Montréal, Canada. Those systems collect automatically at every stop of a trip the corresponding trip identifier, route identifier, direction identifier, stop identifier, date, scheduled departure time, scheduled arrival time, actual departure time, actual arrival time and number of passengers loading or unloading, among other things. The scheduled departure, scheduled arrival, actual departure and actual arrival times are stored in milliseconds. The actual TT of a trip is the difference between its actual arrival time and its actual departure time at the terminals, whereas its primary and secondary delays are the differences between its actual TT and its scheduled TT and between its actual departure time and its scheduled departure time, respectively. Hence, for every trip, only the first and last stops (i.e., terminals) data is kept. Since the APTS were embedded in approximately 20% to 30% of the vehicles at that time, weekends and holidays had an insufficient number of trips recorded. Indeed, during weekends and holidays, the service is reduced and thus the number of trips recorded during those days is too small to conduct relevant data-driven analysis. For that reason, weekends and holidays are not studied and are removed from the dataset. After removing weekends and holidays, the dataset has more than 116,000 trips. Of the 408 routes in the dataset, only the 50 most frequent are kept for the remainder of the study, resulting in a dataset of over 41,000 trips. The 50 selected routes run between 4:00AM to 1:59AM (+1 day) during weekdays. To facilitate the notation, we add two extra hours to the usual 24-hour daily period. Thus, we say that the selected routes run between 4:00AM to 25:59PM.

Figure 3.2 shows the average secondary delay per scheduled departure hour and the average secondary delay plus the standard deviation (σ) of all 41,000 trips. Note that the secondary delay cannot be negative (see equations (3.1.1) - (3.1.3)) and thus the average secondary delay lies between 0 to 1.5 minutes all day long. The variability of the secondary delays is high in the late afternoon (approximately from 15:00PM to 18:59PM) likely because delays accumulate during the day and because this is a period of high mobility.

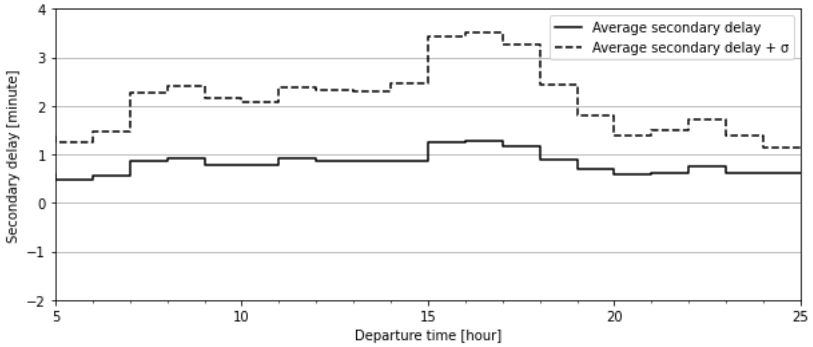


Figure 3.2 – Average secondary delay and variability per hour

3.3.1. Route's characteristics

The distribution of the average TT per scheduled departure hour is presented in Figure 3.3, where each piecewise linear curve represents the evolution of the average TT of a route. It is possible to distinguish the morning peak approximately from 6:00AM to 8:59AM, and the afternoon peak, approximately from 14:00PM to 17:59PM for all routes. The afternoon peak usually has an average TT higher than the morning peak and the average TT is generally decreasing after 17:00PM.

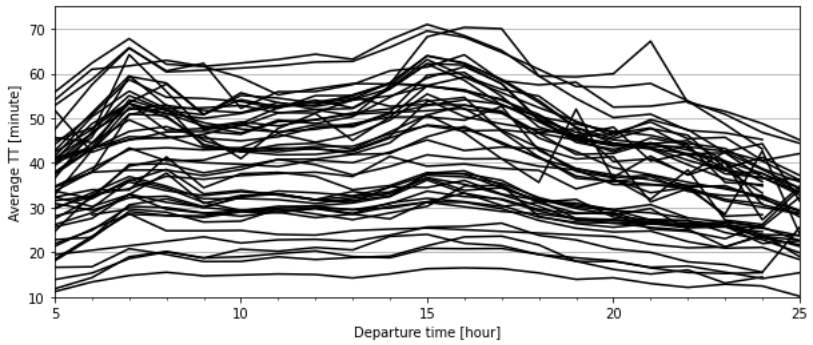


Figure 3.3 – Average TT per scheduled departure hour

The main characteristics of the 50 selected routes, namely the number of stops, the distance traveled and the type of operational region, are presented in Table 3.1. Each route has a unique combination of line identifier and direction, such that *A-East* and *A-West* are two different routes of the same bus line, but in opposite directions. We categorized the type of operational region in 6 categories: residential areas, crossing city center (CC), from city center (to a residential area), to city center (from a residential area), from an industrial (indust.) area (to a residential area), to an industrial area (from a residential area). Bus line B is the only one crossing the city center; it starts in a residential neighborhood, crosses the city center and ends in another residential neighborhood. Industrial areas are characterized by a high density of factories. The city center and industrial areas are usually regions where

a large number of people commute to work every day. The majority of bus routes operate in residential areas (32 out of 50). Those bus routes may, for example, connect two residential areas or a residential area to a subway or train station. The number of stops per bus route ranges from 17 to 74 stops, while the distance traveled ranges from 3.0 km to 15.3 km. In general, as the number of stops goes up, the distance traveled goes up as well. Line P is a counterexample, because it has a large number of stops close to each other. Note that the number of stops and the distance traveled of two routes of the same line are generally not equal, as the path in one direction is usually not symmetric to the path in the other direction (e.g., because some streets are one-way).

Table 3.1 – Characteristics of bus routes studied

Line	Dir.	#stops	Dist. (km)	Type of Region	Line	Dir.	#stops	Dist. (km)	Type of Region
A	East	52	13.9	Residential	M	West	18	4.0	Residential
A	West	49	14.5	Residential	N	East	28	5.3	Residential
B	East	46	13.2	Cross CC	N	West	29	5.3	Residential
B	West	46	12.0	Cross CC	O	North	35	3.0	To indust.
C	North	40	12.1	Residential	O	South	40	3.4	From indust.
C	South	45	12.3	Residential	P	East	74	9.0	Residential
D	North	33	10.3	From indust.	P	West	67	8.5	Residential
D	South	36	10.4	To indust.	Q	East	40	7.0	Residential
E	East	50	14.2	Residential	Q	West	38	7.0	Residential
E	West	52	13.3	Residential	R	East	37	5.3	Residential
F	East	34	7.8	Residential	R	West	35	5.3	Residential
F	West	36	7.7	Residential	S	East	47	11.8	From indust.
G	North	17	4.6	Residential	S	West	51	11.6	To indust.
G	South	19	4.3	Residential	T	North	34	8.5	To indust.
H	North	37	9.3	Residential	T	South	30	8.5	From indust.
H	South	40	10.8	Residential	U	North	46	11.1	Residential
I	East	71	15.3	Residential	U	South	42	10.7	Residential
I	West	68	15.3	Residential	V	East	46	9.5	Residential
J	North	28	7.1	From CC.	V	West	49	8.5	Residential
J	South	30	7.1	To CC	W	East	43	10.3	Residential
K	East	53	11.1	To CC	W	West	47	11.6	Residential
K	West	51	11.4	From CC	X	North	30	6.6	From CC
L	East	35	5.9	Residential	X	South	34	7.5	To CC
L	West	36	6.0	Residential	Y	North	30	8.0	To CC
M	East	18	4.4	Residential	Y	South	28	8.0	From CC

3.3.2. Features analysis

The PDTT has to be based upon features (i.e., explanatory variables) that are available a few days or weeks in advance. For example, meteorological conditions are likely to influence

the TT duration. However, since it is an information that is not available when solving service planning problems, it is not considered. Likewise, the TT of the previous trip is not considered. The list of possible features includes the day of the week, type of region, route identifier, distance, number of stops, scheduled departure time, week number and year. Possible values and types of features (categorical or non-categorical) are listed in Table 3.2.

Table 3.2 – Long-term features

Feature	Type	Possible values
Day of the week	Categorical	{Monday, Tuesday, ..., Friday}
Region	Categorical	{residential, crossing CC, ..., to indust.}
Route identifier	Categorical	{A East, A West, ..., Y South}
Distance (km)	Non-categorical	[3, 15.3]
Number of stops	Non-categorical	{17, 18, . . . , 74}
Scheduled departure time	Non-categorical	[4:00AM, 25:59PM]
Week number	Non-categorical	{35, 36, ..., 44}
Year	Non-categorical	{2017}

The feature year is discarded because our dataset is spread over 2017 only. The statistical significance of the features scheduled departure time, day of the week and week number can be analyzed visually by looking at Figures 3.3, 3.4 and 3.5, which present the average TT per route depending on each of the feature respectively. Figure 3.3 suggests that the scheduled departure time has a high importance. The relationship between the TT and the scheduled departure time is not linear. Generally, the average TT of a bus route increases during peak hours and is steady between the morning and the afternoon peaks. Second, Figure 3.4 suggests that the relationship between the TT and the day of the week is less important. This is hardly surprising given that Saturdays and Sundays are not considered. Interestingly, there is no common pattern between the routes; for example some routes have a slightly higher average TT on Tuesdays than on Mondays and Wednesdays, while some others have an inverse pattern (i.e., the average TT on Tuesdays is slightly lower than on Mondays and Wednesdays). Third, Figure 3.5 suggests that the relationship between the TT and the week number is significant only for a handful of bus routes. In Figures 3.3, 3.4 and 3.5, we can observe that the average TT differs greatly from one bus route to another, unsurprisingly as each route has its own characteristics (as discussed earlier).

3.4. Methodology

Several approaches have been proposed to compute the conditional PDF of a random variable in a probabilistic fashion. Gaussian process based models are among the most popular. The work of Dutordoir et al. (2018) proposed a Gaussian process-based model to estimate a conditional PDF using latent variables in order to model non-Gaussian probability

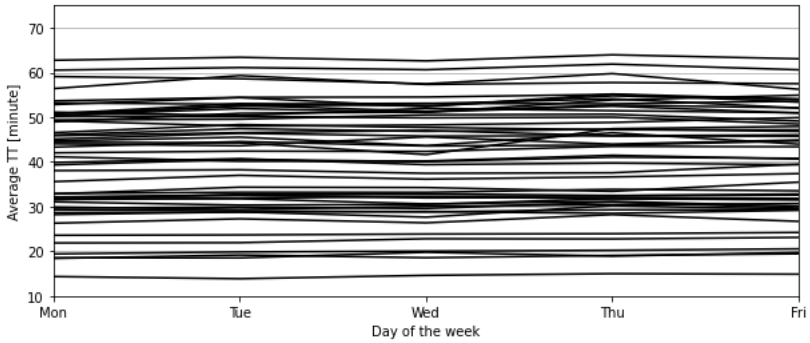


Figure 3.4 – Average TT per day of the week

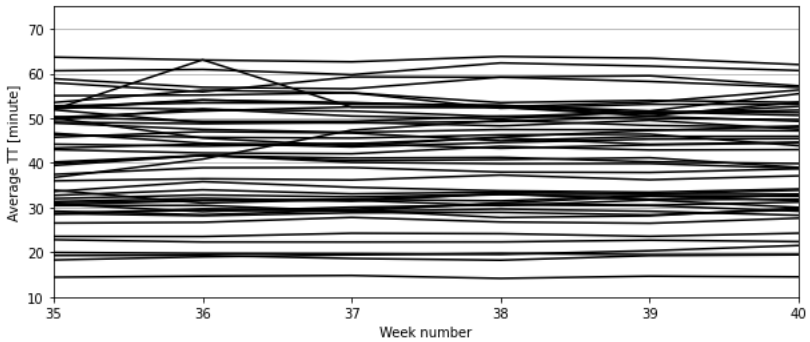


Figure 3.5 – Average TT per week number

distributions. Also, Bishop (1994) developed Mixture Density Networks, which is a type of artificial neural network predicting multimodal conditional density distributions. The main drawback of Mixture Density Networks is that they perform poorly when the size of the dataset is not large enough. In the work of Yeo et al. (2018), the prediction of a continuous PDF is converted into a classification task by using a discretization technique. This simplifies the learning task and traditional probabilistic classifiers can be used to predict the probability mass function, which can be smoothed later on into a PDF. The focus of this paper is on frequentist models (Koller and Friedman, 2009). In the remainder of this section two approaches for the PDTT are presented. The first one estimates the PDFs of the TT of a set of similar trips using parametric, semi-parametric or non-parametric density estimation models. The second approach, namely the smoothed Logistic Regression for probabilistic classification, is similar to that of Yeo et al. (2018), but fits a Logistic Regression instead of a Recurrent Neural Network estimator.

3.4.1. Similarity-based density estimation

Similarity-based density estimation models are a two-step process: for each trip, (1) find the set of similar trips and (2) estimate the density of this particular set, by fitting a

parametric, semi-parametric or non-parametric model. Next, we will define two similarity-based methods and introduce some density estimation models.

3.4.1.1. Similarity-based methods. Consider a trip i with a feature vector $(\boldsymbol{x}_1^{(i)}, \ldots, \boldsymbol{x}_d^{(i)})$. We want to select, based upon one of the following similarity-based methods, the set of trips in the (reduced) training set that have similar attributes:

— Equivalent DTW (eDTW) : select all trips from the same route that have a scheduled departure time in the same departure time window (DTW) (Mazloumi et al., 2010; Büchel and Corman, 2018; Ma et al., 2016) as trip i.

— k Nearest Neighbors (kNN): select the k nearest neighbors of the trip i. The distance between trip i and trip j is the Euclidean distance between their feature vectors and is computed as

$$dist(\boldsymbol{x}_i,\boldsymbol{x}_j) = \sqrt{\sum_{\ell=1}^{d}(x_\ell^{(i)} - x_\ell^{(j)})^2}. \tag{3.4.1}$$

3.4.1.2. Density estimation models. The conditional probability $p(T_i = t_i \mid \boldsymbol{x}_i)$ is estimated by fitting a given density estimation model on points close to i in the (reduced) training set, which are either trips in the same eDTW or close neighbors.

Parametric models. Parametric density estimation considers a restricted set of common probability distributions. Each of these distributions has a small number of parameters that have to be estimated from the data. We consider the Normal, Log-Normal, Logistic, Log-Logistic, Gamma, and Cauchy probability distributions. The Gamma distribution is a family of probability distributions containing the Exponential, Erlang and Chi-Squared distributions. In the work of Ma et al. (2016), the first four distributions were successful at modeling the TT distribution at a route level. For each trip, parameters of these probability distributions are found using the Maximum Likelihood Estimation (MLE) algorithm.

Semi-parametric model: Gaussian Mixture Model. GMMs are a sub-category of mixture models composed of K normal components. GMMs are relevant when the population modeled is multimodal and has undefined subpopulations or states, such that each component represents a state. It is common in transport to use three components, one for each of the traffic states: free flow, recurrent and non-recurrent traffic (Ma et al., 2016). The PDF of a K-components GMM is given by

$$\hat{p}(T_i = t_i \mid \boldsymbol{x}_i) = \sum_{k=1}^{K} \pi_{ik}\mathcal{N}(t_i \mid \mu_{ik}, \sigma_{ik}^2). \tag{3.4.2}$$

The vector of positively defined coefficients $\pi_i = (\pi_{i1}, ..., \pi_{iK})$, such that $\sum_{k=1}^{K} \pi_{ik} = 1$, and the vector of the model's k^{th} component's parameters $(\mu_{ik}, \sigma_{ik}^2)$, are found by applying the expectation maximization (EM) algorithm.

Non-parametric model: Kernel Density Estimation (KDE). A KDE model infers the PDF of a random variable based on a sample of its population. To estimate the PDF of a trip i, the model uses m points close to i in the training set. It is a data smoothing problem that allows to find, in a non-parametric fashion, the curve of the PDF given a sample. The Gaussian kernel, $K(\cdot)$, is the most widely used, but any function that integrates to unity ($\int K(t)dt = 1$) can replace it. The smoothness of the estimator is adjusted by the bandwidth parameter h as

$$\hat{p}(T_i = t_i \mid \boldsymbol{x}_i) = \frac{1}{m} \sum_{j=1}^{m} K\left(\frac{t_i - t_j}{h}\right). \tag{3.4.3}$$

3.4.2. Smoothed Logistic Regression for probabilistic classification (LR-PC)

Probabilistic classifiers are a type of machine learning model that can predict the probability that a given input belongs to a set of classes, instead of only predicting the class with the highest probability. When a numerical discretization is applied to the random variable T, such that the TT is categorized in bins of 1 minute, the PDTT task can be translated into a probabilistic classification one: estimate the probability that T takes a value that falls into class $c \in \{0, ..., C-1\}$. A model is fitted per bus route, because this setting yields better experimental results than learning a unique model for all bus routes (see Section 3.6.2).

A question arises: how to choose the number of classes for a bus route? Our approach was to use, for a given bus route, the difference between the trip in the training set with the shortest duration, t_{min}, and the trip with the longest duration, t_{max}, as C, the number of classes. Thus, $P(T_i = c)$ is the probability that T_i takes a value in $[c + t_{min}, c + 1 + t_{min}[$. We disregarded the fact that trips in the test set can have shorter or longer duration than t_{min} and t_{max} respectively, as the smoothing discussed later implicitly solves this issue.

Multinominal Logistic Regression is naturally probabilistic and is commonly used for probabilistic classification tasks. Classes' probabilities of a multinomial Logistic Regression are defined as

$$\hat{P}(\lfloor T_i \rfloor = c + t_{min} \mid \boldsymbol{x}_i, \mathbf{w}_c) = \frac{exp(\mathbf{w}_c^T \boldsymbol{x}_i)}{\sum_{c'=0}^{C-1} exp(\mathbf{w}_{c'}^T \boldsymbol{x}_i)}, \tag{3.4.4}$$

where $\mathbf{w}_c$ is the vector of parameters of the class c, found using a stochastic average gradient descent solver.

Logistic Regression outputs a probability mass function that can be smoothed into a PDF subsequently. As proposed by Yeo et al. (2018), the output of the multinomial Logistic Regression, which takes the form of a p.m.f., is passed through a one-dimensional convolution layer. This step has the effect of enforcing a spatial correlation in the output. The convolution layer is analogous to a KDE with a bandwidth h and a kernel $K(\cdot)$, but uses the probability mass function, $\hat{P}(\lfloor T_i \rfloor \mid \boldsymbol{x}_i, \boldsymbol{w}_c)$, instead of a sample of trips:

$$\hat{p}(T_i = t_i \mid \boldsymbol{x}_i) = \sum_{c'=0}^{C-1} \left[K\left(\frac{t_i - (c' + t_{min})}{h} \right) \times \hat{P}(\lfloor T_i \rfloor = c' + t_{min} \mid \boldsymbol{x}_i, \mathbf{w}_{c'}) \right]. \tag{3.4.5}$$

3.5. Simulation framework to measure the delay tolerance of a vehicle schedule

Consider again a vehicle schedule $s = \{1, 2, ..., m_s\}$ with m_s trips. After the schedule has been performed, the secondary delay R_i of a trip i can be computed using equations (3.1.1)-(3.1.3), as the actual value of the TT (T_{i-1}) and the actual departure time (D_{i-1}) of the previous trip are then available. However, before the schedule is performed, $\mathbb{E}(R_1), \mathbb{E}(R_2) \ldots, \mathbb{E}(R_{m_s})$ must be computed using the PDFs of the TT and it is impossible to do this exactly. Instead, we propose a Monte Carlo simulation of K iterations, where at each iteration a TT is randomly sampled, when it is possible, from $\hat{p}(T_i|\boldsymbol{x}_i)$ for each $i = 1, \ldots, m_s$ and delays are propagated from the first to the last trip in order to compute the secondary delay of each trip. Before going further, we must state the situation in which it is not possible to sample the TT. This situation occurs because trips in s may not be included in $\mathcal{B}$, the dataset of 41,000 trips and 50 bus routes presented in Section 3.3. A detailed explanation of the reasons for this is provided in Section 3.6.3 and for now just remember that this may be the case. If trip $i \notin \mathcal{B}$, then we have no information about the TT distribution of this trip and we have to use the scheduled duration of the trip directly. The scheduled duration of trip i is the difference between its scheduled arrival time a_i and its scheduled departure time d_i. After running the K iterations, it is possible to approximate the expected secondary delays of all the trips in the schedule. This approximation is given by:

$$\mathbb{E}(R_i) \approx \bar{R}_i = \frac{\sum_{k=1}^{K} R_i^k}{K}, \quad \text{for } i = 1, \ldots, m_s, \tag{3.5.1}$$

with R_i^k the secondary delay of trip i computed at iteration k. The exact expected secondary delay of trip i is approximated by $\bar{R}_i$, its average secondary delay over K iterations. At each iteration, the randomly sampled TTs of trips $1, \ldots, m_s$ must have a duration that lies between $MinTT_i$ and $MaxTT_i$, the smallest and the largest observed TTs of the trips on the same bus route as trip i. Otherwise, a new TT is sampled until that condition is fulfilled. In

other words, we truncate the PDF of the TT below and above the times never recorded and therefore for which we have no information.

The pseudo-code in Algorithm 1 summarizes the Monte Carlo simulation used to compute the approximation of the expected secondary delays for all trips of a schedule s. In essence, at each iteration of the simulation the TT of each trip in the schedule s is either sampled or set to the planned duration and the delays are propagated from the first trip to the last one. The simulation outputs the average secondary delay over K iterations for each trip $i \in s$. Note that we assume that the first trip of a schedule always starts on time, i.e., its secondary delay is null (see equation (3.1.3)).

Algorithm 1: Monte Carlo simulation to approximate the expected secondary delays

$\text{SumR}_i \leftarrow 0, \forall i \in s$
for $k \leftarrow 1$ **to** K **do**
  for trip $i \leftarrow 1$ **to** m_s **do**
    if trip $i = 1$ **then**
      $D_i \leftarrow d_i$
    else
      $D_i \leftarrow \max\{D_{i-1} + T_{i-1} + l_{i-1,i}, d_i\}$
      $\text{SumR}_i \leftarrow \text{SumR}_i + (D_i - d_i)$
    end
    if trip $i \in \mathcal{B}$ **then**
      repeat
        $T_i \leftarrow$ sample from $\hat{p}(T_i|\boldsymbol{x}_i)$
      until $\text{MinTT}_i \leq T_i \leq \text{MaxTT}_i$
    else
      $T_i \leftarrow a_i - d_i$
    end
  end
end
for each trip $i \in s$ **do**
  $\bar{R}_i \leftarrow \text{SumR}_i / K$
end

3.6. Experimental results

Using the dataset described in Section 3.3, we next explain how we fit probabilistic models for the PDTT, before comparing the performance of these models. First, we describe how

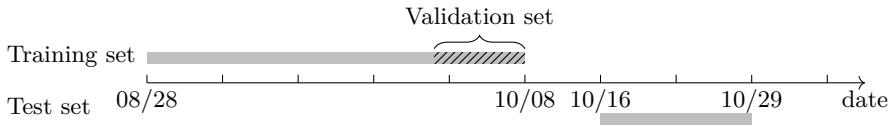
the data is filtered and split for the PDTT. Second, we go through features and parameters selection for each type of model and detail the selection of the temporal aggregation level. Third, metrics to evaluate the performance of the models on the test set are presented. Finally, all probabilistic models are compared to a Random Forests model in terms of their performance on the test set.

3.6.1. Data preparation

The dataset is filtered in order to remove erroneous information and special situations that we do not want to cover in the PDTT. Incomplete trips, vias and trips with detours are discarded. A via is a trip that deviates from the main trip or an express trip. To remove erroneous trips which were not filtered by the previous step, the Median Absolute Deviation or MAD (Hellerstein, 2008) with a 6-delta criterion is used. A trip is discarded if it has a TT longer or shorter than the median TT value of the trips associated with the same route plus (minus) 6 times the corresponding standard deviation. This method also removes trips with extended TT due to exceptional scenarios (e.g., bus failure) that the PDTT problem should not cover, because when such exceptional scenarios occur, an additional bus is usually dispatched to recover the schedule and prevent severe delay propagation.

We use a hold-out method and split the dataset in two sets: a training and a test set. The validation set, a hold-out subset of the training set, is used for features and parameters selection and the test set is used for model evaluation. Figure 3.6 summarizes the dataset split. We split the dataset such that the training data starts on 08/28/2017 and ends on 10/08/2017. The set of test data is composed of the trips from 10/16/2017 to 10/29/2017. Hence, a complete week, from 10/09/2017 to 10/16/2017 is discarded to simulate real-life settings where the planning is done at least a few days ahead. The training set is split again in a validation set and a reduced training set by slicing the last trips recorded per route from the original training set. The reduced training set contains 80% of the trips in the original training set and the validation set contains the remaining 20%.

Figure 3.6 – Dataset split



3.6.2. Models training

In the training process, the features and parameters of the estimator $\hat{p}(\cdot)$ are selected by fitting each model to the reduced training set and evaluating them on unseen data in the

validation set. The performance of all the probabilistic models is evaluated by the negative log-likelihood (NLL) score over the validation set (containing n_{val} points), computed as

$$NLL_{val} = -\sum_{i=1}^{n_{val}} \log(\hat{p}(t_i|\boldsymbol{x}_i)), \tag{3.6.1}$$

with t_i the true TT of trip i. The performance of the Random Forests model, for its part, is evaluated by the mean squared error (MSE) of the output. The pair of features and parameters which obtains the best performance on the validation set is selected.

Similarity-based density estimation models use one of the similarity-based methods, namely the eDTW or the kNN method. While the eDTW method does not require feature selection, as the features used are always the route identifier and the scheduled departure time, the kNN method does require feature selection. Indeed, the distance between neighbors depends on the specified feature vector. Estimating the TT density of a trip i does not require feature selection; it fits a probability density to a sample containing trips similar to trip i. Thus, here the features selection problem is reduced to finding a feature vector for the kNN method.The selection of features is carried out in parallel with the selection of parameters. The parameters of the similarity-based methods are the DTW duration for the eDTW method and the number k of neighbors for the kNN method. For the KDE, the validation set is used to select the bandwidth h and the kernel function.

We found that similarity-based density estimation models and the LR-PC model fit the data better when they are fitted per bus route. By doing so, the features describing the bus route characteristics, namely the number of stops, distance traveled, route identifier and type of region, become uninformative to the model. Indeed, all trips used to train the model of a given bus route have exactly the same values for these features. The remaining features to consider are the scheduled departure time, the week number and the day of the week.

To select the features of the similarity-based density estimation using kNN models as well as the LR-PC and the Random Forests models, we applied the permutation feature importance technique (Breiman, 2001). The latter reports the statistical significance of a set of possible features by measuring the increase of a predictor score when the values of a feature are permuted. The importance of a feature ℓ is the difference between a model's score over the original dataset and the average (over 10 shuffles) score over a corrupted dataset (with the values of the feature ℓ permuted). Results of this analysis are presented in Table 3.3. In line with the preliminary feature analysis presented in Section 3.3.2, the results indicate that the scheduled departure time has a higher statistical significance than the week number and the day of the week. For all models, features with a relative statistical significance of more than 1.00% are selected. Thus, for all similarity-based density estimation models except the Cauchy with kNN and the LR-PC model, the scheduled departure time and the week number

are selected. Only the scheduled departure time is selected for the Cauchy with kNN and the LR-PC models. The number of stops, the distance, the scheduled departure time, the route identifier, the week number and the region are selected for the Random Forests model.

Table 3.3 – Relative statistical significance (%) of features. Cau.- Cauchy, Gam.- Gamma, Nor.- Normal, L.-N.- Log-Normal, Log.- Logistic, L.-L.- Log-Logistic

	kNN*								
Feature	Cau.	Gam.	Nor.	L.-N.	Log.	L.-L.	KDE	LR-PC	RF
Number of stops	-	-	-	-	-	-	-	-	59.03
Distance	-	-	-	-	-	-	-	-	17.75
Sched. dep. time	99.74	96.71	97.02	97.01	97.98	98.21	97.43	99.51	17.37
Route identifier	-	-	-	-	-	-	-	-	2.25
Week number	0.50	2.47	2.10	2.32	1.61	1.49	1.69	-0.46	2.09
Region	-	-	-	-	-	-	-	-	1.20
Day of the week	-0.24	0.81	0.88	0.67	0.41	0.30	0.87	0.95	0.30

Temporal aggregation is a fundamental aspect of the PDTT since it has been shown to affect the shape and nature of the TT probability distribution (Mazloumi et al., 2010; Ma et al., 2016). Thus, the parameter associated with it, namely the DTW duration, is studied carefully. We select the DTW duration by analyzing how the density estimation models perform on the validation set for different levels of temporal aggregation. DTWs considered are, going from the most aggregated to the least aggregated, 5 periods per day (before morning peak, morning peak, in-between morning and afternoon peaks, afternoon peak and after afternoon peak), 60 minutes and 30 minutes. Table 3.4 compares the performance of models using the eDTW method, with the values in bold indicating the best NNL of the validation set for each level of temporal aggregation. For all models except the GMM, the NLL score over the validation set is better at DTWs of 60 minutes. The Log-Logistic model obtains the best results at all aggregation levels, matched by the Gamma, Log-Normal and KDE models at an aggregation level of 5 periods per day. The GMM has a similar performance to the parametric models for the most aggregated level, but it also has a poor performance for lower levels of temporal aggregation, both 60 and 30 minutes. Since the performance on the reduced training set is good, it indicates that the GMM overfits the training data. For the DTWs considered, the conditional PDF of the TT is most likely not multimodal. Thus, this model is discarded for the rest of the study. Interestingly, the training NLL decreases when the temporal aggregation level increases for all models except the KDE, while the NLL over the validation set increases from DTWs of 60 minutes to DTWs of 30 minutes for all models. This suggests that models are overfitting more at DTWs of 30 minutes than at DTWs of 60 minutes.

Table 3.4 – NLL (lower is better) of the similarity-based density estimation models using eDTW method with different levels of temporal aggregation on the validation set

	5 periods		60 minutes		30 minutes	
Model	Train	Validation	Train	Validation	Train	Validation
Cauchy	3.24	3.30	2.72	2.89	2.60	2.96
Gamma	3.08	**3.15**	2.58	2.83	2.47	5.98
Normal	3.09	3.16	2.59	2.87	2.48	6.57
Log-Normal	3.09	**3.15**	2.57	2.81	2.46	5.72
Logistic	3.09	3.16	2.59	2.77	2.48	2.91
Log-Logistic	3.09	**3.15**	2.58	**2.75**	2.47	**2.88**
GMM	3.07	3.16	2.26	5.81	1.82	24.96
KDE	2.99	**3.15**	2.49	2.76	2.77	2.89

We can conclude that, between the three levels of temporal aggregation compared, the best one is the one with DTWs of 60 minutes. We denote the eDTW method with DTWs of 60 minutes as eDTW*. For the second similarity-based method, namely the kNN, the value of k can be chosen similarly to the duration of DTWs, by assessing the performance of the similarity-based density estimation models on the validation set for different values of k. Figure 3.7 shows that the NLL over the validation set decreases significantly for all models when the value of k increases up to approximately $k = 13$. After that, the NLL stays almost constant. Thus, we set the number k of neighbors to 13 and denote the kNN method with $k = 13$ as kNN*.

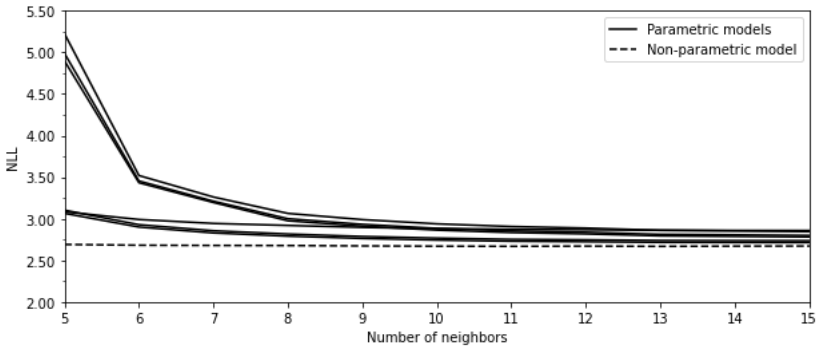


Figure 3.7 – NLL (lower is better) of the similarity-based density estimation models using kNN method with different numbers of neighbors

The LR-PC model yields better performance when it considers transformations of the scheduled departure time to capture a non-linear relationship with the TT. First, the scheduled departure time is categorized in bins of 1 hour and 30 minutes using a one-hot encoding. Second, the sine and cosine of the scheduled departure time are computed. The total dimension of the feature vector is 69 (22 for the one-hot encoding of bins of 1 hour, 44 for the one-hot encoding of bins of 30 minutes, 2 for the sine and cosine of the scheduled departure

time and 1 for the original scheduled departure time). The bandwidth and the kernel function of the LR-PC model are selected based on the performance on the validation set, along with the regularization strength of the Logistic Regression.

The number of trees in the Random Forest model, the maximum number of features considered when branching, the maximum depth of each tree and the minimum number of samples required to split an internal node are selected based on the performance on the validation set.

3.6.3. Models evaluation

After models training, the reduced training set is combined with the validation set and each model is trained on the complete training set using their selected features and parameters. The performance of the models for the PDTT is evaluated by the NLL score and the MSE of the expected secondary delay over the test set. The NLL score over the test set is analogue to the NLL over the validation set (NLL_{val}) and is computed as

$$NLL_{test} = -\sum_{i=1}^{n_{test}} \log(\hat{p}(t_i|\boldsymbol{x}_i)). \tag{3.6.2}$$

It quantifies the likelihood of the PDFs of the TT predicted by the models, with respect to the test points. The second metric, the MSE of the expected secondary delay over the test set, measures the accuracy of the approximation of $\mathbb{E}(R_i)$ for $i = 1, \ldots, n_{test}$. It is given by

$$MSE_R = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} \left(r_i - \bar{R}_i\right)^2, \tag{3.6.3}$$

with r_i the true secondary delay of trip i and $\bar{R}_i$ the average predicted secondary delay of trip i, for $i = 1, \ldots, n_{test}$. The latter values are obtained by running the simulation presented in Section 3.5. This simulation takes as input a complete vehicle schedule $s = \{1, \ldots, m_s\}$, the scheduled departure and arrival times of trips $i = 1, \ldots, m_s$ and the probability distributions of the TT of trips $i \in s$ for which the information is available. Indeed, as mentioned in Section 3.3, the original dataset of 166,000 trips and 408 bus routes is reduced to a dataset of 41,000 trips and the 50 most frequent bus routes. Thus, many, if not nearly all, of the trips in the test set are part of a vehicle schedule that contains some trips that are discarded and for which we therefore have no information about the uncertainty of their TTs. In order to be able to propagate delays from the first trip to the last trip of s using the recursive equations (3.1.1)-(3.1.3), the TTs of the trips for which no information is available are considered deterministic and equal to the planned duration.

3.6.4. Models comparison

Table 3.5 presents the NLL and the MSE of the expected secondary delay of all models over the test set. On the one hand, it is interesting to see that models using a non-Gaussian probability distribution, either a Gamma, Log-Normal, Logistic or Log-Logistic distribution, yield a lower NLL over the test set than those using a Normal distribution, which calls the normality of the conditional PDF of the TT into question. For the parametric models, the Cauchy, Logistic and Log-Logistic distributions have better NLL score over the test set when using the kNN* method, while the Gamma, Normal and Log-Normal distributions have rather the opposite results. The KDE models also have a better NLL over the test set when using the kNN* method than when using the eDTW* method. Overall, the Log-Logistic with kNN* model is the one which yields the best test NLL.

On the other hand, all parametric models predict more accurate expected secondary delays when using the kNN* method. On the contrary, the KDE models generate more accurate expected secondary delays when using the eDTW* method. The KDE using kNN* method and the LR-PC models surprisingly generate poorly accurate approximations of expected secondary delays, even though they achieved good NLL scores. It appears that smooth distributions, like parametric distributions, produce better approximations of the expected secondary delays. From the perspective that the end-goal of the PDTT is to approximate the expected secondary delays, the Log-Logistic with kNN* model should be selected because it yields the lowest MSE of the expected secondary delay, with a MSE of 4.31. Furthermore, the good test NLL of this model confirms this choice.

Similarity-based density estimation models and the LR-PC model are compared with two different interpretations of the Random Forests model. The first interpretation, which is our benchmark, considers the Random Forests model as non-probabilistic, i.e., used exclusively for *point prediction* (Dutordoir et al., 2018). The emphasis is on modeling the mapping between an input $\boldsymbol{x}$ to its output y rather than on predicting the conditional PDF $p(y \mid \boldsymbol{x})$ (Dutordoir et al., 2018). For this interpretation, the NLL over the test set is not computed because the prediction is not a PDF, but rather a point. Moreover, the expected secondary delays are computed slightly differently than for the other models because the TTs are considered deterministic in this interpretation. Thus, the expected secondary delays can be computed directly using equations (3.1.1) - (3.1.3) and without using the simulation presented in Section 3.5. As shown in Table 3.5 (see row "Random Forests (*point prediction*)"), this interpretation of the Random Forests is outperformed by all the models in terms of the MSE of the expected secondary delay. Thus, the experimental results show that there is an added value in modeling the conditional PDF of the TT using probabilistic models. The second interpretation considers the Random Forests model as probabilistic. In order to turn the Random Forests model into a probabilistic model compatible with our framework,

Table 3.5 – NLL (lower is better) and MSE (lower is better) of the expected secondary delay of similarity-based density estimation, LR-PC and Random Forests models on the test set

Model	Similarity method	NLL_{test}	MSE_R
Cauchy	eDTW*	2.87	4.44
	kNN*	2.84	4.33
Gamma	eDTW*	2.79	4.45
	kNN*	2.80	4.33
Normal	eDTW*	2.82	4.47
	kNN*	2.84	4.37
Log-Normal	eDTW*	2.77	4.43
	kNN*	2.78	4.33
Logistic	eDTW*	2.74	4.42
	kNN*	2.73	4.34
Log-Logistic	eDTW*	2.72	4.39
	kNN*	**2.70**	**4.31**
KDE	eDTW*	2.75	4.41
	kNN*	2.73	4.43
LR-PC	-	2.71	4.72
Random Forests (*point prediction*)	-	-	4.75
Random Forests (probabilistic interpretation)	-	2.83	4.59

*DTWs = 60 minutes or $k = 13$

we proceed as follows. We assume the conditional PDF $p(y|x)$ is normally distributed with constant variance σ^2, that is, $p(y|x) = \mathcal{N}(\mu(x),\sigma^2)$. Then, because the RF model is trained to optimize a sum-of-squares loss function, both the data-dependent mean $\mu(x)$, and the constant variance σ^2 can be recovered from the Random Forest model, by using the prediction of the Random Forest model for $\mu(x)$, and the mean squared error of the model for σ^2. The results of this second interpretation can be found in row "Random Forests (probabilistic interpretation)" in Table 3.5. Note that the MSE of the expected secondary delay over the test set is better for this second interpretation than for the *point prediction* one, supporting our earlier assertion about the added value of probabilistic models. However, this interpretation is still outperformed by all models except the LR-PC model, showing the poor adequacy of the Random Forests model with a probabilistic interpretation to actual TTs and delays. One possible interpretation could be that the Random Forests model, which assumes Gaussian noise, performs poorly both in terms of the NLL and the MSE of the expected secondary delay over the test set because it assumes assumes Gaussian noise. However, as outlined by the NLL over the test set of parametric models before, the conditional PDF of the TT is most likely not Gaussian. Another key factor that might explain the dominance of the Random Forests model by truly probabilistic models could be that the variance of probabilistic

models is not constant for every trip, whereas it is the case for the Random Forests model as we interpreted it. In sum, because both interpretations of the Random Forests model, the non-probabilistic and the probabilistic ones, generate less accurate expected secondary delays than truly probabilistic models, the latter prevail for the prevision of the expected secondary delays of bus trips.

3.7. Preview of an integration in an optimization problem

The conditional PDF of the TT can be integrated in many service planning problems in an attempt to improve the delay tolerance of the service. For example, we are currently working on a variant of the vehicle scheduling problem with stochastic TTs that aims at computing vehicle schedules based on the expected secondary delay of their timetabled trips. The complete methodology and results of this work will be presented in a subsequent work. Nevertheless, we provide below a preview of the formulation of this optimization problem for interested readers.

The vehicle scheduling problem has been widely studied over the last half-century (Bunte and Kliewer, 2010) and consists of assigning vehicles to cover a set of timetabled trips, in such a way that every timetabled trip is covered exactly once and at minimal costs. When the operator's fleet is spread in two or more depots, it is referred to as the Multiple Depot Vehicle Scheduling Problem (MDVSP). We introduce an extension of the MDVSP, namely the reliable MDVSP with stochastic TTs, that exploits the long-term prediction of the PDFs of the TT studied in this work. This model takes the set $\mathcal{V}$ of n timetabled trips and the long-term prediction of the PDF of the TT of each of these timetabled trips in input in order to output cost-efficient and delay tolerant vehicle schedules. Let D be the set of depots, S the set of all feasible vehicle schedules, and S^d the subset of schedules starting and ending at depot d. The problem is to find a subset of vehicle schedules in S that covers exactly once each timetabled trip while respecting the number of available buses b_d at each depot $d \in D$ and minimizing a weighted sum of the total planned vehicle operating cost and the total expected secondary delay. To formulate this problem, we define for each timetabled trip $i \in V$ and schedule $s \in S$ a binary parameter a_{is} which is equal to 1 if schedule s covers timetabled trip i and 0 otherwise, and denote by c_s the cost of schedule s (including delay penalties). Furthermore, we introduce for each schedule $s \in S$, a binary variable y_s that takes value 1 if schedule s is selected in the solution and 0 otherwise. The reliable MDVSP with stochastic TTs can then be expressed as the following integer linear program:

$$\min \quad \sum_{s \in \mathcal{S}} c_s y_s \tag{3.7.1}$$

$$\text{s.t.} \quad \sum_{s \in \mathcal{S}} a_{is} y_s = 1, \quad \forall i \in \mathcal{V} \tag{3.7.2}$$

$$\sum_{s \in \mathcal{S}^d} y_s \leq b_d, \quad \forall d \in \mathcal{D} \tag{3.7.3}$$

$$y_s \in \{0,1\}, \quad \forall s \in \mathcal{S}. \tag{3.7.4}$$

Constraints (3.7.2) ensure that each timetabled trip is covered by a selected vehicle schedule, whereas constraints (3.7.3) impose vehicle availability at each depot.

The objective function (3.7.1) minimizes the total cost of the selected schedules which combines planned operational costs and delay penalties. The planned costs usually include a fix cost per vehicle used and a variable cost that depends on the traveled distance and the waiting time attended by a bus driver. Consider a vehicle schedule $s = \{1, 2, ..., m_s\} \in \mathcal{S}$ of m_s timetabled trips. The total cost c_s of the vehicle schedule s is a weighted sum of the planned costs q_s and the sum of the expected secondary delay $\mathbb{E}(R_i)$ of all timetabled trips i covered by the schedule s, weighted by a factor β:

$$c_s = q_s + (\beta \sum_{i=1}^{m_s} \mathbb{E}(R_i)). \tag{3.7.5}$$

Note that the PDTT is trained and tested using trips data whereas the reliable MDVSP with stochastic TTs deals with timetabled trips. Fortunately, the selected model, namely the Log-Logistic with kNN*, can easily compute the PDF of the TT of timetabled trips, used to approximate the expected secondary delays. To that end, the model training is done as presented in Section 3.3.2 using data on past trips. The selected model for the PDTT is based on three features, the route identifier, the scheduled departure time and the week number. Thus, the prediction of the PDF of the TT of a timetabled trip $i \in \mathcal{V}$ depends on its feature vector $x_i = (x_1^{(i)}, x_2^{(i)}, x_3^{(i)})$, with $x_1^{(i)}$, $x_2^{(i)}$ and $x_3^{(i)}$ the route identifier, the scheduled departure time and the week number of timetabled trip i, respectively. Since timetabled trips are not associated with a given date, it is not clear how to define $x_3^{(i)}$, the week number. However, it is straightforward to see that, for the model selected, the PDF of the TT of a timetabled trip i is the same regardless of the value of $x_3^{(i)}$, as long as it is a week number in the future planning horizon. Thus, the PDF of the TT of a timetabled trip can be computed by setting its week number to any week number in the future planning horizon. Then, an approximation of the expected secondary delay $\mathbb{E}(R_i)$ of all timetabled trips i covered by a given schedule s can be computed based on the PDFs of the TT by running the Monte Carlo simulation detailed in Section 3.5.

Since the number of feasible schedules is typically huge, it is impossible to enumerate them all. Instead, schedules are generated using column generation. The costs c_s are computed during the solution process by taking into account the PDFs of the TT.

When solving the reliable MDVSP with stochastic TTs, the scheduled departure and arrival times are fixed for every timetabled trip. Thus, the choice of timetabled trip connections is the only lever to tackle reliability. The sequence of timetabled trips in each vehicle schedule must take into account the uncertainty of the TT of these trips, i.e., the connection between an uncertain timetabled trip and the next should allow enough idle time to avoid delay propagation. The β factor can be modulated according to the operator's level of aversion to delay propagation. Of course, the higher the β factor, the greater the tolerance to delay, but the higher the planned costs.

3.8. Conclusions

In public transport, reliability has become a key challenge for operators wishing to attract new users. In this work, we proposed a method to measure, in order to eventually improve, the reliability of bus schedules. To that end, we presented a simulation model to approximate the delay tolerance of a vehicle schedule based on the long-term conditional PDF of the TT. We framed the prediction of this conditional probability distribution, that we referred to as the PDTT, as a supervised learning problem. We verified if probabilistic models could predict more accurately the complete conditional PDF of the TT and generate more accurate approximations of the expected secondary delays than a Random Forests model. In fact, the latter is not inherently probabilistic and is typically used for *point prediction*. Also, we compared the performance of several probabilistic models.

To train and test the PDTT models, we used a 2-month dataset collected by buses equipped with APTS in the city of Montréal, involving 50 bus routes and a total of over 41,000 trips. The bus routes studied have various attributes (e.g., number of stops, frequency, traveled distance, etc.) and constitute a diverse sample from which we hope to obtain results relevant to other bus networks. Based on previous works on TT variability analysis, we determined a set of features, the number of stops, distance, scheduled departure time, route identifier, week number, type of region and day of the week, which we ranked in order of statistical significance for each model.

We proposed two types of probabilistic models for the PDTT, namely similarity-based density estimation models and the LR-PC model. The former is a two-step process that firstly find, for each trip, the set of similar trips and then estimate the density of this set using parametric, semi-parametric or non-parametric density estimation models. We proposed two types of similarity-based methods, namely the eDTW and the kNN, for which the temporal aggregation level and the number of neighbors had to be set, respectively. The LR-PC model

applies a numerical discretization to the TT before fitting a Logistic Regression classifier per bus route. The output of the Logistic Regression is then smoothed into a PDF using a convolution layer analogous to a KDE.

Previous works on TT distribution modeling indicated that the level of temporal aggregation greatly affects the shape and nature of the TT distribution. Thus, we carefully selected the DTW duration based on the performance on the validation set. The GMM model had a poor performance for DTWs of 60 minutes and 30 minutes and thus we concluded that the conditional PDF of the TT is most likely not multimodal. This result is aligned with the one of (Ma et al., 2016) which observed that the multimodality of the TT decreases with spatial aggregation.

Models were compared in terms of both of their NLL and their MSE of the expected secondary delay over a test set. The first metric measures the likelihood of the probability distribution of the TT predicted while the second metric measures the accuracy of the approximations of the expected secondary delays outputted by the simulation model using the predicted probability distributions of the TT. From all the models tested, the density-based estimation model using kNN method and a Log-Logistic distribution yielded the best NLL and MSE of the expected secondary delay over the test set. Precisely, it produced approximations of the expected secondary delays that are about 9% more accurate than the benchmark model, the Random Forests. This result indicates that there is an added value in modeling the conditional PDF of the TT using probabilistic models. In particular, probabilistic models account for the variability of the TT whereas the Random Forests model does not intrinsically. Also, the Random Forests model as we interpreted it assumes, as many other *point prediction* models, that the noise of the TT is Gaussian. However, the normality of the TT was questioned because several similarity-based density estimation models using parametric distributions had better NLL over the test set than the models using the Normal distribution.

The Log-Logistic with kNN model generated accurate approximations of the expected secondary delays that schedulers can use to compare few bus schedule alternatives in terms of their reliability or to recommend changes to customers in the service planning parameters (e.g., minimum idle time between timetabled trips). Moreover, the expected secondary delays can be used to solve a reliable version of the MDVSP. We introduced this problem and proposed an integer programming model for it. In a forthcoming paper, we will propose a column generation algorithm for solving it.

3.9. Corrections

In this section, we present supplementary computational experiments aimed at rectifying two errors identified in the methodology of the article. It is worth noting that while these

errors, as explained later, do not significantly impact the conclusions drawn from this study, addressing them contributes to the overall rigor of our analysis.

The first error lies in the choice of the distance metric within the kNN similarity-based method (see Section 3.4.1). Our decision to use the Euclidean distance falls short, particularly due to the diverse nature of the feature vectors composed of mixed-type data (both categorical and non-categorical). A more fitting alternative would be the Gower distance (Gower, 1971), which takes mixed data types into account. The Gower distance between the feature vectors of two trips i and j is computed as

$$dist_{Gower}(\boldsymbol{x}_i,\boldsymbol{x}_j) = \sqrt{1 - \frac{\sum_{\ell=1}^{d} s_{ij\ell}\delta_{ij\ell}}{\sum_{\ell=1}^{d} \delta_{ij\ell}}}, \tag{3.9.1}$$

where $\delta_{ij\ell}$ takes value 1 if $x_{i\ell}$ and $x_{j\ell}$ can be compared and 0 otherwise and $s_{ij\ell}$ is the similarity between $x_{i\ell}$ and $x_{j\ell}$. If the values are categorical, the similarity is defined as the indicator function

$$s_{ij\ell} = \begin{cases} 1 & \text{if } x_{i\ell} = x_{j\ell} \\ 0 & \text{else,} \end{cases} \tag{3.9.2}$$

and if the values are numerical, it is defined as

$$s_{ij\ell} = 1 - \frac{|x_{i\ell} - x_{j\ell}|}{r_\ell}, \tag{3.9.3}$$

where r_ℓ is the range of the feature ℓ.

The second error pertains to a modeling choice in the LR-PC model (see Section 3.4.2). Specifically, we opted for a multinomial Logistic Regression model to predict travel time probabilities. Here, the classes correspond to discrete 1-minute intervals. However, the multinomial Logistic Regression model is designed for nominal dependent variables and relies on the assumption of the independence of irrelevant alternatives (IIA). In our context, the dependent variable is ordinal, implying it can be ordered in a meaningful way, and the labels are not independent. As a result, a more appropriate model for computing class probabilities would be the ordered multinomial probit model (see, e.g., Daykin and Moffatt, 2002). This model is explicitly tailored for ordinal outcomes and free from the assumption of IIA.

We conducted additional experiments to address the aforementioned corrections. Specifically, we explored two distinct approaches: the kNN similarity-based method using the Gower distance and a smoothed ordered multinomial probit model for probabilisitic classification (OP-PC). Data preparation, models training, and models evaluation were carried out as explained in Section 3.6.1, 3.6.2, and 3.6.3, respectively. The updated results are presented in Table 3.6.

Table 3.6 – NLL (lower is better) and MSE (lower is better) of the expected secondary delay of similarity-based density estimation, LR-PC, OP-PC, and Random Forests models on the test set - Corrections

Model	Similarity method	NLL_{test}	MSE_R
Cauchy	eDTW*	2.87	4.44
	kNN*	2.85	4.33
Gamma	eDTW*	2.79	4.45
	kNN*	2.76	4.27
Normal	eDTW*	2.82	4.47
	kNN*	2.79	4.30
Log-Normal	eDTW*	2.77	4.43
	kNN*	2.75	4.26
Logistic	eDTW*	2.74	4.42
	kNN*	2.72	**4.17**
Log-Logistic	eDTW*	2.72	4.39
	kNN*	**2.70**	**4.17**
KDE	eDTW*	2.75	4.41
	kNN*	2.77	4.50
LR-PC	-	2.71	4.72
OP-PC	-	2.76	5.20
Random Forests (*point prediction*)	-	-	4.75
Random Forests (probabilistic interpretation)	-	2.83	4.59

*DTWs = 60 minutes or $k = 15$

The corrected results reaffirm the fundamental takeaways discussed in Section 3.6.4. Specifically, they validate the preference for the Log-logistic with kNN model, which presents the lowest NLL score and MSE of the expected secondary delay over the test set among the models considered. Also, using the Gower distance improved the performance of all models employing the kNN similarity technique, with exceptions being the Cauchy distribution and the KDE model. Furthermore, the outcomes achieved by the OP-PC model are inferior to those of the LR-PC model.

Chapter 4

Article 2 - Increasing schedule reliability in the multi-depot vehicle scheduling problem with stochastic travel time

Prologue

This work was presented at the Institute for Operations Research and the Management Sciences (INFORMS) annual meeting, among other conferences, and was submitted to Omega-The International Journal of Management Science:

> Increasing schedule reliability in the multi-depot vehicle scheduling problem with stochastic travel time. Léa Ricard, Guy Desaulniers, Andrea Lodi, Louis-Martin Rousseau. Omega-The International Journal of Management Science, under revisions.

The general idea of this paper came from Guy Desaulniers, Andrea Lodi, and Louis-Martin Rousseau, while most of the specific research ideas, particularly the modeling choices, were contributed by me. I was responsible for modeling the problem mathematically, developing an efficient algorithm, writing an implementation guide for the programmer in charge of modifying GENCOL according to the developed algorithm, building test instances, running tests, analyzing the results, and writing the article. Guy Desaulniers revised the model and the algorithm. Guy Desaulniers, Andrea Lodi, and Louis-Martin Rousseau revised and edited the article.

4.1. Introduction

One of the specificities of public transport is its social mission; most transit agencies aim at providing a fair access to essential services and jobs, and strive to offer a service competitive

to single occupant vehicles, that is both time- and cost-efficient for users. The main goal of the agencies, usually centered on the quality of service, reflects this mission, whereas almost all other transport organizations, for example trucking or airline companies, are primarily concerned with making a profit (Desaulniers and Hickman, 2007). However, agencies have access to a limited budget and therefore seek to provide the best possible service within this budget. To reach this goal, budget constraints must be considered throughout the whole planning process which is commonly divided into strategic, tactical, and operational planning steps. Indeed, the planning process cannot be addressed as a whole due to tractability issues and is therefore often divided into the following sequential problems. Strategic planning includes the definition of transit routes and networks, tactical planning includes setting stops and service frequencies, whereas operational planning concerns vehicle scheduling, duty scheduling, and crew rostering among others (Desaulniers and Hickman, 2007; Ibarra-Rojas et al., 2015). As opposed to the strategic and tactical planning steps, the operational planning step is traditionally addressed only with the objective of cost-efficient running, completely or partially obliterating the main concern of agencies, namely, service quality. The vehicle scheduling problem (VSP), one of the most studied operational planning problems and the one we are interested in this work, makes no exception. When the bus fleet is spread in two or more depots, the VSP is referred to as the multiple depot vehicle scheduling problem (MDVSP) and is proven to be NP-hard (Bertossi et al., 1987). It takes as input a timetable of trips - a trip is defined by a start time and a location, an itinerary composed of a sequence of stops, and an end time and a location - and aims at finding a set of bus schedules that covers exactly once every timetabled trip while minimizing the operational costs and respecting the capacity at each depot. The operational costs usually consist of a fixed cost per vehicle used and a variable cost per kilometer and/or minute spent outside the depot. Bus schedules outputted by the MDVSP are sequences of trips (pull-in, pull-out, deadhead, or timetabled trips) interspersed by waiting times (or idle times) that must start and end at the same depot. After completing a timetabled trip, a bus can either stay at its current location before starting another timetabled trip or move to another bus terminal to start a subsequent timetabled trip, sometimes after some idle time. This move without passengers is called a deadhead trip. Pull-in and pull-out trips are special cases of deadhead trips where the departure or arrival terminal is a depot, respectively.

Because the MDVSP is traditionally tailored to minimize only operational costs without considering the quality of the underlying service, near-optimal bus schedules outputted by the MDVSP are often difficult to comply with on the day of operation because they typically do not contain much buffer time (Amberg et al., 2011). Indeed, disruptions and operational variability, for example heavy traffic (Amberg et al., 2011; Naumann et al., 2011) and fluctuation of passenger volume (Amberg et al., 2011; Chen et al., 2022), can cause delays that, in turn, can make the planned schedule infeasible when the buffer times cannot absorb these

delays. Thus, dense bus schedules (i.e., schedules with sparse buffer times) are generally less reliable. In this work, we define schedule reliability in terms of the invariability of service attributes and, specifically, vehicle timeliness.

In the literature, two types of delays are distinguished: *primary* (or exogenous) and *secondary* (or propagated) delays (Kramkowski et al., 2009; Amberg et al., 2011; Naumann et al., 2011; Amberg et al., 2019). Primary delays are a measure of the additional time required to complete a trip due to a disruption or operational variability or, in other words, the difference between the actual and the planned duration of a trip. When the buffer times of a bus schedule do not allow a full recovery of primary delays, secondary delays are encountered. The secondary delay of a trip is the difference between its actual departure time and its planned departure time. Because primary delays are considered unavoidable (Kramkowski et al., 2009; Amberg et al., 2019) and agencies only have control over secondary delays during the operational planning step, secondary delays are commonly used in the literature as a measure of the reliability of a bus schedule. In what follows, this measure is also used.

This work approaches the MDVSP from a stochastic perspective in an attempt to reduce the propagation of delays in vehicle schedules and hence increase their reliability. Our main contributions are (i) we formulate a data-driven model for the reliable MDVSP with stochastic travel time (R-MDVSP-STT) that considers an objective function that combines the operational costs and a penalty for the secondary delays, (ii) we introduce a column generation algorithm for solving the R-MDVSP-STT that gives an exact lower bound on the solution value, is fast enough to be suitable for large-scale instances and is highly adaptable, (iii) we define three reliability metrics to evaluate the solutions of the R-MDVSP-STT, and (iv) we show throughout computational tests on three real-world instances of the city of Montréal that our approach provides solutions that form an approximate Pareto front with good trade-offs between operational costs and reliability. The best probabilistic model for predicting the travel time distributions of the Montréal bus network, as well as its features and parameters, was selected in a previous work (Ricard et al., 2022) based on a dataset of various attributes of more than 41,000 trips collected over a two-month period.

The remainder of this paper is organized as follows. In Section 4.2, we discuss related works on MDVSP and methods for addressing MDVSP reliability. The R-MDVSP-STT model and the integration of a reliability measurement and control method into the model are detailed in Section 4.3. Section 4.4 introduces the branch-and-price algorithm used to solve the R-MDVSP-STT. Notably, we explain how vehicle schedules are generated in the column generation algorithm by solving shortest path problems with stochasticity. Three reliability metrics to evaluate the quality of the R-MDVSP-STT solutions are presented in Section 4.5 with a Monte Carlo simulation to compute them. Section 4.6 evaluates the heuristic performance of the R-MDVSP-STT as well as the trade-off between operational costs and

reliability, in terms of the three reliability metrics proposed, and compares the results of the R-MDVSP-STT with those of the MDVSP with mandatory buffer times. Section 4.7 summarizes our findings.

4.2. Literature review

The MDVSP has been studied since the 1970s (Dell'Amico et al., 1993), but exact algorithms have been proposed only since the end of the 1980s including those of Carpaneto et al. (1989), Ribeiro and Soumis (1994), Löbel (1998), Fischetti et al. (1999), Hadjar et al. (2006), and Kliewer et al. (2006). The work of Ribeiro and Soumis (1994) was a breakthrough that has paved the way for other exact algorithms. The authors proposed to first model the MDVSP using an integer multicommodity flow formulation and then reformulated it by applying a Dantzig-Wolfe decomposition to obtain a set-partitioning formulation, where each variable is associated with a feasible vehicle schedule. This model can be solved by a column generation algorithm (branch-and-price) that generates new columns (i.e., vehicle schedules) by solving shortest path problems. This algorithm was later enhanced to a branch-price-and-cut algorithm by Hadjar et al. (2006) who introduced valid inequalities and a variable fixing technique. Based on a multicommodity flow formulation similar to that of Ribeiro and Soumis (1994), Löbel (1998) developed a column generation algorithm that dynamically generates the arc flow variables of this formulation. However, instead of pricing these variables individually, a so-called Lagrangian pricing which prices groups of variables based on two different Lagrangian relaxations is performed. Later, Kliewer et al. (2006) modeled the MDVSP using a time-space network. This type of network contains far fewer arcs and variables than the networks considered by the above authors. The resulting arc-flow model can then be solved using a commercial mixed-integer programming (MIP) solver. Furthermore, there are a number of extensions for the VSP and MDVSP including, among others, problems with time windows (Desaulniers et al., 1998) and mixed-fleet (Li et al., 2019; Rinaldi et al., 2020; Zhou et al., 2020). We refer interested readers to Bunte and Kliewer (2010) and Desaulniers and Hickman (2007) for a detailed review on the VSP, the MDVSP, and some of their extensions.

Methods found in the literature to address the timeliness of buses when forming bus schedules can be divided into two families: online (or real-time) and offline methods. The focus of this work is on the second family of methods as we are interested in the operational planning step, an offline step. Nevertheless, it is noteworthy to mention the works of Huisman et al. (2004) and He et al. (2018), who developed solution approaches to the dynamic VSP. After a major disruption, dynamic VSP helps recover feasible schedules by rescheduling online new vehicle itineraries.

Kramkowski et al. (2009), Naumann et al. (2011), Shen et al. (2016), van Kooten Niekerk (2018), Amberg et al. (2011), and Amberg et al. (2019) proposed offline models and solution approaches for the reliable MDVSP and VSP. In particular, Kramkowski et al. (2009) presented an offline metaheuristic to increase the reliability (or delay-tolerance) of the solutions of the VSP with multiple vehicle types. From an initial solution computed as in Kliewer et al. (2006), a simulated annealing for noisy environments (SANE) seeks valid neighboring solutions. The method provided small reliability improvements, but SANE appears to be unable to find solutions with higher reliability gains. A stochastic programming framework for the VSP with stochastic travel time was proposed by Naumann et al. (2011). The network of Kliewer et al. (2006) (i.e., a time-space network) is expanded with extra waiting and deadhead arcs in order to add a penalty in the arc cost for delays between pairwise consecutive timetabled trips (full delay propagation was not implemented). In experiments on a real-life instance and 100 delay scenarios, several solutions using the same number of vehicles and with lower penalty costs than the deterministic approach were found. Shen et al. (2016) introduced two models for the VSP with stochastic travel time featuring stochastic trip compatibility (i.e., negative buffer times are allowed), stochastic idle time, and a penalty for arc infeasibility. The first model considers only delays between pairwise consecutive trips, whereas the second enhanced model considers full delay propagation. A hybrid solution approach combining a matching-based heuristic and a greedy local search is proposed for the model considering full delay propagation. Experimental results showed that both models provide more reliable solutions than the deterministic model while using the same number of vehicles. The model considering full delay propagation achieved higher punctuality than the first model with a little increase in costs. van Kooten Niekerk (2018) introduced the stochastic departure time dependent vehicle scheduling problem. The model allows negative buffer times and the cost of the arcs between pairs of trips is modified to include a cost for secondary delays. Different cost calculations with delay propagation between a maximum of two trips were compared to assess the potential of an approach incorporating full delay propagation. After carrying out computational experiments, the authors concluded that accounting for the propagation of delays over longer trip sequences in their model promised little benefit. Solutions 2 to 3% more reliable than the baseline approach of imposing minimum buffer times were achieved. The work of Amberg et al. (2019) is an extension of Amberg et al. (2011) and they both address the sequential, partially integrated, and integrated vehicle and crew scheduling problems. In both works, the VSP and the MDVSP are modeled as multi-commodity problems with underlying time-space networks. In the first paper, mandatory buffer times between trips covered by the same vehicle are imposed. Furthermore, novel decomposition schemes of flows (i.e., bundles of equal-cost solutions) taking into account secondary delays were proposed. In Amberg et al. (2019), the solution approach of Amberg

et al. (2011) is used to find a good initial solution to the Lagrangian relaxation and column generation-based algorithm that takes into account deterministic delay propagation in vehicle and crew duties. Experiments on real-life instances from Germany showed that the integrated scheduling scheme provides the best trade-offs between reliability and operational costs, compared to the sequential and partially integrated schemes. Moreover, a robust electric MDVSP (E-MDVSP) model with stochastic travel time was formulated in Jiang et al. (2021). Energy consumption is assumed to be proportional to the travel time deviation. A solution to the E-MDVSP is deemed feasible if, in addition to the usual requirements of the MDVSP, the departure time of each timetabled trip is always respected and the vehicle state-of-charge always lies in a given safety range when the travel times vary in a cardinality constrained set.

The only stochastic model proposed in the literature that accounts for full delay propagation is the enhanced model of Shen et al. (2016). This model is based on a network flow formulation and thus is not solvable using a conventional MIP solver. Indeed, the cost and the penalty on each arc cannot be computed beforehand as they are a function of the previous arcs. Our work addresses this issue by introducing an alternative set partitioning formulation for the MDVSP with stochastic travel time that can be solved using a branch-and-price algorithm. The choice of our approach was guided by the needs of our industrial partner, GIRO Inc., for a solution approach scalable to instances of real-life size and easily adaptable to the various rules and constraints that their customers may impose on the vehicle schedules.

4.3. Mathematical model

We first present our model for the R-MDVSP-STT and then describe how the reliability is integrated into the model.

4.3.1. The reliable MDVSP with stochastic travel time

Let $\mathcal{V}$ be a timetable of n trips and $\mathcal{D}$ be a set of depots. The number of vehicles available at depot $d \in \mathcal{D}$ is denoted b_d. Given the long-term prediction of the probability distributions of the travel time and the expected ridership for all trips $i \in \mathcal{V}$, the R-MDVSP-STT consists of finding feasible vehicle schedules over a one-day horizon that cover exactly every trip $i \in \mathcal{V}$ while respecting vehicle availability at each depot $d \in \mathcal{D}$. The R-MDVSP-STT is a bi-objective optimization problem that aims at minimizing the operational costs (including a fixed cost per vehicle used and variable costs) and, at the same time, maximizing service reliability. In order to find near Pareto-efficient solutions, we use a linear scalarization method (Hwang and Masud, 1979). Consequently, the R-MDVSP-STT is reformulated as

a single-objective optimization problem by balancing the values of the two objectives by a weighting factor β.

A vehicle schedule is defined as a sequence of trips starting and ending at a depot. The first trip of a vehicle schedule is a *pull-out* trip from a depot $d \in \mathcal{D}$ to the starting location of the first timetabled trip of the schedule and the last trip is a *pull-in* trip from the end location of the last timetabled trip of the schedule to d. Meanwhile, the vehicle performs *connections* between one or several pairs of trips i and $j \in \mathcal{V}$. If trip i ends at the same location as the starting location of trip j, the vehicle remains at the same location and may have to wait at the terminal. Otherwise, the vehicle performs a deadhead trip (i.e., a trip without passengers) from the end location of trip i to the starting location of trip j. A vehicle schedule is deemed feasible if it starts and ends at the same depot $d \in \mathcal{D}$ and contains a sequence of pairwise compatible timetabled trips. Let $\kappa_{i,j}$ be the deadhead travel time between the end location of trip i and the starting location of trip j and let τ be the minimum layover time between two trips. Trips i and $j \in \mathcal{V}$ are deemed compatible if $d_i^1 + \kappa_{i,j} + \tau \leq d_j^0$, where d_j^0 and d_i^1 are the departure and the arrival time of trips j and i, respectively. Let $\mathcal{S}$ be the set of all feasible vehicle schedules and $\mathcal{S}^d \subset \mathcal{S}$ be the subset of these schedules starting and ending at the depot $d \in \mathcal{D}$, such that $\mathcal{S} = \bigcup_{d \in \mathcal{D}} \mathcal{S}^d$.

The proposed model for the R-MDVSP-STT uses the following additional notation. For every vehicle schedule $s \in \mathcal{S}$ and trip $i \in \mathcal{V}$, let y_s be a binary variable that takes value 1 if vehicle schedule s is used in the solution and $a_{i,s}$ be a binary parameter equal to 1 if schedule s covers trip i.

The R-MDVSP-STT can be expressed as the following integer linear program:

$$\min \quad \sum_{s \in \mathcal{S}} c_s y_s \tag{4.3.1}$$

$$\text{s.t.} \quad \sum_{s \in \mathcal{S}} a_{is} y_s = 1, \qquad \forall i \in \mathcal{V} \tag{4.3.2}$$

$$\sum_{s \in \mathcal{S}^d} y_s \leq b_d, \qquad \forall d \in \mathcal{D} \tag{4.3.3}$$

$$y_s \in \{0,1\}, \qquad \forall s \in \mathcal{S}, \tag{4.3.4}$$

where c_s is the total cost (weighted sum of the operational costs and the cost for delays) of vehicle schedule s.

This set partitioning formulation is similar to those proposed by Ribeiro and Soumis (1994) and Hadjar et al. (2006). The only difference is in the definition of the cost coefficients $c_s, s \in \mathcal{S}$, which take into account the risks of delay propagation in our model. The objective function (4.3.1) minimizes the total cost of the selected vehicle schedules, while constraints (4.3.2) ensure that each timetabled trip is covered exactly once by a schedule and constraints (4.3.3) ensure that vehicle availability is respected at each depot.

Because $\mathcal{S}$ typically contains a huge number of schedules, it is not computationally efficient to enumerate them a priori. Rather, as described in Section 4.4, a column generation algorithm is used to identify vehicle schedules to include in a restricted version of the model. This is done by solving shortest path problems by dynamic programming on so-called connection networks similar to those first introduced by Ribeiro and Soumis (1994) and defined as follows. With every depot $d \in \mathcal{D}$, we associate a network $G^d = (V_d, A_d)$ with node set $V_d = \mathcal{V} \cup \{n_0^d, n_1^d\}$ and arc set A_d. Nodes n_0^d and n_1^d represent depot d at the beginning and the end of the day, respectively. Three types of arcs (i,j) are defined: pull-out arcs (i.e., (n_o^d, i) for $i \in \mathcal{V}$), pull-in arcs (i.e., (i,n_1^d) for $i \in \mathcal{V}$), and arcs between pairs of compatible timetabled trips (i.e., (i,j) for $i, j \in \mathcal{V}$). We have previously determined the compatibility of trips using deterministic values for travel time, departure time and arrival time. However, since we are taking into account the variability of travel time, an alternative approach would be to establish a compatibility probability between any pair of trips i and $j \in \mathcal{V}$ and expand our network to include all possible connections between trips with a non-zero compatibility probability. This probability would represent the likelihood of trip j starting on time after trip i. Despite considering this option, we deliberately chose not to incorporate these additional arcs. This decision was driven by the fact that, on average, they would result in positive secondary delays, which is not desirable in practice.

A path in G^d starting at the source node n_o^d and ending at the sink node n_1^d is a feasible vehicle schedule. We define the cost of a feasible schedule s associated with the set of arcs $A(s)$ as

$$c_s = \sum_{(i,j) \in A(s)} c_{ij}^s, \tag{4.3.5}$$

where c_{ij}^s is the cost of the arc (i,j) when it is covered by vehicle schedule s, defined as shown in Table 4.1. In this table, η is the cost per vehicle used, r^T is the cost per minute of travel, $\kappa_{d,i}$ and $\kappa_{i,d}$ are the deadhead travel times between the depot d and the starting location of trip i and the end location of trip i and the depot d, respectively, r^W is the cost per minute of waiting outside a depot, and q_i^s is the cost associated with potential delays in trip i when it is covered by vehicle schedule s. The cost of connection arcs is a weighted sum of the operational costs and the cost for delays (q_i^s), with β as the weighting factor. The latter cost q_i^s is path-dependent and its value will be discussed in Section 4.3.2.

Table 4.1 – Cost of the arcs $(i,j) \in A(s)$

Arc (i,j)	Type	c_{ij}^s
(n_o^d,j)	pull-out	$\eta + r^T \kappa_{d,j}$
(i,j)	connection	$r^T \kappa_{i,j} + r^W (d_j^0 - d_i^1) + \beta q_j^s$
(i,n_1^d)	pull-in	$r^T \kappa_{i,d}$

To avoid excessive congestion at the terminals and long idle time for the drivers, we impose a threshold Δ on the waiting time $d_j^0 - d_i^1$ between two consecutive trips i and $j \in \mathcal{V}$. If the waiting time exceeds Δ, the vehicle must move to the nearest depot after completing trip i, wait at the depot, and then move to the starting location of trip j. We modified slightly the connection network of Ribeiro and Soumis (1994) to account for this constraint by adding a new type of connection that artificially includes the wait-at-depot. This avoids the use of pull-in and pull-out arcs for the wait-at-depot.

4.3.2. Controlling schedule reliability

Delay propagation between all consecutive timetabled trips of a schedule is penalized in the objective function of the R-MDVSP-STT in an attempt to increase the reliability of the selected vehicle schedules. For each $i \in \mathcal{V}$ and $s \in \mathcal{S}$, we define the cost for delays as $q_i^s = \alpha_i \mathbb{E}(X_i^s)$, where α_i is the relative passenger flow (or demand volume) on trip i such that $\sum_{i \in \mathcal{V}} \alpha_i = 1$, X_i^s is a random variable representing the secondary delay of trip i when it is covered by schedule s, and $\mathbb{E}(X_i^s)$ its expectation. The parameters $\alpha_i, i \in \mathcal{V}$, put more weight on delays incurred during timetabled trips with a high ridership in the cost function so that, for example, the penalty assigned to a delayed peak hour trip is larger than the one assigned to a delayed off-peak hour trip, when the delays of both trips are of the same magnitude.

To be able to compute the expected secondary delay of each trip $i \in \mathcal{V}$ covered by any schedule $s \in \mathcal{S}$, an estimate of the discretized probability density function of the travel time of each timetabled trip is given in input to the R-MDVSP-STT. These estimations are taken from Ricard et al. (2022) that compared many probabilistic models for the long-term prediction of the density of the travel time (PDTT). The PDTT is framed as a supervised learning problem that aims at predicting, for each trip in a set of unseen (or future) trips, the probability density function of its travel time based on its attributes (e.g., day of the week, distance, scheduled departure time, etc.). Several models were trained and tested on a 2-month dataset of the Montréal bus network including more than 41,000 trips collected by in-car Advanced Public Transport Systems (APTS). A similarity-based density estimation model using a k Nearest Neighbors method and a Log-Logistic distribution provided the best results, both in terms of the estimation of the true conditional probability density function of the travel time and the approximation of the expected secondary delays, on this dataset. Thus, this model (and its selected features and parameters) is used here to estimate the probability density function of the travel time of each timetabled trip. Then, for any given vehicle schedule $s \in \mathcal{S}$, the probability mass function of X_i^s for all $i \in s$ is recursively computed based on the latter travel time distributions from the first to the last timetabled trip of the schedule. Note that we consider deterministic travel times for pull-in, pull-out,

and deadhead trips because these trips are usually short and do not involve passengers, thus eliminating one main source of travel time variability. Next, we discuss the discretization of the probability density functions of the travel time and the derivation of the probability mass functions of X_i^s for each $i \in \mathcal{V}$ and $s \in \mathcal{S}$.

First, for each trip $i \in \mathcal{V}$, let $\dot{T}_i$ be a random variable representing its actual travel time and let $\dot{h}_i(t) = Pr(\dot{T}_i = t)$ be the probability density function of $\dot{T}_i$. The actual travel time $\dot{T}_i$ for all $i \in \mathcal{V}$ varies from day-to-day due to random disruptions and demand or capacity reasons that can either be internal or external to the bus network (Yetiskul and Senbil, 2012; Mazloumi et al., 2010). We assume that this uncertainty is exogenous to resource allocation or, in other words, that $\dot{h}_i(t)$, for each $i \in \mathcal{V}$, stays the same regardless of the selected bus schedules.

In order to obtain a finite number of possible outcomes, we transpose $\dot{T}_i$, for all $i \in \mathcal{V}$, onto a discrete space by allocating the density of all non-integer travel times to the closest rounded down travel time (using minutes as time unit). Furthermore, for each $i \in \mathcal{V}$, we truncate the distribution of $\dot{T}_i$ below its 5th percentile and over its 95th percentile. Let $\dot{t}_i^5$ and $\dot{t}_i^{95}$ be the value of $\dot{T}_i$ at the 5th and 95th percentile of its probability distribution, respectively. For each $i \in \mathcal{V}$, let T_i be defined over Φ_i as

$$\Phi_i = \{\left\lfloor \dot{t}_i^5 \right\rfloor, \left\lfloor \dot{t}_i^5 \right\rfloor + 1, ..., \left\lfloor \dot{t}_i^{95} \right\rfloor - 1, \left\lfloor \dot{t}_i^{95} \right\rfloor\}, \tag{4.3.6}$$

and let $h_i(t)$ be the probability mass function with a discrete finite support of T_i, where the density of $Pr(\dot{T}_i < \left\lfloor \dot{t}_i^5 \right\rfloor)$ and $Pr(\dot{T}_i \geq \left\lfloor \dot{t}_i^{95} \right\rfloor + 1)$ is uniformly redistributed to $Pr(T_i = \left\lfloor \dot{t}_i^5 \right\rfloor), Pr(T_i = \left\lfloor \dot{t}_i^5 \right\rfloor + 1), \ldots, Pr(T_i = \left\lfloor \dot{t}_i^{95} \right\rfloor)$ in order to obtain a proper probability distribution (i.e., $\sum_{t=\left\lfloor \dot{t}_i^5 \right\rfloor}^{\left\lfloor \dot{t}_i^{95} \right\rfloor} h_i(t) = 1$).

Second, based on the above probability mass functions and considering that d_i^0, for all $i \in \mathcal{V}$, $\kappa_{i,j}$, for all $i, j \in \mathcal{V}$, and τ are also stored to the nearest minute, it is possible to derive, for a given vehicle schedule $s \in \mathcal{S}$, the probability mass function of X_i^s for all $i \in s$ as follows. For each $i \in s$, let Y_i^s be a random variable representing the actual departure time of trip i and let $f_i^s(y) = Pr(Y_i^s = y)$ be the probability mass function of Y_i^s. We have developed an exact procedure inspired by the works of Errico et al. (2018) and Shen et al. (2016) to recursively compute $f_i^s(y)$. This procedure is as follows.

We assume that the first timetabled trip of a schedule is never delayed, i.e., for every schedule $s \in \mathcal{S}$, $f_{v_0^s}^s(d_{v_0^s}^0) = 1$, where $v_0^s \in s$ is the first timetabled trip of s. For the other trips $j \in s \backslash \{v_0^s\}$, three cases are distinguished: when the bus starts trip j on time (i.e., $y = d_j^0$), late (i.e., $y > d_j^0$), and early (i.e., $y < d_j^0$). The last case has zero probability because we assume a bus cannot start ahead of time. The distribution $f_j^s(y)$ of a trip $j \in s \backslash \{v_0^s\}$ preceded by trip $i \in s$ can be computed as

$$f_j^s(y) = Pr(Y_j^s = y) = \begin{cases} \sum_{k\in\Phi_i} h_i(k) \times \sum_{y'=d_i^0}^{d_j^0-\kappa_{i,j}-\tau-k} f_i^s(y'), & \text{if } y = d_j^0; \\ \sum_{k\in\Phi_i} h_i(k) \times f_i^s(y-\kappa_{i,j}-\tau-k), & \text{if } y > d_j^0; \\ 0, & \text{otherwise.} \end{cases} \tag{4.3.7}$$

Let $g_j^s(x) = Pr(X_j^s = x) = Pr(Y_j^s = d_j^0 + x) = f_j^s(d_j^0 + x)$ be the probability mass function of X_j^s. Its expectation is given by

$$\mathbb{E}(X_j^s) = \sum_{x=0}^{x_{j,s}^{max}} x \times g_j^s(x) = \sum_{x=0}^{x_{j,s}^{max}} x \times f_j^s(x + d_j^0), \tag{4.3.8}$$

where i is the trip that precedes the trip j in schedule s and $x_{j,s}^{max} = d_i^0 + x_{i,s}^{max} + \kappa_{i,j} + \tau + \left\lfloor t_i^{95} \right\rfloor - d_j^0$ is the maximum secondary delay of trip j.

Note that, by definition, Y_i^s and X_i^s for all $i \in \mathcal{V}$ depend on the travel times of the previous timetabled trips in s; their uncertainty is therefore endogenous to resource allocation. In other words, the random variables Y_i^s and X_i^s for $s \in \mathcal{S}$ and $i \in \mathcal{V}$ are likely to have different probability distributions than $Y_i^{s'}$ and $X_i^{s'}$ for $s \neq s' \in \mathcal{S}$. Hence, it is not possible to compute Y_i^s and X_i^s for all $i \in \mathcal{V}$ and $s \in \mathcal{S}$ beforehand because, as mentioned earlier, vehicle schedules are not enumerated but rather generated dynamically when solving the R-MDVSP-STT.

Overall, we have shown in this section how to compute, for a given vehicle schedule, the convolution of the probability mass function of the secondary delay of every timetabled trip in the schedule. These distributions are used to assess the reliability of the schedule, measured by the total expected secondary delay per passenger. With this information, a decision maker can then address the trade-off between operational costs and reliability by adjusting the weighting factor β that controls the importance given to the penalty for unreliability.

4.4. Heuristic branch-and-price algorithm for the R-MDVSP-STT

In real-life R-MDVSP-STT instances, there exists a very large number of feasible vehicle schedules. Instead of explicitly enumerating the corresponding variables in the integer program (4.3.1)-(4.3.4), we propose to solve the R-MDVSP-STT using a column generation algorithm (Irnich and Desaulniers, 2005; Lübbecke and Desrosiers, 2005) embedded in a branch-and-bound tree. This solution method is also referred to as branch-and-price (Barnhart et al., 1998). Furthermore, we propose to use acceleration strategies to obtain integer solutions in a reasonable amount of time. On the one hand, we use a heuristic branching strategy and, on the other hand, we apply a perturbation method to reduce the strong

degeneracy inherent to the set partitioning model (4.3.1)-(4.3.4). The column generation algorithm as well as these two acceleration strategies are detailed in the following.

4.4.1. Column generation

Column generation is an iterative algorithm that generates variables (columns) as needed. In this context, the linear relaxation of (4.3.1)-(4.3.4) is called the master problem (MP). At each iteration, the algorithm solves a restricted MP (RMP) that is defined as the MP restricted to a small subset $\mathcal{S}' \subseteq \mathcal{S}$ of the schedule variables y_s. This resolution provides a primal and a dual solution. To identify potentially useful columns to add to the RMP, a set of pricing problems is solved with the goal of finding negative reduced cost variables. In our case, there is one pricing problem per depot and it corresponds to a shortest path problem with stochasticity (Wellman et al., 2013; Boland et al., 2015). If no columns with a negative reduced cost are identified, the algorithm stops and the current RMP solution is guaranteed to be also optimal for the MP. Otherwise, some columns with a negative reduced cost are added to $\mathcal{S}'$ and a new iteration is started.

4.4.1.1. Shortest path problem with stochasticity. The pricing problem for depot d is defined on the acyclic network G^d (see Section 4.3.1) with modified arc costs as detailed next. Let $(u_i)_{i\in\mathcal{V}}$ and $(\pi_d)_{d\in\mathcal{D}}$ be the dual variables associated with constraints (4.3.2) and (4.3.3), respectively. The reduced cost $\tilde{c}_s$ of a vehicle schedule $s \in \mathcal{S}^d$ housed in depot d is given by

$$\tilde{c}_s = c_s - \sum_{i\in\mathcal{V}} a_{is} u_i - \pi_d, \tag{4.4.1}$$

with the following cost breakdown per arc

$$\tilde{c}^s_{ij} = \begin{cases} c^s_{ij} - \pi_d, & \text{if } i = n^d_0; \\ c^s_{ij} - u_i, & \text{if } i \in \mathcal{V}. \end{cases} \tag{4.4.2}$$

Because the arc costs are stochastic and path-dependent in the R-MDVSP-STT, the standard labeling algorithm (Irnich and Desaulniers, 2005) cannot be used directly to solve the pricing problems. The dependence assumption is crucial because, as explained by Wellman et al. (2013), if the arc costs were stochastic but *independent* (i.e., if the probability mass function of the secondary delay at each node did not depend on the ones of the previous nodes in the path), the expected arc costs could be used directly in the standard labeling algorithm. Faced with path-dependent uncertainty, we must therefore use the modified version of the labeling algorithm proposed by Wellman et al. (2013) and Boland et al. (2015) that uses a stochastic dominance criterion.

This modified version of the labeling algorithm simultaneously extends multiple paths in G^d until they reach the sink node (i.e., the depot). Each of these paths is obtained by starting from a trivial path p and extending it by adding one vertex at a time. A vertex can be added to a path if the corresponding extended path is feasible with respect to path-structural constraints. The probability mass functions of the propagated delay and the reduced costs are used in the algorithm to "discard paths which are not useful either to build a Pareto-optimal set of paths or to be extended into Pareto-optimal paths"(Irnich and Desaulniers, 2005). This ability is indeed essential to efficient dynamic programming algorithms (Irnich and Desaulniers, 2005). In the R-MDVSP-STT, two paths are not comparable if one has a lower reduced cost than the other, but higher chances of being unreliable. In this case, no path dominates the other and therefore no path can be discarded.

The labeling algorithm efficiently encodes paths using labels. Each label contains information useful to identify paths that can be safely discarded. We refer interested readers to the work of Ahuja et al. (1993) for an overview of the subject. Next, we define the labels, the extension functions of these labels (i.e., a procedure to compute a label from a previous label), and the stochastic dominance rule used in the dynamic programming algorithm.

Labels. Each label stores a representation of the actual departure time and the accumulated reduced cost. As of now, many variables and parameters, notably the arc costs and the probability distributions of both the actual departure time and the secondary delay of each timetabled trip, have been defined in terms of the vehicle schedule covering them. We extend these definitions to consider their counterparts in terms of the path that covers each timetabled trip. To simplify notation, subscripts s and p, associated with a vehicle schedule and a path, respectively, are used interchangeably in the following.

For each path p in G^d and each trip $i \in \mathcal{V}$ included in p (written as $i \in p$ afterwards), let $F_i^p(z)$ be the cumulative distribution function of Y_i^p defined as

$$F_i^p(z) = \sum_{y=d_i^0}^{z} f_i^p(y). \tag{4.4.3}$$

The actual departure time of a trip $i \in p$ is represented by $F_i^p(z)$ instead of $f_i^p(y)$ because, as we will explain in the following, the former is directly used in the dominance rule. The label L_i^p of path p at node i is defined as

$$L_i^p = (F_i^p(d_i^0), ..., F_i^p(d_i^0 + x_{i,p}^{max}), C_i^p), \tag{4.4.4}$$

where C_i^p is the current accumulated reduced cost of path p.

Extension functions. Consider the extension of a label $L_i^{p'} = (F_i^{p'}(d_i^0), ..., F_i^{p'}(d_i^0 + x_{i,p'}^{max}), C_i^{p'})$ associated with node i along the arc (i,j) to create a new label L_j^p at node

j. First, the equations for the $x_{j,p}^{max}$ components of type $F_j^p(\cdot)$ are derived from equation (4.3.7) as detailed in Appendix A.1. This derivation results in

$$F_j^p(z) = \sum_{k \in \Phi_i} h_i(k) \times F_i^{p'}(z - \kappa_{i,j} - \tau - k). \tag{4.4.5}$$

When extending a label to the sink node, it is not necessary to update information on the actual departure time, because a bus cannot be delayed once it is back at the depot.

Second, C_j^p can be decomposed in route segments (see equation (4.4.2) for the cost breakdown per arc) as

$$C_j^p = C_i^{p'} + \tilde{c}_{i,j}^p, \tag{4.4.6}$$

where $\tilde{c}_{i,j}^p$ is path-dependent if $i,j \in \mathcal{V}$.

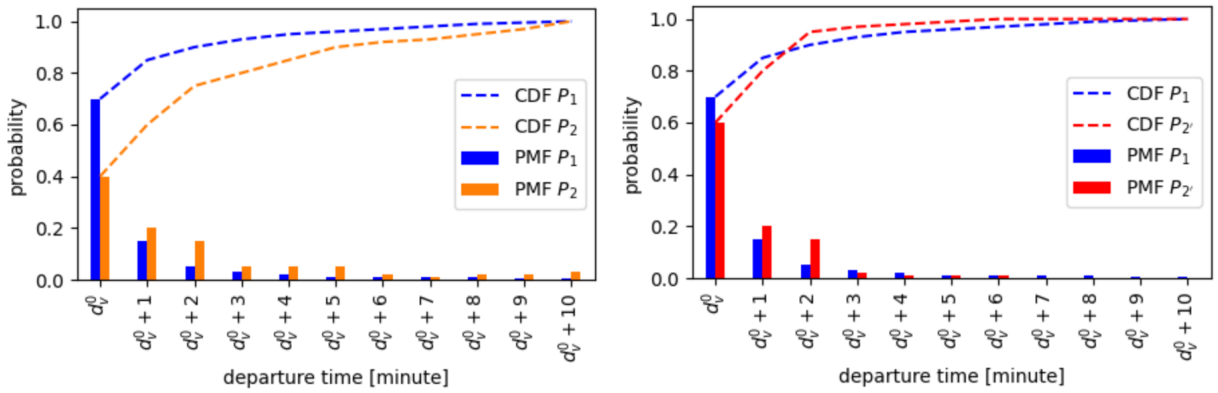
Stochastic dominance rule. Consider two paths p_1 and p_2 in G^d both ending at node $i \in \mathcal{V}$ (that is, i is the resident node of p_1 and p_2) with labels $L_i^{p_1} = (F_i^{p_1}(d_i^0), ..., F_i^{p_1}(d_i^0 + x_{i,p_1}^{max}), C_i^{p_1})$ and $L_i^{p_2} = (F_i^{p_2}(d_i^0), ..., F_i^{p_2}(d_i^0 + x_{i,p_2}^{max}), C_i^{p_2})$, respectively. The path p_1 dominates p_2 (and therefore p_2 can be discarded) when the following two conditions hold:

(1) $C_i^{p_1} \leq C_i^{p_2}$,

(2) $F_i^{p_1}(z) \geq F_i^{p_2}(z)$, for all $z \in \{d_i^0, d_i^0 + 1, \ldots, d_i^0 + \max\{x_{i,p_1}^{max}, x_{i,p_2}^{max}\}\}$.

The first condition is straightforward, but we will explain in further detail the second one. In the shortest path problem with stochasticity, the uncertain element is the cost of the arcs and, more specifically, the cost for delays. Even though the arc costs are computed using the expected secondary delays, the dominance condition cannot be based on the mathematical expectation, because the probability distributions of secondary delay are path-dependent. For example, consider again the two paths p_1 and p_2. If $\mathbb{E}(X_i^{p_1}) \leq \mathbb{E}(X_i^{p_2})$, it does not necessarily imply if we extend p_1 and p_2 with node $j \in \mathcal{V}$ that $\mathbb{E}(X_j^{p_1 \oplus j}) \leq \mathbb{E}(X_j^{p_2 \oplus j})$, where $p_k \oplus j, k = 1, 2$, denotes the path resulting from appending node j to path p_k. Thus, a stochastic dominance condition based on the cumulative distribution functions of the actual departure time is used instead. When $F_i^{p_1}(z) \geq F_i^{p_2}(z)$ for some $z \geq d_i^0$, it means that $Pr(Y_i^{p_1} > z) \leq Pr(Y_i^{p_2} > z)$ (i.e., path p_1 is less likely to start trip i after time z). The latter situation is desirable as it means that p_1 is more likely to start on time. If it holds for all $z \in \{d_i^0, d_i^0 + 1, \ldots, d_i^0 + \max\{x_{i,p_1}^{max}, x_{i,p_2}^{max}\}\}$ (i.e., if the second condition holds), then p_1 is undoubtedly more reliable than p_2 and if we extend p_1 and p_2 along the same path, the extension of p_1 will be at least as reliable as the extension of p_2. An example of this case is illustrated in Figure 4.1a. This figure and Figure 4.1b display the probability mass functions (PMF) and the cumulative distribution functions (CDF) of the actual departure time of the resident node $i \in \mathcal{V}$ of two paths p_1 and p_2 in G^d. If both conditions hold, then p_2 and the extension of p_2 are dominated by p_1 and its extension. Path p_2 can thus be discarded. If the second condition does not hold for some $z \in \{d_i^0, d_i^0 + 1, \ldots, d_i^0 + \max\{x_{i,p_1}^{max}, x_{i,p_2}^{max}\}\}$,

as in the example in Figure 4.1b, it is not clear if p_1 or p_2 is more reliable. For example, $Pr(Y_i^{p_1} > d_i^0 + 1) < Pr(Y_i^{p_2} > d_i^0 + 1)$, but $Pr(Y_i^{p_1} > d_i^0 + 2) > Pr(Y_i^{p_2} > d_i^0 + 2)$. Thus, both paths are kept.



(a) Second condition holds

(b) Second condition does not hold

Figure 4.1 – Examples of the second dominance condition applied to the trip i of paths p_1 and p_2

4.4.2. Heuristic branching

An exact branch-and-price algorithm (Barnhart et al., 1998) can derive an optimal solution to a MDVSP. However, medium- and large-scale MDVSP instances are difficult to solve using exact methods because too many branch-and-bound nodes need to be explored. To avoid exploring too many nodes, we apply one type of branching decision combined with a variable rounding strategy. First, we branch on the number of vehicles used per depot. This decision leads to the creation of one or two child nodes with upper and lower bounds on the capacity of a given depot. Second, we impose the rounding of multiple schedule variables. When this strategy is selected, a single node with one or several schedule variables fixed to 1 is created. To select the variable(s) fixed to 1, all the variables are first sorted in descending order of their fractional value. Then, a maximum of three variables with a fractional part greater than or equal to 0.9, namely, those with the largest fractional parts, are selected. If there are fewer than three variables fulfilling this condition, only one or two variables are selected, and if no variable has a fractional part greater than or equal to 0.9, the first variable in the list, i.e., the one with the largest fractional part, is selected. Note that the values of the maximum number of variables to round and of the fractional part threshold have been determined empirically during a preliminary test campaign.

For medium-sized instances (less than 1,500 timetabled trips), the first technique (i.e., branching on the number of vehicles used per depot) is applied in priority. When this number is integer for all depots, the algorithm switches to rounding schedule variables to

yield integer solutions. Our experimental results suggest that when this branching decision and this strategy are used to partially explore the branch-and-bound search tree, a good trade-off between computing time and solution quality is achieved. When the number of timetabled trips exceeds 1,500, the latter approach is not sufficient to reduce the computing time and only the rounding strategy is applied. Thus, only one branch is explored, which is equivalent to a diving heuristic. Our experimental results show that this does not significantly sacrifice the quality of the derived solutions.

4.4.3. Constraint perturbation

Problems like (4.3.1) - (4.3.4) generally exhibit a high degree of degeneracy. In order to cope with this issue, we use a constraint perturbation strategy (Charnes, 1952). We introduce perturbation variables η_i^+ and η_i^- that allow limited under- and over-covering of trip i for all $i \in \mathcal{V}$, respectively. The perturbed MP is defined as

$$\min \quad \sum_{s \in \mathcal{S}} c_s y_s + \sum_{i \in \mathcal{V}} (\delta_i^+ \eta_i^+ + \delta_i^- \eta_i^-) \tag{4.4.7}$$

$$\text{s.t.} \quad \sum_{s \in \mathcal{S}} a_{is} y_s + \eta_i^+ - \eta_i^- = 1, \qquad \forall i \in \mathcal{V} \tag{4.4.8}$$

$$\sum_{s \in \mathcal{S}^d} y_s \leq b_d, \qquad \forall d \in \mathcal{D} \tag{4.4.9}$$

$$0 \leq \eta_i^+ \leq \xi_i^+, \qquad \forall i \in \mathcal{V} \tag{4.4.10}$$

$$0 \leq \eta_i^- \leq \xi_i^-, \qquad \forall i \in \mathcal{V} \tag{4.4.11}$$

$$0 \leq y_s \leq 1, \qquad \forall s \in \mathcal{S}, \tag{4.4.12}$$

where δ_i^+ and δ_i^- are the penalties in the objective function for under- and over-covering trip $i \in \mathcal{V}$, respectively, and ξ_i^+ and ξ_i^- are the upper bounds of η_i^+ and η_i^-, respectively, for every trip $i \in \mathcal{V}$. Perturbation is removed when no other branching decision or strategy can be applied.

4.5. Assessing schedule reliability

To address delay propagation, the R-MDVSP-STT selects vehicle schedules that are such that they usually contain buffers to absorb recurring delays. Pushed to the extremes, the lowest level of reliability is achieved when these schedules contain no buffer and the highest level when a different vehicle is assigned to each timetabled trip (i.e., the number of vehicles is equal to the number of timetabled trips). However, the former solution is likely to displease passengers and the latter is highly cost-inefficient. The problem is thus to address the trade-off between operational costs and reliability. This is done by adjusting the factor β in the

cost of connection arcs (see Table 4.1). After solving several times the R-MDVSP-STT while adjusting the factor β, it is possible to compare more carefully the solutions thereby obtained. In this respect, we define next three reliability metrics that decision-makers can use to thoroughly compare multiple schedules and choose the one with the most appropriate balance, in their view, between operational costs and reliability and we provide a Monte Carlo simulation framework to compute these metrics.

4.5.1. Reliability metrics

The R-MDVSP-STT solutions can be compared based on the following metrics: the expected secondary delay per passenger, the probability that a passenger boards a delayed timetabled trip, and the average number of timetabled trips needed to recover from secondary delays. These three metrics are computed as follows.

Expected secondary delay per passenger (γ):

$$\gamma = \sum_{s \in \mathcal{S}^*} \sum_{i \in \mathcal{V}} a_{is} \alpha_i \mathbb{E}(X_i^s), \tag{4.5.1}$$

where $\mathcal{S}^*$ is the set of vehicle schedules selected in a solution.

Probability that a passenger boards a delayed timetabled trip (ψ):

$$\psi = \sum_{s \in \mathcal{S}^*} \sum_{i \in \mathcal{V}} a_{is} \alpha_i Pr(X_i^s > \epsilon), \tag{4.5.2}$$

where ϵ is a grace period (i.e., if the secondary delay of a trip $i \in \mathcal{V}$ is less than or equal to ϵ, the trip is considered on time and otherwise it is considered delayed).

Average number of timetabled trips needed to recover from secondary delays (θ):

$$\theta = \frac{1}{|\mathcal{S}^*|} \sum_{s \in \mathcal{S}*} \bar{\Omega}_s, \tag{4.5.3}$$

where Ω_s is a set containing, for every timetabled trip in the schedule s, the expected number of trips needed to get back on schedule every time it is delayed (i.e., if its secondary delay is larger than ϵ) and $\bar{\Omega}_s$ its average. To approximate this metric, two counters, $\varphi^{s,k}$ and $\rho^{s,k}$, counting the number of first delayed timetabled trips and subsequent delayed timetabled trips in schedule s at iteration k, respectively, are used. To better understand how these counters work, let us consider the toy example presented in Figure 4.2. The timetabled trips highlighted in red are delayed whereas the gray ones are on time. In this example, it takes one timetabled trip to recover the first delay impacting trip v_1^s, whereas the second delay does not affect the departure time of the next timetabled trip, and the last delay impacting trip v_7^s is never fully recovered before the end of the schedule. Thus, $\varphi^{s,k} = 3$ and $\rho^{s,k} = 1+0+2 = 3$ in this example.

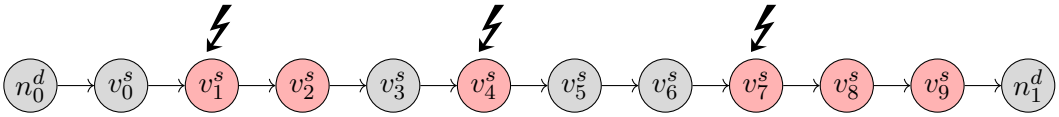


Figure 4.2 – Example of the potential delay propagation in schedule $s = \{v_0^s, v_1^s, \ldots, v_9^s\}$ of 10 timetabled trips at iteration k. Delayed trips are highlighted in red

Metrics γ and ψ reflect the secondary delay a passenger is expected to encounter in the bus network and how likely a passenger is to travel on a timetabled trip that is late, respectively. The parameters $\alpha_i, i \in \mathcal{V}$, weight the contribution of each trip $i \in \mathcal{V}$ to these metrics by their relative passenger flow (or demand volume). Thus, these metrics are user-oriented, a point of view often overlooked in the evaluation of public transportation services (van Oort, 2011). The third metric is more agency-oriented as it provides useful information for assessing the potential savings on recovering methods. Indeed, an extra bus is often dispatched when a severe cascade effect of delays is observed, i.e., when the number of timetabled trips needed to get back on schedule is relatively large. The first metric is minimized when solving (4.3.1) - (4.3.4), whereas the two other metrics are not directly minimized. Thus, the first metric should improve as β increases and we expect the other two metrics to follow this trend as well, since the three metrics are interrelated.

4.5.2. Monte Carlo simulation

In this section, two simulation cases will be presented. We start with the first one. The pseudo-code in Algorithm 2 summarizes the Monte Carlo simulation of $K = 1{,}000$ iterations performed to compute the approximations of the three metrics: $\hat{\gamma}$, $\hat{\psi}$, and $\hat{\theta}$. At each iteration $k = 1, \ldots, K$, delay propagation in each vehicle schedule $s \in \mathcal{S}^*$ is assessed. In Step 8, a travel time t_i^k is randomly sampled from $\dot{h}_i(t)$ for each $i \in s$ except for the last timetabled trip in s. Note that, in the simulation, we use the travel time probability density functions $\dot{h}_i(t)$ instead of the probability mass functions $h_i(t)$ like in the R-MDVSP-STT because the probability density functions offer more complete information. Then, in Steps 6 and 9, the value of $y_i^{s,k}$, the actual departure time of trip i covered by schedule s at iteration k, is recursively computed for all $i \in s$, from the first timetabled trip of the schedule to the last timetabled trip, as

$$y_{v_0^s}^{s,k} = d_{v_0^s}^0 \tag{4.5.4}$$

$$y_j^{s,k} = \max\{y_i^{s,k} + t_i^k + \kappa_{i,j} + \tau, d_i^o\}, \tag{4.5.5}$$

where, for each $j \in \mathcal{V}$, i is the trip preceding j in s and v_0^s is the first timetabled trip of s. In Step 10, the secondary delay $x_j^{s,k}$ of trip j covered by schedule s at iteration k is computed for all $j \in s \setminus \{v_o^s\}$ as

$$x_j^{s,k} = y_j^{s,k} - d_j^0. \tag{4.5.6}$$

Algorithm 2: Monte Carlo simulation to compute $\hat{\gamma}, \hat{\psi}$ and $\hat{\theta}$

1 $\hat{\gamma} \leftarrow 0$, $\hat{\psi} \leftarrow 0$, $\hat{\theta} \leftarrow 0$
2 **for** $k \leftarrow 1$ **to** K **do**
3 **for** $s \in \mathcal{S}^*$ **do**
4 $\varphi^{s,k} \leftarrow 0$ // counter of the number of first delayed timetabled trips
5 $\rho^{s,k} \leftarrow 0$ // counter of the number of subsequent delayed timetabled trips
6 $y_{v_0^s}^{s,k} \leftarrow d_{v_0^s}^o$
7 **for** $i \in s \setminus \{v_0^s\}$ **do**
8 Randomly generate t_{i-1}^k from $\dot{h}_{i-1}(t)$ or $\bar{h}_{i-1}(t)$
9 $y_i^{s,k} \leftarrow \max\{y_{i-1}^{s,k} + t_{i-1}^k + \kappa_{i-1,i} + \tau, d_i^0\}$
10 $x_i^{s,k} \leftarrow y_i^{s,k} - d_i^0$
11 $\hat{\gamma} \leftarrow \hat{\gamma} + (\alpha_i \times x_i^{s,k})$
12 **if** $x_i^{s,k} > \epsilon$ **then**
 // delayed timetabled trip
13 $\hat{\psi} \leftarrow \hat{\psi} + \alpha_i$
14 **if** $x_{i-1}^{s,k} \leq \epsilon$ **then**
15 $\varphi^{s,k} \leftarrow \varphi^{s,k} + 1$
16 **else**
17 $\rho^{s,k} \leftarrow \rho^{s,k} + 1$
18 **end**
19 **end**
20 **end**
21 $\hat{\theta} \leftarrow \hat{\theta} + \frac{\rho^{s,k}}{\varphi^{s,k}}$
22 **end**
23 **end**
24 $\hat{\gamma} \leftarrow \hat{\gamma}/K$
25 $\hat{\psi} \leftarrow \hat{\psi}/K$
26 $\hat{\theta} \leftarrow \hat{\theta}/(K \times |\mathcal{S}^*|)$

Finally, the approximations of the three metrics are computed in Steps 11-26. When a trip $i \in s$ preceded by trip $i-1$ is delayed at iteration k (i.e., when it has a secondary delay larger than ϵ) and when the trip $i-1$ was not delayed, the counter $\varphi^{s,k}$ is incremented by one (see Step 15). Otherwise, the counter $\rho^{s,k}$ is incremented by one (see Step 17).

Furthermore, we study the effect of external and extraordinary factors (e.g., a severe snowstorm) resulting in delays of the same magnitude on all timetabled trips in the so-called second case of the simulation. For each trip $i \in \mathcal{V}$, let $\dot{t}_i^{75}$ and $\dot{t}_i^{95}$ be the values of $\dot{T}_i$

at the 75th and 95th percentiles of its probability distribution, respectively. To generate random travel time values, we truncate the probability density function $\dot{h}_i(t)$ below $\dot{t}_i^{75}$ and above $\dot{t}_i^{95}$, utilizing the procedure proposed in Thomopoulos (2012). At each iteration and for each timetabled trip, a random value $u \sim U(0,1)$ is generated. Then, we compute $v = \dot{H}_i(\dot{t}_i^{75}) + u \times (\dot{H}_i(\dot{t}_i^{95}) - \dot{H}_i(\dot{t}_i^{75}))$, where $\dot{H}_i(t)$ is the CDF of $\dot{h}_i(t)$, before finding the travel time value t such that $\dot{H}_i(t) = v$.

4.6. Computational results

In this section, we test our model on three real-life instances, I1, I2, and I3 with 1,175, 1,916, and 2,195 timetabled trips, respectively, taken from the Montréal bus network and provided by our industrial partner. The probability density functions of the travel time of all the timetabled trips in these instances have been computed as in Ricard et al. (2022) (see Section 4.3.2). The main properties of instances I1, I2, and I3, namely the instance name (Instance), the number of timetabled trips ($|\mathcal{V}|$), the number of arcs ($|A|$), the number of depots ($|\mathcal{D}|$), and the number of bus lines (# lines), are listed in Table 4.2.

Table 4.2 – Properties of real-life instances I1 - I3

Instance	$\lvert\mathcal{V}\rvert$	$\lvert A\rvert$	$\lvert\mathcal{D}\rvert$	# lines
I1	1,175	628,064	2	8
I2	1,916	1,622,134	3	8
I3	2,195	2,119,534	2	8

Throughout our experiments, the cost per vehicle used η, the cost per minute of travel r^T (either deadhead or pull-in/pull-out trips), the cost per minute of waiting time outside a depot r^W, the threshold on the maximum waiting time outside a depot Δ, and the grace period ϵ are set to 1,000, 0.4, 0.2, 45 minutes, and 3 minutes, respectively. The penalties for under- and over-covering trip $i \in \mathcal{V}$, δ_i^+ and δ_i^-, are set to 1 for every trip $i \in \mathcal{V}$ and the upper bounds ξ_i^+ and ξ_i^- are randomly selected in $[0, 0.1]$ for every trip $i \in \mathcal{V}$.

We conduct our experiments on a Linux machine equipped with 12 Intel core i7-8700 processors running at 3.20 GHz and a RAM of 65 GB. The branch-and-price algorithm is implemented using the GENCOL library, version 4.5, and the pricing problems are solved by the commercial solver CPLEX 12.8.

Next, we compare the R-MDVSP-STT to the traditional MDVSP and the MDVSP with minimum buffer time. On the one hand, we evaluate the heuristic performance of our algorithm and analyze how the operational costs and the reliability metrics change with the factor β in the MDVSP and the R-MDVSP-STT solutions. On the other hand, we compare the values of the reliability metrics of the solutions of the R-MDVSP-STT to those of the

MDVSP with minimum buffer time to be able to establish the best approach for improving bus schedule reliability.

4.6.1. MDVSP results

In this section, we provide baseline results obtained by solving heuristically instances I1-I3 without considering reliability (i.e., $\beta = 0$). This is equivalent to using the traditional MDVSP formulation. We first study the performance of our algorithm in terms of computing time and solution quality and then we assess the trade-off between operational costs and reliability of MDVSP solutions.

Table 4.3 reports the upper bound (UB) and the lower bound (LB) obtained, the relative difference in percentage between the UB and the LB (Gap), the number of branching nodes explored (# Nodes), and the computing times (CPU time) in seconds, including the total CPU time (Total), the time to solve the root node (Root), and the time to solve the pricing problems (Pricing).

Table 4.3 – Heuristic performance without considering reliability (MDVSP)

					CPU time (seconds)		
Instance	UB	LB	Gap (%)	# Nodes	Total	Root	Pricing
I1	78,861.8	78,857.3	0.01	68	3,035.5	1,046.0	1,481.1
I2	135,919.2	135,912.2	0.01	96	6,934.2	3,327.4	3,252.2
I3	167,727.2	167,718.5	0.01	128	8,235.1	3,642.2	3,933.9

All instances are solved in less than 140 minutes with approximately half of the computing time spent on solving the root node. Also, the optimality gaps are small (below 0.01%), suggesting that our heuristic algorithm can find near-optimal solutions.

The trade-offs between operational costs and reliability for the instances I1-I3 solved without considering reliability are presented in Table 4.4. The columns display the operational costs (Op. costs), the number of vehicles used (# Bus), the average number of timetabled trips per bus (# trips/bus), and the reliability metrics (γ, ψ and θ) based on the first and the second cases (i.e., normal conditions and external and extraordinary factors, respectively). The definition of the two simulation cases as well as the method to approximate these metrics using the travel time probability density functions were presented in Section 4.5.

Of instances I1, I2, and I3, the second is the most prone to unreliability. It has an expected secondary delay per passenger of 2.14 minutes, a probability that a passenger boards a delayed timetabled trip of 23%, and an average number of timetabled trips needed to recover from secondary delays of 2.08 trips. These values are not negligible, especially for

Table 4.4 – Operational costs versus reliability of solutions obtained without considering reliability (MDVSP)

Instance	Op. costs	# Bus	# trips/bus	Reliability					
				1st case			2nd case		
				γ	ψ	θ	γ	ψ	θ
I1	78,861.8	75	16	0.61	0.07	1.38	1.96	0.22	2.18
I2	135,919.2	131	15	2.14	0.23	2.08	12.31	0.64	2.92
I3	167,727.2	162	14	1.46	0.17	1.89	9.08	0.58	4.19

the last metric if we consider that each bus performs an average of 15 timetabled trips per day, but were expected because no buffer time nor stochastic optimization is considered.

4.6.2. R-MDVSP-STT results

This section investigates the performance of our algorithm when solving the R-MDVSP-STT, i.e., when taking into account reliability ($\beta > 0$). Moreover, we present the operational costs and the reliability metrics of several R-MDVSP-STT solutions. This allows us to compare the solutions of our model with the baseline MDVSP solutions and assess how the reliability metrics and operational costs fluctuate. Given the bi-criteria nature of the objective function, we obtained multiple R-MDVSP-STT solutions (an approximate Pareto frontier) for each instance by solving the problem multiple times with a different β value each time. A total of 8 values within the range of $\eta/10$ to 10η, where $\eta = 1{,}000$ is the cost per vehicle used, have been tested.

Table 4.5 provides the heuristic performance of our algorithm for different β values. The upper bounds (UB) and lower bounds (LB) on the optimal values include the cost related to delays, whereas it is not the case in Table 4.3 because $\beta = 0$. It may be noted that the optimality gaps of the R-MDVSP-STT are similar to those of the MDVSP. Furthermore, the total computing times do not increase much compared to those obtained for the MDVSP, thanks to the use of a branching heuristic that controls the number of branching nodes explored (see Section 4.4.2). However, the proportion of time dedicated to the pricing problems is increased. For example, 47% of instance I2 computing time is spent on the princing problems for the MDVSP and this proportion increases to 77% to 86% for the R-MDVSP-STT.

The trade-offs between operational costs and reliability of the vehicle schedules obtained with different β values are provided in Table 4.6. The operational costs displayed in this table are computed by subtracting to the solution values (UBs) the costs related to delays. Since, in our experiments, all the solutions to the R-MDVSP-STT use the same number of vehicles as the corresponding MDVSP schedules, the cost increase is evaluated in terms of the variable operational costs (that exclude the cost per vehicle used). The variable

Table 4.5 – Heuristic performance considering reliability (R-MDVSP-STT)

						CPU time (seconds)		
Instance	β	UB	LB	Gap (%)	# Nodes	Total	Root	Pricing
I1	$\eta/10$	78,916.8	78,912.3	0.01	42	2,820.7	1,504.3	2,205.3
	$\eta/5$	78,962.0	78,956.8	0.01	44	2,630.0	1,468.0	2,071.7
	$\eta/2$	79,062.1	79,057.6	0.01	43	2,803.7	1,507.5	2,174.0
	η	79,183.8	79,179.2	0.01	43	2,903.8	1,474.0	2,240.4
	2η	79,359.4	79,354.9	0.01	43	3,066.6	1,524.3	2,310.0
	4η	79,615.2	79,610.7	0.01	43	3,364.1	1,552.9	2,495.6
	8η	80,039.4	79,997.7	0.05	133	6,214.5	1,617.3	4,561.5
	10η	80,170.5	80,165.4	0.01	50	3,852.5	1,544.3	2,813.9
	avg.	**79,356.9**	**79,346.8**	**0.01**	**57**	**3,487.2**	**1,517.0**	**2,645.3**
I2	$\eta/10$	136,114.0	136,107.1	0.01	76	10,979.1	6,601.6	9,423.7
	$\eta/5$	136,275.0	136,267.3	0.01	95	11,601.2	6,494.8	9,868.2
	$\eta/2$	136,605.0	136,595.5	0.01	94	11,174.1	6,221.3	9,445.9
	η	136,966.0	136,955.6	0.01	106	10,314.0	5,473.6	8,560.3
	2η	137,452.0	137,442.9	0.01	92	10,652.6	6,175.8	8,499.1
	4η	138,153.0	138,144.0	0.01	87	9,467.8	5,816.2	7,430.0
	8η	139,235.0	139,226.5	0.01	81	7,666.6	4,863.0	5,932.5
	10η	139,717.0	139,707.6	0.01	90	7,232.0	4,766.6	5,564.8
	avg.	**137,393.1**	**137,384.5**	**0.01**	**87**	**9,963.7**	**5,902.4**	**8,201.4**
I3	$\eta/10$	167,862.0	167,852.2	0.01	94	10,215.8	6,366.2	8,395.6
	$\eta/5$	167,977.0	167,967.8	0.01	111	10,599.1	5,986.0	8,586.8
	$\eta/2$	168,229.0	168,219.3	0.01	112	10,075.9	5,998.4	8,083.8
	η	168,504.0	168,493.7	0.01	128	10,957.0	5,786.1	8,602.5
	2η	168,832.0	168,822.9	0.01	100	10,064.7	6,558.3	7,771.3
	4η	169,212.0	169,201.8	0.01	107	9,978.4	6,129.7	7,460.4
	8η	169,654.0	169,644.4	0.01	92	8,045.2	5,341.3	5,963.6
	10η	169,804.0	169,794.2	0.01	75	7,409.8	5,446.4	5,714.0
	avg.	**168,652.3**	**168,642.8**	**0.01**	**97**	**9,634.9**	**6,001.1**	**7,591.3**

operational costs increase (Var. op. costs increase) values are obtained by comparing the R-MDVSP-STT solutions to the corresponding solutions of the MDVSP found in Table 4.4.

The R-MDVSP-STT solutions have variable operational costs up to 29.53% higher than the base solutions, but all use the same number of buses. This is desirable in practice because it implies that taking reliability into account does not affect the optimal fleet size. Significant reliability gains can be achieved without much deterioration in operational costs. For example, with $\beta = \eta/5$, the probability that a passenger boards a delayed timetabled trip in instance I2 is reduced from 23% to 15% with a variable operational costs increase of

Table 4.6 – Operational costs versus reliability of the R-MDVSP-STT solutions

Instance	β	Var. op. costs increase(%)	# Bus	Reliability					
				1st case			2nd case		
				γ	ψ	θ	γ	ψ	θ
I1	$\eta/10$	0.16	75	0.40	0.05	0.96	1.49	0.18	2.26
	$\eta/5$	0.54	75	0.33	0.04	0.78	1.01	0.13	1.41
	$\eta/2$	1.51	75	0.22	0.02	0.60	0.69	0.08	1.30
	η	2.82	75	0.16	0.02	0.43	0.43	0.04	1.00
	2η	5.28	75	0.11	0.01	0.36	0.31	0.03	0.70
	4η	7.97	75	0.08	0.01	0.34	0.22	0.02	0.56
	8η	12.82	75	0.07	0.01	0.24	0.20	0.02	0.49
	10η	13.57	75	0.06	0.01	0.24	0.20	0.02	0.49
I2	$\eta/10$	0.31	131	1.51	0.18	1.50	9.48	0.62	3.11
	$\eta/5$	1.38	131	1.20	0.15	1.29	7.96	0.60	2.70
	$\eta/2$	5.04	131	0.71	0.09	0.96	5.04	0.50	2.56
	η	8.81	131	0.49	0.06	0.76	3.08	0.37	2.28
	2η	14.51	131	0.33	0.04	0.56	1.58	0.21	1.39
	4η	20.34	131	0.25	0.03	0.46	1.02	0.13	1.12
	8η	27.49	131	0.20	0.02	0.40	0.75	0.10	0.98
	10η	29.53	131	0.19	0.02	0.38	0.71	0.09	0.86
I3	$\eta/10$	0.22	162	1.03	0.13	1.32	7.23	0.56	3.95
	$\eta/5$	0.59	162	0.90	0.11	1.16	6.09	0.53	2.99
	$\eta/2$	2.74	162	0.57	0.07	0.88	3.89	0.43	2.59
	η	5.54	162	0.36	0.04	0.73	2.39	0.29	2.13
	2η	10.62	162	0.18	0.02	0.54	1.02	0.12	1.57
	4η	15.02	162	0.11	0.01	0.40	0.53	0.06	1.10
	8η	22.37	162	0.06	0.01	0.23	0.23	0.02	0.54
	10η	24.15	162	0.05	<0.01	0.20	0.19	0.02	0.42

1.38%. Furthermore, in the first simulation case (i.e., under normal conditions), all three reliability metrics, namely γ, ψ, and θ, improve monotonically when the penalty factor β increases (compare columns under "1st case" in Tables 4.4 and 4.6). In the second simulation case (i.e., under extraordinary factors), the latter remark holds for most of the solutions with a few exceptions (compare columns under "2nd case" in Tables 4.4 and 4.6). All the exceptions concern the third metric, θ, which is the metric the least connected to the function minimized in (4.3.1)-(4.3.4). Specifically, the value of θ (second case) is higher than the one of the corresponding MDVSP solution for instances I1 and I2 with $\beta = \eta/10$. Still, our model produces overall good trade-offs between reliability and operational costs, even if

some inconsistencies are observed during extraordinary events. We will analyze in the next section how our model compares to a classical approach to address bus schedules' reliability.

4.6.3. Comparison with the MDVSP with minimum buffer time

In Naumann et al. (2011), van Kooten Niekerk (2018), Amberg et al. (2011), and Amberg et al. (2019), models that are aimed to improve the reliability of vehicle schedules are compared to a simple approach: adding *hard* minimum buffer time constraints to the traditional MDVSP. As in Naumann et al. (2011), van Kooten Niekerk (2018), Amberg et al. (2011), and Amberg et al. (2019), we have observed that all solutions found using the latter approach are very costly, use many additional vehicles, and most of them are largely dominated by the R-MDVSP-STT solutions (see Appendix A.2). Instead, we propose to compare our model to the MDVSP with *soft* minimum buffer time constraints. The soft constraints penalize, in the cost function, connections that do not meet the minimum buffer time. In our experiments, we set the numerical value of this penalty to 0.4 per minute below the minimum buffer time.

We tested several minimum buffer time rules that are drawn from the literature. The rules and the numerical values tested are the following:

— Global buffer time, i.e., the same buffer time is imposed after each timetabled trip (1, 2, 3, 4, 5, and 10 minutes);

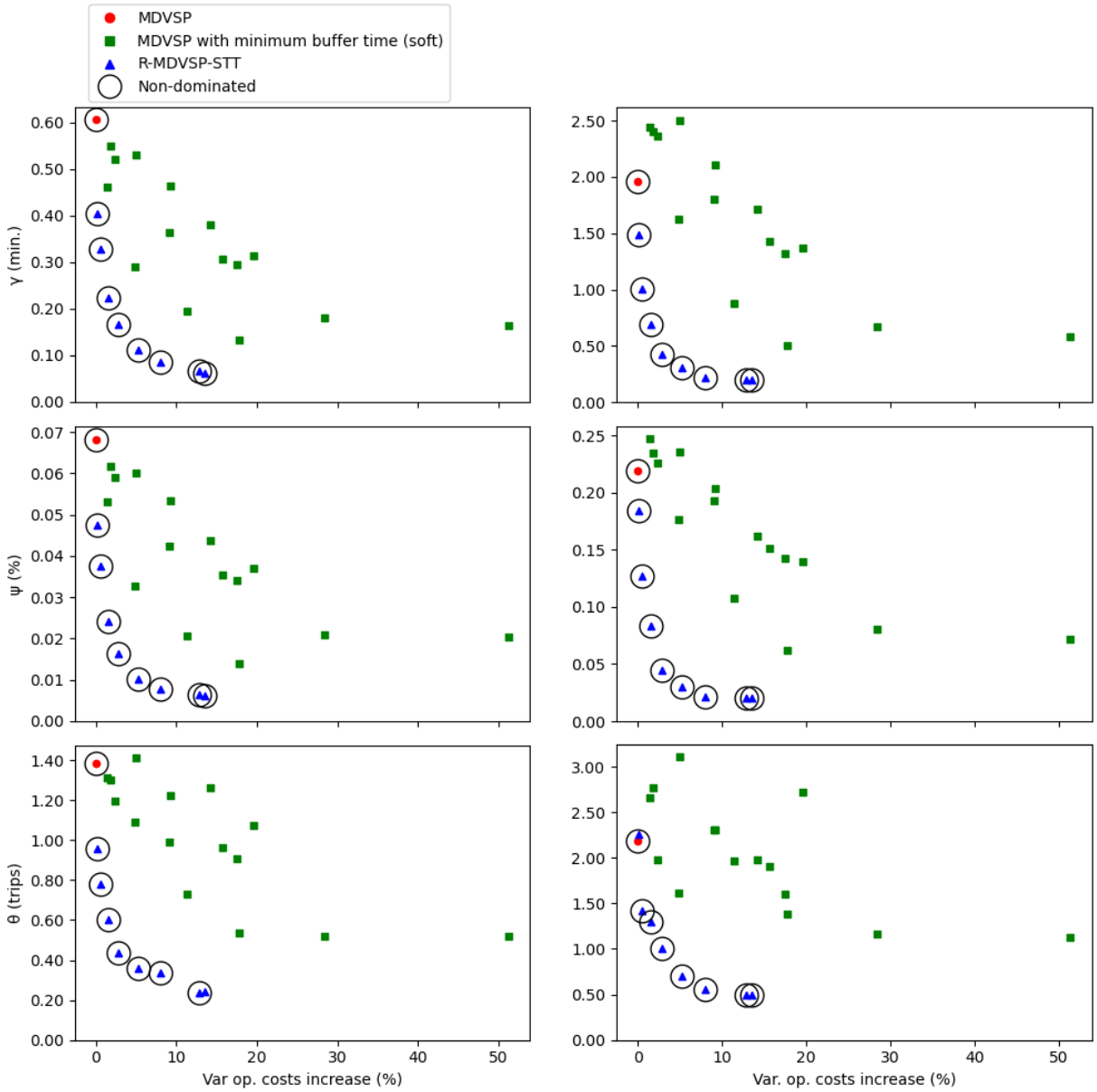
— Buffer times proportional to the duration of each timetabled trip (5%, 10%, 15%, and 20% of the expected trip duration);

— Buffer times to cover the primary delay of each timetabled trip x% of the time (50th, 75th, 90th, and 95th percentiles of $\dot{T}_i$, for $i \in \mathcal{V}$);

— Buffer times provided by our industrial partner.

Figures 4.3, 4.4 and 4.5 (one figure per instance) show the relationship between the value of the reliability metrics and the increase in variable operational costs (relative to the corresponding baseline solution) of the MDVSP, MDVSP with soft minimum buffer time and R-MDVSP-STT solutions. Non-dominated solutions are circled. The simulation results under normal conditions and under external and extraordinary factors are displayed in the left-hand and right-hand side graphs, respectively.

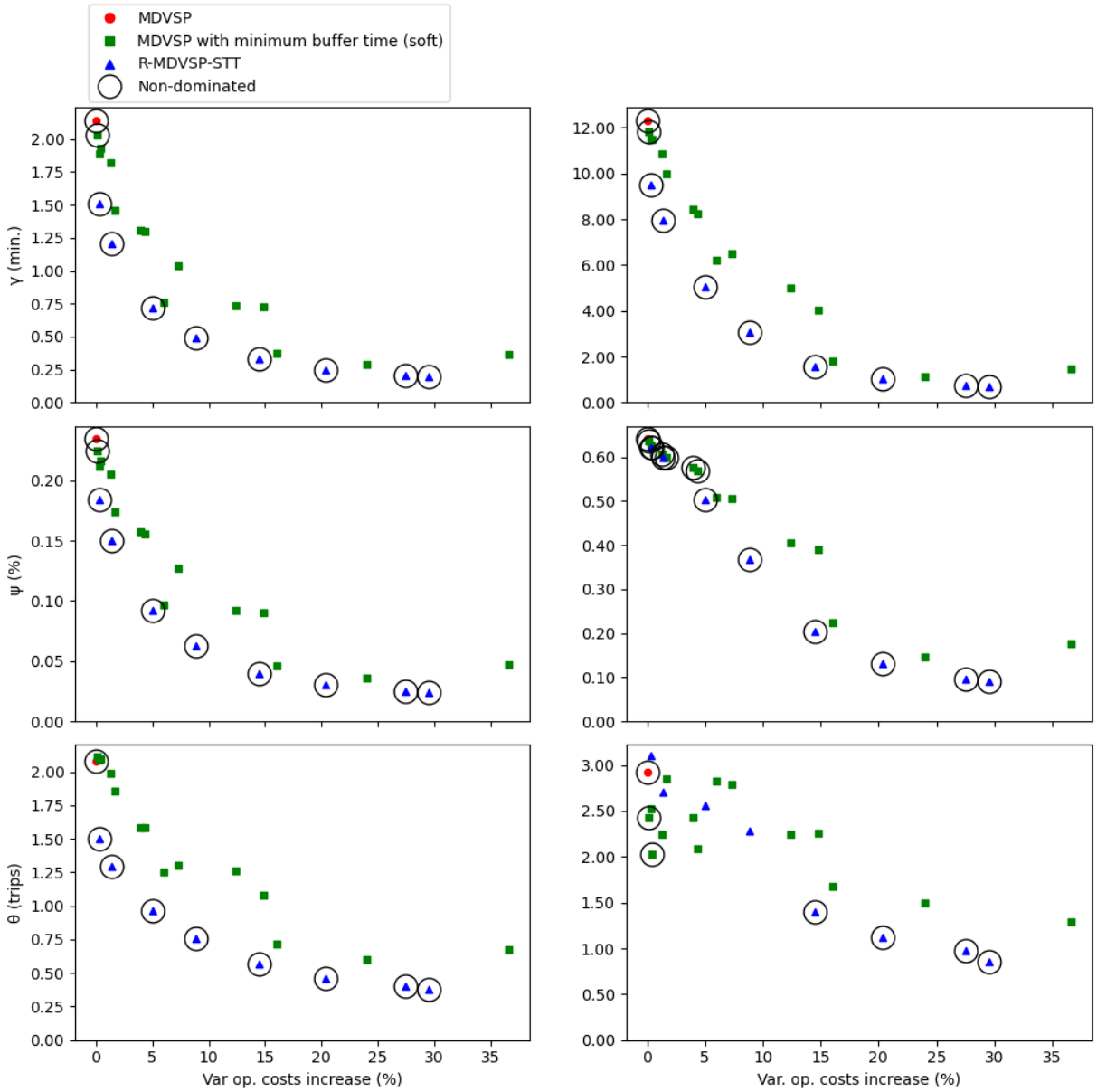
On the one hand, Figures 4.3a, 4.4a and 4.5a show that, under normal conditions, all the R-MDVSP-STT solutions are non-dominated with one exception (see the graph at the bottom of Figure 4.3a). On the other hand, Figures 4.3b, 4.4b and 4.5b show that all the solutions of the R-MDVSP-STT are non-dominated, in terms of metrics γ and ψ, for the second simulation case. In terms of θ, many of the R-MDVSP-STT solutions of instance I2 are dominated by the solutions of the MDVSP or the MDVSP with soft minimum buffer time, whereas all except one R-MDVSP-STT solutions of instance I1 and all R-MDVSP-STT



(a) 1st case **(b)** 2nd case

Figure 4.3 – Reliability metrics - I1

solutions of instance I3 are non-dominated in the second simulation case. The proportion of all non-dominated solutions that are R-MDVSP-STT solutions for each instance, scenario, and metric is summarized in Table 4.7. Each entry of this table represents the number of non-dominated R-MDVSP-STT solutions (where the maximum possible value is 8) on the total number of non-dominated solutions (including MDVSP, MDVSP with soft minimum buffer time, and S-MDVSP-STT).

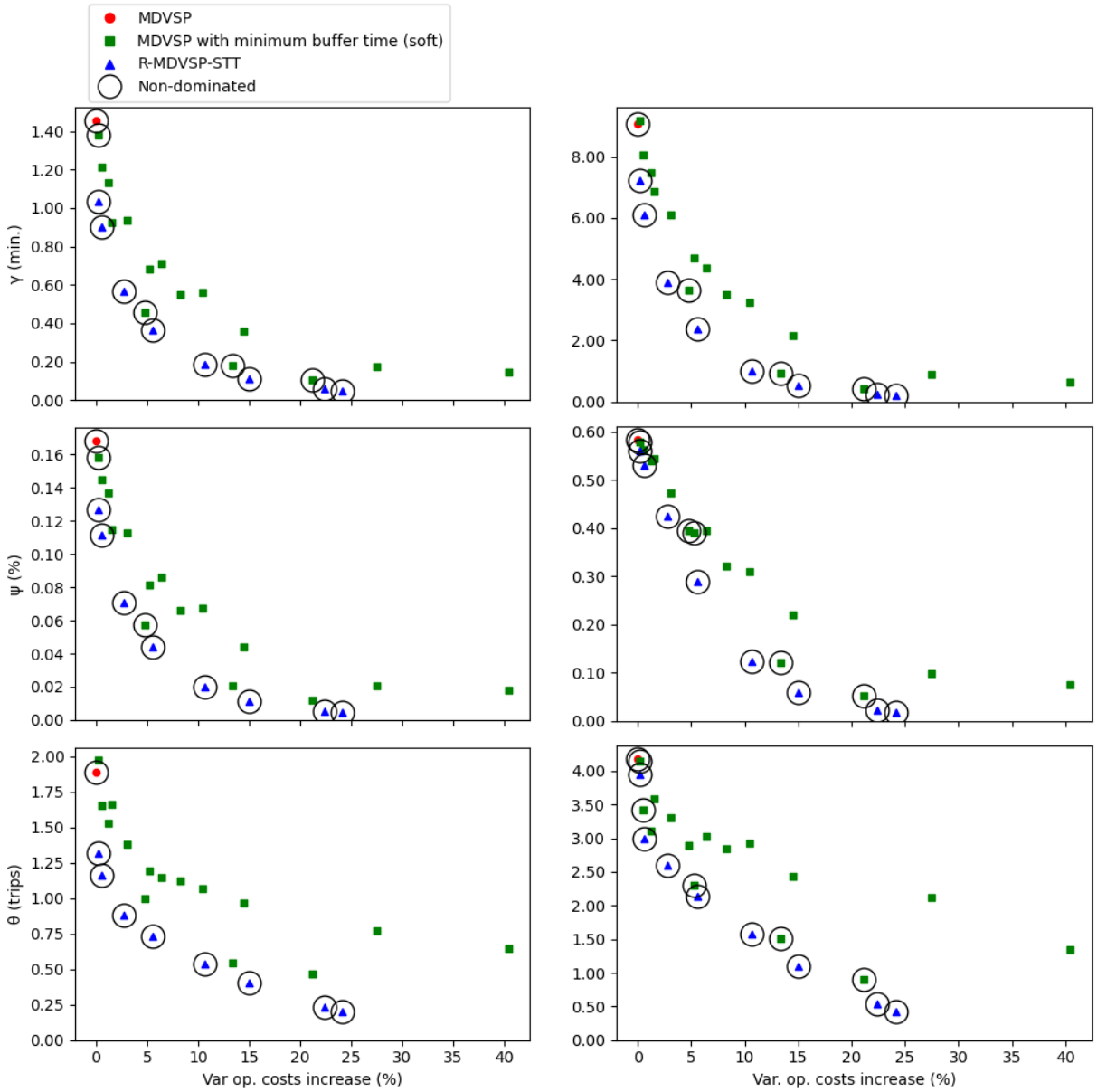


(a) 1st case **(b)** 2nd case

Figure 4.4 – Reliability metrics - I2

Table 4.7 – Proportion of all non-dominated solutions that are R-MDVSP-STT solutions

Instance	1st case			2nd case		
	γ	ψ	θ	γ	ψ	θ
I1	8/9	8/9	7/8	8/9	8/9	7/8
I2	8/10	8/10	8/9	8/10	8/15	4/7
I3	8/13	8/11	8/9	8/12	8/14	8/14



(a) 1st case **(b)** 2nd case

Figure 4.5 – Reliability metrics - I3

Furthermore, the solutions of the R-MDVSP-STT form approximate Pareto-fronts in both simulation cases. This feature is interesting for transport agencies because it allows them to easily adjust the level of reliability.

4.7. Conclusions

In this work, we proposed a model for the R-MDVSP-STT that can be solved using MIP technology featuring column generation. This model addresses the trade-off between two

conflicting objectives, namely minimizing the operational costs and minimizing the expected secondary delay per passenger. To evaluate the second objective, we introduced a method to compute the convolution of the probability mass function of the secondary delay of every timetabled trip in a vehicle schedule based on the discretized probability density functions of the travel time. Furthermore, a heuristic branch-and-price algorithm for solving the R-MDVSP-STT is proposed. In order to generate new columns (i.e., vehicle schedules) that are both cost- and delay-efficient, a modified version of the labeling algorithm that features a stochastic dominance criterion is used to solve the pricing problems.

We introduced three reliability metrics, two of which are passenger-oriented, and a simulation framework to compute them after solving the R-MDVSP-STT. Two simulation cases are tested. Delay propagation is assessed, first, under normal conditions (i.e., when travel times are subject to day-to-day variability) and second, under external and extraordinary factors (e.g., a severe snowstorm).

We conducted our experiments on three real-life instances of 1,175, 1,916, and 2,195 timetabled trips and 2 or 3 depots from the city of Montréal. Our experimental results indicate that our approach provides near-optimal trade-offs between operational costs and reliability in a reasonable amount of time. Specifically, under normal conditions, the R-MDVSP-STT solutions are more reliable in terms of the three reliability metrics than the corresponding solutions of the MDVSP. Moreover, all but one of the solutions of our model were not dominated by a MDVSP or MDVSP with minimum buffer time (enforced by soft or hard constraints) solution. In the presence of external and extraordinary factors, our model also generally provided better trade-offs between operational costs and reliability than traditional MDVSP with minimum buffer time in terms of the first and second metrics and still good solutions in terms of the third metric. What is more, our approach allows to easily reach a targeted reliability level by tuning the value of a weighing factor and selecting the solution in the approximate Pareto frontier with the desired trade-off between operational costs and reliability.

Further research avenues include extending this work to electric buses where recharging operations can induce even more delays, especially if energy consumption is also considered stochastic.

Chapter 5

Article 3 - The stochastic multi-depot electric vehicle scheduling problem with recourse

Prologue

> The stochastic multi-depot electric scheduling problem with recourse. Léa Ricard, Guy Desaulniers, Andrea Lodi, Louis-Martin Rousseau. This paper has been accepted at the 11th symposium of the European Association for Research in Transportation.

I contributed to the original idea of the paper and was responsible for modeling the problem mathematically, developing an efficient algorithm, writing an implementation guide for the programmer in charge of modifying GENCOL according to the developed algorithm, building test instances, running rests, analyzing the results, and writing the article. Guy Desaulniers, Andrea Lodi, and Louis-Martin Rousseau provided valuable feedback and contributed to the original idea of the paper.

5.1. Introduction

The multi-depot electric vehicle scheduling problem (MDEVSP) is an extension of the multi-depot vehicle scheduling problem with additional limitations, including shorter driving range, longer refueling time, and special charging infrastructure. It aims at finding a set of vehicle schedules that covers each timetabled trip exactly once while minimizing the operational costs and respecting energy feasibility and depot capacity constraints. These vehicle schedules are subject to operational uncertainties (e.g., traffic jams, extreme weather conditions, or special happenings in the city) that impact travel time and energy consumption. Nevertheless, the MDEVSP is generally solved without taking these uncertainties into account. This strong assumption may compromise schedule adherence and lead to solutions

with sub-optimal true costs (including recourse costs). A simple way to guarantee energy feasibility is to adopt a robust optimization approach, i.e., ensuring that energy feasibility is respected for the worst case energy consumption scenarios (see for example the work of Bie et al., 2021). Some less conservative approaches, that we group into stochastic optimization (Li et al., 2021), robust optimization with cardinality constrained set (Jiang et al., 2021), and dynamic optimization (Tang et al., 2019), have been proposed in the literature to address the MDEVSP with uncertain travel time and/or energy consumption.

This work presents the first stochastic model for the MDEVSP with stochastic travel time and energy consumption (S-MDEVSP). We formulate the S-MDEVSP as a two-stage stochastic program and introduce a recourse policy to recover energy feasibility when the vehicle schedules outputted a priori turn out to be infeasible. The main idea of our approach is to take advantage of the fact that charging time can be adjusted from day-to-day to cope with energy consumption deviations. This flexibility in the charging time could allow us to output less conservative vehicle schedules than the robust optimization approach while guaranteeing energy feasibility. However, this flexibility may also induce delays. To control the build-up of delays, that can also be caused by travel time deviations, we add a penalty for delays in the objective function as in Ricard et al. (2022). Our objective is to assess the relevance of our two-stage stochastic model for commercially available battery electric buses (BEBs). Precisely, we want to verify if a substantial reduction in the optimal fleet size can be archived by introducing a recourse policy. We propose a branch-and-price algorithm to solve this challenging optimization problem and test our solution approach on a real-life instance of the city of Montréal.

This paper is organized as follows. Section 5.2 deals with the problem definition and a two-stage stochastic program is introduced. We devise a method to compute the second stage cost analytically. A column generation-based solution approach is presented in Section 5.3. We present the results of computation in Section 5.4 and discuss the relevance of our approach for different commercially available BEBs. Our main conclusions are stated in Section 5.5.

5.2. Mathematical model

Let $\mathcal{V}$ be a timetable of trips, where trip $i \in \mathcal{V}$ is scheduled to start at d_i^0, $\mathcal{D}$ be a set of depots, such that $|\mathcal{D}| \geq 2$, $\mathcal{H}$ be a set of charging stations, and $\mathcal{R}$ be a set of k time intervals each of duration ρ. Let $\mathcal{H}^E$ be a set of nodes, where each node is associated with a charging station in $\mathcal{H}$ and a time interval in $\mathcal{R}$. For example, denote $h_1^E r_5$ the node associated with charging station $h_1 \in \mathcal{H}$ and time interval $r_5 \in \mathcal{R}$. The S-MDEVSP is defined on the acyclic connection-based networks $G^d(V_d,A_d)$, for $d \in \mathcal{D}$, with node set $V_d = \mathcal{V} \cup \{n_0^d, n_1^d\} \cup \mathcal{H}^E$, where n_0^d and n_1^d represent depot d at the beginning and the end of the day, respectively, and

arc set A_d. Given the probability mass function (PMF) with finite supports of the travel time ($h_i(t)$) and the PMF of the energy consumption ($e_i(\mu)$) of each timetabled trip $i \in \mathcal{V}$ as well as the travel time κ_{ij} and the energy consumption ι_{ij} between the end location of node i and the start location of node j, for all pairs of nodes i,$j \in V_d$, the first stage problem of the S-MDEVSP consists of finding an a priori set of vehicle schedules $\mathcal{S}^*$ that covers exactly once each trip $i \in \mathcal{V}$ and respects the capacity b_d of each depot $d \in \mathcal{D}$. A vehicle schedule is defined as a sequence of timetabled trips and time-expanded charging nodes starting and ending at a depot $d \in \mathcal{D}$. The amount of energy recharged at each time-expanded charging node included in a vehicle schedule is derived from a piecewise linear function similar to the one used in Montoya et al. (2017). In the second stage, the travel time and energy consumption values are revealed and the a priori plan is modified with respect to a recourse policy to guarantee energy feasibility. A vehicle schedule is considered feasible if the state of charge (SoC) of the BEB never falls below φ^{min} (e.g., 0%), or if one or several recourse actions can be taken to regain energy feasibility. A recourse action is taken at the second stage if the SoC of a bus is under ω (e.g., 50%) after a charging activity. It consists in extending the charging activity by one or several time intervals in order to reach a SoC of at least ω.

Our model for the S-MDEVSP uses the following notation. Let $\mathcal{S}$ be the set of all feasible vehicle schedules, $\mathcal{S}^d$ be the subset of these schedules starting and ending at the depot d, y_s be a binary variable equal to 1 if vehicle schedule s is selected, and a_{is} be a binary parameter equal to 1 if schedule s covers trip $i \in \mathcal{V}$. The S-MDEVSP can be formulated as the following integer linear program:

$$\min \quad \sum_{s \in \mathcal{S}} \bar{c}_s y_s \tag{5.2.1}$$

$$\text{s.t.} \quad \sum_{s \in \mathcal{S}} a_{is} y_s = 1, \qquad \forall i \in \mathcal{V} \tag{5.2.2}$$

$$\sum_{s \in \mathcal{S}^d} y_s \leq b_d, \qquad \forall d \in \mathcal{D} \tag{5.2.3}$$

$$y_s \in \{0,1\}, \qquad \forall s \in \mathcal{S}, \tag{5.2.4}$$

where $\bar{c}_s = c_s + \beta \mathbb{E}[Q_s(t,\mu)]$ is the expected cost of vehicle schedule s, c_s is the operational costs of s, β is a weighting factor, and $\mathbb{E}[Q_s(t,\mu)]$ is the expected second-stage cost of s. This latter cost penalizes the delay a passenger is likely to encounter in schedule s. Specifically, $\mathbb{E}[Q_s(t,\mu)] = \sum_{i \in s \cup \mathcal{V}} \alpha_i \mathbb{E}(X_i^s)$, where α_i is the relative passenger flow (or demand volume) on timetabled trip i and X_i^s is the secondary delay of timetabled trip i covered by schedule s (in minutes). A vehicle schedule s may be delayed because the travel times of its trips deviate from the planned time, because buffer times before trips are not sufficient, or because recourse actions are required. By adjusting the weighting factor β, one can find solutions

with different trade-offs between operational costs and the expected second-stage cost. In general, the larger the β the more reliable the S-MDEVSP solutions. Analytical equations to compute $\mathbb{E}[Q_s(t,\mu)]$ in the first stage, for all $s \in \mathcal{S}$ generated, are developed in the following two sections.

5.2.1. Probability of using a recourse action

Consider a vehicle schedule $s = (1,2,\ldots,i,i+1,\ldots,j-1,j,\ldots,n)$ with trips i and $j \in \mathcal{V}$ interspersed by a charging activity of $j-i$ time intervals (i.e., $i+1, i+2, \ldots, j-1$ are time-expanded charging nodes). Let $m_j^s(z)$ be the PMF with finite supports of the SoC of bus s just before trip j. The probability of not having to extend the charging time is

$$\Pr\{0 \text{ charge period before } j \in s\} = \sum_{z=\omega}^{100} m_j^s(z), \tag{5.2.5}$$

and the probability of having to extend the charging time of ϕ charge periods is

$$\begin{aligned}\Pr\{\phi \text{ charge periods before } j \in s\} &= \Pr\{z|\Lambda_\omega(z) = \phi\} \\ &= \sum_{z=\varphi^{min}}^{\omega-1} m_j^s(z)[\Lambda_\omega(z) = \phi], \qquad \phi = 1,2,\ldots,k,\end{aligned} \tag{5.2.6}$$

where $\Lambda_\omega(z)$ is a function outputting the minimum number of additional charge time periods to be performed when the initial SoC of a BEB is equal to z in order to get an updated SoC of at least ω. We use the Iverson bracket (Iverson, 1962) notation (i.e., $[P]$ is equal to 1 if P is true and 0 otherwise).

5.2.2. Delay propagation

Let $f_i^s(y)$ be the PMF with finite supports of the actual start time of activity i assigned to schedule s and $g_i^s(x)$ be the PMF with finite supports of X_i^s, i.e., of the secondary delay of trip i, such that $g_i^s(x) = f_i^s(x + d_i^0)$ when $i \in \mathcal{V}$. For $i \notin \mathcal{V}$, $g_i^s(x)$ is not defined.

Consider a schedule $s = (1,2\ldots,n)$ and denote $P_{is}^0 := \Pr\{0 \text{ extra chr per. before } i \in s\}$ and $P_{is}^\phi := \Pr\{\phi \text{ extra chr per. before } i \in s\}$. We assume that the first timetabled trip of a vehicle schedule s is never delayed (i.e., $f_1^s(d_1)) = 1$). Consider a trip $j \in (2,3,\ldots,n)$ preceded by a trip i. The distribution of the actual start time of trip j can be recursively computed as

$$f_j^s(y) = \begin{cases} \sum_{t=t_i^{min}}^{t_i^{max}} h_i(t) \sum_{y'=d_i^0}^{d_j^0-t-\Upsilon(i,j)} \left[f_i^s(y') P_{js}^0 + \sum_{\phi=1}^{k} f_i^s(y'-\rho\phi) P_{js}^{\phi} \right], & \text{if } y = d_j^0; \\ \sum_{t=t_i^{min}}^{t_i^{max}} h_i(t) \left[f_i^s(y-t-\Upsilon(i,j)) P_{jrs}^0 + \sum_{\phi=1}^{k} f_i^s(y-t-\Upsilon(i,j)-\rho\phi) P_{js}^{\phi} \right], & \text{if } y > d_j^0; \\ 0, & \text{otherwise,} \end{cases} \tag{5.2.7}$$

where t_i^{min} and t_i^{max} are the minimum and the maximum possible travel time values of timetabled trip $i \in \mathcal{V}$, respectively, and $\Upsilon(i,j)$ is equal to $\kappa_{ij} + \tau$ if there is no charging activity between trips i and j, or $\kappa_{iq} + \kappa_{qj} + \tau + (j-i)\rho$ if there is a charging activity of $j-i$ time intervals at station $h \in \mathcal{H}$ between trips i and j. Here, τ is the minimum layover time before each timetabled trip.

The expected secondary delay of a trip j assigned to s is expressed as $\mathbb{E}(X_j^s) = \sum_{x=0}^{x_{js}^{max}} x \times f_j^s(x + d_j^0)$, where $x_{js}^{max} = d_i^0 + x_{is}^{max} + \kappa_{ij} + \tau + t_i^{max} - d_j^0$ is the maximum possible secondary delay of trip j when covered by vehicle schedule s. It should be observed that $f_j^s(y)$, $m_j^s(z)$, P_{is}^0, and P_{is}^{ϕ} are, by definition, schedule-dependent. Since the schedules are not enumerated but rather generated in our algorithm, is it impossible to compute $f_j^s(y)$, $m_j^s(z)$, P_{is}^0, and P_{is}^{ϕ} for all $i \in \mathcal{V}$ and $s \in \mathcal{S}$ beforehand. Instead, the latter are dynamically generated throughout the solution process.

Every time a trip i is delayed of 1 minute, a penalty of $\beta\alpha_i$ is paid. Depending on the transport agency's level of delay aversion, the weighting factor β can be adjusted to find an appropriate trade-off between the operational costs and reliability. Generally speaking, the larger the β, the more reliable (or delay-tolerant) the S-MDEVSP solutions.

5.3. Heuristic branch-and-price algorithm for the S-MDEVSP

Since there is generally a very large number of feasible vehicle schedules in the S-MDEVSP, we propose a branch-and-price solution approach that generates columns (i.e., vehicle schedules) instead of enumerating them. We use the same heuristic branching strategy as in Ricard et al. (2022) to obtain integer solutions in a reasonable amount of time.

To identify columns that could be potentially useful to add, we solve one pricing problem per depot $d \in \mathcal{D}$ at each iteration. These pricing problems are defined on the networks G^d, for $d \in \mathcal{D}$, with modified arc costs $\tilde{c}_{ij}^s$ defined as

$$\tilde{c}_{ij}^s = \begin{cases} \bar{c}_{ij}^s - \pi_d, & \text{if } i = n_o^d \\ \bar{c}_{ij}^s - u_i, & \text{if } i \in \mathcal{V}, \end{cases} \tag{5.3.1}$$

where $(u_i)_{i\in\mathcal{V}}$ and $(\pi_d)_{d\in\mathcal{D}}$ are dual variables associated with constraints (5.2.2) and (5.2.3), respectively.

Since the cost of the arcs is stochastic and path-dependent in the S-MDEVSP, these pricing problems correspond to shortest path problems with stochasticity (Boland et al., 2015; Wellman et al., 2013) that can be solved by a modified version of the labeling algorithm (see Ahuja et al., 1993, for more details on this algorithm). Next, we specify the main characteristics of the dynamic programming algorithm, namely the labels, the extension functions, and the stochastic dominance criteria.

5.3.1. Labels

Each label stores a representation of the actual start time cumulative distribution function (CDF), a representation of the SoC CDF, and the accumulated reduced cost. Let $F_j^p(y)$ be the CDF of $f_j^p(y)$ at node j defined as

$$F_j^p(y) = \sum_{y'=d_j^0}^{y} f_j^p(y'), \tag{5.3.2}$$

and let $M_j^p(z)$ be the CDF of $m_j^p(z)$ at node j defined as

$$M_j^p(z) = \sum_{z'=\varphi^{min}}^{z} m_j^p(z'). \tag{5.3.3}$$

The label L_j^p of path p at node j is defined as $L_j^p = (F_j^p(d_j^0), \ldots, F_j^p(y_{jp}^{max}), M_j^p(\varphi^{min}), \ldots, M_j^p(100), C_j^p)$, where y_{jp}^{max} is the maximum value of $F_j^p(y)$ and C_j^p is the accumulated reduced cost.

5.3.2. Extension functions

We want to extend a label $L_i^{p'} = (F_i^{p'}(d_i^0), \ldots, F_i^{p'}(y_{ip'}^{max}), M_i^{p'}(\varphi^{min}), \ldots, M_i^{p'}(100), C_i^{p'})$ associated with node i along arc (i,j) to create label L_j^p. The accumulated reduced cost C_j^p at node j is given by

$$C_j^p = C_i^{p'} + \tilde{c}_{ij}^p. \tag{5.3.4}$$

In Section 5.2, we devised a method to analytically compute the propagation of delays in a sequence of timetabled trips. Here, we specify this method in the form of an extension function. The PDF of the actual start time of trip j covered by path p is given by

$$f_j^p(y) = \begin{cases} \sum_{t=t_i^{min}}^{t_i^{max}} h_i(t) \sum_{y'=d_i^0}^{d_j^0-t-\kappa_{ij}-\tau} f_i^{p'}(y'), & \text{if } i,j \in \mathcal{V} \text{ and } y = d_j^0 \\ \sum_{t=t_i^{min}}^{t_i^{max}} h_i(t) f_i^{p'}(y-t-\kappa_{ij}-\tau), & \text{if } i,j \in \mathcal{V} \text{ and } y > d_j^0 \text{ or } i \in \mathcal{V} \text{ and } j \in \mathcal{H}^E \\ f_i^{p'}(y-\rho), & \text{if } i,j \in \mathcal{H}^E \\ \sum_{y'=d_i^0}^{d_j^0-\kappa_{ij}-\tau} \left[f_i^{p'}(y') P_{jp}^0 + \sum_{\phi=1}^{k} f_i^{p'}(y'-\rho\phi) P_{jp}^{\phi} \right], & \text{if } i \in \mathcal{H}^E, j \in \mathcal{V}, \text{ and } y = d_j^0 \\ f_i^{p'}(y-\kappa_{ij}-\tau) P_{jp}^0 + \sum_{\phi=1}^{k} f_i^{p'}(y-\kappa_{ij}-\tau-\rho\phi) P_{jp}^{\phi}, & \text{if } i \in \mathcal{H}^E, j \in \mathcal{V}, \text{ and } y > d_j^0 \\ 0, & \text{otherwise.} \end{cases} \tag{5.3.5}$$

The components $M_j^p(\cdot)$ are computed as

$$m_j^p(z) = \begin{cases} \sum_{\mu=\varphi^{min}}^{100} e_i(\mu) m_i^{p'}(z+\mu+\iota_{ij}), & \text{if } i \in \mathcal{V} \\ \sum_{z'=\varphi^{min}}^{100} m_i^{p'}(z')[\lambda(z', \rho \times \Lambda_\omega(z')) = z], & \text{if } i,j \in \mathcal{H}^E \\ m_i^{p'}(z+\iota_{ij}), & \text{if } i \in \mathcal{H}^E, j \in \mathcal{V}, P_{jp}^0 = 1 \\ m_i^{p'}(z+\iota_{ij}) + \sum_{z'=0}^{\omega-1} m_i^{p'}(z')[\lambda(z', \rho \times \Lambda_\omega(z')) = z], & \text{if } i \in \mathcal{H}^E, j \in \mathcal{V}, z \geq \omega, 1 - P_{jp}^0 > 0 \\ 0, & \text{otherwise,} \end{cases} \tag{5.3.6}$$

where $\lambda(z,t)$ is a piecewise linear function giving the final SoC of a battery after a charge of t minutes that started with an initial SoC of z. We assume all BEBs start the day fully charged.

5.3.3. Stochastic dominance criteria

Consider two paths p^1 and p^2 with resident node i. Path p^1 dominates path p^2 when the following conditions hold:

(1) $C_i^{p^1} \leq C_i^{p^1}$

(2) $F_i^{p^1}(y) \geq F_i^{p^2}(y)$, for all $y \in \{d_i^0, d_i^0+1, \ldots, d_i^0 + \max\{y_{ip^1}^{max}, y_{ip^2}^{max}\}\}$

(3) $M_i^{p^1}(z) \leq M_i^{p^1}(z)$, for all $z \in \{\varphi^{min}, \varphi^{min}+1, \ldots, 100\}$

All dominated paths can be safely discarded because they are not part of the Pareto-optimal set of paths or will not be extended into Pareto-optimal paths.

5.4. Computational results

We test our model on a real-life instance from the city of Montréal of 273 timetabled trips, 2 depots, and 2 charging stations. To minimize battery degradation, φ^{min} is set to $\varphi^{min} = 25\%$. We compare our approach for two different types of commercially available BEBs; an electric shuttle with battery capacity (W) of $W = 80$ kWh, charger power (P) of $P = 60$ kW, and an average consumption rate of 0.76 kWh/km (Gao et al., 2017) and a 35-foot transit bus with $W = 492$ kWh, $P = 221$ kW, and an average consumption rate of 1.57 kWh/km (Proterra, 2022). We assume the energy consumption distributions follow a normal distribution.

The heuristic performance of our solution approach and the quality of the solutions are reported in Table 5.1 and 5.2, for the first and second type of BEB, respectively, for β values ranging from 0 to η, where η is the cost per bus used, and ω values ranging from φ^{min} to 75% of the battery capacity. When $\omega = \varphi^{min}$, our approach is equivalent to a robust optimization approach (i.e., no corrective actions). The columns display the relative difference in percentage between the upper bound and the lower bound (Gap), the number of branching nodes explored (Nodes), the computing times (CPU time), including the total time in seconds (Total), the portion of the total time dedicated to solve the root node (Root) and the pricing problems (Pricing), the operational costs (Op. costs), the fleet size (bus) and the total penalty for delays ($\sum_{s\in\mathcal{S}^*}\mathbb{E}[Q_s(t,z)]$).

Table 5.1 – S-MDEVSP heuristic performance and quality of the solutions, with $W = 80$ kWh and $P = 60$ kW

		Heuristic performance					Quality of the solutions		
				CPU time					
β	ω	Gap (%)	Nodes	Total (s)	Root (%)	Pricing(%)	Op. costs	Bus	$\sum_{s\in\mathcal{S}^*}\mathbb{E}[Q_s(t,z)]$
0	φ^{min}	0.05	28	1,361	39.0	99.4	32,301.8	30	0.57
	35	0.05	28	1,362	39.4	99.5	32,301.8	30	0.57
	50	0.05	27	1,567	42.1	99.5	31,374.0	29	0.70
	75	0.09	27	2,054	38.5	99.6	31,438.4	29	0.67
	avg.	**0.06**	**28**	**1,586**	**39.8**	**99.5**	**31,854.0**	**30**	**0.63**
$\eta/2$	φ^{min}	0.13	28	3,264	34.7	99.8	32,364.0	30	0.24
	35	0.13	28	3,152	34.2	99.8	32,364.0	30	0.24
	50	0.14	28	5,170	24.9	99.9	31,451.6	29	0.26
	75	0.16	29	6,341	28.3	99.9	31,430.8	29	0.33
	avg.	**0.14**	**28**	**4,482**	**30.5**	**99.8**	**31,902.6**	**30**	**0.27**
η	φ^{min}	0.09	27	3,236	38.2	99.8	32,431.2	30	0.12
	35	0.09	27	3,226	37.8	99.8	32,431.2	30	0.12
	50	0.14	30	5,681	25.8	99.9	31,526.0	29	0.15
	75	0.08	25	6,032	36.3	99.9	31,471.0	29	0.22
	avg.	**0.10**	**27**	**4,544**	**34.5**	**99.9**	**31,964.8**	**30**	**0.15**

For both vehicle types, all problems are solved in less than 2 hours with almost all the computing time spent on solving the pricing problems. Also, the solutions obtained with our approach are at most 0.16 % more expensive than their corresponding lower bound, suggesting that our heuristic can find near-optimal solutions. Generally speaking, when β increases, the operational costs increase and the reliability improves.

For the first type of BEB, namely the shuttle with $W = 80$ kWh and $P = 60$ kW, the introduction of the recourse policy provides significant cost savings. Indeed, the fleet size can

Table 5.2 – S-MDEVSP heuristic performance and quality of the solutions, with $W = 492$ kWh and $P = 221$ kW

		Heuristic performance					Quality of the solutions		
				CPU time					
β	ω	Gap (%)	Nodes	Total (s)	Root (%)	Pricing(%)	Op. costs	Bus	$\sum_{s\in\mathcal{S}^*} \mathbb{E}[Q_s(t,z)]$
0	φ^{min}	0.02	21	1,059	49.7	99.6	27,290.4	26	0.49
	35	0.02	21	1,434	53.8	99.6	27,290.4	26	0.49
	50	0.02	21	1,145	49.4	99.6	27,290.4	26	0.49
	75	0.01	19	1,106	53.3	99.6	27,286.4	26	0.41
	avg.	**0.02**	**21**	**1,186**	**51.5**	**99.6**	**27,289.4**	**26**	**0.47**
$\eta/2$	φ^{min}	0.01	19	2,025	49.2	100.0	27,325.2	26	0.16
	35	0.01	19	2,855	53.4	100.0	27,325.2	26	0.16
	50	0.01	19	2,051	49.1	100.0	27,325.2	26	0.16
	75	0.01	21	2,567	49.1	99.9	27,325.2	26	0.16
	avg.	**0.01**	**20**	**2,374**	**50.2**	**100.0**	**27,325.2**	**26**	**0.16**
η	φ^{min}	0.01	21	2,091	46.7	100.0	27,362.6	26	0.11
	35	0.01	21	3,002	50.1	100.0	27,362.6	26	0.11
	50	0.01	21	2,294	47.0	99.9	27,362.6	26	0.11
	75	0.01	22	2,051	47.5	99.9	27,364.0	26	0.11
	avg.	**0.01**	**21**	**2,360**	**47.8**	**99.9**	**27,363.0**	**26**	**0.11**

be reduced from 30 BEBs to 29 BEBs by introducing a recourse policy with $\omega \geq 50$, which could be considered as a substantial reduction since the number of vehicles used constitutes the major part of the operational costs. Furthermore, the deterioration in reliability that the charging policy introduces can be compensated for by a higher weighting factor β.

For the second type of BEB, namely the 35-foot transit bus with $W = 492$ kWh and $P = 221$ kW, introducing a recourse policy does not improve the cost of the solutions found nor does it reduce the size of the fleet. Thus, for this second type of vehicle with larger battery capacity and higher charging power, our approach is not useful and a simple robust optimization approach should be used to find vehicle schedules such that the vehicles never run out of energy. Indeed, for this type of BEB, the battery capacity is large enough that vehicles only need to charge once or twice a day. Because timetables of trips typically include off-peak periods with fewer trips to make, charging activities can easily be scheduled during these periods and batteries are often charged to their maximal capacity. In this context, the recourse policy we introduced is never activated.

5.5. Conclusions

In this work, we introduced a stochastic model for the MDEVSP that we formulated as a two-stage stochastic program with a recourse action. We proposed an efficient branch-and-price algorithm to solve this challenging problem. Our results indicated that the use of recourse actions is beneficial for shuttle BEBs with relatively small battery capacity and charging power, but not for 35-foot transit BEBs with larger battery capacity and charging power. Medium- to large-scale transit agencies are typically equipped with up-to-date BEBs that resemble the second type of vehicle tested, so our approach is probably not relevant for them. However, our two-stage stochastic model may be relevant for smaller transit agencies or those with access to fewer resources. Future work includes translating our approach to other routing problems with smaller electric vehicles, for example the electric dial-a-ride problem.

Chapter 6

Article 4 - Strategies for the electric bus scheduling problem with battery degradation: a stochastic perspective

Prologue

> Strategies for the electric bus scheduling problem with battery degradation: a stochastic perspective. Léa Ricard, Guy Desaulniers, Andrea Lodi, Louis-Martin Rousseau. To be submitted.

I originated the idea for the paper, while Guy Desaulniers, Andrea Lodi, and Louis-Martin Rousseau contributed to shaping it into its current form by providing valuable feedback. I was responsible for modeling the problem mathematically, developing an efficient algorithm, writing an implementation guide for the programmer in charge of modifying GENCOL according to the developed algorithm, building test instances, running tests, analyzing the results, and writing the article. Guy Desaulniers revised the model and the algorithm. Guy Desaulniers, Andrea Lodi, and Louis-Martin Rousseau revised and edited the article.

6.1. Introduction

The transition of public transport bus fleet towards battery electric buses (BEBs) is gaining momentum worldwide as a solution to reduce local air and noise pollution. Cities like Paris and Copenhagen have already committed to converting their entire bus fleets to electric buses by 2025 (Perumal et al., 2022), and it is projected that nearly half of the buses globally will be electric in the near future (Abdelaty et al., 2021). However, the adoption of BEBs brings its share of challenges and additional constraints. Compared to traditional internal combustion engine buses, BEBs require longer refueling times, have shorter driving

ranges, and rely on charging infrastructure that is often limited in capacity. Additionally, the battery packs used in BEBs, predominantly lithium-ion batteries (Zhang et al., 2021), are expensive and represent a significant portion of these vehicles' total cost of ownership. In some cases, the daily cost of battery degradation is even more than twice the daily charging cost (Zhou et al., 2022). Lithium-ion battery packs must be replaced periodically due to the gradual fading during charging and discharging cycles (cycle aging) and storage time (calendar aging). Typically, a battery is considered to have reached its end of life (EoL) when its capacity drops to 70-80% of its initial capacity (Lam and Bauer, 2012; Zhang et al., 2019). Cluzel and Douglas (2012) reported that the typical lifespan of electric car batteries ranges between five to ten years, depending on the operating and storage conditions. However, the lifespan of BEBs tends to be even shorter due to their extensive usage. After this period, the battery packs of the BEBs must be replaced, which results in high maintenance costs. From a price perspective, it is thus essential to consider battery aging mechanisms when scheduling BEBs. This approach is also relevant from an environmental point of view, as the production of lithium-ion batteries accounts for about 20% of CO_2 equivalent emitted during the manufacture of BEBs (Nordelöf et al., 2019).

The degradation mechanisms of lithium-ion batteries are influenced by various factors, including storage and operating conditions, as well as cell chemistry, all of which interact with one another (Pelletier et al., 2017). Consequently, degradation mechanisms are highly complex, and modeling them is difficult. Nevertheless, certain factors have been identified as accelerators of battery aging, such as overcharging, over-discharging, extreme temperatures, high state-of-charge (SoC) during storage, large depth-of-discharge (DoD), high charging and discharging rates, high average SoC, and high SoC deviation (Pelletier et al., 2017; Millner, 2010; Lam and Bauer, 2012). Aside from temperature and the charging process, battery aging is predominantly influenced by the SoC, whether it involves extreme SoC values or the SoC range experienced throughout the day. Figure 6.1 illustrates an example of capacity fading (measured in kWh/year) and the lifetime of a BEB's battery cycled to different SoC ranges obtained using the capacity fading model for $LiFePO_4$ batteries of Lam and Bauer (2012). We consider a BEB with a battery capacity of 300 kWh, an average energy consumption of 400 kWh per day, and an EoL threshold of 20%. By reducing the SoC range from 100%-0% to 70%-20%, the battery's lifetime of a BEB can be extended by more than 10 years. This simple example highlights the importance of considering battery degradation when planning vehicle schedules.

At the planning stage, vehicle schedules are determined by solving the vehicle scheduling problem (VSP). This problem consists of finding the optimal bus assignment to timetabled trips, each defined by a start time and location, an itinerary composed of a sequence of stops, and an end time and location - such that each timetabled trip is covered exactly one by a vehicle schedule. This problem and some of its extensions, for example, the VSP

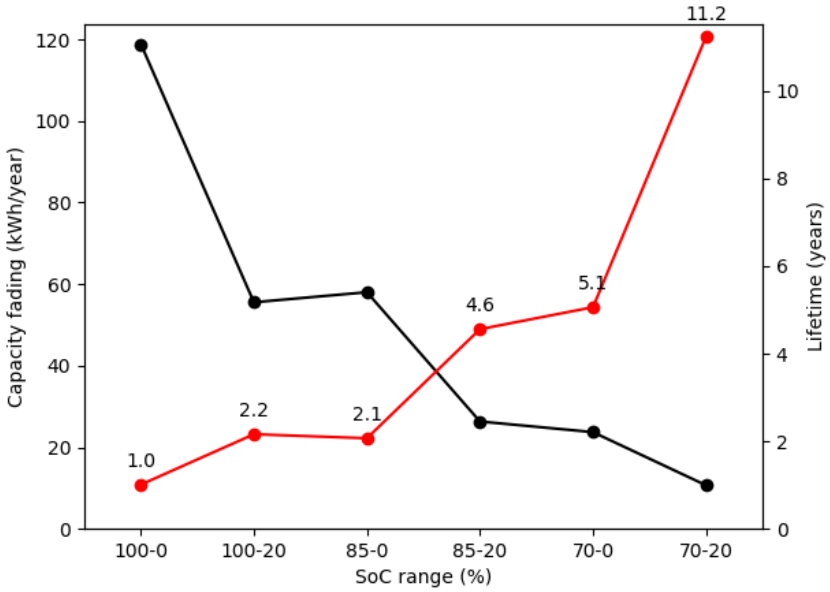


Figure 6.1 – Example of the capacity fading (in black) and lifetime (in red) of a BEB cycled at SoC ranges of $100\% - 0\%, 100\% - 20\%, \ldots, 70\% - 20\%$

with multiple depots, have been extensively studied in the past decades. Several models and algorithms (exact and heuristic ones) have been proposed, notably those of Ribeiro and Soumis (1994), Löbel (1998), Hadjar et al. (2006), Freling et al. (2001), and Kliewer et al. (2006). A detailed subject overview is provided in Desaulniers and Hickman (2007) and Bunte and Kliewer (2010).

In recent years, the extension of the VSP to BEBs, namely the electric VSP (E-VSP), has gained significant attention in the literature. In the E-VSP, BEBs must satisfy driving range constraints and charging requirements. If the single-depot VSP can be solved in polynomial time, the single-depot E-VSP is NP-hard (Sassi and Oulamara, 2017). Several works addressed the single-depot case, in particular, those of Li (2014), van Kooten Niekerk et al. (2017), Janovec and Kohání (2019). In their work, Li (2014) provided a model for the VSP that considers BEBs with either fast charging or battery swapping. Their model uses a fixed battery service time sufficiently large to swap or fully recharge the batteries. Each charging station is time-expanded, and a time discretization technique is used so that the capacity of each charging station can be constrained. A column generation-based algorithm is devised, and computational experiments on real-world and randomly generated instances are carried out. In van Kooten Niekerk et al. (2017), two alternative models allowing BEBs to be partially charged are proposed. The first and second models are with continuous and discrete SoC, respectively. The latter can also account for time-of-day electricity pricing, nonlinear charging time, and battery lifetime. A regular column generation-based algorithm

is compared with a faster version with Lagrangian relaxation over four datasets. The discrete model provided solutions using the same number of buses but with more waiting time than their continuous model. Janovec and Kohání (2019) introduced in a short proceeding paper an E-VSP model with partial charging and limited charger capacity. A set of "charging events", one per timetabled trip, is defined for each charger. The number of consecutive charging events in a vehicle schedule determines the charging time. An extension of the E-VSP, namely the E-VSP with a mixed fleet, is addressed in Olsen and Kliewer (2020), Rinaldi et al. (2020), and Alvo et al. (2021). Furthermore, Perumal et al. (2021) proposed an algorithm combining an adaptive large neighborhood search and column generation to tackle the integrated vehicle and crew scheduling problem. Recently, Wu et al. (2022) presented a bi-objective model for the multi-depot E-VSP (MDEVSP) considering grid characteristics, namely time-of-use pricing and peak load risk. Wang et al. (2021) also studied the MDEVSP and proposed a column generation-based genetic algorithm using an elite strategy.

The E-VSP with battery degradation has been relatively overlooked in the existing literature. To the best of our knowledge, it has only been addressed by Zhang et al. (2021) and Zhou et al. (2022). In their work, Zhang et al. (2021) proposed a set partitioning model for the E-VSP considering a single-terminal transit network and nonlinear charging profile. Each trip in this single-terminal transit network is a loop starting and ending at the terminal. They defined the total cost of a vehicle schedule as the sum of the charging fees, the cost incurred by battery degradation, and a fixed cost for vehicle acquisition. The capacity fading rate model of Lam and Bauer (2012) is employed to estimate the battery degradation cost of a schedule. An exact branch-and-price algorithm is devised, and a dual stabilization technique accelerates their algorithm. Their model does not accommodate partial charging, a limitation that Zhou et al. (2022) overcomes in their work. They addressed the electric bus charging scheduling problem (EB-CSP), which involves assigning BEBs to trips and scheduling the charge periods for a single-terminal transit network with roundtrips. A mixed-integer nonlinear nonconvex program and a mixed-integer linear programming (MILP) approximation are introduced. The MILP uses linear functions to approximate the charging and battery degradation functions. Zeng et al. (2022) also proposed a model for the EB-CSP (under predetermined bus-to-trip assignments) with battery degradation using peak-to-average power ratio, time-of-use electricity price, and battery wear cost.

6.1.1. Contributions

Our analysis of the literature on the E-VSP shows that E-VSP with battery degradation has been little studied despite the significant impact that taking the SoC range into account can have on battery life. Additionally, existing literature on the E-VSP with battery degradation focuses solely on deterministic approaches, which may underestimate the probability

of encountering low SoC, high SoC, or large SoC deviation. Indeed, energy consumption varies from day to day due to factors like weather conditions and traffic congestion. Since average SoC and SoC deviation greatly influence battery aging, a comprehensive and representative E-VSP model considering battery degradation should incorporate information about energy stochasticity.

This work aims at filling this gap by presenting the first model for the E-VSP with battery degradation and stochastic energy consumption. To ensure a realistic system representation, we also consider other important E-VSP constraints, including partial en-route charging, nonlinear charging profile, and charging stations with limited capacity. Our proposed model introduces a chance-constraint approach for the E-VSP with battery degradation, limiting the probability of overusing the bus fleet's batteries. Avoiding excessive battery usage can minimize SoC variations throughout the day, reducing battery degradation. The average SoC is controlled by a parameter that defines the maximum allowable accumulated energy consumption or, equivalently, the maximum SoC. To solve the introduced chance-constraint model, we present a tailored branch-and-price algorithm that involves stochastic pricing problems and an exact stochastic dominance rule. We test our approach on randomly generated single-depot instances considering practical settings. Nevertheless, the model and algorithm we present can be readily adapted to address the multi-depot case.

The remainder of this paper is organized as follows. Section 6.2 presents a general definition of the E-VSP and introduces our chance-constrained model for the E-VSP with battery degradation and stochastic energy consumption. Furthermore, a method to decompose our chance-constraint by vehicle schedule is provided. Section 6.3 presents our heuristic branch-and-price algorithm. Section 6.4.1 details a simulation method to compute the capacity fading of a solution to our stochastic problem. We compare in Section 6.5 our approach to a deterministic baseline and analyze the tradeoff between operational costs and battery degradation for both approaches. Section 6.6 summarizes our results.

6.2. The E-VSP with battery degradation and stochastic energy consumption

We first define the E-VSP and present the non-linear charging function approximation used in Sections 6.2.1 and 6.2.3, respectively. The second part focuses on the stochastic case. In Section 6.2.4, we introduce the chance-constrained model for the E-VSP with battery degradation and stochastic energy consumption. A method to compute the probability of overuse of bus batteries is provided in Section 6.2.5.

6.2.1. General definition

Let $\mathcal{V}$, $\mathcal{D}$, and $\mathcal{H}$ be a timetable of trips, a set of depots, and a set of charging stations, respectively. For each trip in $\mathcal{V}$, a departure time and location, an arrival time and location, and some information about its energy consumption are given. Furthermore, the travel time and the energy consumption between any two locations in the bus network (starting or ending location of a trip in $\mathcal{V}$, a depot in $\mathcal{D}$, or a charging station in $\mathcal{H}$) is known. Given this information, the E-VSP consists in finding a set of feasible vehicle schedules $\mathcal{S}^*$ that covers exactly once each timetabled trip while respecting the capacity b_d (number of available buses) of each depot $d \in \mathcal{D}$ and the capacity g_h (number of available chargers) of each charging station $h \in \mathcal{H}$. In short, each vehicle schedule is an ordered feasible sequence of timetabled trips and charging activities starting and ending at the same depot in $\mathcal{D}$. We provide a more precise definition later in this section after discussing the charging process and the underlying graphical representation.

We model the capacity of each charging station with three additional sets. Let $\mathcal{R}$ be a set of $k+1$ time intervals, each of duration ρ. Let also $\mathcal{H}^C$ and $\mathcal{H}^W$ be two sets of nodes, where each node is associated with a charging station in $\mathcal{H}$ in the charging or the waiting state, respectively, and a time interval in $\mathcal{R}$. For example, we denote by $h_1^C r_5$ and $h_1^W r_5$ the nodes associated with charging station $h_1 \in \mathcal{H}$ and time interval $r_5 \in \mathcal{R}$ in $\mathcal{H}^C$ and $\mathcal{H}^W$, respectively.

Our E-VSP model is defined on connection-based networks (Ribeiro and Soumis, 1994) with time-expanded charging nodes, denoted $G^d(V_d, A_d)$, for $d \in \mathcal{D}$. The node set of G^d is $V_d = \mathcal{V} \cup \{n_0^d, n_1^d\} \cup \mathcal{H}^C \cup \mathcal{H}^W$, where n_0^d and n_1^d represent depot d at the beginning and the end of the day, respectively, and its arc set is A_d. This network contains three types of arcs (i,j), namely *pull-out* and *pull-in* arcs, (n_0^d, i) and (i, n_1^d), $\forall i \in \mathcal{V}$, respectively, *connection* arcs (i,j), for $i,j \in \mathcal{V}$, and *charging* arcs. Figure 6.2 illustrates an example of such network with 6 trips v_1 to v_6, 1 charging station h_1, and 7 time intervals, r_0 to r_6.

Two timetabled trips i and j are connected by an arc (i,j) in G^d, $d \in \mathcal{D}$, only if the departure time of trip j is larger than or equal to the arrival time of trip i plus the deadhead travel time between i and j and a minimum layover time of τ minutes to board passengers at the beginning of the trip. To minimize congestion at the terminals and reduce idle time for drivers, we implement a waiting time threshold of 45 minutes between consecutive trips i and $j \in \mathcal{V}$. If the waiting time exceeds 45 minutes, the vehicle must proceed to the nearest depot after completing trip i. At the depot, it will wait before moving to the starting location of trip j. Charging arcs take many forms. First, for each charging station $h \in \mathcal{H}$, all consecutive nodes in $\mathcal{H}^C$ and $\mathcal{H}^W$ associated with h are connected, such that two disjoint chains are formed per charging station. The nodes in $\mathcal{H}^W$ associated with the latest time interval are connected to n_1^d. Second, we connect each node in $\mathcal{H}^W$, except the ones associated with the

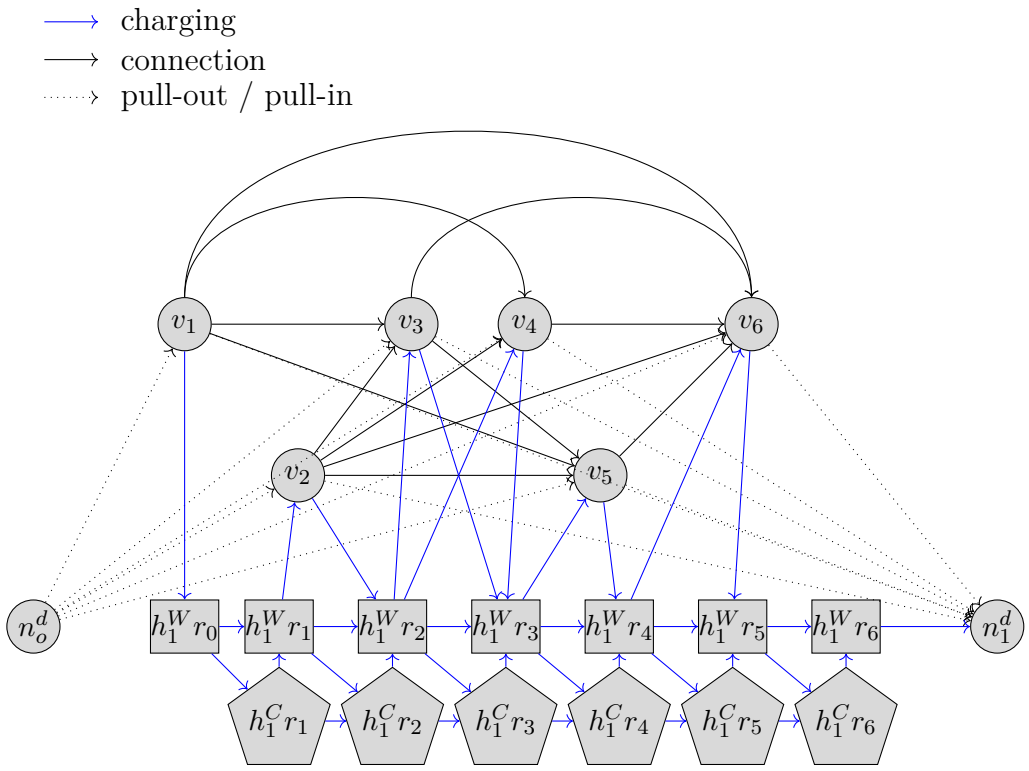


Figure 6.2 – Connection-based network with time-expanded charging station nodes

k^{th} time interval, to the node in $\mathcal{H}^C$ associated with the same charging station and the next time interval. Third, an arc connects each node in $\mathcal{H}^C$ to the node in $\mathcal{H}^W$ that belongs to the same time interval and charging station. These first three types of charging arcs form the lower block of Figure 6.2. This block is then connected to the timetabled trips as follows. We create an arc between a trip in $\mathcal{V}$ and a node in $\mathcal{H}^W$, which we call a "waiting node", if the earliest bus arrival time at the charging station following this trip falls within the time interval associated with the waiting node. Similarly, we add an arc between a waiting node and a trip if the latest departure time of a bus from the charging station, ensuring the bus arrives precisely on time for the trip's starting time, falls within the time interval associated with the previous waiting node.

A path in the graph G^d starting at the source node n_0^d and ending at the sink node n_1^d is a feasible vehicle schedule if it respects energy-feasibility and charging-feasibility constraints. Let a bus's accumulated energy consumption (AEC) be the sum of the energy used or gained (when charging) over a given time. We assume that each vehicle begins the day with an initial AEC of $\sigma^{init} = 0$ (i.e., fully charged). The energy-feasibility constraints ensure that the AEC of a bus remains below the maximum allowed AEC, denoted as σ^{max}, throughout the day. We detail in Section 6.2.2 the relationship between the AEC and the SoC of a battery. Energy consumption values, AEC values, and σ^{max} are expressed as a percentage of the total battery capacity. All values are rounded to the nearest integer, so admissible

values are within $\{0\%, 1\%, \ldots, 100\%\}$. Moreover, charging-feasibility constraints ensure that every time a bus visits a charging station in $\mathcal{H}$, exactly one recharging activity (composed of consecutive nodes in $\mathcal{H}^C$) is planned.

The cost of a vehicle schedule s passing through the set of arcs $A(s)$ is given by

$$c_s = \sum_{(i,j)\in A(s)} c_{ij}, \tag{6.2.1}$$

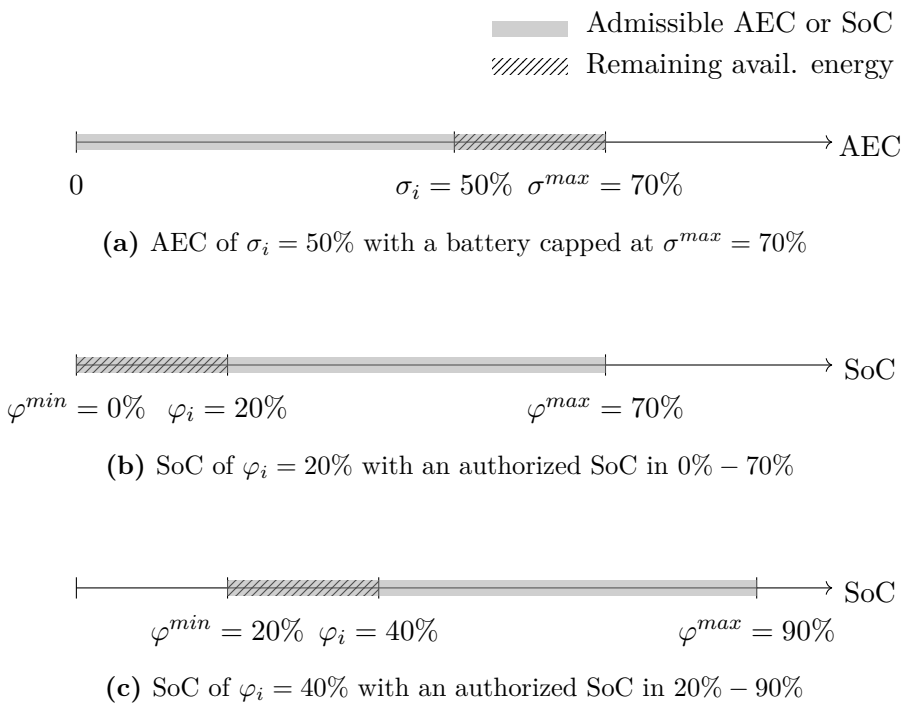
where c_{ij} is the cost of arc (i,j), consisting of a cost per vehicle used, a cost for each minute waiting outside the depot, a cost per unit of distance traveled, and a cost per charging activity. This latter penalty for each charging activity is used to, on the one hand, discourage unnecessary visits to the charging stations when the AEC is still low and, on the other hand, take into consideration operational costs, such as the salary of employees dedicated to each charging station.

6.2.2. The AEC compared with the SoC

Let σ_i and φ_i be the AEC and the SoC after node $i \in s$, $s \in \mathcal{S}$, respectively. For a given σ_i and a maximum allowed AEC of σ^{max}, there are several equivalent φ_i values. We provide an example of a case where an AEC level can be associated with two perfectly equivalent SoC levels in Figure 6.3. Figure 6.3a illustrates the admissible AEC range of a battery for $\sigma^{max} = 70\%$. In Figure 6.3b, higher SoC values are prohibited by setting the maximum allowed SoC to $\varphi^{max} = 70\%$ and the minimum allowed SoC to $\varphi^{min} = 0\%$. In Figure 6.3c, $\varphi^{max} = 90\%$ and $\varphi^{min} = 20\%$. Note that all cases are equivalent since the difference between the battery's total capacity and the capacity that can be used is the same, i.e., $100\% - \sigma^{max} = 30\%$. The remaining available energy equals 20% in all three figures. However, the AEC of σ_i is equivalent to $\varphi_i = 20\%$ in Figure 6.3b and to $\varphi_i = 40\%$ in Figure 6.3c.

Under the hypothesis that the wear cost of a schedule is increasing with its capacity fading and that the capacity fading function of Lam and Bauer (2012) correctly represents the capacity fading rate of a battery, we can show that the optimal battery management policy with respect to the wear cost and the operational costs is always to prohibit SoC greater than or equal to σ^{max}, rather than prohibiting smaller SoC values. Indeed, the capacity fading function of Lam and Bauer (2012) indicates that battery capacity degradation increases with average SoC. Therefore, given a maximum allowed AEC of σ^{max}, it is more favorable to limit high SoC values instead of low SoC values. The average SoC is lowered by truncating high SoC values, reducing capacity fade.

In what follows, we use the AEC to represent energy level. Note that, for this work, optimizing vehicle schedules by taking AEC levels into account will always produce better or equivalent solutions than if we took SoC levels into account.



(a) AEC of $\sigma_i = 50\%$ with a battery capped at $\sigma^{max} = 70\%$

(b) SoC of $\varphi_i = 20\%$ with an authorized SoC in $0\% - 70\%$

(c) SoC of $\varphi_i = 40\%$ with an authorized SoC in $20\% - 90\%$

Figure 6.3 – Example of the comparison between AEC and SoC. The SoC levels in (b) and (c) are equivalent to the AEC in (a)

6.2.3. Non-linear charging profile

The potential risks associated with excessive voltage levels that could permanently damage the battery can be mitigated by charging the battery using the constant current - constant voltage (CC-CV) scheme (Pelletier et al., 2017). In this work, we adopt the approach introduced by Montoya et al. (2017), which involves approximating the non-linear charging function of the CC-CV scheme with a piecewise linear function.

After charging t minutes, we approximate the AEC by a piecewise linear function $\lambda(z,t)$, where z represents the initial AEC. In our approach, a bus is recharged for ρ minutes at each charging node. Suppose a path in G^d contains several consecutive charging nodes, the function $\lambda(z,\rho)$ is then applied a corresponding number of times equivalent to the length of the charging node sequence. With each application, the initial AEC z is updated. The output of $\lambda(z,t)$ is capped at 0, ensuring that the presence of additional charging nodes when the battery is fully charged does not create any issues.

6.2.4. Chance-constrained E-VSP with battery degradation

From a battery aging perspective, it is advantageous to decrease both the average SoC and the deviation in SoC throughout the day (Lam and Bauer, 2012). Moreover, energy consumption varies daily for the same bus trips, which can impact the amplitude of the

SoC deviation and energy feasibility. Therefore, we aim to consider the energy consumption uncertainty and these ideal SoC profiles in the E-VSP to reduce battery wear and ensure energy feasibility for virtually all scenarios.

A deterministic approach to controlling the average SoC and the deviation in SoC throughout the day is to limit the maximum allowed AEC by decreasing σ^{max}. This approach is equivalent to reducing the battery's available capacity and effectively reduces the SoC deviation. Furthermore, considering that the capacity reduction is applied to the highest SoC values, this approach also decreases the average SoC.

Now, consider that the probability mass function (PMF) with finite support of the energy consumption of every trip $i \in \mathcal{V}$, denoted $e_i(\mu)$, is known. Then, an alternative approach is to limit the probability of exceeding a recommended upper limit on the AEC in addition to the strict bound on the maximum allowed AEC, denoted as σ^{rec}. As for σ^{max}, σ^{rec} is expressed as a percentage of the total battery capacity and rounded to the nearest integer. We consider deterministic energy consumption for pull-in, pull-out, and deadhead trips because these trips are usually short and do not involve passengers, eliminating several sources of uncertainty that affect energy usage. For instance, factors such as passenger loading and heating or air conditioning requirements within the bus are not present during these trips. To easily refer to this approach later, we will say that a battery is overused when the AEC of a bus exceeds σ^{rec}. We formulate the limit on the probability of overuse of a battery as a chance-constraint.

Our model uses the following additional notation. Let $\mathcal{S}$ be the set of all feasible vehicle schedules and $\mathcal{S}^d \subset \mathcal{S}$ be the subset of feasible vehicle schedules for depot $d \in \mathcal{D}$. Let also a_{is} be a binary parameter equal to 1 if and only if schedule $s \in \mathcal{S}$ covers trip i, y_s be a binary variable equal to 1 if and only if schedule s is part of the solution, and w_s^{hr} be a binary parameter equal to 1 if and only if schedule s includes a charging activity at station h in time interval r. The E-VSP with battery degradation and stochastic energy consumption can then be formulated as the following chance-constrained integer program:

$$\min \quad \sum_{s\in\mathcal{S}} c_s y_s \tag{6.2.2}$$

$$\text{s.t.} \quad \sum_{s\in\mathcal{S}} a_{is} y_s = 1, \qquad \forall i \in \mathcal{V} \tag{6.2.3}$$

$$\sum_{s\in\mathcal{S}^d} y_s \leq b_d, \qquad \forall d \in \mathcal{D} \tag{6.2.4}$$

$$\sum_{d\in\mathcal{D}} \sum_{s\in\mathcal{S}^d} w_s^{rh} y_s \leq g_h, \qquad \forall h \in \mathcal{H}, r \in \mathcal{R} \tag{6.2.5}$$

$$\Pr\{\text{overuse of the battery of one bus or more}\} \leq \epsilon \tag{6.2.6}$$

$$y_s \in \{0,1\}, \qquad \forall s \in \mathcal{S}. \tag{6.2.7}$$

The objective function (6.2.2) minimizes the total operational costs, while constraints (6.2.3) ensure that each timetabled trip is covered exactly once by a schedule, constraints (6.2.4) ensure that vehicle availability is respected at each depot, and constraints (6.2.5) guarantee that for each charging station $h \in \mathcal{H}$, a maximum of g_h chargers are used at the same time. Finally, the chance-constraint (6.2.6) ensures that the probability that at least one bus battery is overused during the day is less than or equal to a maximum threshold, ϵ.

We assume that the probability of overuse of the battery of any two schedules in $\mathcal{S}$ are independent. Let the probability $P_s = Pr\{\text{not overuse the battery of bus assigned to schedule } s\}$. Using this notation and the previous independence assumption, we can rewrite constraint (6.2.6) as

$$1 - \prod_{s \in \mathcal{S}: y_s = 1} P_s \leq \epsilon, \tag{6.2.8}$$

which, by the properties of logs and extending the summation to all vehicle schedules, is equivalent to

$$\sum_{s \in \mathcal{S}} y_s \ln(P_s) \geq \ln(1 - \epsilon). \tag{6.2.9}$$

Let $\beta_s = \ln(P_s)$ and $\beta = \ln(1 - \epsilon)$. Then, we have

$$\sum_{s \in \mathcal{S}} y_s \beta_s \geq \beta. \tag{6.2.10}$$

In what follows, we provide a method to compute P_s for any given vehicle schedule $s \in \mathcal{S}$. If we allow a bus to use more than σ^{rec} percent of the battery capacity, we strictly enforce that no more than σ^{max} percent be used. This guarantees the energy feasibility of the system.

6.2.5. Computing probabilities of overuse of bus batteries

Consider a schedule $s = (0,1,\ldots,n,n+1)$ where 0 and $n+1$ are the nodes n_0^d and n_1^d, respectively, associated with a depot $d \in \mathcal{D}$ and $1,\ldots,n$ are other nodes in V_d. Let X_i^s be the AEC at the end of node i in schedule s and $f_i^s(x)$ be the truncated PMF associated with event $X_i^s = x$ and the event that the bus associated with schedule s has not overused its battery up until node $i-1$. Furthermore, we define $\bar{f}_i^s(x)$ as

$$\bar{f}_i^s(x) = \begin{cases} f_i^s(x) & x \leq \sigma^{rec} \\ 0 & \text{otherwise.} \end{cases} \tag{6.2.11}$$

Note that, by definition, $\sum_{x=0}^{\sigma^{rec}} \bar{f}_i^s(x) \leq 1$. We can compute $f_i^s(x)$ recursively with $f_0^s(0) = 1$ as

$$f_i^s(x) = \begin{cases} \sum_{\mu=0}^{\sigma^{max}} e_i(\mu) \bar{f}_{i-1}^s(x - \mu - \iota_{i-1,i}) & \text{if } i \in \mathcal{V} \\ \bar{f}_{i-1}^s(\lambda^{-1}(x,\rho)) & \text{if } i \in \mathcal{H}^C \\ \bar{f}_{i-1}^s(x - \iota_{i-1,i}) & \text{otherwise,} \end{cases} \tag{6.2.12}$$

where $\lambda^{-1}(z,t)$ is the inverse of the charging function and outputs the initial AEC such that the final AEC after a charge of t minutes is z. Also, $\iota_{i-1,i}$ is the deterministic energy consumption between nodes $i-1$ and i. Observe that if $1 - \sum_{\mu=0}^{\sigma^{max}} e_i(\mu) > 0$ (i.e., the probability that the energy consumption of trip i is greater than σ^{max} is positive), the E-VSP is infeasible. Therefore, we do not consider this case. The probability of overuse of the battery of the bus assigned to s is thus

$$P_s = \sum_{x=0}^{\sigma^{rec}} \bar{f}_{n+1}^s(x) = \bar{F}_{n+1}^s(\sigma^{rec}), \tag{6.2.13}$$

where $\bar{F}_i^s(x)$ is the cumulative distribution function (CDF) of $\bar{f}_i^s(x)$, computed as

$$\bar{F}_i^s(x) = \sum_{x'=0}^{x} \bar{f}_i^s(x'), \qquad x = 0, 1, \ldots, \sigma^{rec}. \tag{6.2.14}$$

6.3. Heuristic branch-and-price algorithm

Variants of the VSP formulated as a set partitioning problem, including the E-VSP, are readily solved by the branch-and-bound algorithm, given their tight lower bound. However, the number of possible variables, i.e., bus schedules, in such formulation is typically very large. It is, therefore, not relevant nor efficient to enumerate all possible bus schedules. Instead, a column generation algorithm (see, e.g., Desaulniers et al., 2005; Lübbecke and Desrosiers, 2005) can be embedded in the branch-and-bound tree to generate variables as needed. The resulting approach is called branch-and-price (Barnhart et al., 1998; Costa et al., 2019).

One of the most challenging components of the branch-and-price algorithm for problems like the E-VSP is the pricing problem, i.e., the generation of columns of minimum reduced cost. In the case of the E-VSP, the pricing problem is separable by depot and consists in solving one shortest path problem in G^d, for all $d \in \mathcal{D}$. This problem is complexified in our case since we consider stochastic energy consumption. Consequently, the pricing problems transform from shortest path problems with resource constraints (SPPRC), as described in Irnich and Desaulniers (2005), to SPPRC with stochasticity (Boland et al., 2015; Wellman et al., 2013). The label components, extension functions, and dominance rule of the stochastic dynamic programming algorithm used to solve the pricing problems are detailed in Section 6.3.1. Then, we explain in Sections 6.3.2 and 6.3.3 the branching strategy and the perturbation method adopted to accelerate computing time, respectively.

6.3.1. Stochastic pricing problems

Column generation is used to solve the linear relaxation of (6.2.2)-(6.2.7), which is called the master problem (MP). At each iteration of the column generation algorithm, the MP restricted to a small subset of its variables is solved to yield a primal and a dual solution. Then, the pricing problems come into play to identify potentially useful columns to add to the restricted MP. These pricing problems are defined on the networks G^d, for $d \in \mathcal{D}$, where the arc costs have been modified to incorporate dual information obtained from the most recent solution of the restricted MP. Let $(u_i)_{i\in\mathcal{V}}$, $(\pi_d)_{d\in\mathcal{D}}$, $(\alpha^{hr})_{h\in\mathcal{H},r\in\mathcal{R}}$, and θ be the dual variables associated with constraints (6.2.3), (6.2.4), (6.2.5), and (6.2.6), respectively. The reduced cost $\tilde{c}_s$ of schedule s is

$$\tilde{c}_s = c_s - \sum_{i\in\mathcal{V}} a_{is}u_i - \pi_d - \sum_{h\in\mathcal{H}}\sum_{r\in\mathcal{R}} w_s^{hr}\alpha^{hr} - \theta \ln \bar{F}_{n+1}^s(\sigma^{rec}). \tag{6.3.1}$$

As previously mentioned, schedules can be seen as paths in a network G^d, $d \in \mathcal{D}$, i.e., as sequences of arcs. Each pricing problem involves finding a path in G^d, $d \in \mathcal{D}$, with minimal reduced cost. The pricing problems verify if the solution to the restricted MP is optimal by searching for vehicle schedules with negative reduced costs. If negative reduced cost paths are found, all or some of these columns are added to the restricted MP. Otherwise, the solution to the restricted MP is optimal for the MP.

Paths in G^d, $d \in \mathcal{D}$, are constructed by dynamic programming, specifically by a labeling algorithm (Irnich and Desaulniers, 2005). The general framework for such an algorithm is as follows. Starting from an initial label at the source node n_0^d, labels are extended throughout the network using resource extension functions to create partial paths (schedules) until reaching the sink node where the paths correspond to complete schedules. A label represents a partial path starting from the source node, including all necessary information to assess the feasibility of any extension and its reduced cost. Infeasible paths are discarded when they are identified as such. Furthermore, a dominance procedure is used to compare the labels and eliminate non-Pareto-optimal labels to avoid enumerating all feasible paths.

The main components of the labeling algorithm used in our tailored branch-and-price algorithm, namely the label components, the extension functions, and the dominance rule, are detailed next.

6.3.1.1. Labeling procedure. In the E-VSP, a path p in G^d, $d \in \mathcal{D}$, with resident node i is usually represented by its accumulated reduced cost C_i^p and its expected AEC (or, equivalently, its expected SoC). However, our model has three specificities: (i) the energy consumption is stochastic, (ii) it involves a chance-constraint, and (iii) the capacity of each

charging station is limited. Therefore, it is necessary to include more information in the labels to describe the paths, as detailed next fully.

First, instead of keeping track of the expected AEC, we will store the worst case AEC, that we denote ζ_i^p for path p with resident node i. Second, to compute the probability of overuse of a bus battery associated with a path p ending at node i and its extensions, it is necessary to store $\bar{F}_i^p(x)$ for integer values of x between 0 and σ^{rec}. Simply relying on the CDF of the AEC is insufficient for this purpose. The probability that the AEC of a bus at a given node is above σ^{rec} indicates the probability that the battery is overused at this specific point along the bus route, but it lacks memory. By definition, $\bar{F}_i^p(\cdot)$ incorporates this memory by capturing the probability of overuse of the bus battery for the entire path. One must therefore store this latter CDF in a path's label. Third, we add two resources to ensure realistic charging activity plans. Specifically, let E_i^p be the number of charging activities completed since the last timetabled trip in p and R_i^p be the number of times a bus has waited at a recharging station without being recharged since the last timetabled trip in p. The first resource ensures that at most one recharge occurs when visiting a charging station, and the second ensures that a recharge occurs when visiting a charging station. Both resources are initialized at a value of 0.

Given a label $L_i^{p'} = (\bar{F}_i^{p'}(0), \ldots, \bar{F}_i^{p'}(\sigma^{rec}), \zeta_i^{p'}, C_i^{p'}, R_i^{p'}, E_i^{p'})$ representing a partial path p' ending at node i, we can extend path p' along an arc $(i,j) \in A_d$ to create label $L_j^p = (\bar{F}_j^p(0), \ldots, \bar{F}_j^p(\sigma^{rec}), \zeta_j^p, C_j^p, R_j^p, E_j^p)$ by applying the following method.

The extension function to compute $\bar{F}_j^p(\cdot)$ based on $\bar{F}_i^{p'}(\cdot)$ can easily be retrieved using equations (6.2.11), (6.2.12), and (6.2.14). It is given by

$$\bar{F}_j^p(x) = \begin{cases} \sum_{\mu=0}^{\sigma^{max}} e_j(\mu) \bar{F}_i^{p'}(\min\{x, \sigma^{rec}\} - \mu - \iota_{i,j}) & \text{if } j \in \mathcal{V} \\ \bar{F}_i^{p'}(\lambda^{-1}(x,\rho)) & \text{if } j \in \mathcal{H}^C \\ \bar{F}_i^{p'}(\min\{x, \sigma^{rec}\} - \iota_{i,j}) & \text{otherwise.} \end{cases} \tag{6.3.2}$$

Furthermore, ζ_j^p can be computed using

$$\zeta_j^p = \begin{cases} \zeta_i^{p'} + \max\{\mu \text{ st. } e_j(\mu) > 0\} + \iota_{i,j} & \text{if } j \in \mathcal{V} \\ \lambda(\zeta_i^{p'}, \rho) & \text{if } j \in \mathcal{H}^C \\ \zeta_i^{p'} + \iota_{i,j} & \text{otherwise,} \end{cases} \tag{6.3.3}$$

where $\max\{\mu \text{ st. } e_j(\mu) > 0\}$ outputs the largest energy consumption value with positive probability. The accumulated reduced cost C_j^p is updated as

$$C_j^p = C_i^{p'} + \tilde{c}_{ij}^p, \tag{6.3.4}$$

where the reduced cost breakdown per arc is

$$\tilde{c}_{ij}^{s} = \begin{cases} c_{ij} - \pi_d & \text{if } i = n_0^d \\ c_{ij} - u_i & \text{if } i \in \mathcal{V} \text{ and } j \neq n_1^d \\ c_{ij} - u_i - \theta \ \ln \bar{F}_{n+1}^{s}(\sigma^{rec}) & \text{if } i \in \mathcal{V} \text{ and } j = n_1^d \\ c_{ij} - \alpha^{hr} & \text{if } i = h^C r \in \mathcal{H}^C \\ c_{ij} - \theta \ \ln \bar{F}_{n+1}^{s}(\sigma^{rec}) & \text{if } i = h^W r \in \mathcal{H}^W \text{ and } j = n_1^d \\ c_{ij} & \text{otherwise.} \end{cases} \tag{6.3.5}$$

Observe that the part of the reduced cost attributable to the chance-constraint (6.2.6), i.e., $\theta \ln \bar{F}_{n+1}^{s}(\sigma^{rec})$, is not decomposable by arc. Indeed, the probability of overuse of the bus battery associated with schedule s is only revealed when the bus schedule is fully formed. Hence, we add this cost to the last arc of the schedule, the pull-in arc.

Lastly, the resources $R_i^{p'}$ and $E_i^{p'}$ are extended as

$$R_j^p = \begin{cases} R_i^{p'} + 1 & \text{if } i = h^W r \in \mathcal{H}^W \text{ and } j = h^C r \in \mathcal{H}^C \\ R_i^{p'} - 1 & \text{if } i = h^W r \in \mathcal{H}^W \text{ and } j \in \mathcal{V} \\ R_i^{p'} & \text{otherwise,} \end{cases} \tag{6.3.6}$$

and

$$E_j^p = \begin{cases} E_i^{p'} + 1 & \text{if } i \in \mathcal{V} \text{ and } j = h^W r \in \mathcal{H}^W \\ E_i^{p'} - 1 & \text{if } i = h^W r \in \mathcal{H}^W \text{ and } j \in \mathcal{H}^C \\ E_i^{p'} & \text{otherwise.} \end{cases} \tag{6.3.7}$$

A path p ending at node i is discarded to ensure energy feasibility if it has a positive probability of running out of energy, i.e., $\zeta_i^p > \sigma^{max}$. Moreover, all paths such that the chance-constraint (6.2.6) would be automatically violated if we were to include any extension of these paths in the E-VSP solution, i.e., all paths for which $\ln(\bar{F}_i^p(\sigma^{rec})) < \ln(1-\epsilon)$, are excluded. This latter rule helps speed up the algorithm by removing the paths we know will not be part of the solution. Moreover, a path p ending at node i is discarded when the resources E_i^p and R_i^p exceed their respective resource windows. These windows require E_i^p and R_i^p to be less than or equal to 0 if $i \in \mathcal{V}$ and less than or equal to 1 otherwise.

6.3.1.2. Label dominance. Consider two feasible paths in G^d, p_1 and p_2, both ending at node $i \in V_d$. For pricing problems in the family of the SPPRC, we say that path p_1 dominates path p_2 when:

(a) Any feasible extension e of p_2 ending at node $j \in V_d$ is also feasible for p_1;

(b) For any feasible extension e of p_1 and p_2 ending at node $j \in V_d$, the inequality $C_j^{p_1 \oplus e} \leq C_j^{p_2 \oplus e}$ holds, where $p_i \oplus e$, $i = 1, 2$, denotes a path resulting from appending a feasible extension e to path p_i.

In practice, these two conditions are hard to use directly to identify dominated labels. So, next, we derive a stricter dominance rule which ensures that the conditions (a) and (b) are met.

Consider two paths p^1 and p^2, both ending at node i. Path p^1 dominates path p^2 when the following conditions hold:

(i) $C_i^{p_1} \leq C_i^{p_2}$

(ii) $R_i^{p_1} \leq R_i^{p_2}$

(iii) $E_i^{p_1} \leq E_i^{p_2}$

(iv) $\zeta_i^{p_1} \leq \zeta_i^{p_2}$

(v) $\bar{F}_i^{p_1}(x) \geq \bar{F}_i^{p_2}(x)$, for all $x \in \{0, 1, \ldots, \sigma^{rec}\}$

These conditions are valid since all the resource extension functions, namely (6.3.2), (6.3.3), (6.3.4, (6.3.6), and (6.3.7), are non decreasing for all types of arcs, except the extension of the accumulated reduced cost when the arc (i,j) is such that $j = n_1^d$. Indeed, the function $\min\{x, \sigma^{rec}\}$, CDFs, and the logarithmic function are non-decreasing by definition. However, the logarithmic function is multiplied by -1 in the third and fourth cases of the equation (6.3.5). Since we favor smaller accumulated reduced costs, we thus also favor larger probabilities of not overusing BEBs' batteries (given by $\bar{F}_{n+1}^s(\sigma^{rec})$, for $s \in \mathcal{S}$). Larger probabilities are obtained with condition *(v)*.

6.3.2. Branching strategy

To find integer solutions in a reasonable amount of time, we employ a heuristic branching strategy, namely a diving strategy. On the one hand, we branch on the total number of vehicles used. A single branch with a lower bound constraint on the minimum number of vehicles used is created. On the other hand, we apply two rounding strategies - one for schedule variables and another for connection arcs (i.e., arcs between two timetabled trips). When either of these rounding strategies is chosen, we create a single node with one or several schedule variables or connection arcs fixed to 1. Fixing an arc between trips i and j to 1 is done by removing all arcs (i,k) and $(k,j) \in A_d$, $d \in \mathcal{D}$, such that $k \in V_d \setminus \{i,j\}$. The variables or arcs with the largest fractional values are rounded up. A maximum of three variables or arcs with a fractional part greater than or equal to 0.99 are selected at each branch-and-bound node. If fewer than 3 schedule variables or arcs are above this threshold, all variables or arcs with a fractional part greater than 0.99 are selected. At worst, the variable or arc with the largest fractional part (less than 0.99) is selected. Note that the

minimum and maximum number of selected variables and the fractional part threshold have been determined through empirical testing in a preliminary experimental campaign.

We prioritize the first technique, which involves branching on the total number of vehicles used. Subsequently, we alternate between the two rounding strategies. At each node, we choose the strategy with the highest score, where the score is determined by the selected variable(s) or arc(s) with the smallest fractional part.

6.3.3. Constraint perturbation

MDVSPs formulated as a set partitioning problem are typically degenerate, and this degeneracy generally increases with the time horizon considered or, equivalently, the average number of trips per schedule (see Oukil et al., 2007; Benchimol et al., 2012). We use the simple constraint perturbation strategy of Charnes (1952) to reduce the degeneracy in our experiments. Let η_i^+ and η_i^- be perturbation variables that allow a under- and over-covering of trip $i \in \mathcal{V}$ of up to ξ_i^+ and ξ_i^-, respectively. The perturbed MP is given by:

$$\min \quad \sum_{s\in\mathcal{S}} c_s y_s + \sum_{i\in\mathcal{V}} (\delta_i^+ \eta_i^+ + \delta_i^- \eta_i^-) \tag{6.3.8}$$

$$\text{s.t.} \quad \sum_{s\in\mathcal{S}} a_{is} y_s + \eta_i^+ - \eta_i^- = 1, \qquad \forall i \in \mathcal{V} \tag{6.3.9}$$

$$\sum_{s\in\mathcal{S}^d} y_s \leq b_d, \qquad \forall d \in \mathcal{D} \tag{6.3.10}$$

$$\sum_{d\in\mathcal{D}} \sum_{s\in\mathcal{S}^d} w_s^{rh} y_s \leq g_h, \qquad \forall h \in \mathcal{H}, r \in \mathcal{R} \tag{6.3.11}$$

$$\sum_{s\in\mathcal{S}} y_s \beta_s \geq \beta \tag{6.3.12}$$

$$0 \leq \eta_i^+ \leq \xi_i^+, \qquad \forall i \in \mathcal{V} \tag{6.3.13}$$

$$0 \leq \eta_i^- \leq \xi_i^-, \qquad \forall i \in \mathcal{V} \tag{6.3.14}$$

$$0 \leq y_s \leq 1, \qquad \forall s \in \mathcal{S}, \tag{6.3.15}$$

where δ_i^+ and δ_i^- are the penalties for under- and over-covering trip $i \in \mathcal{V}$, respectively.

6.4. Computing the battery degradation level of a bus schedule

We evaluate the degradation level of a BEB associated with a schedule $s \in \mathcal{S}^*$ by analyzing the capacity fading encountered during each discharging and charging cycle $\psi_t \in s$. A discharging and charging cycle refers to utilizing and replenishing electrical energy stored in the vehicle's battery pack. It involves two primary stages: 1) discharging, which occurs when the vehicle is in use and draws power from the battery to

propel itself, and 2) charging, which happens when the battery is connected to an external power source to restore its energy. Specifically, let us consider a vehicle schedule $s = (n_o^d, v_1, h_1^W r_0, h_1^C r_1, h_1^W r_1, v_2, v_3, h_1^W r_3, h_1^C r_4, h_1^W r_4, v_6, n_1^d)$ in the network illustrated in Figure 6.2. This vehicle schedule consists of three discharging and charging cycles: $\psi_1 = (n_o^d, v_1, h_1^W r_0, h_1^C r_1, h_1^W r_1)$, $\psi_2 = (v_2, v_3, h_1^W r_3, h_1^C r_4, h_1^W r_4)$, and $\psi_3 = (v_6, n_1^d)$. Figure 6.4 presents an example of the potential simplified AEC profile for these cycles. Although the actual discharging and charging functions are nonlinear, we represent them as linear functions in time for simplicity.

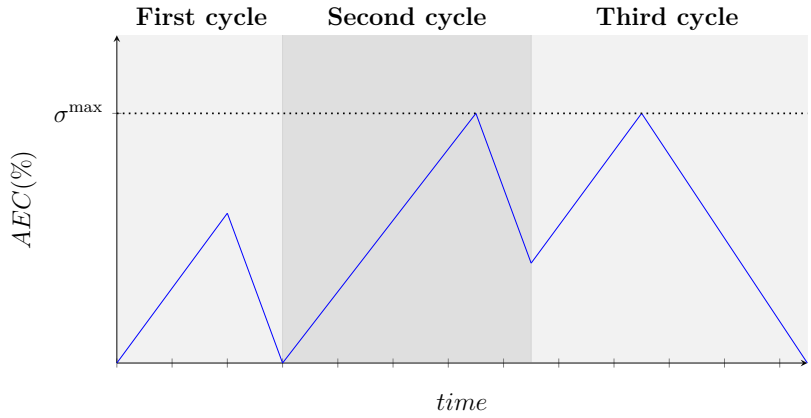


Figure 6.4 – Example of the SoC profile a vehicle schedule with three discharging and charging cycles

In this paper, we use the capacity fading function developed in Lam and Bauer (2012) to approximate the degradation level of a solution to the E-VSP with battery degradation and stochastic energy consumption. This model quantifies the rate at which a battery's capacity degrades, i.e., the loss in kWh of the battery's capacity per 1 kWh of charge processed. The degradation rate depends on both the average SoC and the SoC deviation. However, in real-world scenarios, the discharging and charging cycles differ daily due to the randomness of energy consumption. Consequently, the average SoC and SoC deviation also fluctuate. To address this stochasticity, we need to specify a method for approximating the expected capacity fade of bus batteries. In Section 6.4.1, we present our adaptation of the function of Lam and Bauer (2012) to meet our specific scenario. This adapted function calculates capacity fading for cycles characterized by AEC levels instead of SoC. Additionally, we introduce in Section 6.4.2 a Monte Carlo simulation that approximates the expected capacity fading of a BEB over the entire planning horizon (typically one day).

6.4.1. Battery capacity fading function

Let $\sigma^b_{\psi_t}$, $\sigma^m_{\psi_t}$, and $\sigma^e_{\psi_t}$ be the AEC at the beginning, middle (after the discharging regime), and end of the discharging and charging cycle ψ_t, respectively. The average SoC in cycle ψ_t is defined as

$$\varphi^{avg}_{\psi_t} = \sigma^{max} - \frac{\sigma^b_{\psi_t} + \sigma^m_{\psi_t} + \sigma^e_{\psi_t}}{3}, \tag{6.4.1}$$

and the SoC deviation is defined as

$$\varphi^{dev}_{\psi_t} = \varphi^{avg}_{\psi_t} - (\sigma^{max} - \sigma^m_{\psi_t}). \tag{6.4.2}$$

Note that equations (6.4.1) and (6.4.2) assume for simplicity that the charging and discharging processes are linear, even though the SoC varies non-linearly with time. The capacity fading rate (in kWh/1 kWh processed) can be computed as

$$\phi(\sigma^b_{\psi_t}, \sigma^m_{\psi_t}, \sigma^e_{\psi_t}) = \gamma_1 \varphi^{dev}_{\psi_t} e^{\gamma_2 \varphi^{avg}_{\psi_t}} + \gamma_3 e^{\gamma_4 \varphi^{dev}_{\psi_t}}, \tag{6.4.3}$$

where γ_1 to γ_4 are model parameters borrowed from Lam and Bauer (2012) (see Table 6.1 for the numerical values).

Table 6.1 – Capacity fading rate function parameters

Parameter	Value	Parameter	Value
γ_1	-4.092e-4	γ_3	1.408e-5
γ_2	-2.167	γ_4	6.130

Given a charge processed of $\Sigma_{\psi_t} = (\sigma^b_{\psi_t} - \sigma^m_{\psi_t}) + (\sigma^e_{\psi_t} - \sigma^m_{\psi_t})$ in cycle ψ_t, the daily capacity fade of cycle ψ_t (in kWh) can be computed as

$$\Phi(\sigma^b_{\psi_t}, \sigma^m_{\psi_t}, \sigma^e_{\psi_t}) = \Sigma_{\psi_t} \times \phi(\sigma^b_{\psi_t}, \sigma^m_{\psi_t}, \sigma^e_{\psi_t}). \tag{6.4.4}$$

The daily capacity fade of schedule s is obtained by summing the daily capacity fade of every cycle $\psi_t \in s$ as

$$Q_s = \sum_{\psi_t \in s} \Phi(\sigma^b_{\psi_t}, \sigma^m_{\psi_t}, \sigma^e_{\psi_t}), \tag{6.4.5}$$

where we assume the AEC at the end of the last cycle $\psi_{q_s-1} \in s$ is $\sigma^e_{s_q-1} = \sigma^{init}$, i.e., the BEBs are recharged overnight at the depot.

6.4.2. Monte Carlo simulation

This section proposes a Monte Carlo simulation to approximate the daily capacity fading. After solving the E-VSP with battery degradation and stochastic energy consumption, this simulation is performed offline and getting a solution $\mathcal{S}^*$.

The Monte Carlo simulation is presented in Algorithm 3 and can be summarized as follows. For each vehicle schedule $s \in \mathcal{S}^*$ and in each iteration $k = 1, \ldots, K$, a capacity fading value Q_s^k is computed by iterating through all the nodes in the discharging and charging cycles $\psi_t \in s$ in Steps 7-28. The capacity fading $\phi(\sigma_{\psi_t}^{b,k}, \sigma_{\psi_t}^{m,k}, \sigma_{\psi_t}^{e,k})$ of each cycle $\psi_t \in s$ is computed and added to Q_s^k in Step 26. To do this, random energy consumption values μ_i^k are generated for each timetabled trip $i \in s$ in Step 11. This enables the computation of $\sigma_{\psi_t}^{b,k}$, $\sigma_{\psi_t}^{m,k}$, and $\sigma_{\psi_t}^{e,k}$ for each cycle $\psi_t \in s$. After iterating through all cycles in s, the algorithm adds the capacity fading of schedule s in iteration k, Q_s^k, to the capacity fading of schedule s (Q_s) in Step 29. This process is repeated K times for each vehicle schedule $s \in \mathcal{S}^*$. Finally, in Step 33, the algorithm returns the approximated daily capacity fading per vehicle of the E-VSP solution.

6.5. Experimental results

We evaluate our approach using simulated instances of bus line 95 in the city of Montréal, which consists of 42 stops covering a distance of 8.5 kilometers. The random instances have approximately 60, 150, and 250 trips distributed over 19 hours, ranging from 5 am to midnight. To simplify the analysis, we assume a constant frequency throughout the time horizon, adjusting it according to the instance size. We conduct five tests for each instance size, denoted as I1, I2, and I3, where the start times of the trips are slightly shifted in each of the five tests. Table 6.2 presents the average characteristics of instances I1, I2, and I3, namely the instance family name (Instance), the average number of timetabled trips ($|\mathcal{V}|$), the number of charging stations ($|\mathcal{H}|$), the average number of arcs ($|A|$), the number of depots ($|\mathcal{D}|$), and the number of chargers per charging station (Char. station capacity). Note that while the model and algorithm presented in Sections 6.2 and 6.3, respectively, can potentially be applied to cases where the number of depots or charging stations is larger than or equal to 2, our tests only address the single-depot and single-charging station scenario. This simplified case is already challenging to solve due to stochastic energy consumption and charging station with limited capacity, and it provides sufficient information to conduct a meaningful analysis of the potential advantages of employing a stochastic approach for the E-VSP with battery degradation.

We assume that the energy consumption rate (kWh/km) is normally distributed based on the findings from previous studies (see Bie et al., 2021; Abdelaty et al., 2021). The PDFs of the energy consumption rate of each trip in $\mathcal{V}$ is generated using the following procedure.

Algorithm 3: Monte Carlo simulation to approximate the total daily capacity fade of a solution

$W \leftarrow 0$
for $s \in \mathcal{S}^*$ **do**
  $Q_s \leftarrow 0$
  for $k \leftarrow 1$ **to** K **do**
    $\sigma_{\psi_1}^{b,k} \leftarrow \sigma^{\text{init}}$
    $Q_s^k \leftarrow 0$
    for $\psi_t \in s$ **do**
      $\sigma_{\psi_t}^{m,k} \leftarrow \sigma_{\psi_t}^{b,k}$
      for $i \in \psi_t$ **do**
        if $i \in \mathcal{V}$ **then**
          Randomly generate μ_i^k from $e_i(\mu)$
          $\sigma_{\psi_t}^{m,k} \leftarrow \sigma_{\psi_t}^{m,k} - \mu_i^k - \iota_{i-1,i}$
        end
        if $i = h^W r \in \mathcal{H}^W$ *and* $i - 1 \in \mathcal{V}$ **then**
          $\sigma_{\psi_t}^{m,k} \leftarrow \sigma_{\psi_t}^{m,k} - \iota_{i-1,i}$
          $\sigma_{\psi_t}^{e,k} \leftarrow \sigma_{\psi_t}^{m,k}$
        end
        if $i = h^C r \in \mathcal{H}^C$ **then**
          $\sigma_{\psi_t}^{e,k} \leftarrow \lambda(\sigma_{\psi_t}^{e,k}, \rho)$
        end
        if $i = n_1^d$ **then**
          $\sigma_{\psi_t}^{m,k} \leftarrow \sigma_{\psi_t}^{m,k} - \iota_{i-1,i}$
          $\sigma_{\psi_t}^{e,k} \leftarrow \sigma^{init}$
        end
      end
      $Q_s^k \leftarrow Q_s^k + \phi(\sigma_{\psi_t}^{b,k}, \sigma_{\psi_t}^{m,k}, \sigma_{\psi_t}^{e,k})$
      $\sigma_{\psi_{t+1}}^{b,k} \leftarrow \sigma_{\psi_t}^{e,k}$
    end
    $Q_s \leftarrow Q_s + Q_s^k$
  end
  $W \leftarrow W + Q_s/K$
end
Return $W/|\mathcal{S}^*|$

First, for each timetabled trip, we draw the mean of its energy consumption rate distribution from an exponential distribution with parameters Exp(1.57,0.26), where 1.57 and 0.26 are the location and scale of the distribution, respectively. This distribution has a mean of 1.83 kWh/km. Second, we determine the variance of the normal distribution by sampling a value from a uniform distribution U(0.35,0.5). The location and the scale of the exponential and uniform distributions were calibrated using the data from Basma et al. (2020). The authors

Table 6.2 – Average properties of the families of instances I1 - I3

Instance	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{H}\rvert$	$\lvert A\rvert$	$\lvert\mathcal{D}\rvert$	Char. station capacity
I1	60	1	2,190	1	1
I2	155	1	11,900	1	2
I3	248	1	29,499	1	3

modeled the energy consumption rate associated with propulsion, parking brakes, doors, suspension system, steering pumps, and air compressors. Over the ten simulation cases carried out in Basma et al. (2020) (for a temperature of 20 degrees), the average energy consumption rate varied from 1.74 to 2.77 kWh/km. Furthermore, the authors noticed that heavy traffic and weather conditions could increase the average energy consumption rate of 35% and 200%, respectively.

The following model parameters are used in our tests. The costs per vehicle used, per minute of travel (excluding the travel time of timetabled trips), per minute of waiting outside the depot, and per charging activity are set to 1,000 0.4, 0.2, and 10, respectively. The penalties for under- and over-covering trip $i \in \mathcal{V}$, δ_i^+ and δ_i^-, are set to 1 for every trip $i \in \mathcal{V}$ and the upper bounds ξ_i^+ and ξ_i^- are randomly chosen from the interval $[0, 0.1]$ for every trip $i \in \mathcal{V}$. We consider single deck-12m BEBs equipped with 300 kWh battery packs (the same type of BEBs as those studied in Basma et al. (2020) to ensure consistency) and time intervals of $\rho = 15$ minutes. Fast chargers with an approximate power of 400 kW are used. The chargers process 2.5 kWh per minute for AEC levels from 20%-100%, 2 kWh per minute for AEC levels from 10%-20%, and 1.25 kWh per minute for AEC levels from 0%-10%. The number of iterations of the Monte Carlo simulations is set to $K = 1{,}000$.

We conduct our experiments on a Linux machine equipped with 16 Intel Xeon ES-2637 v4 processors running at 3.50 GHz and a RAM of 125 GB. The branch-and-price algorithm is implemented using the GENCOL library, version 4.5, and all linear programs are solved by the commercial solver CPLEX 22.1.

The remainder of this section is organized as follows. In Section 6.5.1, we analyze the algorithm behavior on I1, I2, and I3 for the deterministic case. Specifically, we analyze the optimality gap for all instances and specify for which instances our heuristic yields exact solutions. Section 6.5.2 compares our chance-constraint model for the E-VSP with battery degradation and stochastic energy consumption to the deterministic approach regarding operational costs and battery degradation. We present a visualization of the SoC profile over a day of solutions to a given test obtained by the deterministic and stochastic approach and highlight the added value of the latter for better battery management.

6.5.1. Algorithm behavior for the deterministic approach

When $\sigma^{rec} = \sigma^{max}$ or $\epsilon = 100\%$, our approach is equivalent to a deterministic approach considering only the worst-case energy consumption scenario. In other words, when $\sigma^{rec} = \sigma^{max}$ or $\epsilon = 100\%$, the chance-constraint in (6.2.2)-(6.2.7) becomes redundant. Since the PDFs of the energy consumption are only used to compute the probability of overuse of a bus battery, one could replace, for every trip in $\mathcal{V}$, its energy consumption PDF by a single value equal to its worst-case energy consumption. This deterministic approach also guarantees energy feasibility. We will use it as a baseline in what follows.

Table 6.3 presents the heuristic performance of our algorithm for the deterministic case. The table reports the average upper bound (Avg. UB), average lower bound (Avg. LB), the average relative difference in percentage between the UB and the LB (Avg. gap), and the largest relative difference in percentage between the UB and the LB (Largest gap). These metrics are averaged over the five tests conducted per family of instances. Note that our heuristic algorithm successfully finds the optimal solution for all the tests in the instance family I1, where σ^{max} is set to 100% or 70%. Additionally, all the tests in our experiments achieve optimality gaps less than or equal to 0.25%. For the instance families I1, I2, and I3, the average optimality gaps are 0.0033%, 0.017%, and 0.065%, respectively. These results indicate that our heuristic algorithm provides near-optimal solutions for the deterministic approach in most cases, providing a solid basis for further comparison with our stochastic approach.

6.5.2. Comparison with the E-VSP with battery degradation and stochastic energy consumption

This section compares the deterministic E-VSP with battery degradation to our stochastic approach. Table 6.4 presents the tradeoff between operational costs and battery degradation. Each value in the table represents an average over the five tests conducted for each instance family. The columns display the family of instances name (Inst.), the maximum allowed AEC (σ^{max}), the difference between the maximum allowed and the recommended AEC ($\sigma^{max} - \sigma^{rec}$), the chance-constraint threshold (ϵ), the operational costs (Op. costs), the operational costs increase (Op. costs incr.), the number of buses (# bus), the capacity fading per year and BEB (Cap. fade), and the capacity fading improvement (Cap. fade imp.). The operational costs increase and the capacity fading improvement are calculated relative to the baseline case where $\sigma^{max} = 100\%$.

The average capacity fading decreases with σ^{max} for all instances. However, this reduction in capacity fading comes at the cost of increased operational costs. For example, the average capacity fading of instance I2 is 23.10 kWh per year and BEB with $\sigma^{max} = 100\%$ and 4.00 kWh per year and BEB with $\sigma^{max} = 55\%$ and $\sigma^{max} - \sigma^{rec} = 0\%$. This decrease

Table 6.3 – Heuristic performance of the deterministic approach to the E-VSP with battery degradation

Instance	σ^{max}	Avg. UB	Avg. LB	Avg. gap (%)	Largest gap (%)
I1	100	3,293.28	3,293.28	0.0	0.0
	85	3,889.44	3,889.20	0.0056	0.028
	70	4,097.00	4,097.00	0.0	0.0
	55	4,310.28	4,309.96	0.0074	0.037
	avg.	**3,897.50**	**3,897.36**	**0.0033**	**0.016**
I2	100	7,465.04	7,463.36	0.023	0.051
	85	7,873.92	7,872.04	0.023	0.076
	70	8,280.68	8,280.04	0.0076	0.024
	55	8,523.04	8,521.64	0.016	0.077
	avg.	**8,035.67**	**8,034.27**	**0.017**	**0.057**
I3	100	11,607.00	11,602.25	0.041	0.091
	85	11,649.96	11,646.33	0.031	0.039
	70	11,918.72	11,903.40	0.13	0.25
	55	12,568.84	12,561.28	0.061	0.13
	avg.	**11,936.13**	**11,928.32**	**0.065**	**0.13**

of 82.7% in average capacity fading comes at the cost of a 14.2% increase in operational costs. Interestingly, for most instances (except I1 and I2 with $\sigma^{max} = 85\%$ and $\epsilon = 0.2$), we found that the stochastic approach can provide additional capacity fading gains with a slight increase in operational costs. To illustrate this, let us consider the family of instances I2 with $\sigma^{max} = 70\%$. In the deterministic approach, the average capacity fading is measured at 9.21 kWh per year and per vehicle for an operational cost of 8,280.68, whereas the stochastic approach with $\epsilon = 0.1$ yields a reduced average capacity fading of 7.82 kWh per year and per vehicle for an operational cost of 8,290.76. Although these improvements in capacity fading may appear modest, they could impact the lifetime of a BEB fleet. For instance, with an EOL threshold of 20%, a BEB experiencing a capacity fading of 9.21 kWh per year would have a projected lifetime of 6.51 years. However, by employing the stochastic approach, we can extend the lifetime to 7.67 years, representing an increase of 1.16 years. This improvement in BEB lifetime comes with a marginal operational cost increase of only 1.2E-3%. To validate the benefits of our chance-constrained model over the deterministic model, controlled field experiments over a sufficiently long period of time should be carried out. Furthermore, we also observe that the stochastic approach with $\epsilon = 0.1$ generally produced solutions with smaller capacity fading compared to the stochastic approach with $\epsilon = 0.2$. However, it should be noted that the operational costs associated with the former approach were usually higher.

Table 6.4 – Operational costs versus battery degradation of solutions to the E-VSP with battery degradation and stochastic energy consumption

Inst.	σ^{max} (%)	$\sigma^{max} - \sigma^{rec}$(%)	ϵ	Op. costs	Op. costs incr. (%)	# bus	Cap. fade (kWh/year)	Cap. fade imp. (%)
I1	100	0	-	3,293.28	0.0	3.0	11.26	0.0
	85	0	-	3,889.44	18.1	3.6	11.08	1.6
		20	0.2	3,889.44	18.1	3.6	11.22	0.3
		20	0.1	3,889.52	18.1	3.6	10.82	3.9
	70	0	-	4,097.00	24.4	3.8	6.62	41.2
		20	0.2	4,098.12	24.4	3.8	6.13	45.5
		20	0.1	4,097.40	24.4	3.8	5.95	47.1
	55	0	-	4,310.28	30.9	4.0	2.95	73.8
		20	0.2	4,317.00	31.1	4.0	2.58	77.1
		20	0.1	4,322.32	31.2	4.0	2.49	77.9
I2	100	0	-	7,465.04	0.0	7.0	23.10	0.0
	85	0	-	7,873.92	5.5	7.4	15.54	32.7
		20	0.2	7,872.12	5.4	7.4	15.65	32.2
		20	0.1	7,872.08	5.4	7.4	15.53	32.8
	70	0	-	8,280.68	10.9	7.8	9.21	60.1
		20	0.2	8,285.00	11.0	7.8	8.39	63.7
		20	0.1	8,290.76	11.1	7.8	7.82	66.1
	55	0	-	8,523.04	14.2	8.0	4.00	82.7
		20	0.2	8,567.72	14.8	8.0	3.22	86.1
		20	0.1	8,576.20	14.9	8.0	3.16	86.3
I3	100	0	-	11,607.00	0.0	11.0	28.50	0.0
	85	0	-	11,649.96	0.37	11.0	17.38	39.0
		20	0.2	11,659.12	0.45	11.0	17.26	39.4
		20	0.1	11,850.72	2.1	11.2	17.09	40.0
	70	0	-	11,918,72	2,7	11.2	9.15	67.9
		20	0.2	12,124.76	4.5	11.4	8.35	70.7
		20	0.1	11,934.56	2.8	11.2	8.13	71.5
	55	0	-	12,568.84	8.3	11.8	4.26	85.0
		20	0.2	12,842.48	10.6	12.0	3.37	88.2
		20	0.1	12,864.60	10.8	12.0	3.27	88.5

Figure 6.5 summarizes the results presented in Table 6.4, with each point in the graph corresponding to a line in the table.

For each family of instances, represented by red for I1, blue for I2, and green for I3, three distinct "groups" of points can be observed. Each group corresponds to a different σ^{max} value, namely $\sigma^{max} = 85, 70, 55$. The large gaps between the operational costs of these groups indicate an increase in fleet size, which accounts for a significant portion of the operational costs. Notably, as the instance size increases from I1 to I3, smaller operational

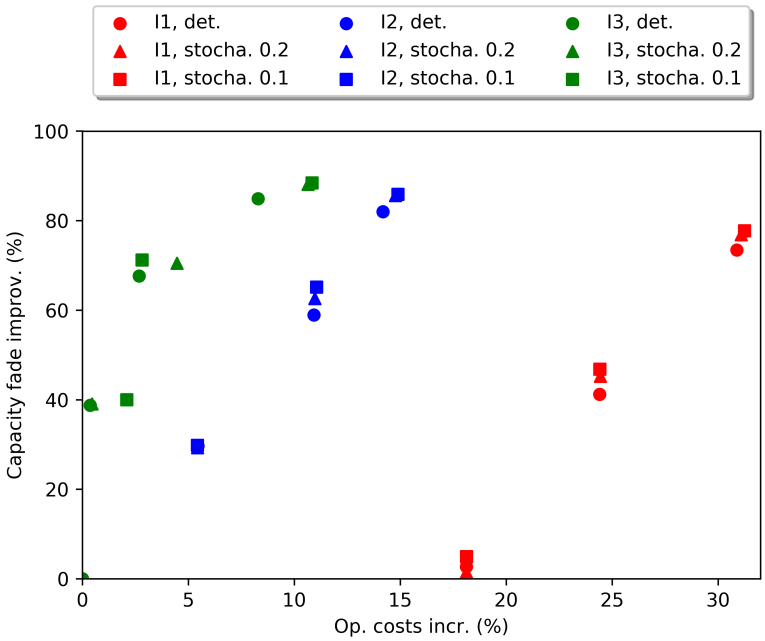
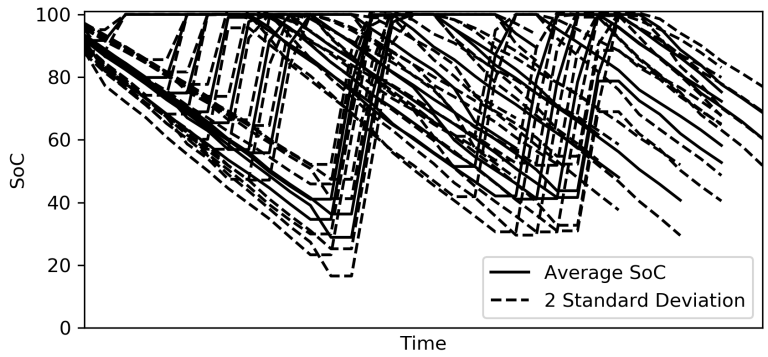


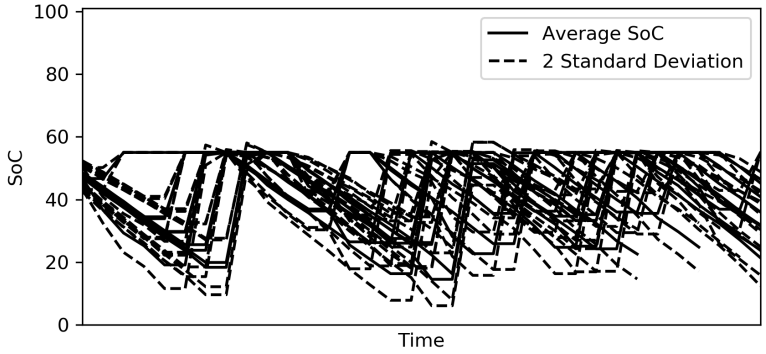
Figure 6.5 – Capacity fade improvement per operational costs increase for the deterministic and the stochastic cases (with $\epsilon = 0.1, 0.2$)

costs increases yield larger capacity fading improvements. This observation is particularly interesting and suggests that, with larger instances, more substantial gains in capacity fading improvement could potentially be achieved with relatively smaller increases in operational costs.

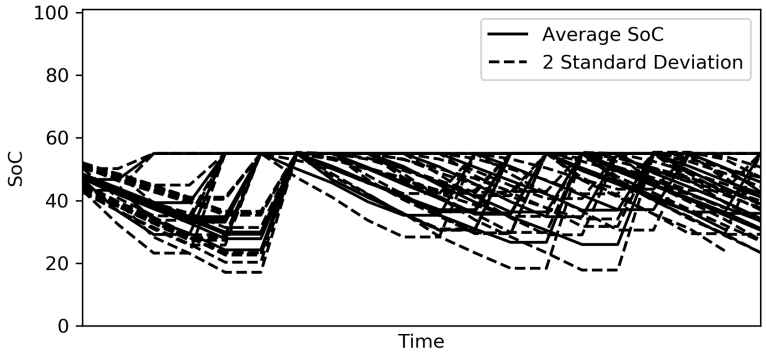
Figure 6.6 provides examples of the SoC profiles of the vehicle schedules selected in the solutions, found using a deterministic and a stochastic approach, for a test conducted in I3. Each solid line in Figures 6.6a, 6.6b, and 6.6c represents the average SoC as a function of time for a BEB associated with a selected vehicle schedule. The dotted lines represent the average SoC plus or minus two times the standard deviation. In Figure 6.6a, we depict the baseline case where battery degradation is not considered. In this case, the average SoC remains high, and the SoC level exhibits significant variation throughout the day. However, by reducing σ^{max} from 100% to 55%, both the average SoC and the SoC deviation are significantly reduced (see Figure 6.6b). Introducing our chance-constrained model (see Figure 6.6c), we observe a further reduction in SoC deviation by imposing a threshold on the maximum probability of one or more vehicles experiencing SoC levels below $100\% - \sigma^{rec} = 20\%$. This reduction in the average SoC and SoC deviation is desirable from a battery management point of view, as it extends the lifespan of BEBs.



(a) Baseline, $\sigma^{max} = 100\%$



(b) Deterministic approach with and $\sigma^{max} = 55\%$



(c) Chance-constrained model with $\sigma^{max} = 55\%$, $\sigma^{rec} = 35\%$, and $\epsilon^{low} = 0.1$

Figure 6.6 – Example of SoC profiles of BEBs - I3

6.6. Conclusions

This work introduced a chance-constraint model to address the E-VSP with battery degradation and stochastic energy consumption. Our proposed model can limit the maximum

probability of overusing BEB's batteries throughout the day. Combined with a tunable parameter that controls the SoC cutoff during charging activities (en-route and overnight), our approach efficiently reduces the average SoC and the SoC deviation of BEBs. These two factors have been identified as significant accelerators of battery aging. The reduction in average SoC and SoC deviation achieved by our approach can be controlled through two parameters: the maximum allowed AEC (σ^{max}) and the recommended maximum AEC (σ^{rec}). When determining the values of these parameters, one should consider the tradeoff between battery degradation and operational costs.

To solve the chance-constraint problem, we developed a branch-and-price algorithm incorporating stochastic pricing problems. In our computational experiments, we first validated the quality of solutions obtained for the deterministic case, where the recommended maximum AEC equals the maximum allowed AEC. Subsequently, we compared this deterministic approach, which allows for a tunable maximum allowed AEC, with our stochastic approach. Our results demonstrate that the proposed chance-constraint model offers additional gains in capacity fading for small operational costs increases. For example, for the analyzed BEBs and operational settings, our stochastic approach can extend the lifetime of each BEB in a fleet by more than one year, with a cost increase of only 1.2E-3% compared to the deterministic approach.

Chapter 7

Conclusions and outlook

This thesis proposed probabilistic models for predicting uncertainties in public transport using APTS data. Additionally, it introduced three stochastic models and column generation-based algorithms for variants of the VSP, namely the MDVSP and the E-VSP. These stochastic models are considered *data-driven* because they incorporate, or have the potential to incorporate, information predicted by statistical learning models trained on historical APTS data, namely the PDFs of the travel time and the energy consumption of bus trips.

The research advancements in this thesis represent a step toward more comprehensive and accurate public transport service planning. Specifically, we contributed to the existing literature by providing valuable insights into understanding and incorporating the inherent uncertainty of transport networks in the VSP, which has traditionally been tackled deterministically. The papers presented in Chapters 3 - 4 and Chapters 5 - 6 aim to improve the quality of public transport service and battery management, respectively. This agenda is of utmost importance for public transport agencies and policymakers in the context of the current climate crisis. We now summarize the main contributions of this thesis and conclude by identifying the limitations of our work and potential future research.

7.1. Synthesis of work

Chapter 3 established the foundation for the data-driven approaches employed in this thesis. In particular, it allows selecting a probabilistic model for the long-term prediction of the PDF of travel time, which performs well in predicting the true conditional PDF of travel time and in estimating the expected reliability of a bus schedule (i.e., tolerance to delays). The latter serves as the ultimate objective of this research: to measure the accuracy of a prospective bus schedule. This information can then by used by the operators' schedulers to evaluate and compare vehicle schedules in terms of their reliability or by a computer to optimize over a large number of potential vehicle schedules. Hence, our approach

benefits public transport agencies, their optimization-based software providers, and public transportation passengers by improving the quality of their local bus service.

Our second contribution built on the insights gained from our first paper and introduced a data-driven model for the MDVSP with stochastic travel time, which we refer to as the R-MDVSP-STT (Chapter 4). The objective is to find vehicle schedules that strike a balance between operational costs and reliability. We proposed a column generation-based algorithm that incorporates several acceleration strategies to solve this complex stochastic program. Additionally, we define novel reliability metrics that enable practitioners to comprehend better the tradeoff between operational costs and reliability in MDVSP solutions. The impact of this project extends beyond theoretical contributions, as our industrial partner, GIRO Inc., has recently successfully implemented our findings into their software. Ultimately, this project will assist public transport agencies in enhancing the quality of their service, a particularly relevant objective in the context of deregulation and liberalization of public transport markets, where transport agencies are often required to commit to a minimum level of quality or face penalties.

Chapter 5 extended the knowledge obtained from our previous papers. It introduced a novel two-stage stochastic model with recourse, incorporating stochastic travel time and energy consumption. This model, named the S-MDEVSP, focuses on addressing battery management challenges. Efficient energy management of BEBs is crucial for public transport agencies with a (partial) fleet of electric buses. Our research aims to help these agencies address energy feasibility constraints while using as few vehicles as possible. To achieve this objective, we introduced a recourse policy that enables extending the charge time when deviations in energy consumption are observed. By allowing flexible adjustments to the charging duration, we ensure that energy feasibility constraints are respected while optimizing the utilization of BEBs.

In Chapter 6, we addressed the issue of battery degradation in BEBs by introducing a novel chance-constraint model for the E-VSP with battery degradation and stochastic energy consumption. This model limits the maximum probability of overusing the BEBs' batteries throughout the day. Combined with a parameter that controls the charger cutoff SoC (i.e., the threshold at which chargers stop transferring energy to batteries), our approach effectively balances operational costs with battery degradation. Computational experiments of a tailored branch-and-price algorithm demonstrated that our approach can significantly extend the lifetime of BEBs, with only marginal increases in operational costs. Since replacing a battery in a BEB is expensive and carries a substantial environmental burden, addressing the tradeoff between battery degradation and operational costs is particularly important to public transport agencies. Hence, our tool is a valuable resource for assisting public transport agencies in making informed decisions tailored to their settings.

7.2. Limitations and future research

This thesis makes significant progress towards modeling uncertainties in public transport planning by proposing innovative data-driven models for the VSP under uncertainties and tailored column generation-based algorithms. Our contributions and the insights they provide open many opportunities for interesting research directions that could be addressed in future work.

The VSP models presented in this thesis assume uncorrelated uncertainties. However, trips are likely to have correlated travel time or energy consumption, especially among trips assigned to the same vehicle with similar spatiotemporal attributes. Recent work by Rostami et al. (2021) proposes a parametric convex binary quadratic program for the capacitated vehicle routing problem (VRP) with uncertain and statistically correlated travel time, but this approach is not directly applicable to the VSP. Indeed, in the VSP, each task is associated with a single-valued *time window*, i.e., a start time. Although the VRP with time windows (VRPTW) and stochastic travel time has been studied previously (see, e.g., Errico et al., 2018), the literature has yet to address the VRPTW with uncertain and statistically correlated travel time. To address this gap in the VSP and VRP literature, simulation-based optimization seems to be the most relevant methodology. In our second contribution, we compared solutions to the R-MDVSP-STT in terms of reliability metrics estimated offline using a Monte Carlo simulation. This procedure allowed us to assess the impact of a special case of statistically correlated travel times, namely when external and extraordinary factors, such as a severe snowstorm, cause travel time deviations of the same magnitude for all timetabled trips. Future research could explore incorporating the same type of Monte Carlo simulation directly into the optimization problem as part of a simulation-based optimization model.

As transportation networks move from ICE buses to BEBs, taking into account the uncertainty inherent in VSPs will raise more computational issues. Indeed, we identified that the E-VSP is much harder to solve than the MDVSP, which can be explained by two main causes. On the one hand, the underlying networks of the E-VSP typically contain a larger number of arcs and nodes than those of the MDVSP. These additional arcs and nodes are added to model charging activities. On the other hand, the E-VSP contains additional constraints related to the energy and charging station capacity. Our approach for the E-VSP with stochastic energy consumption presented in Chapter 6 can solve instances of approximately 250 trips and one depot, significantly extending existing works on the E-VSP under uncertainties and with capacity-limited charging stations. However, as transport system electrification continues to progress, larger problems are likely to arise. To address these computational issues, several potential avenues could be explored. For instance, decomposing the E-VSP into two easier problems—the VSP and the charge scheduling—could allow

for sequential or partially integrated solution schemes, similar to the approaches adopted for other combinations of transit planning sub-problems (see, e.g., Amberg et al., 2019; Perumal et al., 2021). Another approach involves developing a more concise representation of charging activities compared to the current time-expanded charging nodes.

In recent years, advancements in telecommunication technologies have made it possible to utilize online APTS data for real-time adjustments to the planned schedule based on the current state of the bus network. As a result, the dynamic VSP has garnered significant attention in the literature since the early work of Huisman et al. (2004). Our approaches were developed in a static setting, which also holds relevance to transport agencies in planning ahead of time for their bus service. Nevertheless, we could adapt our models to the dynamic case to extend our current work. The literature on dynamic and stochastic VSP remains largely unexplored, apart from the notable exception of He et al. (2018). Investigating this setting could provide great benefits for policymakers.

References

Abdelaty, H., Al-Obaidi, A., Mohamed, M., Farag, H.E., 2021. Machine learning prediction models for battery-electric bus energy consumption in transit. Transportation Research Part D: Transport and Environment 96, 102868. doi:https://doi.org/10.1016/j.trd.2021.102868.

Abkowitz, M.D., Engelstein, I., 1983. Factors affecting running time on transit routes. Transportation Research Part A: General 17A, 107 – 113. doi:https://doi.org/10.1016/0191-2607(83)90064-X.

Adler, J.D., Mirchandani, P.B., 2017. The vehicle scheduling problem for fleets with alternative-fuel vehicles. Transportation Science 51, 441–456. doi:https://doi.org/10.1287/trsc.2015.0615.

Ahuja, R.K., Magnanti, T.L., Orlin, J.B., 1993. Network Flows: Theory, Algorithms, and Applications. Prentice Hall.

Alvo, M., Angulo, G., Klapp, M.A., 2021. An exact solution approach for an electric bus dispatch problem. Transportation Research Part E: Logistics and Transportation Review 156, 102528. doi:https://doi.org/10.1016/j.tre.2021.102528.

Amaran, S., Sahinidis, N.V., Sharda, B., Bury, S.J., 2016. Simulation optimization: a review of algorithms and applications. Annals of Operations Research 240, 351–380. doi:10.1007/s10479-015-2019-x.

Amberg, B., Amberg, B., Kliewer, N., 2011. Increasing delay-tolerance of vehicle and crew schedules in public transport by sequential, partial-integrated and integrated approaches. Procedia - Social and Behavioral Sciences 20, 292–301. doi:https://doi.org/10.1016/j.sbspro.2011.08.035.

Amberg, B., Amberg, B., Kliewer, N., 2019. Robust efficiency in urban public transportation: Minimizing delay propagation in cost-efficient bus and driver schedules. Transportation Science 53, 89–112. doi:https://doi.org/10.1287/trsc.2017.0757.

Amnesty international, 2016. "This is what we die for": Human rights abuses in the Democratic Republic of the Congo power the global trade in cobalt. Amnesty international and African Resources Watch.

Barnhart, C., Johnson, E.L., Nemhauser, G.L., Savelsbergh, M.W.P., Vance, P.H., 1998. Branch-and-price: Column generation for solving huge integer programs. Operations Research 46, 316–329. doi:https://doi.org/10.1287/opre.46.3.316.

Barton, R.R., Meckesheimer, M., 2006. Chapter 18 metamodel-based simulation optimization, in: Henderson, S.G., Nelson, B.L. (Eds.), Simulation. Elsevier. volume 13 of *Handbooks in Operations Research and Management Science*, pp. 535–574. doi:https://doi.org/10.1016/S0927-0507(06)13018-2.

Basma, H., Mansour, C., Haddad, M., Nemer, M., Stabat, P., 2020. Comprehensive energy modeling methodology for battery electric buses. Energy 207, 118241. doi:https://doi.org/10.1016/j.energy.2020.118241.

Bates, J., Polak, J., Jones, P., Cook, A., 2001. The valuation of reliability for personal travel. Transportation Research Part E: Logistics and Transportation Review 37, 191 – 229. doi:https://doi.org/10.1016/S1366-5545(00)00011-9.

Ben-Tal, A., Ghaoui, L.E., Nemirovski, A., 2009. Robust Optimization. Princeton University Press, Princeton. doi:doi:10.1515/9781400831050.

Benchimol, P., Desaulniers, G., Desrosiers, J., 2012. Stabilized dynamic constraint aggregation for solving set partitioning problems. European Journal of Operational Research 223, 360–371. doi:https://doi.org/10.1016/j.ejor.2012.07.004.

Bertossi, A.A., Carraresi, P., Gallo, G., 1987. On some matching problems arising in vehicle scheduling models. Networks 17, 271–281. doi:https://doi.org/10.1002/net.3230170303.

Bianco, L., Mingozzi, A., Ricciardelli, S., 1994. A set partitioning approach to the multiple depot vehicle scheduling problem. Optimization Methods and Software 3, 163–194. doi:https://doi.org/10.1080/10556789408805563.

Bie, Y., Ji, J., Wang, X., Qu, X., 2021. Optimization of electric bus scheduling considering stochastic volatilities in trip travel time and energy consumption. Computer-Aided Civil and Infrastructure Engineering 36, 1530–1548. doi:https://doi.org/10.1111/mice.12684.

Bishop, C.M., 1994. Mixture Density Networks. Technical Report NCRG 4288. Neural Computing Research Group, Department of Computer Science, Aston University.

Boland, N., Dickson, S., Savelsbergh, M., Smilowitz, K., 2015. Dominance in pricing problems with stochasticity.

Bongiovanni, C., Kaspi, M., Cordeau, J.F., Geroliminis, N., 2022. A machine learning-driven two-phase metaheuristic for autonomous ridesharing operations. Transportation Research Part E: Logistics and Transportation Review 165, 102835. doi:https://doi.org/10.1016/j.tre.2022.102835.

Breiman, L., 1996. Bagging predictors. Machine Learning 24, 123–140.

Breiman, L., 2001. Random forests. Machine Learning 45, 5–32. doi:https://doi.org/10.1023/A:1010933404324.

Büchel, B., Corman, F., 2018. Modelling probability distributions of public transport travel time components, in: 18th Swiss Transport Research Conference, pp. 1–27. doi:https://doi.org/10.3929/ETHZ-B-000263929.

Bunte, S., Kliewer, N., 2010. An overview on vehicle scheduling models. Public Transport 1, 299–317. doi:https://doi.org/10.1007/s12469-010-0018-5.

Carney, M., Cunningham, P., Dowling, J., Lee, C., 2005. Predicting probability distributions for surf height using an ensemble of mixture density networks, in: Proceedings of the 22nd International Conference on Machine Learning, pp. 113–120. doi:10.1145/1102351.1102366.

Carpaneto, G., Dell'amico, M., Fischetti, M., Toth, P., 1989. A branch and bound algorithm for the multiple depot vehicle scheduling problem. Networks 19, 531–548. doi:https://doi.org/10.1002/net.3230190505.

Charnes, A., 1952. Optimality and degeneracy in linear programming. The Econometric Society 20, 160–170.

Chen, C.M., Liang, C.C., Chu, C.P., 2020. Long-term travel time prediction using gradient boosting. Journal of Intelligent Transportation Systems 24, 109–124. doi:https://doi.org/10.1080/15472450.2018.1542304.

Chen, Z., Li, S., D'Ariano, A., Yang, L., 2022. Real-time optimization for train regulation and stop-skipping adjustment strategy of urban rail transit lines. Omega 110, 102631. doi:https://doi.org/10.1016/j.omega.2022.102631.

Chen, Z., Yin, Y., Song, Z., 2018. A cost-competitiveness analysis of charging infrastructure for electric bus operations. Transportation Research Part C: Emerging Technologies 93, 351–366. doi:https://doi.org/10.1016/j.trc.2018.06.006.

Cluzel, C., Douglas, C., 2012. Cost and performance of EV batteries. Element Energy, Axeon, and EaStCHEM, Commissioned by The Committee on Climate Change. URL: https://www.element-energy.co.uk/wordpress/wp-content/uploads/2012/06/CCC-battery-cost_-Element-Energy-report_March2012_Finalbis.pdf.

Comi, A., Nuzzolo, A., Brinchi, S., Verghini, R., 2017. Bus travel time variability: some experimental evidences. Transportation Research Procedia 27, 101 – 108. doi:https://doi.org/10.1016/j.trpro.2017.12.072.

Costa, L., Contardo, C., Desaulniers, G., 2019. Exact branch-price-and-cut algorithms for vehicle routing. Transportation Science 53, 946–985. doi:10.1287/trsc.2018.0878.

Cox, P., 2013. Moving People - Sustainable Transport Development. Bloomsbury Publishing.

Daykin, A.R., Moffatt, P.G., 2002. Analyzing ordered responses: A review of the ordered probit model. Understanding Statistics 1, 157–166. doi:10.1207/S15328031US0103_02.

Dell'Amico, M., Fischetti, M., Toth, P., 1993. Heuristic algorithms for the multiple depot vehicle scheduling problem. Management Science 39, 115–125.

Desaulniers, G., Desrosiers, J., Solomon, M., 2005. Column Generation. doi:https://doi.org/10.1007/b135457.

Desaulniers, G., Hickman, M.D., 2007. Chapter 2 public transit, in: Barnhart, C., Laporte, G. (Eds.), Transportation. Elsevier. volume 14 of *Handbooks in Operations Research and Management Science*, pp. 69–127. doi:https://doi.org/10.1016/S0927-0507(06)14002-5.

Desaulniers, G., Lavigne, J., Soumis, F., 1998. Multi-depot vehicle scheduling problems with time windows and waiting costs. European Journal of Operational Research 111, 479–494. doi:https://doi.org/10.1016/S0377-2217(97)00363-9.

Duan, T., Anand, A., Ding, D.Y., Thai, K.K., Basu, S., Ng, A., Schuler, A., 2020. NGBoost: Natural gradient boosting for probabilistic prediction, in: International Conference on Machine Learning, pp. 2690–2700.

Ducan, K., 2011. Fairness in a Car-dependent Society. Sustainable Development Commission Reports. Sustainable Development Commission.

Dutordoir, V., Salimbeni, H., Deisenroth, M.P., Hensman, J., 2018. Gaussian process conditional density estimation, in: 32nd Conference on Neural Information Processing Systems, pp. 1–19.

Environment and Climate Change Canada, 2022. National inventory report 1990-2020: Greenhouse gas sources and sinks in canada. URL: https://www.canada.ca/en/environment-climate-change/services/climate-change/greenhouse-gas-emissions/inventory.html.

Errico, F., Desaulniers, G., Gendreau, M., Rei, W., Rousseau, L.M., 2018. The vehicle routing problem with hard time windows and stochastic service times. EURO Journal on Transportation and Logistics 7, 223–251. doi:https://doi.org/10.1007/s13676-016-0101-4.

Fischetti, M., Lodi, A., Toth, P., 1999. A branch-and-cut algorithm for the multiple depot vehicle scheduling problem.

Forbes, M., Holt, J., Watts, A., 1994. An exact algorithm for multiple depot bus scheduling. European Journal of Operational Research 72, 115–124. doi:https://doi.org/10.1016/0377-2217(94)90334-4.

Freling, R., Wagelmans, A.P.M., Paixão, J.M.P., 2001. Models and algorithms for single-depot vehicle scheduling. Transportation Science 35, 165–180. doi:https://doi.org/10.1287/trsc.35.2.165.10135.

Freund, Y., Schapire, R.E., 1996. Experiments with a new boosting algorithm, in: Proceedings of the Thirteenth International Conference on International Conference on Machine Learning, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. p. 148–156.

Gao, Z., Lin, Z., LaClair, T.J., Liu, C., Li, J.M., Birky, A.K., Ward, J., 2017. Battery capacity and recharging needs for electric buses in city transit service. Energy 122, 588–600. doi:https://doi.org/10.1016/j.energy.2017.01.101.

Gintner, V., Kliewer, N., Suhl, L., 2005. Solving large multiple-depot multiple-vehicle-type bus scheduling problems in practice. OR Spectrum 27, 507–523. doi:https://doi.org/10.1007/s00291-005-0207-9.

Gower, J.C., 1971. A general coefficient of similarity and some of its properties. Biometrics 27, 857–871.

Gössling, S., 2016. Urban transport justice. Journal of Transport Geography 54, 1–9. doi:https://doi.org/10.1016/j.jtrangeo.2016.05.002.

Hadjar, A., Marcotte, O., Soumis, F., 2006. A branch-and-cut algorithm for the multiple depot vehicle scheduling problem. Operations Research 54, 130–149. doi:https://doi.org/10.1287/opre.1050.0240.

Hannah, L.A., 2015. Stochastic optimization, in: Wright, J.D. (Ed.), International Encyclopedia of the Social & Behavioral Sciences (Second Edition). second edition ed.. Elsevier, Oxford, pp. 473–481. doi:https://doi.org/10.1016/B978-0-08-097086-8.42010-6.

Hastie, T., Tibshirani, R., 1986. Generalized additive models. Statistical Science 1, 297–310.

He, F., Yang, J., Li, M., 2018. Vehicle scheduling under stochastic trip times: An approximate dynamic programming approach. Transportation Research Part C: Emerging Technologies 96, 144–159. doi:https://doi.org/10.1016/j.trc.2018.09.010.

Hellerstein, J.M., 2008. Quantitative Data Cleaning for Large Databases. Technical Report. EECS Computer Science Division, UC Berckley.

Huisman, D., Freling, R., Wagelmans, A.P.M., 2004. A robust solution approach to the dynamic vehicle scheduling problem. Transportation Science 38, 447–458. doi:https://doi.org/10.1287/trsc.1030.0069.

Hwang, C.L., Masud, A.S.M., 1979. Multiple Objective Decision Making — Methods and Applications. Springer, Berlin, Heidelberg. doi:https://doi.org/10.1007/978-3-642-45511-7.

Häll, C.H., Ceder, A.A., Ekström, J., Quttineh, N.H., 2019. Adjustments of public transit operations planning process for the use of electric buses. Journal of Intelligent Transportation Systems 23, 216–230. doi:https://doi.org/10.1080/15472450.2018.1488131.

Ibarra-Rojas, O., Delgado, F., Giesen, R., Muñoz, J., 2015. Planning, operation, and control of bus transport systems: A literature review. Transportation Research Part B: Methodological 77, 38–75. doi:https://doi.org/10.1016/j.trb.2015.03.002.

Irnich, S., Desaulniers, G., 2005. Shortest path problems with resource constraints, in: Desaulniers, G., Desrosiers, J., Solomon, M.M. (Eds.), Column Generation. Springer, pp. 33–65. doi:https://doi.org/10.1007/0-387-25486-2_2.

Iverson, K.E., 1962. A Programming Language. John Wiley, USA.

Janovec, M., Kohání, M., 2019. Exact approach to the electric bus fleet scheduling. Transportation Research Procedia 40, 1380–1387. doi:https://doi.org/10.1016/j.trpro.2019.07.191.

Jiang, M., Zhang, Y., Zhang, Y., 2021. Optimal electric bus scheduling under travel time uncertainty: A robust model and solution method. Journal of Advanced Transportation 2021, 19. doi:https://doi.org/10.1155/2021/1191443.

Jordan, M., Jacobs, R., 1993. Hierarchical mixtures of experts and the EM algorithm, in: Proceedings of 1993 International Conference on Neural Networks (IJCNN-93-Nagoya, Japan), pp. 1339–1344 vol.2. doi:10.1109/IJCNN.1993.716791.

Kiefer, J., Wolfowitz, J., 1952. Stochastic estimation of the maximum of a regression function. The Annals of Mathematical Statistics 23, 462–466.

Kieu, L.M., Bhaskar, A., Chung, E., 2015. Public transport travel-time variability definitions and monitoring. Journal of Transportation Engineering 141, 04014068. doi:https://doi.org/10.1061/(ASCE)TE.1943-5436.0000724.

Kliewer, N., Mellouli, T., Suhl, L., 2006. A time–space network based exact optimization model for multi-depot bus scheduling. European Journal of Operational Research 175, 1616–1627. doi:https://doi.org/10.1016/j.ejor.2005.02.030.

Klunder, G., Baas, P., Op de Beek, F., 2007. A long-term travel time prediction algorithm using historical data. Technical Report. TNO.

Kolda, T.G., Lewis, R.M., Torczon, V., 2003. Optimization by direct search: New perspectives on some classical and modern methods. SIAM Review 45, 385–482. doi:10.1137/S003614450242889.

Koller, D., Friedman, N., 2009. Probabilistic Graphical Models: Principles and Techniques. The MIT press.

van Kooten Niekerk, M.E., 2018. Optimizing for Reliable and Sustainable Public Transport. Ph.D. thesis. Univeristeit Utrecht. Netherlands.

van Kooten Niekerk, M.E., van den Akker, J.M., Hoogeveen, J.A., 2017. Scheduling electric vehicles. Public Transport 9, 155–176. doi:https://doi.org/10.1007/s12469-017-0164-0.

Kramkowski, S., Kliewer, N., Meier, C., 2009. Heuristic methods for increasing delay-tolerance of vehicle schedules in public bus transport, in: VIII Metaheuristics International Conference, pp. 1–9.

Kulkarni, S., Krishnamoorthy, M., Ranade, A., Ernst, A.T., Patil, R., 2018. A new formulation and a column generation-based heuristic for the multiple depot vehicle scheduling problem. Transportation Research Part B: Methodological 118, 457–487. doi:https://doi.org/10.1016/j.trb.2018.11.007.

Kumar, B.A., Vanajakshi, L., Subramanian, S.C., 2014. Pattern-based bus travel time prediction under heterogeneous traffic conditions, in: Colloquium on Transportation Systems

Engineering and Management, pp. 1–16. doi:https://doi.org/10.13140/RG.2.1.2338.5448.

Lam, L., Bauer, P., 2012. Practical capacity fading model for li-ion battery cells in electric vehicles. IEEE Transactions on Power Electronics 28, 5910–5918. doi:10.1109/TPEL.2012.2235083.

Lamatsch, A., 1992. An approach to vehicle scheduling with depot capacity constraints, in: Desrochers, M., Rousseau, J.M. (Eds.), Computer-Aided Transit Scheduling, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 181–195.

Laurent, B., Hao, J.K., 2009. Iterated local search for the multiple depot vehicle scheduling problem. Computers & Industrial Engineering 57, 277–286. doi:https://doi.org/10.1016/j.cie.2008.11.028.

Li, J.Q., 2014. Transit bus scheduling with limited energy. Transportation Science 48, 521–539.

Li, J.Q., 2016. Battery-electric transit bus developments and operations: A review. International Journal of Sustainable Transportation 10, 157–169. doi:https://doi.org/10.1080/15568318.2013.872737.

Li, L., Lo, H.K., Huang, W., Xiao, F., 2021. Mixed bus fleet location-routing-scheduling under range uncertainty. Transportation Research Part B: Methodological 146, 155–179. doi:https://doi.org/10.1016/j.trb.2021.02.005.

Li, L., Lo, H.K., Xiao, F., 2019. Mixed bus fleet scheduling under range and refueling constraints. Transportation Research Part C: Emerging Technologies 104, 443–462. doi:https://doi.org/10.1016/j.trc.2019.05.009.

Li, X., Wang, T., Li, L., Feng, F., Wang, W., Cheng, C., 2020. Joint optimization of regular charging electric bus transit network schedule and stationary charger deployment considering partial charging policy and time-of-use electricity prices. Journal of Advanced Transportation 2020, 1–16. doi:10.1155/2020/8863905.

Liu, T., Ceder, A., 2020. Battery-electric transit vehicle scheduling with optimal number of stationary chargers. Transportation Research Part C: Emerging Technologies 114, 118–139. doi:https://doi.org/10.1016/j.trc.2020.02.009.

Lübbecke, M.E., Desrosiers, J., 2005. Selected topics in column generation. Operations Research 53, 1007–1023. doi:https://doi.org/10.1287/opre.1050.0234.

Löbel, A., 1998. Vehicle scheduling in public transit and lagrangean pricing. Management Science 44, 1637–1649.

Ma, Z., Ferreira, L., Mesbah, M., 2014. Measuring service reliability using automatic vehicle location data. Mathematical Problems in Engineering 2014, 1–12. doi:https://doi.org/10.1155/2014/468563.

Ma, Z., Ferreira, L., Mesbah, M., Zhu, S., 2016. Modeling distributions of travel time variability for bus operations. Journal of Advanced Transportation 50, 6–24. doi:https:

//doi.org/10.1002/atr.1314.

Mazloumi, E., Currie, G., Rose, G., 2010. Using gps data to gain insight into public transport travel time variability. Journal of Transportation Engineering 136, 623–631. doi:https://doi.org/10.1061/(ASCE)TE.1943-5436.0000126.

Mesquita, M., Paixão, J., 1999. Exact algorithms for the multi-depot vehicle scheduling problem based on multicommodity network flow type formulations, in: Wilson, N.H.M. (Ed.), Computer-Aided Transit Scheduling, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 221–243.

Millner, A., 2010. Modeling lithium ion battery degradation in electric vehicles, in: 2010 IEEE Conference on Innovative Technologies for an Efficient and Reliable Electricity Supply, pp. 349–356. doi:10.1109/CITRES.2010.5619782.

Montoya, A., Guéret, C., Mendoza, J.E., Villegas, J.G., 2017. The electric vehicle routing problem with nonlinear charging function. Transportation Research Part B: Methodological 103, 87–110. doi:https://doi.org/10.1016/j.trb.2017.02.004.

Moreira, J., Jorge, A., Sousa, J., Soares, C., 2012. Comparing state-of-the-art regression methods for long term travel time prediction. Intelligent Data Analysis 16, 427 – 449. doi:https://doi.org/10.3233/IDA-2012-0532.

Moreira-Matias, L., Mendes-Moreira, J., de Sousa, J.F., Gama, J., 2015. Improving mass transit operations by using avl-based systems: A survey. IEEE Transactions on Intelligent Transportation Systems 16, 1636–1653. doi:https://doi.org/10.1109/TITS.2014.2376772.

Moreno, C.A.M., Falcón, L.M.E., Bolaños, R.I., Subramanian, A., Zuluaga, A.H.E., Echeverri, M.G., 2019. A hybrid algorithm for the multi-depot vehicle scheduling problem arising in public transportation. International Journal of Industrial Engineering Computations 10, 361–374. doi:https://doi.org/10.5267/j.ijiec.2019.2.002.

Murphy, K., 2012. Machine Learning: A Probabilistic Perspective. volume 58. MIT Press.

Naumann, M., Suhl, L., Kramkowski, S., 2011. A stochastic programming approach for robust vehicle scheduling in public bus transport. Procedia - Social and Behavioral Sciences 20, 826–835. doi:https://doi.org/10.1016/j.sbspro.2011.08.091.

Nelder, J.A., Wedderburn, R.W.M., 1972. Generalized linear models. Journal of the Royal Statistical Society. Series A (General) 135, 370–384.

Nordelöf, A., Romare, M., Tivander, J., 2019. Life cycle assessment of city buses powered by electricity, hydrogenated vegetable oil or diesel. Transportation Research Part D: Transport and Environment 75, 211–222. doi:https://doi.org/10.1016/j.trd.2019.08.019.

Olsen, N., Kliewer, N., 2020. Scheduling electric buses in public transport: Modeling of the charging process and analysis of assumptions. Logistics Research 13, 18. doi:https://doi.org/10.23773/2020_4.

van Oort, N., 2011. Service Reliability and Urban Public Transport Design. Ph.D. thesis. Delft University of Technology.

Osorio, C., Bierlaire, M., 2013. A simulation-based optimization framework for urban transportation problems. Operations Research 61, 1333–1345. doi:10.1287/opre.2013.1226.

Otsuki, T., Aihara, K., 2016. New variable depth local search for multiple depot vehicle scheduling problems. Journal of Heuristics 22, 567–585. doi:https://doi.org/10.1007/s10732-014-9264-z.

Oukil, A., Amor, H.B., Desrosiers, J., El Gueddari, H., 2007. Stabilized column generation for highly degenerate multiple-depot vehicle scheduling problems. Computers & Operations Research 34, 817–834. doi:https://doi.org/10.1016/j.cor.2005.05.011.

Pelletier, S., Jabali, O., Laporte, G., 2016. 50th anniversary invited article—goods distribution with electric vehicles: Review and research perspectives. Transportation Science 50, 3–22. doi:10.1287/trsc.2015.0646.

Pelletier, S., Jabali, O., Laporte, G., 2018. Charge scheduling for electric freight vehicles. Transportation Research Part B: Methodological 115, 246–269. doi:https://doi.org/10.1016/j.trb.2018.07.010.

Pelletier, S., Jabali, O., Laporte, G., Veneroni, M., 2017. Battery degradation and behaviour for electric vehicles: Review and numerical analyses of several models. Transportation Research Part B: Methodological 103, 158–187. doi:10.1016/j.trb.2017.01.020.

Pepin, A.S., Desaulniers, G., Hertz, A., Huisman, D., 2009. A comparison of five heuristics for the multiple depot vehicle scheduling problem. Journal of Scheduling 12, 17–30. doi:https://doi.org/10.1007/s10951-008-0072-x.

Perumal, S.S., Dollevoet, T., Huisman, D., Lusby, R.M., Larsen, J., Riis, M., 2021. Solution approaches for integrated vehicle and crew scheduling with electric buses. Computers & Operations Research 132, 105268. doi:https://doi.org/10.1016/j.cor.2021.105268.

Perumal, S.S., Lusby, R.M., Larsen, J., 2022. Electric bus planning & scheduling: A review of related problems and methodologies. European Journal of Operational Research 301, 395–413. doi:https://doi.org/10.1016/j.ejor.2021.10.058.

Pillac, V., Gendreau, M., Guéret, C., Medaglia, A.L., 2013. A review of dynamic vehicle routing problems. European Journal of Operational Research 225, 1–11. doi:https://doi.org/10.1016/j.ejor.2012.08.015.

Powell, B.W., 2022. Reinforcement Learning and Stochastic Optimization: A Unified Framework for Sequential Decisions. John Wiley & Sons, Inc., Hoboken, USA. doi:https://doi.org/10.1002/9781119815068.

Proterra, 2022. ZX5 35-foot battery-electric transit bus - platform specifications. URL: https://www.proterra.com/wp-content/uploads/2022/09/SPEC_35_001_METRIC_Q4_2022_V1_09_09_22-1.pdf.

Raff, S., 1983. Routing and scheduling of vehicles and crews: The state of the art. Computers & Operations Research 10, 63–211. doi:https://doi.org/10.1016/0305-0548(83)90030-8.

Rasmussen, C.E., Williams, C.K.I., 2005. Gaussian Processes for Machine Learning. The MIT Press. doi:10.7551/mitpress/3206.001.0001.

Ribeiro, C.C., Soumis, F., 1994. A column generation approach to the multiple-depot vehicle scheduling problem. Operations Research 42, 41–52.

Ricard, L., Desaulniers, G., Lodi, A., Rousseau, L.M., 2022. Predicting the probability distribution of bus travel time to measure the reliability of public transport services. Transportation Research Part C: Emerging Technologies 138, 103619. doi:https://doi.org/10.1016/j.trc.2022.103619.

Rigby, R.A., Stasinopoulos, D.M., 2005. Generalized additive models for location, scale and shape. Journal of the Royal Statistical Society: Series C (Applied Statistics) 54, 507–554. doi:https://doi.org/10.1111/j.1467-9876.2005.00510.x.

Rinaldi, M., Picarelli, E., D'Ariano, A., Viti, F., 2020. Mixed-fleet single-terminal bus scheduling problem: Modelling, solution scheme and potential applications. Omega 96, 102070. doi:https://doi.org/10.1016/j.omega.2019.05.006.

Robbins, H., Monro, S., 1951. A stochastic approximation method. The Annals of Mathematical Statistics 22, 400–407.

Rostami, B., Desaulniers, G., Errico, F., Lodi, A., 2021. Branch-price-and-cut algorithms for the vehicle routing problem with stochastic and correlated travel times. Operations Research 69, 436–455. doi:10.1287/opre.2020.2037.

Sassi, O., Oulamara, A., 2017. Electric vehicle scheduling and optimal charging problem: complexity, exact and heuristic approaches. International Journal of Production Research 55, 519–535. doi:10.1080/00207543.2016.1192695.

Schlosser, L., Hothorn, T., Stauffer, R., Zeileis, A., 2019. Distributional regression forests for probabilistic precipitation forecasting in complex terrain. The Annals of Applied Statistics 13, 1564–1589. doi:https://doi.org/10.1214/19-aoas1247.

Shen, Y., Xu, J., Li, J., 2016. A probabilistic model for vehicle scheduling based on stochastic trip times. Transportation Research Part B: Methodological 85, 19–31. doi:https://doi.org/10.1016/j.trb.2015.12.016.

Strathman, J.G., Hopper, J.R., 1993. Empirical analysis of bus transit on-time performance. Transportation Research Part A: Policy and Practice 27, 93 – 100. doi:https://doi.org/10.1016/0965-8564(93)90065-S.

Sutton, R.S., Barto, A.G., 2018. Reinforcement Learning: An Introduction. Second ed., The MIT Press.

Søndergaard, J., 2003. Optimization Using Surrogate Models—by the Space Mapping Technique. Ph.D. thesis. Technical University of Denmark. Denmark.

Tang, X., Lin, X., He, F., 2019. Robust scheduling strategies of electric buses under stochastic traffic conditions. Transportation Research Part C: Emerging Technologies 105, 163–182. doi:https://doi.org/10.1016/j.trc.2019.05.032.

Thomopoulos, N., 2012. Essentials of Monte Carlo Simulation: Statistical Methods for Building Simulation Models. Springer New York.

Vij, A., Gorripaty, S., Walker, J.L., 2017. From trend spotting to trend 'splaining: Understanding modal preference shifts in the san francisco bay area. Transportation Research Part A: Policy and Practice 95, 238–258. doi:https://doi.org/10.1016/j.tra.2016.11.014.

Wang, C., Guo, C., Zuo, X., 2021. Solving multi-depot electric vehicle scheduling problem by column generation and genetic algorithm. Applied Soft Computing 112, 107774. doi:https://doi.org/10.1016/j.asoc.2021.107774.

Wang, H., Liu, Y., Fu, H., Li, G., 2013. Estimation of state of charge of batteries for electric vehicles. International Journal of Control and Automation 6, 185–194.

Weigl, C., 2021. An investigation into deep seabed mining and materials. Worldwide Fund for Nature.

Wellman, M.P., Ford, M., Larson, K., 2013. Path planning under time-dependent uncertainty. arXiv:1302.4987.

Wen, M., Linde, E., Ropke, S., Mirchandani, P., Larsen, A., 2016. An adaptive large neighborhood search heuristic for the electric vehicle scheduling problem. Computers & Operations Research 76, 73–83. doi:https://doi.org/10.1016/j.cor.2016.06.013.

Wu, W., Lin, Y., Liu, R., Jin, W., 2022. The multi-depot electric vehicle scheduling problem with power grid characteristics. Transportation Research Part B: Methodological 155, 322–347. doi:https://doi.org/10.1016/j.trb.2021.11.007.

Yeo, K., Melnyk, I., Nguyen, N., Lee, E.K., 2018. DE-RNN: Forecasting the probability density function of nonlinear time series, in: 2018 IEEE International Conference on Data Mining (ICDM), pp. 697–706. doi:https://doi.org/10.1109/ICDM.2018.00085.

Yetiskul, E., Senbil, M., 2012. Public bus transit travel-time variability in ankara (turkey). Transport Policy 23, 50 – 59. doi:https://doi.org/10.1016/j.tranpol.2012.05.008.

Zeng, Z., Wang, S., Qu, X., 2022. On the role of battery degradation in en-route charge scheduling for an electric bus system. Transportation Research Part E: Logistics and Transportation Review 161, 102727. doi:https://doi.org/10.1016/j.tre.2022.102727.

Zhang, L., Wang, S., Qu, X., 2021. Optimal electric bus fleet scheduling considering battery degradation and non-linear charging profile. Transportation Research Part E: Logistics and Transportation Review 154, 102445. doi:https://doi.org/10.1016/j.tre.2021.102445.

Zhang, Y., Xiong, R., He, H., Qu, X., Pecht, M., 2019. State of charge-dependent aging mechanisms in graphite/li(nicoal)o2 cells: Capacity loss modeling and remaining useful

life prediction. Applied Energy 255, 113818. doi:https://doi.org/10.1016/j.apenergy.2019.113818.

Zhou, G.J., Xie, D.F., Zhao, X.M., Lu, C., 2020. Collaborative optimization of vehicle and charging scheduling for a bus fleet mixed with electric and traditional buses. IEEE Access 8, 8056–8072. doi:10.1109/ACCESS.2020.2964391.

Zhou, Y., Meng, Q., Ong, G.P., 2022. Electric bus charging scheduling for a single public transport route considering nonlinear charging profile and battery degradation effect. Transportation Research Part B: Methodological 159, 49–75. doi:https://doi.org/10.1016/j.trb.2022.03.002.

Appendix A

Article 2 Appendix

A.1. Derivation of the cumulative distribution functions of the actual departure time

$$F_j^p(z) = \sum_{y=d_j^0}^{z} f_j^p(y) \tag{A.1.1}$$

$$= \left[\sum_{k \in \Phi_i} h_i(k) \times \sum_{y'=d_i^0}^{d_j^0 - \kappa_{i,j} - \tau - k} f_i^{p'}(y') \right] + \sum_{y=d_j^0+1}^{z} \left[\sum_{k \in \Phi_i} h_i(k) \times f_i^{p'}(y - \kappa_{i,j} - \tau - k) \right] \tag{A.1.2}$$

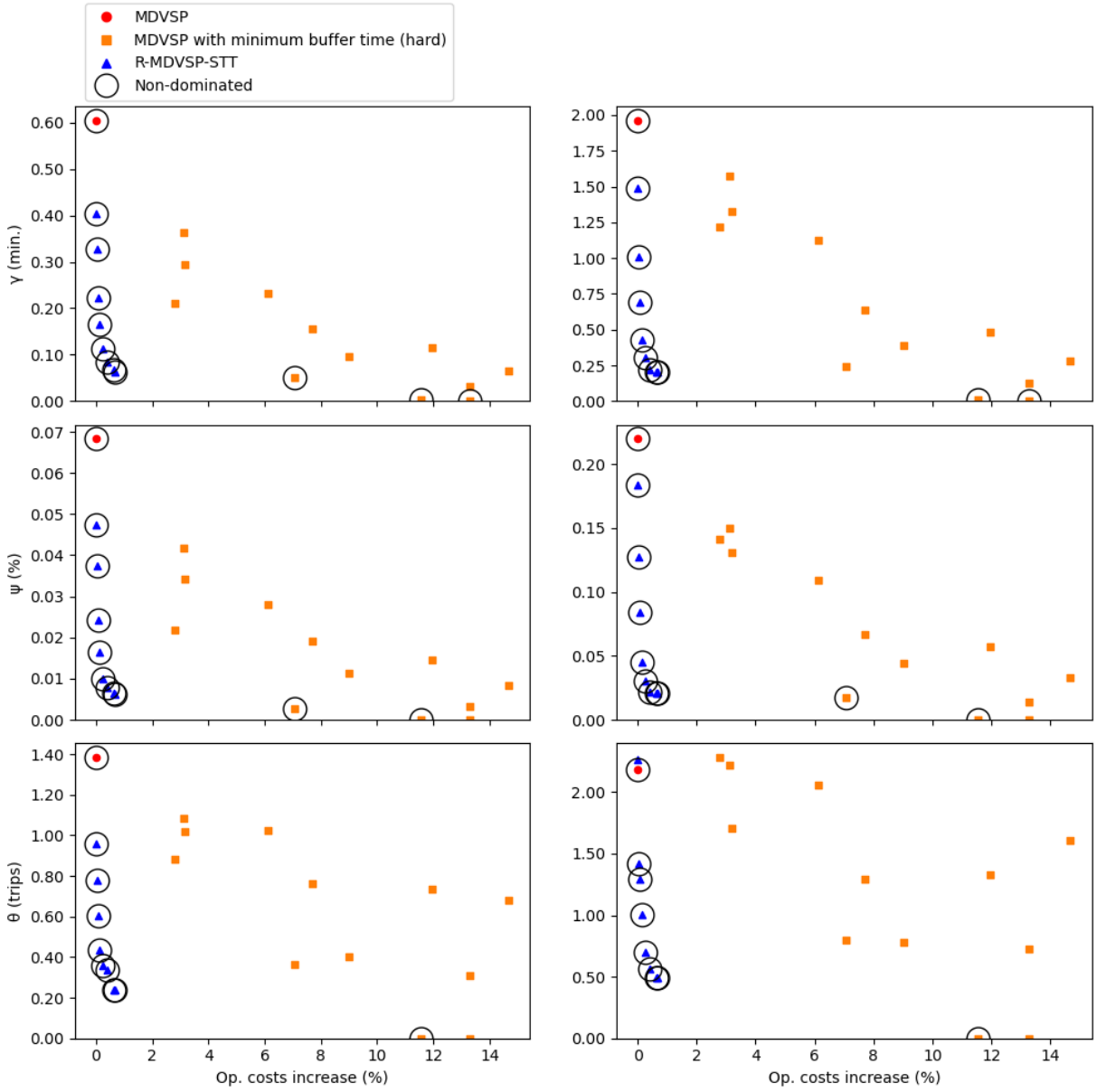
$$= \sum_{k \in \Phi_i} h_i(k) \left[\sum_{y'=d_i^0}^{d_j^0 - \kappa_{i,j} - \tau - k} f_i^{p'}(y') + \sum_{y=d_j^0+1}^{z} f_i^{p'}(y - \kappa_{i,j} - \tau - k) \right] \tag{A.1.3}$$

$$= \sum_{k \in \Phi_i} h_i(k) \sum_{y'=d_i^0}^{z - \kappa_{i,j} - \tau - k} f_i^{p'}(y') \tag{A.1.4}$$

$$= \sum_{k \in \Phi_i} h_i(k) \times F_i^{p'}(z - \kappa_{i,j} - \tau - k) \tag{A.1.5}$$

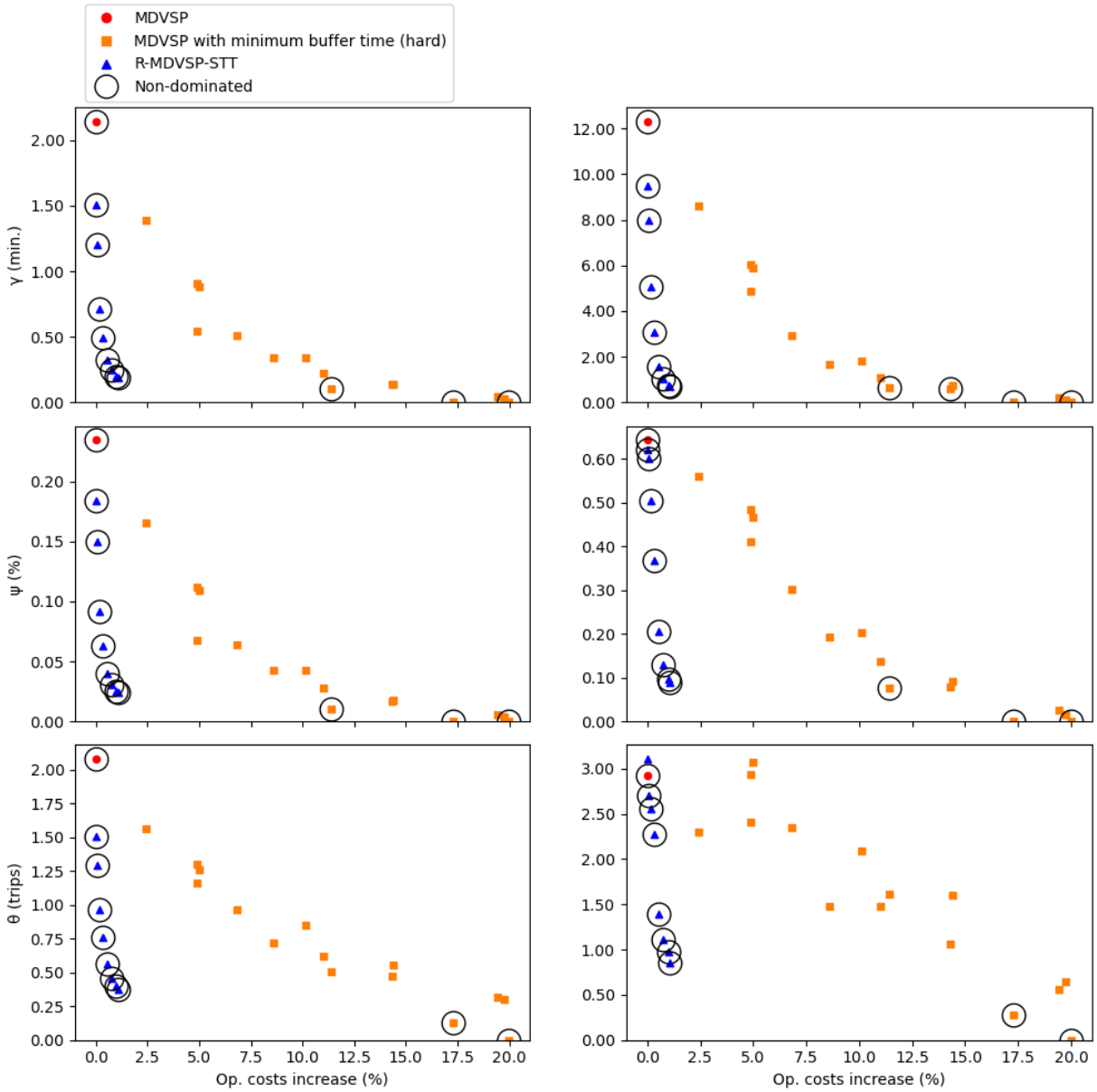
The term on the right-hand side of (A.1.3) reduces to $\sum_{y'=d_i^0}^{z-\kappa_{i,j}-\tau-k} f_i^{p'}(y')$ because $\sum_{y'=d_i^0}^{d_j^0-\kappa_{i,j}-\tau-k} f_i^{p'}(y')$ can be expressed as $f_i^{p'}(d_i^0)+\cdots+f_i^{p'}(d_j^0-\kappa_{i,j}-\tau-k)$ and $\sum_{y=d_j^0+1}^{z} f_i^{p'}(y-\kappa_{i,j}-\tau-k)$ can be expressed as $f_i^{p'}(d_j^0+1-\kappa_{i,j}-\tau-k)+\cdots+f_i^{p'}(z-\kappa_{i,j}-\tau-k)$.

A.2. Additional results



(a) 1st case **(b)** 2nd case

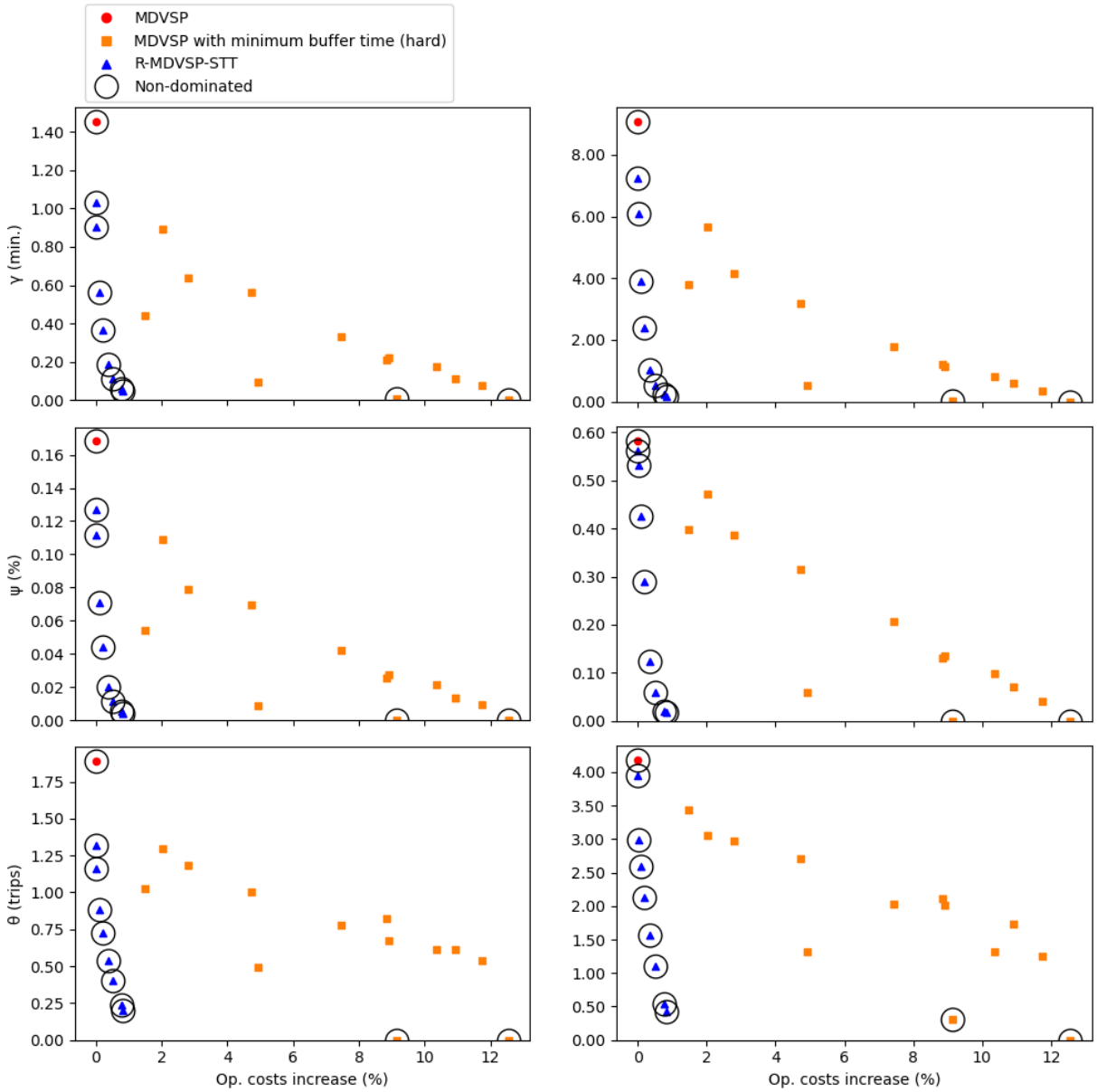
Figure A.1 – Reliability metrics - I1



(a) 1st case

(b) 2nd case

Figure A.2 – Reliability metrics - I2



(a) 1st case **(b)** 2nd case

Figure A.3 – Reliability metrics - I3