Les Cahiers du GERAD ISSN: 0711–2440

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

L. Ricard, G. Desaulniers, A. Lodi, L.-M. Rousseau

G–2022–30

July 2022

Citation suggérée : L. Ricard, G. Desaulniers, A. Lodi, L.-M. Rousseau (Juillet 2022). Increasing schedule reliability in the multi-depot vehicle scheduling problem with stochastic travel time, Rapport technique, Les Cahiers du GERAD G– 2022–30, GERAD, HEC Montréal, Canada.

Avant de citer ce rapport technique, veuillez visiter notre site Web (https://www.gerad.ca/fr/papers/G-2022-30) afin de mettre à jour vos données de référence, s'il a été publié dans une revue scientifique.

Suggested citation: L. Ricard, G. Desaulniers, A. Lodi, L.-M. Rousseau (July 2022). Increasing schedule reliability in the multi-depot vehicle scheduling problem with stochastic travel time, Technical report, Les Cahiers du GERAD G–2022–30, GERAD, HEC Montréal, Canada.

Before citing this technical report, please visit our website (https://www.gerad.ca/en/papers/G-2022-30) to update your reference data, if it has been published in a scientific journal.

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2022
– Bibliothèque et Archives Canada, 2022

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2022
– Library and Archives Canada, 2022

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

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

Léa Ricard [a, b, c]

Guy Desaulniers [a, d]

Andrea Lodi [a, c, d, e]

Louis-Martin Rousseau [d]

[a] *GERAD, Montréal (Qc), Canada, H3T 1J4*

[b] *Department of Computer Science, Université de Montréal, Montréal, Canada, Montréal (Qc), Canada, H3T 1J4*

[c] *Canada excellence research chair in data science for real-time decision making, Montréal (Qc), Canada, H3C 3A7*

[d] *Department of Mathematics and Industrial Engineering, Polytechnique Montréal, Montréal (Qc), Canada, H3C 3A7*

[e] *Jacobs Technion-Cornell Institute, Cornell Tech and Technion - IIT, New York, NY, 10044, USA*

lea.ricard@umontreal.ca
guy.desaulniers@gerad.ca
andrea.lodi@polymtl.ca

July 2022
Les Cahiers du GERAD
G–2022–30

Abstract : The multi-depot scheduling problem (MDVSP) is one of the most studied problem in public transport service planning. It consists of assigning buses to each timetabled trip while respecting vehicle availability at each depot. Although service quality, and especially reliability, is the core of most transport agencies, the MDVSP is more often than not solved solely in a cost-efficient way. This work introduces a data-driven model to the reliable MDVSP with stochastic travel time (R-MDVSP-STT). The reliability of a schedule is assessed and accounted for by propagating delays using the probability mass function of the travel time of each timetabled trip. We propose a heuristic branch-and-price algorithm to solve this problem and a labeling algorithm with stochastic dominance criterion for the associated subproblems. The solutions obtained are compared based on three passenger-oriented metrics - under normal and extraordinary circumstances. Computational results on real-life instances show that our method can efficiently find good trade-offs between operational costs and reliability, improving the reliability of the solutions with little cost increase.

Keywords : Vehicle scheduling, reliability, cColumn generation, stochastic programming, delay propagation

Acknowledgements: We are thankful to Charles Fleurent and his team at GIRO Inc. for introducing this problem to us, providing the data, and discussing our progress throughout the project. This project was partially supported by GIRO Inc. and the Natural Sciences and Engineering Research Council of Canada under the grant RDCPJ 520349–17. Léa Ricard gratefully acknowledges the financial support of the Natural Sciences and Engineering Research Council of Canada under the grant BESC D3–558645–2021.

1 Introduction

One of the specificities of public transport is its social mission; most transit agencies aim at 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 that 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 in 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 location, an itinerary composed of a sequence of stops, and an end time and 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 on the day of operation because they typically do not contain much buffer time (Amberg et al., 2011). Indeed, disruptions (e.g., heavy traffic) 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 (Naumann et al., 2011; Kramkowski et al., 2009; Amberg et al., 2011, 2019). Primary delays are a measure of the additional time required to complete a trip due to disruptions 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 disruptions (and thus 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.

In this work, we approach the MDVSP from a stochastic perspective in an attempt to increase the reliability of vehicle schedules. We propose a method that is based on the discretized probability density functions of the travel time to compute the convolution of the probability mass function of a timetabled trip's actual departure time, a value used directly to measure its secondary delay. The best probabilistic model to predict the travel time distributions, 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 main contributions of this paper are (i) we formulate a data-driven model to the reliable MDVSP with stochastic travel time (R-MDVSP-STT) which 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. Furthermore, our model takes into account passengers' perspective, a point of view too often forgotten (Oort, 2011), by putting more weight on timetabled trips with a high ridership in the cost function. To our knowledge, this is the first work that incorporates passenger flow information when increasing schedule reliability in the VSP and MDVSP.

The remainder of this paper is organized as follows. In Section 2, we discuss related works on MDVSP and methods for addressing MDVSP reliability. The R-MDVSP-STT model and the derivation of the probability mass function of the secondary delay of a trip covered by a given vehicle schedule is detailed in Section 3. Section 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 5 with a Monte Carlo simulation to compute them. Section 6 evaluates the heuristic performance of the R-MDVSP 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 7 summarizes our findings.

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 to 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 generates dynamically 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 less 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 solver. We refer interested readers to Bunte and Kliewer (2010) and Desaulniers and Hickman (2007) for a detailed review on the VSP and the MDVSP.

Methods found in the literature to address the timeliness of buses when forming bus schedules can be divided in two families: online (or real-time) and offline methods. The focus of this work is on the second family 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.

One of the simplest methods to improve the on-time performance offline is to accurately set travel times (Shen et al., 2016; Salicrú et al., 2011; Furth et al., 2006). With the recent advent of Automatic Vehicle Location (or AVL) systems, many agencies empirically estimate travel times using historical data. The planned travel time can be set, for example, to the mean observed travel time (Ryus et al., 2013) or the mean travel time minus 2 minutes (Salicrú et al., 2011). Another simple method widely used in industry consists in imposing (hard or soft constraints) a mandatory buffer time after each timetabled trip. Amberg et al. (2011), Amberg et al. (2019), Naumann et al. (2011), and van Kooten Niekerk (2018) compared the results of their models for robust VSP to the ones obtained by imposing mandatory buffer times (either a global buffer time for all timetabled trips or buffer times that depend on the primary delay scenarios of each timetabled trip).

Several works tackled the reliability of vehicle schedules while solving the (MD)VSP. First, in Kramkowski et al. (2009) an offline heuristic method to increase the reliability of VSP solutions is developped. From cost-optimal schedules, their simulated annealing heuristic searches for valid neighboring schedules with lower delay costs. Second, Amberg et al. (2011) introduced a few years later in a short proceeding paper a flow decomposition method for the VSP that wisely decomposes the cost-optimal flows of a time-space network in order to distribute buffer times and allow delay absorption. This decomposition method is paired with mandatory buffer time rules to find good trade-offs between operational costs and schedule reliability. The results of the works of Kramkowski et al. (2009) and Amberg et al. (2011) showed that vehicle schedules with lower expected secondary delays can be achieved with virtually no increase in operational costs. Third, Naumann et al. (2011) proposed a stochastic programming framework to address schedule reliability while solving the VSP. They modeled the problem using a multicommodity formulation with a slightly modified time-space network that enables to consider only delays between two timetabled trips and not delay propagation. Fourth, Shen et al. (2016) proposed a two-phase heuristic for the VSP with stochastic travel time. In the first phase, a network flow formulation, where the cost of each arc is redefined based on the compatibility of pairs of timetabled trips, is solved and, in the second phase, an iterative greedy local search that seeks to improve the initial solution both in terms of operational costs and delay propagation is solved. Thus, delay propagation, that is computed based on the exact probability density functions of the departure and arrival times of a timetabled trip, is only considered in the second phase. The two-phase heuristic outputs vehicle schedules with better on-time performance than a deterministic model while using the same number of resources. Fifth, similarly to the first phase of the heuristic of Shen et al. (2016), van Kooten Niekerk (2018) proposed to add delay penalties to the cost of each arc of the network. These penalties account both for pairwise secondary delays and for conditional primary delays, but delay propagation between more than two trips is not considered. The model proposed by van Kooten Niekerk (2018) yielded experimental results that are 2 to 3% more punctual than the baseline approach of imposing minimum buffer times. Sixth, Amberg et al. (2019) expanded the work of Amberg et al. (2011) to include an integrated vehicle and crew scheduling formulation that aims at minimizing both the operational costs and delay propagation. They proposed an algorithm combining column generation and Lagrangian relaxation that accounts for delay propagation in vehicle and crew duties and uses the solution approach of Amberg et al. (2011) to find a good initial solution. Their computational results, that compared sequential, partially integrated and integrated solution schemes, showed that the integrated scheduling scheme provides the most reliable and cost-efficient schedules.

The models and algorithms proposed in the literature to tackle reliability in the VSP either do not consider potential delay propagation between all timetabled trips when computing the (initial) vehicle schedules or flows (when the problem is formulated based on a time-space network) (Naumann et al., 2011; van Kooten Niekerk, 2018; Shen et al., 2016) or consider biased potential delay propagation (Amberg et al., 2019). Indeed, the secondary delay of a timetabled trip is defined as a function of the expected primary delay of the previous timetabled trip in Amberg et al. (2019) and we argued in a previous work that this leads to erroneous approximations of the expected secondary delays (Ricard et al., 2022). Complete probability distributions of the primary delay (or travel time) must be considered to obtain an accurate measure of the expected secondary delays. This work aims to fill the aforementioned shortcomings.

3 Mathematical model

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

3.1 The reliable MDVSP with stochastic travel times

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 ridership for all trips $v \in \mathcal{V}$, the R-MDVSP-STT consists of finding feasible vehicle schedules over a one-day horizon that cover exactly every trip 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 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) that has the effect of balancing the two objectives of the R-MDVSP-STT. Thus, the R-MDVSP-STT is reformulated as a single-objective optimization problem as further discussed in Section 3.2. For now, let us define the total cost c_s of vehicle schedule s as the weighted sum of both objective values, namely the operational costs and the costs for delays.

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 v' and $v \in \mathcal{V}$. If trip v' ends at the same location as the starting location of trip v, the vehicle wait at the terminal or at the nearest depot. Otherwise, the vehicle performs a deadhead trip (i.e., a trip without passengers) from the end location of trip v' to the starting location of trip v. 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 $e_{v',v}$ be the deadhead travel time between the end location of trip v' and the starting location of trip v and let τ be the minimum layover time between two trips. Two timetabled trips are deemed compatible if $d^1_{v'} + e_{v',v} + \tau \leq d^0_v$, where d^0_v and $d^1_{v'}$ are the departure and the arrival time of trips v and v', 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 a depot 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 $v \in \mathcal{V}$, let y_s be a binary variable that takes value 1 if vehicle schedule s is used in the solution and $a_{v,s}$ be a binary parameter equal to 1 if schedule s covers trip v.

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

$$\min \quad \sum_{s \in \mathcal{S}} c_s y_s \tag{1}$$

$$\text{s.t.} \quad \sum_{s \in \mathcal{S}} a_{vs} y_s = 1, \qquad \forall v \in \mathcal{V} \tag{2}$$

$$\sum_{s \in \mathcal{S}^d} y_s \leq b_d, \qquad \forall d \in \mathcal{D} \tag{3}$$

$$y_s \in \{0,1\}, \qquad \forall s \in \mathcal{S}. \tag{4}$$

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 coefficient $c_s, s \in \mathcal{S}$, which takes into account the risks of delay propagation in our model. The objective function (1) minimizes the total cost of the selected vehicle schedules, while constraints (2) ensure that each timetabled trip is covered exactly once by a schedule and constraints (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, 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 introduced first 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, v) for $v \in \mathcal{V}$), pull-in arcs (i.e., (v, n_1^d) for $v \in \mathcal{V}$), and arcs between pairs of compatible timetabled trips (i.e., (v', v) for $v', v \in \mathcal{V}$). Since we consider the travel time of timetabled trips to be stochastic, we could also use an enlarged network where additional arcs are present between incompatible trips, as long as the incompatibility is not larger than a few minutes. However, these connections are not considered because they would imply positive secondary delays on average, which is not welcome 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

$$c_s = \sum_{(i,j) \in A(s)} c_{ij}^s, \tag{5}$$

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 1. In this table, η is the cost per vehicle used, r^T is the cost per minute of travel, $e_{d,v}$ and $e_{v,d}$ are the deadhead travel times between the depot d and the starting location of trip v and the end location of trip v and the depot d, respectively, r^W is the cost per minute of waiting outside a depot, β is a weighing factor, and q_v^s is the cost associated with potential delays in trip v when it is covered by vehicle schedule s. The latter cost q_v^s depends on the schedule s and its value will be discussed in Section 3.2.

Table 1: Cost of the arcs $(i,j) \in A(s)$

Arc (i,j)	Type	c_{ij}^s
(n_o^d, v)	pull-out	$\eta + r^T e_{d,v}$
(v', v)	connection	$r^T e_{v',v} + r^W(d_v^0 - d_{v'}^1) + \beta q_v^s$
(v, n_1^d)	pull-in	$r^T e_{v,d}$

To avoid excessive congestion at the terminals and long idle time for the drivers, we impose a threshold on the waiting time $d_v^0 - d_{v'}^1$ between two trips v' and $v \in \mathcal{V}$. If the waiting time exceeds this threshold, the vehicle must move to the nearest depot after completing trip v', wait at the depot, and then move to the starting point of trip v. 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.

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 $v \in \mathcal{V}$ and $s \in \mathcal{S}$, we define the cost for delays as $q_v^s = \alpha_v \mathbb{E}(X_v^s)$, where α_v is the proportion of passengers boarding trip v such that $\sum_{v \in \mathcal{V}} \alpha_v = 1$, X_v^s is a random variable representing the secondary delay of trip v when it is covered by schedule s, and $\mathbb{E}(X_v^s)$ its expectation.

To be able to compute the expected secondary delay of each trip $v \in \mathcal{V}$ covered by 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 the travel time based on its attributes (e.g., day of the week, distance, scheduled departure time, etc.). The models are trained and tested on a 2-month dataset of 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_v^s for all $v \in s$ 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 of the main sources 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_v^s for each $v \in \mathcal{V}$ and $s \in \mathcal{S}$.

First, for each trip $v \in \mathcal{V}$, let $\dot{T}_v$ be a random variable representing its actual travel time and let $\dot{h}_v(t) = Pr(\dot{T}_v = t)$ be the probability density function of $\dot{T}_v$. The actual travel time $\dot{T}_v$ for all $v \in \mathcal{V}$ 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 $\dot{h}_v(t)$, for all $v \in \mathcal{V}$, stays the same regardless of the bus schedule covering it.

In order to obtain a finite number of possible outcomes, we transpose $\dot{T}_v$, for all $v \in \mathcal{V}$, 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 $v \in \mathcal{V}$, we truncate the distribution of $\dot{T}_v$ below its 5th percentile and over its 95th percentile. Let $\dot{t}_v^5$ and $\dot{t}_v^{95}$ be the value of $\dot{T}_v$ at the 5th and 95th percentile of its probability distribution, respectively. For each $v \in \mathcal{V}$, let T_v be defined over Φ_v as

$$\Phi_v = \{\lfloor \dot{t}_v^5 \rfloor, \lfloor \dot{t}_v^5 \rfloor + 1, ..., \lfloor \dot{t}_v^{95} \rfloor - 1, \lfloor \dot{t}_v^{95} \rfloor\}, \tag{6}$$

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

Second, based on the above probability mass functions and considering that d_v^0, for all $v \in \mathcal{V}$, $e_{v,v'}$, for all $v, v' \in \mathcal{V}$, 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_v^s for all $v \in s$ as follows. For each $v \in s$, let Y_v^s be a random variable representing the actual departure time of trip v and let $f_v^s(y) = Pr(Y_v^s = y)$ be the probability mass function of Y_v^s. We have developed an exact procedure inspired by the works of Errico et al. (2018) and Shen et al. (2016) to recursively compute $f_v^s(y)$. 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 $v \in s \backslash \{v_0^s\}$, three cases are distinguished: when the bus starts trip v on time (i.e., $y = d_v^0$), late (i.e., $y > d_v^0$), and early (i.e., $y < d_v^0$). The last case has zero probability because we assume a bus cannot start ahead of time. The distribution $f_v^s(y)$ of a trip $v \in s \backslash \{v_0^s\}$ preceded by trip $v' \in s$ can be computed as

$$f_v^s(y) = Pr(Y_v^s = y) = \begin{cases} \sum_{k \in \Phi_{v'}} h_{v'}(k) \times \sum_{y'=d_{v'}^0}^{d_v^0 - e_{v',v} - \tau - k} f_{v'}^s(y'), & \text{if } y = d_v^0; \\ \sum_{k \in \Phi_{v'}} h_{v'}(k) \times f_{v'}^s(y - e_{v',v} - \tau - k), & \text{if } y > d_v^0; \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Let $g_v^s(x) = Pr(X_v^s = x) = Pr(Y_v^s = d_v^0 + x) = f_v^s(d_v^0 + x)$ be the probability mass function of X_v^s. Its expectation is given by

$$\mathbb{E}(X_v^s) = \sum_{x=0}^{\zeta_v^s} x \times g_v^s(x) = \sum_{x=0}^{\zeta_v^s} x \times f_v^s(x + d_v^0), \tag{8}$$

where v' is the trip that precedes the trip v in schedule s and $\zeta_v^s = d_{v'}^0 + \zeta_{v'}^s + e_{v',v} + \tau + \lfloor \dot{t}_{v'}^{95} \rfloor - d_v^0$ is the maximum secondary delay of trip v.

Note that, by definition, Y_v^s and X_v^s for all $v \in \mathcal{V}$ depend on the travel times of the previous timetabled trips in s so their uncertainty is endogenous to resource allocation. In other words, the random variables Y_v^s and X_v^s for $s \in \mathcal{S}$ and $v \in \mathcal{V}$ are likely to have different probability distributions than $Y_v^{s'}$ and $X_v^{s'}$ for $s \neq s' \in \mathcal{S}$. Hence, it is not possible to compute Y_v^s and X_v^s for all $v \in \mathcal{V}$ 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 schedule, the convolution of the probability mass function of the secondary delay of every timetabled trip in a vehicle 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 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 (1)–(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 (1)–(4). The column generation algorithm as well as these two acceleration strategies are detailed in the following.

4.1 Column generation

Column generation is an iterative algorithm that generates variables (columns) as needed. In this context, the linear relaxation of (1)–(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 (Boland et al., 2015; Wellman et al., 2013). If no columns with a negative reduced cost are identified, the algorithm stops and the current RMP solution is guarenteed 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.

4.1.1 Shortest path problem with stochasticity

The pricing problem for depot d is defined on the acyclic network G^d (see Section 3.1) with modified arc costs as detailed next. Let $(u_v)_{v\in\mathcal{V}}$ and $(\pi_d)_{d\in\mathcal{D}}$ be the dual variables associated with constraints (2) and (3), respectively. The reduced cost $\tilde{c}_s$ of a vehicle schedule $s \in \mathcal{S}^d$ housed in depot d is given by

$$\tilde{c}_s = c_s - \sum_{v\in\mathcal{V}} a_{vs}u_v - \pi_d, \tag{9}$$

with the following cost breakdown per arc

$$\tilde{c}^s_{ij} = \begin{cases} c^s_{ij} - \pi_d, & \text{if } i = n^d_0; \\ c^s_{ij} - u_v, & \text{if } i = v \in \mathcal{V}. \end{cases} \tag{10}$$

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. We would like to put the emphasis on the dependence assumption which is crucial because, as explained by Wellman et al. (2013), if the arc costs are stochastic but *independent* (i.e., if the probability mass function of the secondary delay at each node does 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 Boland et al. (2015) and Wellman et al. (2013) 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 on 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 $v \in \mathcal{V}$ contained in p (written as $v \in p$ afterwards), let $F^p_v(z)$ be the cumulative distribution function of Y^p_v defined as

$$F^p_v(z) = \sum_{y=d^0_v}^{z} f^p_v(y). \tag{11}$$

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

$$L_v^p = (F_v^p(d_v^0), ..., F_v^p(d_v^0 + \zeta_v^p), C^p), \tag{12}$$

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

Extension functions. Consider the extension of a label $L_{v'}^{p'} = (F_{v'}^{p'}(d_{v'}^0), ..., F_{v'}^{p'}(d_{v'}^0 + \zeta_{v'}^{p'}), C^{p'})$ associated with node v' along the arc (v', v) to create a new label L_v^p at node v. First, the ζ_v^p components of type $F_v^p(\cdot)$ are derived using equation (7) as detailed in Appendix A. This derivation results in

$$F_v^p(z) = \sum_{k \in \Phi_{v'}} h_{v'}(k) \times F_{v'}^{p'}(z - e_{v',v} - \tau - k). \tag{13}$$

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 (10) for the cost breakdown per arc) as

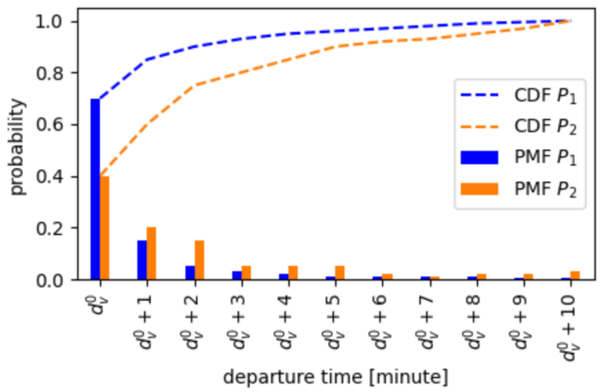
$$C_j^p = C_i^{p'} + \tilde{c}_{i,j}^p, \tag{14}$$

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

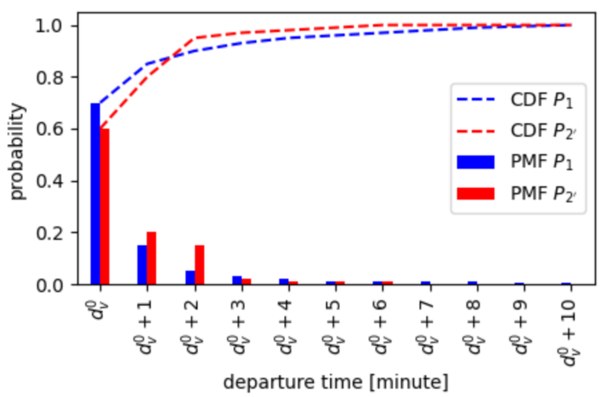
Stochastic dominance rule. Consider two paths p_1 and p_2 in G^d both ending at node $v \in \mathcal{V}$ (that is, v is the resident node of p_1 and p_2) with labels $L_v^{p_1} = (F_v^{p_1}(d_v^0), ..., F_v^{p_1}(d_v^0 + \zeta_v^{p_1}), C_v^{p_1})$ and $L_v^{p_2} = (F_v^{p_2}(d_v^0), ..., F_v^{p_2}(d_v^0 + \zeta_v^{p_2}), C_v^{p_2})$, respectively. The path p_1 dominates p_2 (and therefore p_2 can be discarded) when the following two conditions hold:

1. $C_v^{p_1} \leq C_v^{p_2}$,
2. $F_v^{p_1}(z) \geq F_v^{p_2}(z)$, for all $z \in \{d_v^0, d_v^0 + 1, \ldots, d_v^0 + \max\{\zeta_v^{p_1}, \zeta_v^{p_2}\}\}$.

The first condition is straightforward, but we will explain in further details 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 $\mathbb{E}(X_v^{p_1}) \leq \mathbb{E}(X_v^{p_2})$, it does not necessarily imply if we extend p_1 and p_2 with node $\bar{v} \in \mathcal{V}$ that $\mathbb{E}(X_{\bar{v}}^{p_1 \oplus \bar{v}}) \leq \mathbb{E}(X_{\bar{v}}^{p_2 \oplus \bar{v}})$, where $p_i \oplus \bar{v}, i = 1, 2$, denotes the path resulting from appending node $\bar{v}$ to path p_i. Thus, a stochastic dominance condition based upon the cumulative distribution function of the actual departure time is used instead. When $F_v^{p_1}(z) \geq F_v^{p_2}(z)$ for some $z \geq d_v^0$, it means that $Pr(Y_v^{p_1} > z) \leq Pr(Y_v^{p_2} > z)$ (i.e., path p_1 is less likely to start trip v 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 $z \in \{d_v^0, d_v^0 + 1, \ldots, d_v^0 + \max\{\zeta_v^{p_1}, \zeta_v^{p_2}\}\}$ (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 1a. This figure and Figure 1b display the probability mass functions (PMF) and the cumulative distribution functions (CDF) of the actual departure time of the resident node $v \in \mathcal{V}$ 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 $z \in \{d_v^0, d_v^0+1, \ldots, d_v^0+\max\{\zeta_v^{p_1}, \zeta_v^{p_2}\}\}$, as in the example in Figure 1b, it is not clear if p_1 or p_2 is more reliable. For example, $Pr(Y_v^{p_1} > d_v^0+1) < Pr(Y_v^{p_2} > d_v^0+1)$, but $Pr(Y_v^{p_1} > d_v^0+2) > Pr(Y_v^{p_2} > d_v^0+2)$. Thus, both paths are kept.



(a) Second condition holds



(b) Second condition does not hold

Figure 1: Examples of the second dominance condition applied to the trip v of paths p_1 and p_2

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 (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.

4.3 Constraint perturbation

Problems like (1)–(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_v^+ and η_v^- that allow limited under- and over-covering of trip v for all $v \in \mathcal{V}$, respectively. The perturbed MP is defined as

$$\min \quad \sum_{s\in\mathcal{S}} c_s y_s + \sum_{v\in\mathcal{V}} (\delta_v^+ \eta_v^+ + \delta_v^- \eta_v^-) \tag{15}$$

$$\text{s.t.} \quad \sum_{s\in\mathcal{S}} a_{vs} y_s + \eta_v^+ - \eta_v^- = 1, \qquad \forall v \in \mathcal{V} \tag{16}$$

$$\sum_{s\in\mathcal{S}^d} y_s \le b_d, \qquad \forall d \in \mathcal{D} \tag{17}$$

$$0 \le \eta_v^+ \le \xi_v^+, \qquad \forall v \in \mathcal{V} \tag{18}$$

$$0 \le \eta_v^- \le \xi_v^-, \qquad \forall v \in \mathcal{V} \tag{19}$$

$$0 \le y_s \le 1, \qquad \forall s \in \mathcal{S}, \tag{20}$$

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

5 Assessing schedule reliability

To address delay propagation, the vehicle schedules selected in the R-MDVSP-STT are such that they usually contain buffers allowing the absorption of 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 of the cost of connection arcs (see Table 1). After solving several times the R-MDVSP-STT while adjusting the factor β, it is possible to compare the solutions obtained more carefully. In this respect, we define next three reliability metrics that decision-makers can use to compare thoroughly 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.

5.1 Reliability metrics

The solutions of the R-MDVSP-STT 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 after a secondary delay. These three metrics are computed as follows.

Expected secondary delay per passenger (γ):

$$\gamma = \sum_{s\in\mathcal{S}^*} \sum_{v\in\mathcal{V}} a_{vs} \alpha_v \mathbb{E}(X_v^s), \tag{21}$$

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

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

$$\psi = \sum_{s \in \mathcal{S}^*} \sum_{v \in \mathcal{V}} a_{vs} \alpha_v Pr(X_v^s > \epsilon), \tag{22}$$

where ϵ is a grace period (i.e., if the secondary delay of a trip $v \in \mathcal{V}$ 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 after a secondary delay (θ):

$$\theta = \frac{1}{|\mathcal{S}^*|} \sum_{s \in \mathcal{S}*} \bar{\Omega}_s, \tag{23}$$

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 on schedule s at iteration k, respectively, are used. To better understand how these counters work, let us consider the toy example presented in Figure 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.

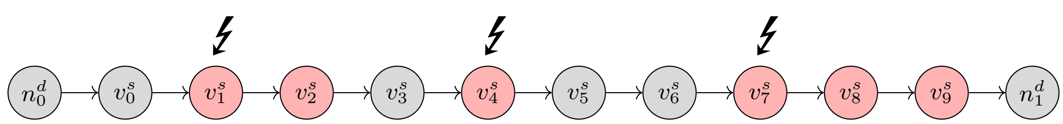


Figure 2: Example of the potential delay propagation in schedule $s = \{v_0^s, v_1^s, \ldots, v_9^s\}$ of 10 timetabled trips at iteration k. Delayed trips are highlighted in red

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. Thus, these metrics are user-oriented, a point of view often overlooked in the evaluation of public transportation services (Oort, 2011). The third metric is more agency-oriented as it provides information useful to assess 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 (1)–(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.

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 1 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 9, a travel time t_v^k is randomly sampled for each $v \in s$ except for the last timetabled trip performed by vehicle schedule s from $\dot{h}_v(t)$. Note that, in the simulation, we use the travel time probability density functions $\dot{h}_v(t)$ instead of the probability mass functions $h_v(t)$ like in the R-MDVSP-STT because the probability density functions offer more complete information. Then, in Steps 6 and 10, the value of $y_v^{s,k}$, the actual departure time of trip v covered by schedule s at iteration k, is recursively computed for all $v \in s$, from the first timetabled trip of the schedule to the last timetabled trip, as

$$y_{v_0^s}^{s,k} = d_{v_0^s}^0 \tag{24}$$

$$y_v^{s,k} = \max\{y_{v'}^{s,k} + t_{v'}^k + e_{v',v} + \tau, d_v^o\}, \tag{25}$$

where, for each $v \in \mathcal{V}$, v' is the trip preceding v in s and v_0^s is the first timetabled trip of s. In Step 11, the secondary delay $x_v^{s,k}$ of trip v covered by schedule s at iteration k is computed for all $v \in s$ as

$$x_v^{s,k} = y_v^{s,k} - d_v^0. \tag{26}$$

Algorithm 1: 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         $v' \leftarrow v_0^s$
8   **for** $v \in s \setminus \{v_0^s\}$ **do**
9   Randomly generate $t_{v'}^k$ from $\dot{h}_{v'}(t)$ or $\bar{h}_{v'}(t)$
10             $y_v^{s,k} \leftarrow \max\{y_{v'}^{s,k} + t_{v'}^k + l_{v',v}, d_v^0\}$
11             $x_v^{s,k} \leftarrow y_v^{s,k} - d_v^0$
12             $\hat{\gamma} \leftarrow \hat{\gamma} + (\alpha_v \times x_v^{s,k})$
13   **if** $x_v^{s,k} > \epsilon$ **then**
                // delayed timetabled trip
14                 $\hat{\psi} \leftarrow \hat{\psi} + \alpha_v$
15   **if** $x_{v'}^{s,k} \leq \epsilon$ **then**
16                     $\varphi^{s,k} \leftarrow \varphi^{s,k} + 1$
17   **else**
18                     $\rho^{s,k} \leftarrow \rho^{s,k} + 1$
19   **end**
20   **end**
21             $v' \leftarrow v$
22   **end**
23         $\hat{\theta} \leftarrow \hat{\theta} + \frac{\rho^{s,k}}{\varphi^{s,k}}$
24   **end**
25 **end**
26 $\hat{\gamma} \leftarrow \hat{\gamma}/K$
27 $\hat{\psi} \leftarrow \hat{\psi}/K$
28 $\hat{\theta} \leftarrow \hat{\theta}/(K \times |\mathcal{S}^*|)$

Finally, the approximations of the three metrics are computed in Steps 12-28. When a trip $v \in s$ preceded by trip v' is delayed at iteration k (i.e., when it has a secondary delay larger than ϵ) and when the trip v' was not delayed, the counter $\varphi^{s,k}$ is incremented by one (see Step 16). Otherwise, the counter $\rho^{s,k}$ is incremented by one (see Step 18).

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. In this case, travel times are randomly sampled from $\bar{h}_v(t)$, a distribution of the longest travel times of $\dot{h}_v(t)$.

For each trip $v \in \mathcal{V}$, let $\bar{T}_v$ be defined over $\bar{\Phi}_v$ such as

$$\bar{\Phi}_v = [\dot{t}_v^{75}, \dot{t}_v^{95}] \tag{27}$$

where $\dot{t}_v^{75}$ and $\dot{t}_v^{95}$ are the values of $\dot{T}_v$ at the 75th and 95th percentile of its probability distribution, respectively. Let $\bar{h}_v(t)$ be the probability density function of $\bar{T}_v$, where the density of $Pr(\dot{T}_v < \dot{t}_v^{75})$ and $Pr(\dot{T}_i > \dot{t}_v^{95})$ are uniformly redistributed to $Pr(\dot{t}_v^{75} \leq \bar{T}_i \leq \dot{t}_v^{95})$ in order to obtain a proper probability distribution (i.e., $\int_{\dot{t}_v^{75}}^{\dot{t}_v^{95}} \bar{h}_v(t)dt = 1$).

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 Montreal bus network and provided by our industrial partner. The probability density functions of the travel time of each timetabled trip in these instances have been computed as in Ricard et al. (2022) as detailed in Section 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 2.

Table 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 B, the cost per minute of travel p^T (either deadhead or pull-in/pull-out trips), the cost per minute of waiting time outside a depot p^W, and the grace period ϵ are set to 1,000, 0.4, 0.2 and 3 minutes, respectively. Furthermore, we enforce that vehicles cannot wait outside a depot more than 45 minutes. If the waiting time between two consecutive timetabled trips is longer than this threshold, the vehicle must perform a pull-in and a pull-out travel to the nearest depot and the cost of the arc in the modified connection network presented in Section 3.1 is adjusted accordingly. The penalties for under- and over-covering trip $v \in \mathcal{V}$, δ_v^+ and δ_v^-, are set to 1 for every trip $v \in \mathcal{V}$ and the upper bounds ξ_v^+ and ξ_v^- are randomly selected in $[0, 0.1]$ for every trip $v \in \mathcal{V}$.

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 scheduling 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.

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 scheduling costs and reliability of MDVSP solutions.

Table 3 reports the upper bound (UB) and lower bound (LB) obtained, the relative difference in percentage between UB and 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).

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.

Table 3: Heuristic performance without considering reliability (MDVSP)

Instance	UB	LB	Gap (%)	# Nodes	CPU time (seconds)		
					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

The trade-offs between operational costs and reliability for the instances I1–I3 solved without considering reliability are presented in Table 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 5.

Table 4: Scheduling 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

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 of timetabled trips needed to recover after a secondary delay of 2.08 trips. These values are not negligible, especially for the last metric if we consider that each bus performs an average of 15 timetabled trip per day, but were expected because no buffer time nor stochastic optimization is considered.

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 the 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 $B/10$ to $10B$, where $B = 1,000$ is the cost per vehicle used, have been tested.

Table 5 provides the results of our algorithm on instances I1–I3 for different β values. The upper bounds (UB) and lower bounds (LB) on the optimal values include the costs related to delays, whereas it is not the case in Table 3 because $\beta = 0$. What can be observed is 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. However, the proportion of the time dedicated to the pricing problems increase. For example, 47% of the computing time is spent on the princing problems for the MDVSP and it increases to 77% to 86% for the R-MDVSP-STT.

Table 5: Results considering reliability (R-MDVSP-STT)

						CPU time (seconds)		
Instance	β	UB	LB	Gap (%)	# Nodes	Total	Root	Pricing
I1	B/10	78,916.8	78,912.3	0.01	42	2,820.7	1,504.3	2,205.3
	B/5	78,962.0	78,956.8	0.01	44	2,630.0	1,468.0	2,071.7
	B/2	79,062.1	79,057.6	0.01	43	2,803.7	1,507.5	2,174.0
	B	79,183.8	79,179.2	0.01	43	2,903.8	1,474.0	2,240.4
	2B	79,359.4	79,354.9	0.01	43	3,066.6	1,524.3	2,310.0
	4B	79,615.2	79,610.7	0.01	43	3,364.1	1,552.9	2,495.6
	8B	80,039.4	79,997.7	0.05	133	6,214.5	1,617.3	4,561.5
	10B	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	B/10	136,114.0	136,107.1	0.01	76	10,979.1	6,601.6	9,423.7
	B/5	136,275.0	136,267.3	0.01	95	11,601.2	6,494.8	9,868.2
	B/2	136,605.0	136,595.5	0.01	94	11,174.1	6,221.3	9,445.9
	B	136,966.0	136,955.6	0.01	106	10,314.0	5,473.6	8,560.3
	2B	137,452.0	137,442.9	0.01	92	10,652.6	6,175.8	8,499.1
	4B	138,153.0	138,144.0	0.01	87	9,467.8	5,816.2	7,430.0
	8B	139,235.0	139,226.5	0.01	81	7,666.6	4,863.0	5,932.5
	10B	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	B/10	167,862.0	167,852.2	0.01	94	10,215.8	6,366.2	8,395.6
	B/5	167,977.0	167,967.8	0.01	111	10,599.1	5,986.0	8,586.8
	B/2	168,229.0	168,219.3	0.01	112	10,075.9	5,998.4	8,083.8
	B	168,504.0	168,493.7	0.01	128	10,957.0	5,786.1	8,602.5
	2B	168,832.0	168,822.9	0.01	100	10,064.7	6,558.3	7,771.3
	4B	169,212.0	169,201.8	0.01	107	9,978.4	6,129.7	7,460.4
	8B	169,654.0	169,644.4	0.01	92	8,045.2	5,341.3	5,963.6
	10B	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**

The trade-offs between operational costs and reliability of the vehicle schedules obtained with different values of β are provided in Table 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 contain 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). Also, the variable operational costs increased (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.

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. Significant reliability gains can be achieved without much deterioration in operational costs. For example, with $\beta = B/5$, the probability that a passenger board a delayed timetabled trip in instance I2 is reduced from 23% to 15% with a cost increase of 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 and 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 and 6). All the exceptions concern the third metric, θ, which is the metric the least related to the function minimized in model (1)–(4). Specifically, the value of θ (second case) is higher than the one of the corresponding MDVSP solution for instances I1 and I2 with $\beta = B/10$. 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 the classical approach to address bus schedules' reliability (i.e., MDVSP with minimum buffer time).

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

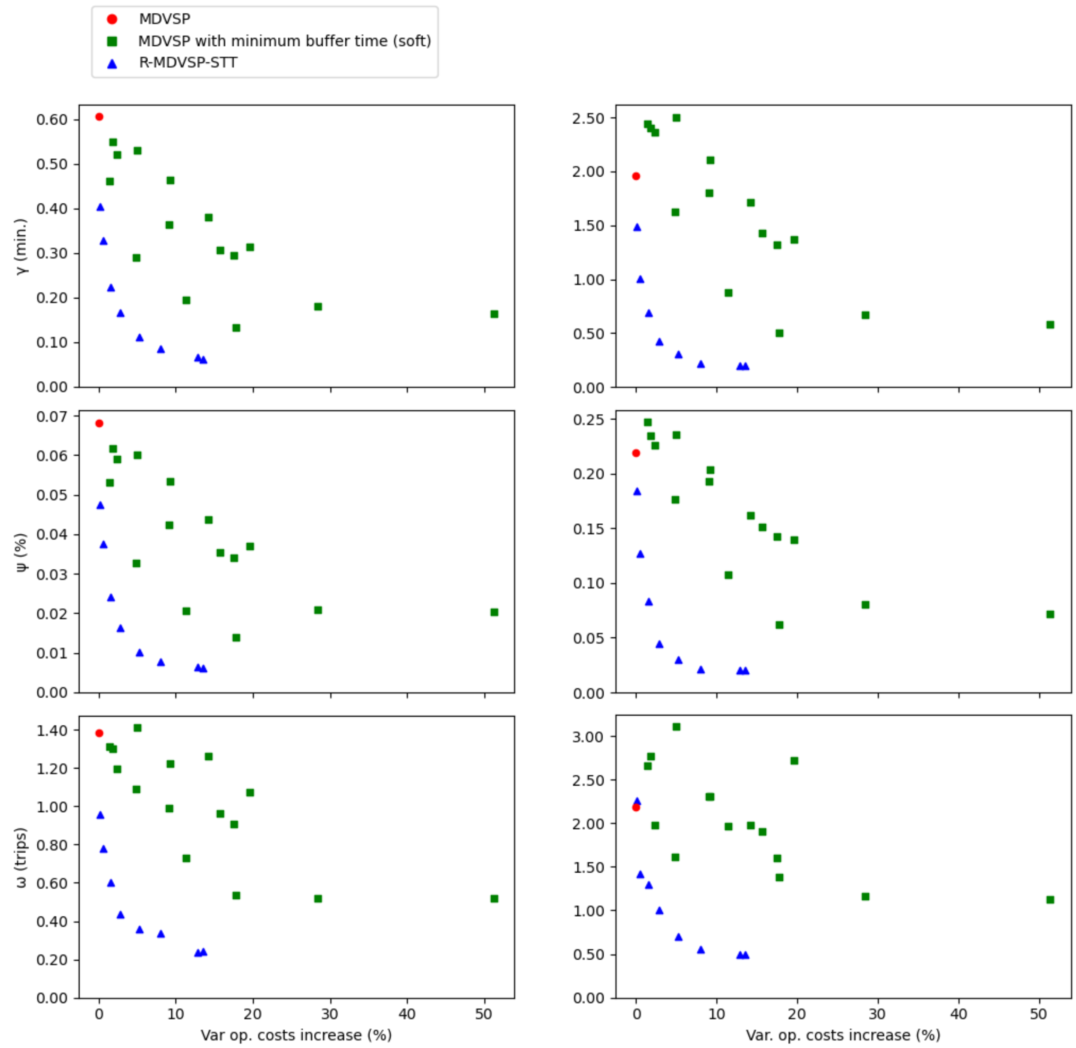
6.3 Comparison with the MDVSP with minimum buffer time

In Amberg et al. (2011), Amberg et al. (2019), Naumann et al. (2011), and van Kooten Niekerk (2018), models aimed at improving the reliability of vehicle schedules are compared to a simple approach: adding *hard* minimum buffer time constraints to the traditional MDVSP. However, this approach leads to poor quality solutions that use many additional vehicles. As in Amberg et al. (2011), Amberg et al. (2019), Naumann et al. (2011), and van Kooten Niekerk (2018), we have observed that all solutions found using the latter approach are very costly and most of them are largely dominated by the R-MDVSP-STT solutions (see Appendix B). 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), imposed after each timetabled trip;
- Buffer times to cover the primary delay of each timetabled trip x% of the time (50th, 75th, 90th and 95th percentiles of $\dot{T}_v$, for $v \in \mathcal{V}$);
- Buffer times provided by our industrial partner.

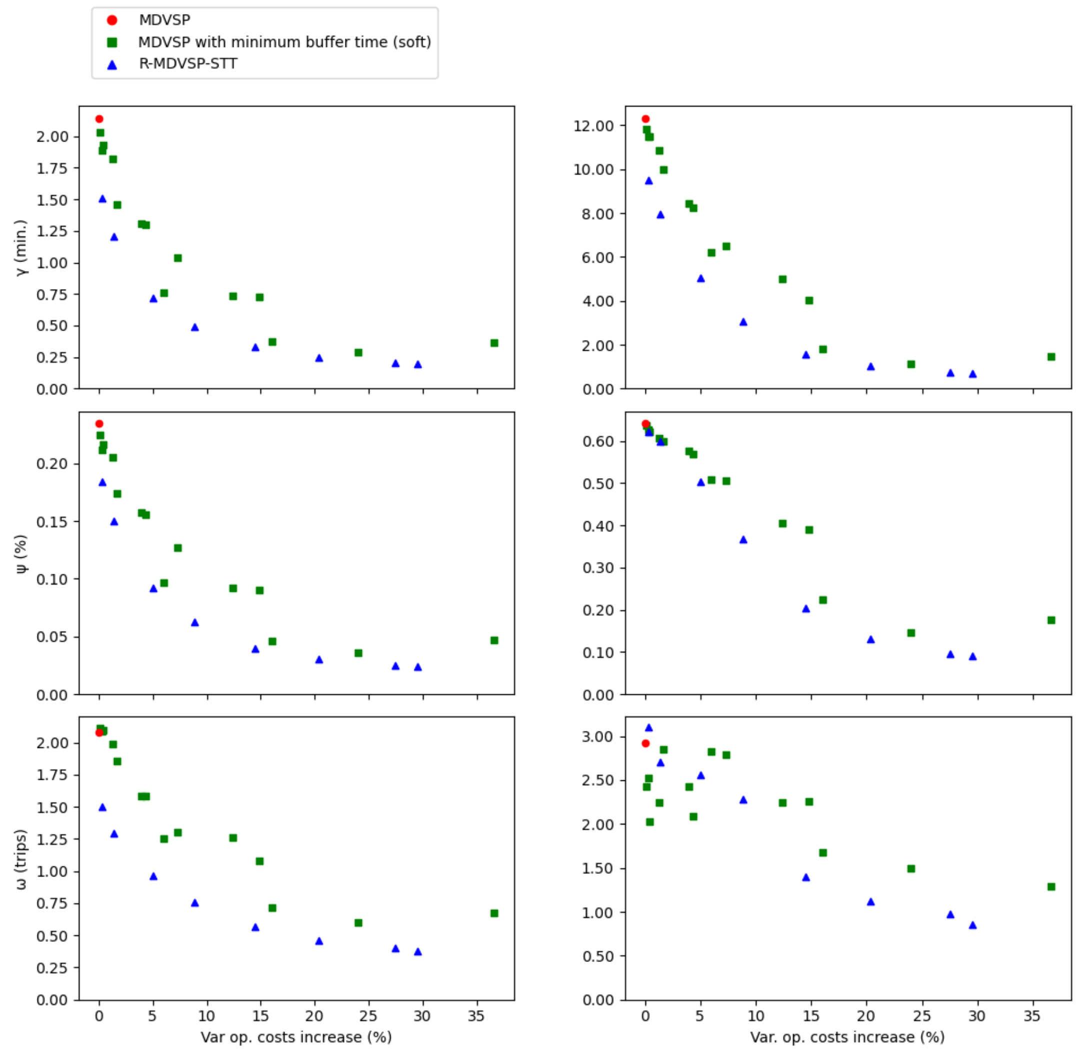
Figures 3, 4 and 5 (one figure per instance) show the relationship between the value of the reliability metrics and the increase in variable operational costs (relative to the corresponding baseline solution) of the MDVSP, MDVSP with soft minimum buffer time and R-MDVSP-STT solutions. The simulation results under normal conditions and under external and extraordinary factors are displayed in the left-hand and right-hand graphs, respectively.



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

Figure 3: Reliability metrics - I1

On the one hand, Figures 3a, 4a and 5a show the dominance of the solutions of the R-MDVSP-STT for all three metrics over those of the MDVSP with soft minimum buffer time under normal conditions. Thus, in our experiments and under normal circumstances, our model achieves better trade-offs between operational costs and reliability than the MDVSP with soft minimum buffer time, independently of the type of buffer time rule. On the other hand, Figures 3b, 4b and 5b show that all the solutions of the R-MDVSP-STT dominate, in terms of both metrics γ and ψ, those of the MDVSP with soft minimum buffer time for the second simulation case. However, some R-MDVSP-STT solutions do not offer the best trade-off between the third metric θ and the operational costs under external and extraordinary factors (i.e., they are dominated by at least one solution of the MDVSP with soft minimum buffer time). Yet, for both simulation cases, the solutions of the R-MDVSP-STT form approximate Pareto-fronts. This feature is of interest to transport agencies, as it allows them to easily adjust the reliability level of their vehicle schedules.



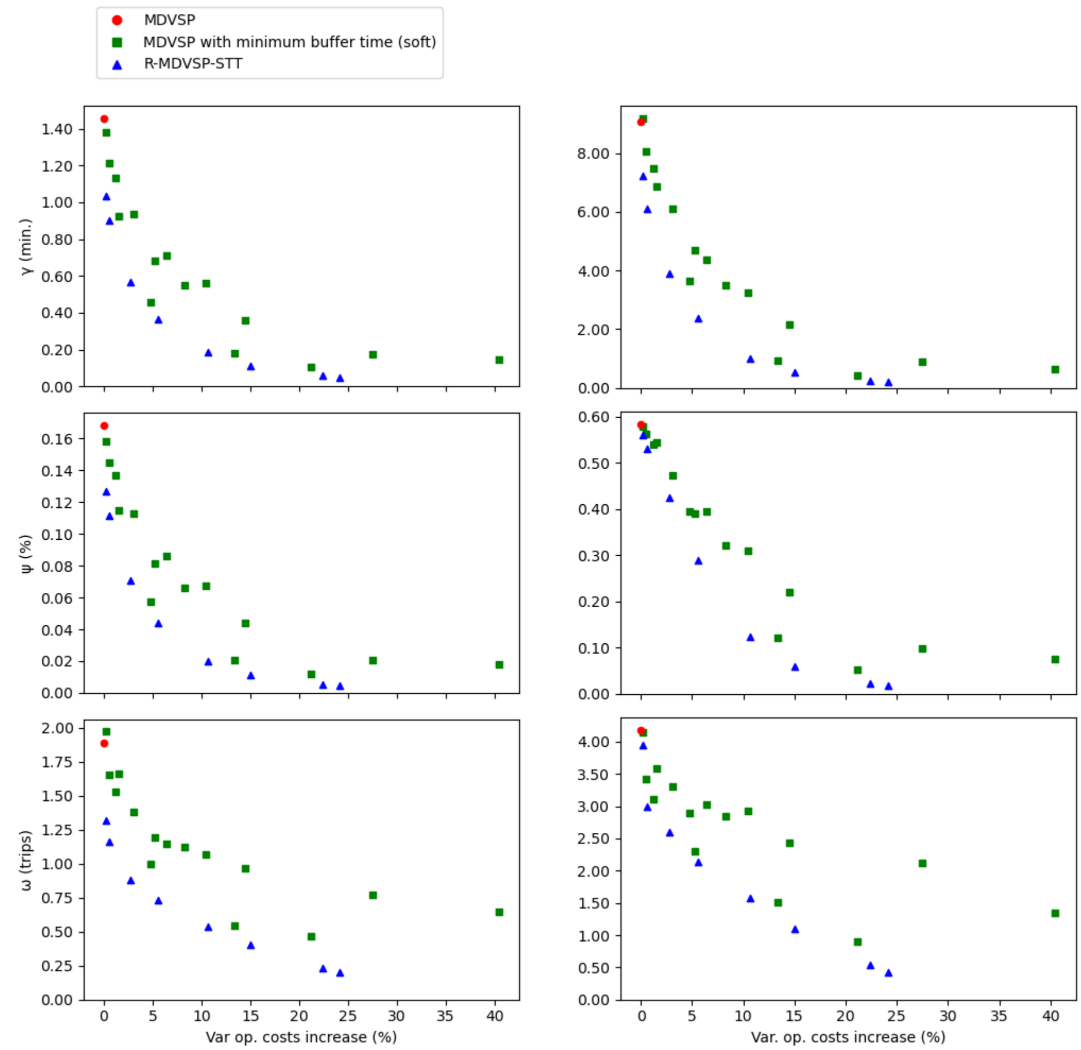
(a) 1st case **(b) 2nd case**

Figure 4: Reliability metrics - I2

7 Conclusions

In this work, we proposed a model for the R-MDVSP-STT that deals with stochastic travel times and 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).



(a) 1st case

(b) 2nd case

Figure 5: Reliability metrics - I3

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, our model provided better solutions than the traditional MDVSP with minimum buffer time (enforced by soft or hard constraints). Indeed, all solutions of the MDVSP with minimum buffer time are dominated by those of our model. In the presence of external and extraordinary factors, our model also 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, since the set of solutions thus obtained forms an approximate Pareto frontier.

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.

A Derivation of the cumulative distribution functions of the actual departure time

$$F_v^p(z) = \sum_{y=d_v^0}^{z} f_v^p(y) \tag{28}$$

$$= \left[\sum_{k\in\Phi_{v'}} h_{v'}(k) \times \sum_{y'=d_{v'}^0}^{d_v^0 - e_{v',v} - \tau - k} f_{v'}^p(y') \right] + \sum_{y=d_v^0+1}^{z} \left[\sum_{k\in\Phi_{v'}} h_{v'}(k) \times f_{v'}^p(y - e_{v',v} - \tau - k) \right] \tag{29}$$

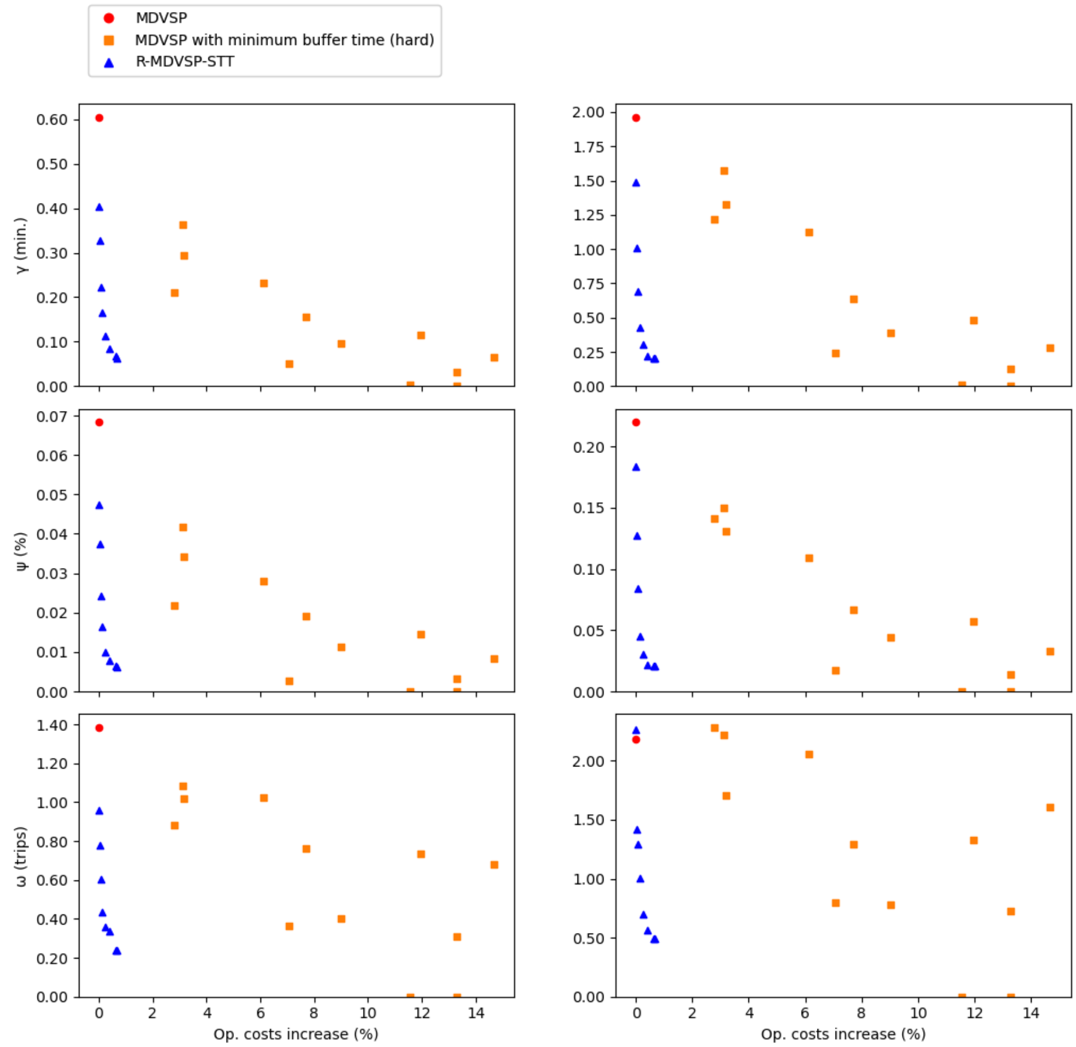
$$= \sum_{k\in\Phi_{v'}} h_{v'}(k) \left[\sum_{y'=d_{v'}^0}^{d_v^0 - e_{v',v} - \tau - k} f_{v'}^p(y') + \sum_{y=d_v^0+1}^{z} f_{v'}^p(y - e_{v',v} - \tau - k) \right] \tag{30}$$

$$= \sum_{k\in\Phi_{v'}} h_{v'}(k) \sum_{y'=d_{v'}^0}^{z - e_{v',v} - \tau - k} f_{v'}^p(y') \tag{31}$$

$$= \sum_{k\in\Phi_{v'}} h_{v'}(k) \times F_v^p(z - e_{v',v} - \tau - k) \tag{32}$$

The term on the right-hand side of (30) reduces to $\sum_{y'=d_{v'}^0}^{z-e_{v',v}-\tau-k} f_{v'}^s(y')$ because $\sum_{y'=d_{v'}^0}^{d_v^0-e_{v',v}-\tau-k} f_{v'}^s(y')$ can be expressed as $f_{v'}^s(d_{v'}^0) + \cdots + f_{v'}^s(d_v^0 - e_{v',v} - \tau - k)$ and $\sum_{y=d_v^0+1}^{z} f_{v'}^s(y - e_{v',v} - \tau - k)$ can be expressed as $f_{v'}^s(d_v^0 + 1 - e_{v',v} - \tau - k) + \cdots + f_{v'}^s(z - e_{v',v} - \tau - k)$.

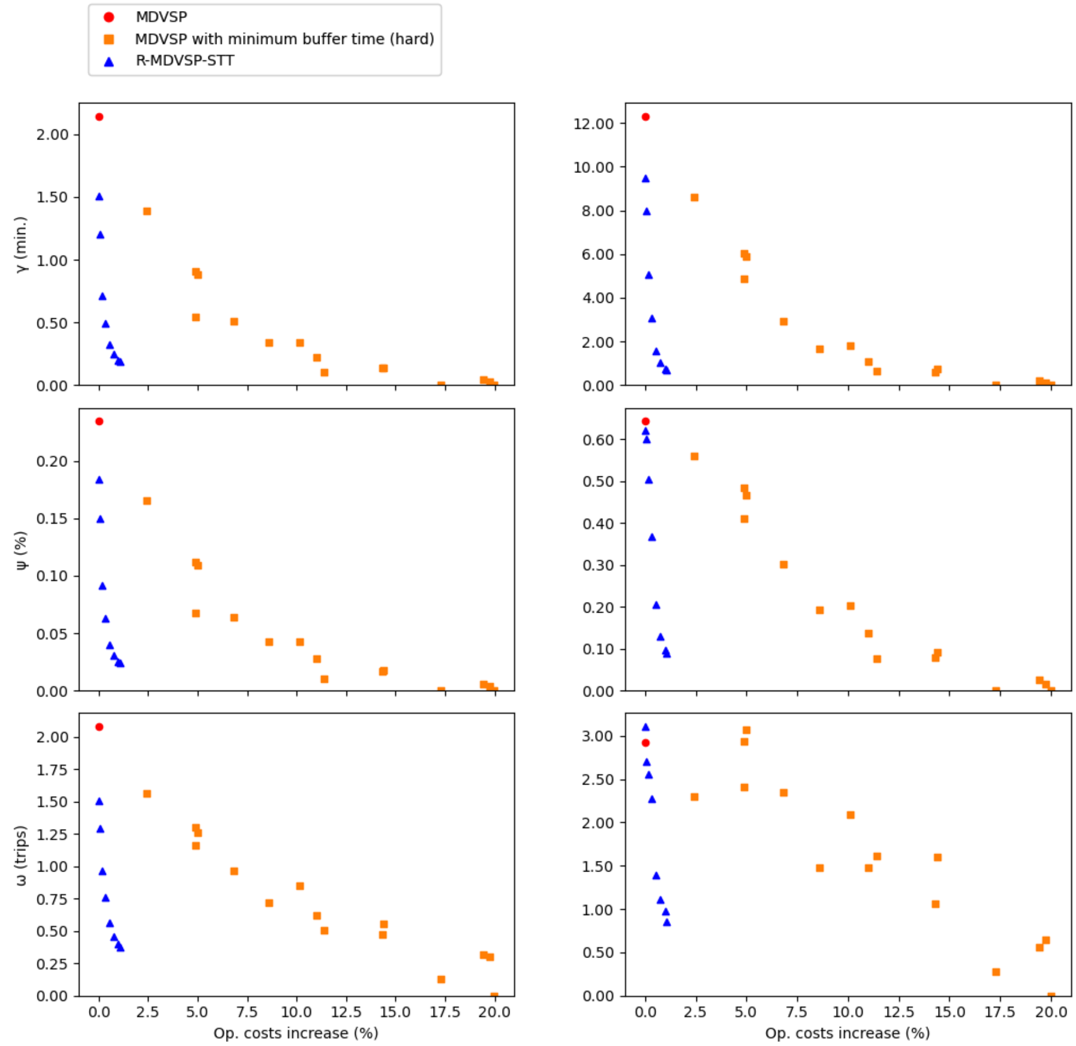
B Additional results



(a) 1st case

(b) 2nd case

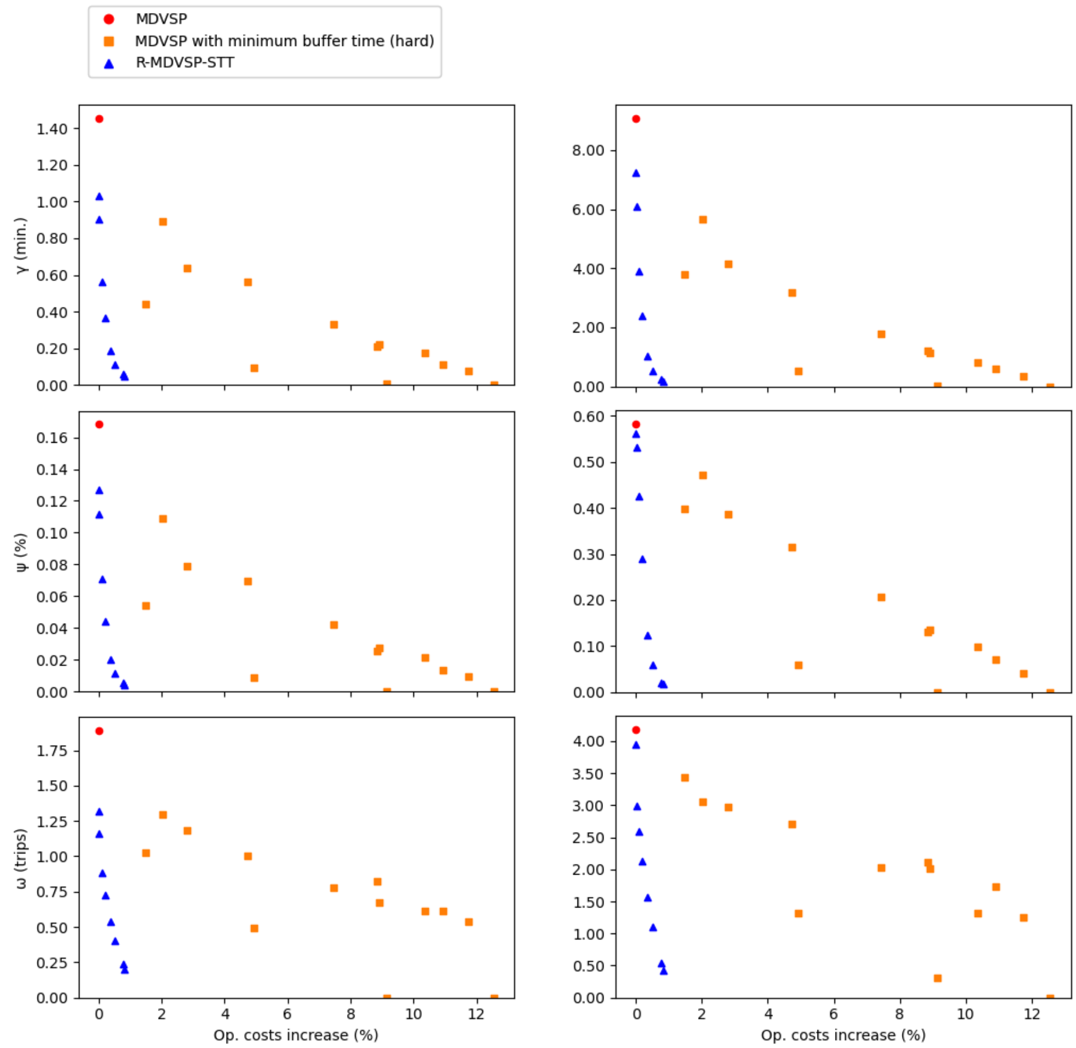
Figure 6: Reliability metrics - I1



(a) 1st case

(b) 2nd case

Figure 7: Reliability metrics - I2



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

Figure 8: Reliability metrics - I3

References

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

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.

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.

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.

Boland, N., Dickson, S., Savelsbergh, M., Smilowitz, K., 2015. Dominance in pricing problems with stochasticity. (accessed February 22, 2022).

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

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.

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

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

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. Dipartimento di Elittronica e Informatica, Università di Padova, Italy.

Furth, P.G., Hemily, B., Muller, T.H., Strathman, J.G., 2006. Using Archived AVL-APC Data to Improve Transit Performance and Management. Transit Cooperative Research Progrem (TCRP) Report 113. Transport Research Board of the National Academics. Washington, DC, USA. doi:https://doi.org/10.17226/13907.

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.

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.

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.

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.

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.

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

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.

Löbel, A., 1998. Vehicle scheduling in public transit and lagrangean pricing. Management Science 44, 1637–1649.

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.

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.

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.

Oort, N., 2011. Service Reliability and Urban Public Transport Design. Ph.D. thesis. Delft University of Technology. Netherlands.

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.

Ryus, P., Danaher, A., Walker, M., Nichols, F., Carter, B., Ellis, B., Cherrington, L., Bruzzone, A., 2013. Transit Capacity and Quality of Service Manual, 3rd ed. Transit Cooperative Research Progrem (TCRP) Report 165. Transport Research Board of the National Academics. Washington, DC, USA. doi:https://doi.org/10.17226/24766.

Salicrú, M., Fleurent, C., Armengol, J., 2011. Timetable-based operation in urban transport: Run-time optimisation and improvements in the operating process. Transportation Research Part A: Policy and Practice 45, 721–740. doi:https://doi.org/10.1016/j.tra.2011.04.013.

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.

Wellman, M.P., Ford, M., Larson, K., 2013. Path planning under time-dependent uncertainty. arXiv:1302.4987.

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.