Omega 127 (2024) 103100

Contents lists available at ScienceDirect

Omega

journal homepage: www.elsevier.com/locate/omega

Increasing schedule reliability in the multiple depot vehicle scheduling problem with stochastic travel time☆

Léa Ricard [a,*], Guy Desaulniers [b], Andrea Lodi [c], Louis-Martin Rousseau [b]

[a] Department of Computer Science, Université de Montréal, Montréal, Canada
[b] Department of Mathematics and Industrial Engineering, Polytechnique Montréal, Montréal, Canada
[c] Jacobs Technion-Cornell Institute, Cornell Tech and Technion - IIT, NY, USA

ARTICLE INFO

Keywords:
Vehicle scheduling
Reliability
Column generation
Stochastic programming
Delay propagation

ABSTRACT

The multiple depot vehicle scheduling problem (MDVSP) is one of the most studied problems 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 a stochastic dominance criterion for the associated subproblems. The solutions obtained are compared based on three 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.

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 [1]. 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 [1,2]. 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 over two or more depots, the VSP is referred to as the multiple depot vehicle scheduling problem (MDVSP) and is proven to be NP-hard [3]. 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

☆ Area - Production Management, Scheduling and Logistics. This manuscript was processed by Associate Editor Yagiura.

* Corresponding author.
E-mail addresses: lea.ricard@umontreal.ca (L. Ricard), guy.desaulniers@gerad.ca (G. Desaulniers), andrea.lodi@cornell.edu (A. Lodi), louis-martin.rousseau@polymtl.ca (L.-M. Rousseau).

https://doi.org/10.1016/j.omega.2024.103100
Received 1 December 2022; Accepted 15 April 2024
Available online 26 April 2024
0305-0483/© 2024 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

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 [4]. Indeed, disruptions and operational variability, for example heavy traffic [4,5] and fluctuation of passenger volume [4,6], 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 [4,5,7,8]. 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 [7,8] 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 five 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 the three reliability metrics. The best probabilistic model for predicting the travel time distributions of the Montréal bus network, as well as its features and parameters, were selected in a previous work [9] 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 2, we discuss related works on the 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 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-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 on large-scale instances. The robustness and effectiveness of our method are analyzed in Section 7 for edge cases and relatively small instances. Section 8 summarizes our findings.

2. Literature review

The MDVSP has been studied since the 1970s [10], but exact algorithms have been proposed only since the end of the 1980s including those of Carpaneto et al. [11], Ribeiro and Soumis [12], Löbel [13], Fischetti et al. [14], Hadjar et al. [15], and Kliewer et al. [16]. The work of Ribeiro and Soumis [12] 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. [15] who introduced valid inequalities and a variable fixing technique. Based on a multicommodity flow formulation similar to that of Ribeiro and Soumis [12], Löbel [13] 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. [16] 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 VSP and MDVSP extensions including, among others, the (MD)VSP with time windows [17] and the mixed-fleet (MD)VSP [18–20]. We refer interested readers to Bunte and Kliewer [21], Desaulniers and Hickman [1] 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: dynamic (or online, real-time) and static (or offline) methods. This work focuses on static methods, which continue to be essential for many public transport agencies due to technological and logistical considerations. First, dynamic approaches necessitate technological support for real-time communication between buses and decision-makers [22]. However, establishing and maintaining such technological infrastructure can be costly and may not be readily available in all urban environments. Second, vehicle schedules must comply with local employment regulations and collective agreements (where applicable), and respect drivers' transfer points. Any deviation from these rules can be costly; for example, extending a driver's working day can result in high overtime costs. Dynamic approaches require continuous validation of schedules against these criteria, which complicates their implementation in practice. As a result, many transport agencies decide to react dynamically only in the event of major disruptions, and therefore rely mainly on static approaches. Nevertheless, it is noteworthy to mention the works of Huisman et al. [23], He et al. [24], who developed solution approaches to the dynamic VSP. After a major disruption, dynamic VSP helps recover feasible schedules by rescheduling online new vehicle itineraries.

Table 1 provides a summary of the relevant literature on offline models and solution approaches for the reliable (MD)VSP, in particular the works of Kramkowski et al. [7], Naumann et al. [5], Shen et al. [25], van Kooten Niekerk [26], Amberg et al. [4], and Amberg et al. [8]. The columns display the problem type (either single-depot or multi-depot), whether full delay propagation is considered or not (Full delay propa.), whether a stochastic programming approach is used or not (Stoch. program.), the type of model proposed (Model), the solution method (Sol. method), and the size of the largest instance solved. Instances with fewer than 500 trips, between 500 and 1000 trips, and more than 1000 trips are classified as "small", "medium", and "large", respectively. Kramkowski et al. [7] presented an offline

Table 1
Overview of offline methods for the reliable (MD)VSP.

Author(s)	Problem	Full delay propa.	Stoch. program.	Model	Sol. method	Instance size
Kramkowski et al. [7]	single-depot	•		MCF	MH	large
Naumann et al. [5]	single-depot		•	NF	BC	small
Shen et al. [25]	single-depot	•	•	NF	H	small
van Kooten Niekerk [26]	single-depot		•	NF	BC	large
Amberg et al. [4]	single-depot	•		MCF	CG,LR	medium
Amberg et al. [8]	multi-depot	•		MCF	CG,LR	medium
This paper	**multi-depot**	•	•	**SPP**	**CG**	**large**

Model: MCF — multi-commodity, NF — network flow, SPP — set partitioning. *Sol. method*: MH — metaheuristic, BC — branch-and-cut, H — heuristic, CG — column generation, LR — Lagrangian relaxation.

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. [16], 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. [5]. The network of Kliewer et al. [16] (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. [25] 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 [26] 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. [8] is an extension of Amberg et al. [4] and they both address the sequential, partially integrated, and integrated vehicle and crew scheduling problems. In both works, the VSP is modeled as a multi-commodity problem with an underlying time-space network. 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. [8], the solution approach of Amberg et al. [4] 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.

In Table 1, we can observe that among the literature on offline approaches for the reliable (MD)VSP, the work of Shen et al. [25] is the first to introduce a stochastic model that accounts for full delay propagation. In their model, the cost of each arc (including a penalty for delays) is path-dependent and so cannot be computed beforehand. This fact renders the network flow model of Shen et al. [25] impossible to solve using a conventional MIP solver. Consequently, their approach remains limited to small-scale cases, such as the single-depot instance used in their experiments. Our work addresses this scalability issue by introducing an alternative set partitioning formulation for the MDVSP with stochastic travel time and a tailored branch-and-price heuristic that enables us to tackle instances of up to 2,300 trips and three depots. The choice of our approach was guided by the needs of our industrial partner, GIRO Inc., for a solution approach that not only handles the large instances of medium- to large-scale transit agencies effectively but also easily adapts to the diverse rules and constraints imposed by their customers on vehicle schedules.

3. Mathematical model

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

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 every trip $i \in \mathcal{V}$ exactly once 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 a given service reliability metric. In order to find near Pareto-efficient solutions, we use a linear scalarization method [27]. 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. Note that due to data availability constraints, we are considering deterministic travel times for pull-in,

pull-out, and deadhead trips. These trips are usually short and do not involve passengers, thus eliminating one main source of travel time variability. However, if data become available in the future, a similar approach could incorporate stochastic pull-in, pull-out, and deadhead travel times.

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 deterministic 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 complete notation for the R-MDVSP-STT is summarized in Appendix A.

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

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

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

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

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

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 [12] and Hadjar et al. [15]. 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 (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 first introduced by Ribeiro and Soumis [12] 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 $i, j \in \mathcal{V}$). Since we consider the travel time of timetabled trips to be stochastic, we could also use an enlarged network with additional arcs between incompatible trips, as long as the incompatibility does not exceed 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

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

Table 2
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	$r^T \kappa_{i,d}$

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 2. 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 3.2. Notice that, given a schedule, the costs for deadheading and for waiting outside a depot are deterministic, whereas the delay penalty is stochastic. Hence, even if the waiting time may be used to absorb delays due to travel time deviations, we have chosen to consider predefined costs for waiting outside the depot to simplify the problem. Since these costs are relatively small compared to the overall cost for the value of r^W set in our experiments, this simplification does not alter our conclusions.

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 [12] to account for this constraint by adding a new type of connection that artificially includes the wait-at-depot. The cost of such a connection is computed by summing the additional costs for the travel time between the end location of trip i and the depot and between the depot and the starting location of trip j. These modified connections avoid the use of pull-in and pull-out arcs for waiting at a depot, so that an additional vehicle is not counted for each waiting at a 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 $i \in \mathcal{V}$ 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. [9] 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. 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 [28,29]. Due to data availability constraints and for simplicity, 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 percentiles of its probability distribution, respectively. For each $i \in \mathcal{V}$, let T_i be defined over Φ_i as

$$\Phi_i = \{\lfloor \dot{t}_i^5 \rfloor, \lfloor \dot{t}_i^5 \rfloor + 1, \dots, \lfloor \dot{t}_i^{95} \rfloor - 1, \lfloor \dot{t}_i^{95} \rfloor\}, \tag{6}$$

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

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. [30] and Shen et al. [25] 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

$$f_j^s(y) = Pr(Y_j^s = y) = \begin{cases} \sum_{k \in \Phi_i} h_i(k) \times \sum_{y'=d_i^0}^{d_j^0 - \kappa_{i,j} - \tau - k} f_i^s(y'), & \text{if } y = d_j^0; \\ \sum_{k \in \Phi_i} h_i(k) \times f_i^s(y - \kappa_{i,j} - \tau - k), & \text{if } y > d_j^0; \\ 0, & \text{otherwise,} \end{cases} \tag{7}$$

where, in the first case, we sum, for each travel time $k \in \Phi_i$, the travel time probability $h_i(k)$ multiplied by the probability that trip i starts before $y - \kappa_{i,j} - \tau - k$. Indeed, if a bus arrives ahead of schedule at the starting location of a timetabled trip, it must wait until the planned start time of the trip to leave on time. In the second case, we sum, for each travel time $k \in \Phi_i$, the travel time probability $h_i(k)$ multiplied by the probability that trip i starts exactly at $y - \kappa_{i,j} - \tau - k$.

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

$$\mathbb{E}(X_j^s) = \sum_{x=0}^{x_{j,s}^{max}} x \times g_j^s(x) = \sum_{x=0}^{x_{j,s}^{max}} x \times f_j^s(x + d_j^0), \tag{8}$$

where i is the trip that precedes the trip j in schedule s and $x_{j,s}^{max} = d_i^0 + x_{i,s}^{max} + \kappa_{i,j} + \tau + \lfloor \dot{t}_i^{95} \rfloor - d_j^0$ 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 the expected secondary delay per passenger 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 [31, 32] embedded in a branch-and-bound tree. This solution method is also referred to as branch-and-price [33]. 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 algorithmic framework of our tailored branch-and-price algorithm is illustrated in Fig. 1. First, we consider the linear relaxation of (1)–(4). This relaxation, which is called the master problem (MP), is solved at the root node of the branch-and-bound tree using a column generation algorithm. In each iteration of the column generation algorithm, a restricted MP (RMP), i.e., the MP restricted to a small subset $\mathcal{S}' \subseteq \mathcal{S}$ of the schedule variables y_s, is solved. To identify potentially useful columns to add to the RMP, we solve a set of pricing problems. In our case, there is one pricing problem per depot, corresponding to a shortest path problem with stochasticity [34,35]. If we find variables with a negative reduced cost after solving the stochastic pricing problems, we add them to $\mathcal{S}'$, initiating a new column generation iteration. Otherwise, the column generation algorithm terminates, and the current RMP solution is guaranteed to be optimal for the MP. Subsequently, if the solution to the RMP is integer and its cost is less that the current upper bound (UB), we update the UB. If this solution is not integer but its cost is still less than the current UB, we either apply a branching strategy or remove the perturbation and add the corresponding node(s) to the node list. The branch-and-bound algorithm then checks whether there are any nodes left in the node list. If there are, a node is removed from the list, and its corresponding MP is solved using the column generation algorithm. The branch-and-price algorithm stops when the node list is empty.

The stochastic pricing problems as well as the two acceleration strategies used, namely a heuristic branching strategy and a perturbation method, are detailed in the following.

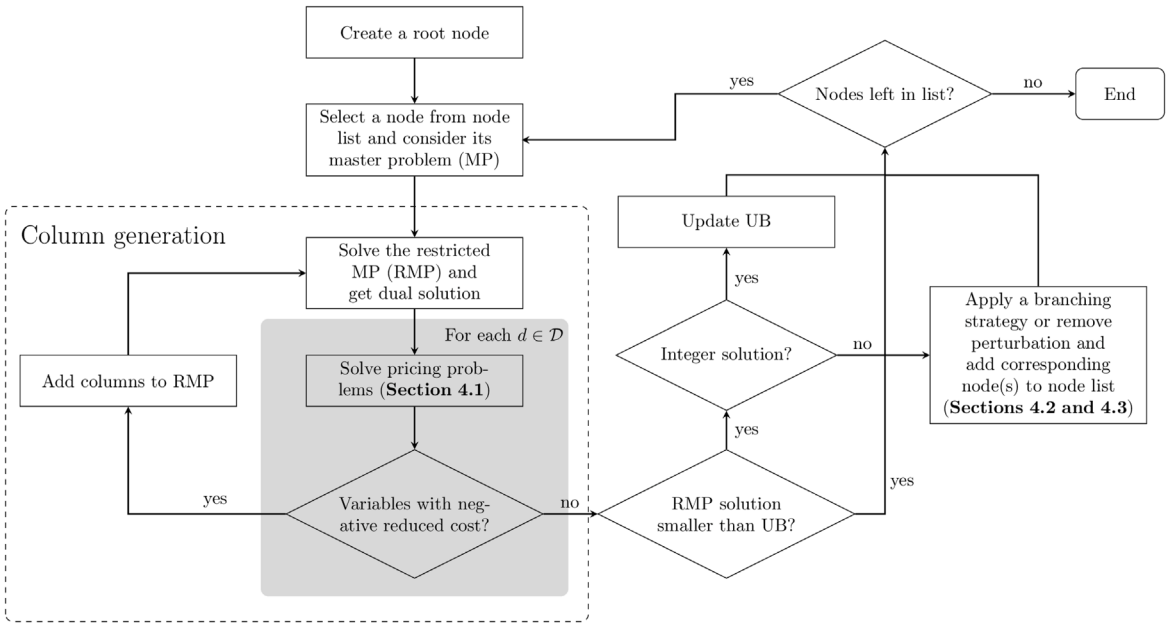


Fig. 1. Flowchart of the branch-and-price algorithm.

4.1. Stochastic pricing problems

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 $(u_i)_{i\in\mathcal{V}}$ 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

$$\tilde{c}_s = c_s - \sum_{i\in\mathcal{V}} a_{is}u_i - \pi_d, \tag{9}$$

with the following cost breakdown per arc

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

Because the arc costs are stochastic and path-dependent in the R-MDVSP-STT, the standard labeling algorithm [31] cannot be used directly to solve the pricing problems. The dependence assumption is crucial because, as explained by Wellman et al. [34], 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. [34] and Boland et al. [35] which employs 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"[31]. This ability is indeed essential to efficient dynamic programming algorithms [31]. 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. [36] 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.

4.1.1. 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 $F^p_i(z)$ be the cumulative distribution function of Y^p_i defined as

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

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

$$L^p_i = (F^p_i(d^0_i), \dots, F^p_i(d^0_i + x^{max}_{i,p}), C^p_i), \tag{12}$$

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

4.1.2. Extension functions

Consider the extension of a label $L^{p'}_i = (F^{p'}_i(d^0_i), \dots, F^{p'}_i(d^0_i + x^{max}_{i,p'}), C^{p'}_i)$ 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 Eq. (7) as detailed in Appendix B. This derivation results in

$$F_j^p(z) = \sum_{k \in \Phi_i} h_i(k) \times F_i^{p'}(z - \kappa_{i,j} - \tau - k). \tag{13}$$

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 into route segments (see Eq. (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 $i, j \in \mathcal{V}$.

4.1.3. 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 $L_i^{p_1} = (F_i^{p_1}(d_i^0), \dots, F_i^{p_1}(d_i^0 + x_{i,p_1}^{max}), C_i^{p_1})$ and $L_i^{p_2} = (F_i^{p_2}(d_i^0), \dots, F_i^{p_2}(d_i^0 + x_{i,p_2}^{max}), C_i^{p_2})$, respectively. The path p_1 dominates p_2 (and therefore p_2 can be discarded) when the following two conditions hold:

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

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 delays are path-dependent. For example, consider again the two paths p_1 and p_2. If $\mathbb{E}(X_i^{p_1}) \le \mathbb{E}(X_i^{p_2})$, it does not necessarily imply if we extend p_1 and p_2 with node $j \in \mathcal{V}$ that $\mathbb{E}(X_j^{p_1 \oplus j}) \le \mathbb{E}(X_j^{p_2 \oplus j})$, 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 $F_i^{p_1}(z) \ge F_i^{p_2}(z)$ for some $z \ge d_i^0$, it means that $Pr(Y_i^{p_1} > z) \le Pr(Y_i^{p_2} > z)$ (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 $z \in \{d_i^0, d_i^0+1, \dots, d_i^0+\max\{x_{i,p_1}^{max}, x_{i,p_2}^{max}\}\}$ (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 Fig. 2(a). This figure and Fig. 2(b) 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 $z \in \{d_i^0, d_i^0 + 1, \dots, d_i^0 + \max\{x_{i,p_1}^{max}, x_{i,p_2}^{max}\}\}$, as in the example in Fig. 2(b), 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.

4.2. Heuristic branching

An exact branch-and-price algorithm [33] 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 1,400 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 1,400, 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 [37]. 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

$$\min \quad \sum_{s \in \mathcal{S}} c_s y_s + \sum_{i \in \mathcal{V}} (\delta_i^+ \eta_i^+ + \delta_i^- \eta_i^-) \tag{15}$$

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

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

$$0 \le \eta_i^+ \le \xi_i^+, \qquad \forall i \in \mathcal{V} \tag{18}$$

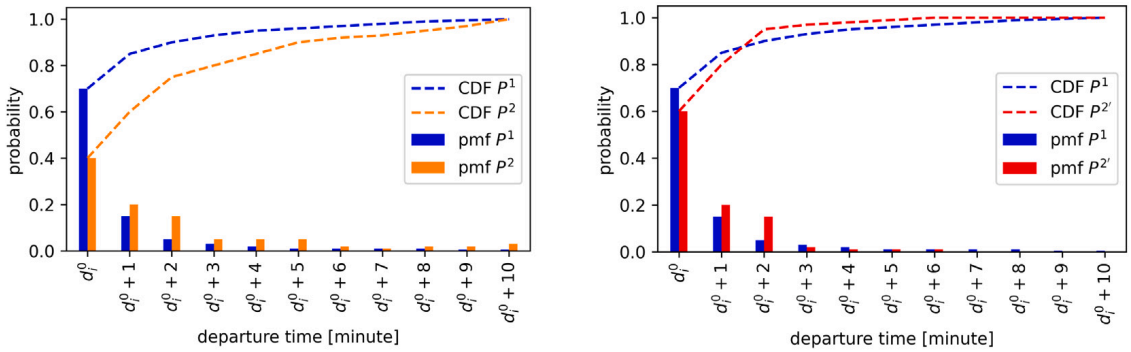
$$0 \le \eta_i^- \le \xi_i^-, \qquad \forall i \in \mathcal{V} \tag{19}$$

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

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.

5. Assessing schedule reliability

To mitigate delay propagation, the R-MDVSP-STT selects vehicle schedules that typically include 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 the expected secondary delay per passenger. This is done by adjusting the factor β in the cost of connection arcs (see Table 2). 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 new reliability metrics that planners can use as a decision-support tool to navigate



(a) Second condition holds (b) Second condition does not hold

Fig. 2. Examples of the second dominance condition applied to the trip i of paths p_1 and p_2.

the trade-off between operational costs and reliability when faced with multiple feasible vehicle scheduling solutions, helping them select the most suitable one. Furthermore, we provide a Monte Carlo simulation framework [38,39] to compute these metrics.

5.1. Reliability metrics

The reliability of a bus service can be assessed at various levels, including the stop, route, and network levels [40]. It can also be evaluated from both the passenger and operator perspectives [41,42]. van Oort [42] pointed out that reliability metrics are typically designed with a focus on the operator's viewpoint, primarily considering vehicles rather than passengers and their perception of service reliability. They recommended that the actual number of passengers be factored into the aggregation of reliability metrics, in order to take account of passengers' interests. Additionally, different reliability metrics are defined in the literature for frequent services (with headways of 10 min or less) and infrequent services. For frequent services, passengers arrive randomly at bus stops, making headway regularity the most critical factor [40]. On the other hand, for infrequent services, on-time performance is typically the primary indicator because passengers plan their arrival time at the bus stop [43]. Various reliability metrics have been proposed in the literature, e.g., metrics related to travel time and waiting time, punctuality, headway variability, additional budgeted travel time, and overcrowding [see, e.g., 40–44].

When solving the MDVSP, operators can only adjust the allocation of buffer time between timetabled trips to enhance schedule reliability. Therefore, in the context of the MDVSP with infrequent service, the most common and relevant approach to measure service reliability is to look at service punctuality, specifically focusing on secondary delays (as seen in works by [4,7,8,26]).

Building upon the insights from the literature, we propose to compare the R-MDVSP-STT solutions using the following punctuality 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. The first two metrics are inspired by the ones proposed in Kramkowski et al. [7] and Amberg et al. [4], whereas the third is derived from information provided by our industrial partner on the challenges faced by transport agencies. These three metrics are computed as follows.

Expected secondary delay per passenger (γ):

$$\gamma = \sum_{s\in S^*}\sum_{i\in\mathcal{V}} a_{is}\alpha_i \mathbb{E}(X_i^s), \tag{21}$$

where S^* is the set of vehicle schedules selected in a solution.

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

$$\psi = \sum_{s\in S^*}\sum_{i\in\mathcal{V}} a_{is}\alpha_i Pr(X_i^s > \epsilon), \tag{22}$$

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):

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

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 in schedule s at iteration k, respectively, are used. If $\sum_{k=1}^{K}\varphi^{s,k} > 0$, where K is the number of iterations of the simulation, $\bar{\Omega}_s$ is approximated by $\sum_{k=1}^{K}(\rho^{s,k}/\varphi^{s,k})$. Otherwise, the approximation of $\bar{\Omega}_s$ takes value 0. To better understand how these counters work, let us consider the toy example presented in Fig. 3. The timetabled trips with a striped background are delayed whereas trips with a plain background 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. Thus, $\varphi^{s,k} = 2$ and $\rho^{s,k} = 1 + 0 = 1$ in this example.

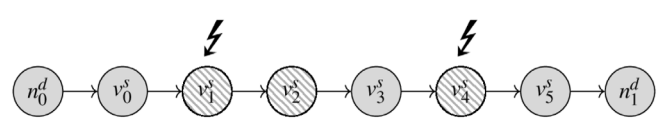


Fig. 3. Example of the potential delay propagation in schedule $s = \{v_0^s, v_1^s, \dots, v_5^s\}$ of 6 timetabled trips at iteration k. Delayed trips are striped.

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 novelty of these metrics lies in the addition of the parameters $\alpha_i, i \in \mathcal{V}$, that weight the contribution of each trip $i \in \mathcal{V}$ 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 [45]. 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 (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. Indeed, γ and ψ are linked by the following relationship $\mathbb{E}(X_i^s) = \mathbb{E}(X_i^s | X_i^s > 0)/Pr(X_i^s > 0)$ [7], whereas θ is positively correlated with $\mathbb{E}(X_i^s | X_i^s > 0)$.

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 $k = 1, \dots, K$, delay propagation in each vehicle schedule $s \in S^*$ 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

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

$$y_j^{s,k} = \max\{y_i^{s,k} + t_i^k + \kappa_{i,j} + \tau, d_i^0\}, \tag{25}$$

where 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 in schedule s at iteration k is computed for all $j \in s \setminus \{v_o^s\}$ as

$$x_j^{s,k} = y_j^{s,k} - d_j^0. \tag{26}$$

Finally, the approximations of the three metrics are computed in Steps 11–28. 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 i_i^{75} and i_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 i_i^{75} and above i_i^{95}, utilizing the procedure proposed in Thomopoulos [38]. At each iteration and for each timetabled trip, a random value $u \sim U(0,1)$ is generated. Then, we compute $v = \dot{H}_i(i_i^{75}) + u \times (\dot{H}_i(i_i^{95}) - \dot{H}_i(i_i^{75}))$, where $\dot{H}_i(t)$ is the CDF of $\dot{h}_i(t)$, before finding the travel time value t such that $\dot{H}_i(t) = v$.

6. Computational results

In this section, we test our model on five real-life instances, I1–I5, with up to 2,341 timetabled trips 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 five instances have been computed as in Ricard et al. [9] (see Section 3.2). The main properties of instances I1–I5, namely the instance name (Instance), the number of timetabled trips ($|\mathcal{V}|$), the number of arcs ($|A|$), the number of depots ($|D|$), and the number of bus lines (# lines), are listed in Table 3.

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 min, and 3 min, 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 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 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. 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.

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** $s \in S^*$ **do**
4         $\varphi^s \leftarrow 0$ `// counter of the number of first delayed timetabled trips`
5         $\rho^s \leftarrow 0$ `// 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 $\dot{h}_{i-1}(t)$
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                     $\varphi^s \leftarrow \varphi^s + 1$
16   **else**
17                     $\rho^s \leftarrow \rho^s + 1$
18   **end**
19   **end**
20   **end**
21   **if** $\varphi^s > 0$ **then**
22             $\hat{\theta} \leftarrow \hat{\theta} + \frac{\rho^s}{\varphi^s}$
23   **end**
24   **end**
25 **end**
26 $\hat{\gamma} \leftarrow \hat{\gamma}/K$
27 $\hat{\psi} \leftarrow \hat{\psi}/K$
28 $\hat{\theta} \leftarrow \hat{\theta}/|S^*|$

Table 3
Properties of real-life instances I1–I5.

Instance	$\lvert\mathcal{V}\rvert$	$\lvert A\rvert$	$\lvert D\rvert$	# lines
I1	1,175	628,064	2	8
I2	1,446	928,539	4	6
I3	1,916	1,622,134	3	8
I4	2,195	2,119,534	2	8
I5	2,341	2,434,675	3	10

Table 4
Heuristic performance without considering reliability (MDVSP).

Inst.	UB	LB	Gap (%)	BBn	CPU time (s)		
					Total	Root (%)	Pric. (%)
I1	78,850.8	78,846.3	0.01	58	2,706.8	28.5	56.1
I2	94,853.6	94,848.5	0.01	56	3,231.2	56.5	67.2
I3	135,920.4	135,912.2	0.01	91	6,297.5	45.6	60.0
I4	167,751.2	167,742.9	<0.01	120	9,000.4	41.6	53.6
I5	182,792.8	182,783.5	0.01	126	10,991.2	40.9	54.2

Table 5
Operational costs versus the three reliability metrics of solutions obtained without considering reliability (MDVSP).

Inst.	Op. costs	# Bus	# Trips/bus	Reliability					
				1st case			2nd case		
				γ	ψ	θ	γ	ψ	θ
I1	78,850.8	75	16	0.67	0.07	0.43	3.03	0.27	1.05
I2	94,853.6	91	16	1.45	0.17	0.65	9.57	0.55	1.63
I3	135,920.4	131	15	2.11	0.23	0.92	11.96	0.64	1.33
I4	167,751.2	162	14	1.45	0.17	0.71	9.24	0.58	1.77
I5	182,792.8	176	13	8.50	0.30	0.68	18.58	0.75	1.52

6.1. MDVSP results

In this section, we provide baseline results obtained by solving heuristically instances I1–I5 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 in terms of the three reliability metrics.

Table 4 reports the 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 (BBn), and the computing times (CPU time) in seconds, including the total CPU time (Total), the percentage of the total time spent to solve the root node (Root), and the percentage of the total time spent to solve the pricing problems (Pricing).

All instances are solved in less than about 3 h with approximately one third to 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 metrics for the instances I1–I5 solved without considering reliability are presented in Table 5. 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). See Section 5 for the definition of the simulation cases and the method to approximate these metrics using the travel time probability density functions.

Of instances I1–I5 , I5 is the most prone to unreliability. It has an expected secondary delay per passenger of 8.50 min, a probability that a passenger boards a delayed timetabled trip of 30%, and an average number of timetabled trips needed to recover from secondary delays of 0.68 trips. These values are not negligible, especially for the last metric if we consider that each bus performs an average of 13 timetabled trips 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 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 10 values within the range of $\eta/20$ to 20η, where $\eta = 1,000$ is the cost per vehicle used, have been tested.

Table 6 provides the heuristic performance of our algorithm for different β values. The upper bounds (UB) on the optimal values include the cost related to delays, whereas it is not the case in Table 4 because $\beta = 0$. It may be noted that the optimality gaps of the R-MDVSP-STT are similar to those of the MDVSP. Furthermore, 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 4.2). However, the proportion of time dedicated to solving pricing problems is increased. For instance I5, 54.2% of the computing time is allocated to solving pricing problems for the MDVSP. This proportion increases to 80.2% to 86.9% for the R-MDVSP-STT.

The trade-offs between operational costs and the three reliability metrics of the vehicle schedules obtained with different β values are provided in Table 7. 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 operational costs increase (Va. cost inc.) values are obtained by comparing the R-MDVSP-STT solutions to the corresponding solutions of the MDVSP found in Table 5.

The R-MDVSP-STT solutions have variable operational costs up to 35.66% 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 I5 is reduced from 30% to 13% with a variable operational costs increase of 2.88%. Furthermore, all three reliability metrics, namely γ, ψ, and θ, improve almost monotonically when the penalty factor β increases in the first and second simulation cases, i.e., under normal conditions and under extraordinary factors, respectively (compare columns under "1st case" and "2nd case" in Tables 5 and 7). Overall, our model produces good trade-offs between the three reliability metrics and operational costs, even if some minor inconsistencies are observed. We will analyze in the next section how our model compares to a classical approach to address bus schedules' reliability.

6.3. Comparison with the MDVSP with minimum buffer time

In Naumann et al. [5], van Kooten Niekerk [26], Amberg et al. [4], and Amberg et al. [8], 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. [5], van Kooten Niekerk [26], Amberg et al. [4], and Amberg et al. [8], 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 C for a comparison of our approach with the MDVSP with hard minimum buffer time on instances I4 and I5). 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,

Table 6
Heuristic performance considering reliability (R-MDVSP-STT).

β	UB	Gap (%)	BBn	CPU time (seconds)		
				Total	Root (%)	Pric. (%)
			Instance I1			
$\eta/20$	78,881.1	0.01	43	3,716.0	43.8	81.6
$\eta/10$	78,907.4	0.01	42	3,940.0	43.1	81.0
$\eta/5$	78,953.9	0.01	49	3,079.1	49.4	81.1
$\eta/2$	79,051.3	0.01	45	3,118.7	49.1	80.3
η	79,169.9	0.01	43	3,352.4	47.0	79.3
2η	79,340.0	0.01	43	3,684.6	41.3	78.8
4η	79,582.1	0.01	42	4,287.8	35.6	78.2
8η	79,940.0	0.01	43	4,216.2	34.4	76.9
10η	80,097.6	0.01	60	5,277.0	27.5	77.2
20η	80,734.6	0.01	63	6,170.7	22.7	74.5
avg.	**79,465.8**	**0.01**	**47**	**4,084.3**	**39.4**	**78.9**
			Instance I2			
$\eta/20$	94,933.1	0.01	49	9,376.0	60.4	91.2
$\eta/10$	95,005.3	0.01	55	10,144.4	57.1	91.4
$\eta/5$	95,111.5	0.01	42	8,820.9	57.2	91.2
$\eta/2$	95,321.4	0.01	65	8,890.2	52.0	90.0
η	95,544.5	0.01	63	7,607.9	56.9	87.9
2η	95,824.9	0.01	69	8,092.0	48.8	86.2
4η	96,184.4	0.01	46	5,718.0	65.4	85.5
8η	96,704.2	0.01	60	6,222.4	59.5	81.3
10η	96,926.9	0.01	51	7,103.8	54.5	79.5
20η	97,922.4	0.01	47	6,547.3	65.0	82.2
avg.	**95,947.9**	**0.01**	**55**	**7,852.3**	**57.7**	**86.7**
			Instance I3			
$\eta/20$	136,021.0	0.01	64	13,812.6	57.0	87.4
$\eta/10$	136,114.5	0.01	76	15,521.0	54.8	87.0
$\eta/5$	136,275.5	0.01	95	15,768.2	51.1	86.3
$\eta/2$	136,605.0	0.01	94	15,513.4	51.1	86.0
η	136,966.4	0.01	106	14,531.5	47.6	84.5
2η	137,451.7	0.01	92	12,320.2	52.0	82.5
4η	138,152.7	0.01	87	11,135.1	53.5	81.6
8η	139,234.8	0.01	81	9,948.8	55.1	80.5
10η	139,716.8	0.01	90	9,569.6	57.3	80.3
20η	141,917.0	0.01	84	10,599.3	53.0	78.4
avg.	**137,845.5**	**0.01**	**87**	**12,872.0**	**53.2**	**83.4**
			Instance I4			
$\eta/20$	167,821.9	<0.01	58	14,512.4	60.8	85.4
$\eta/10$	167,886.7	0.01	74	14,109.5	57.6	84.1
$\eta/5$	168,002.4	0.01	92	14,682.6	54.2	83.0
$\eta/2$	168,250.8	0.01	76	14,188.1	55.1	82.9
η	168,525.4	0.01	126	16,177.3	45.9	80.5
2η	168,852.2	0.01	99	13,696.4	53.1	79.7
4η	169,232.0	0.01	106	13,342.7	51.8	77.6
8η	169,667.9	0.01	70	12,058.3	59.1	79.3
10η	169,814.9	0.01	85	11,466.7	59.7	78.6
20η	170,370.0	<0.01	62	13,661.5	61.6	78.0
avg.	**168,842.4**	**0.01**	**85**	**13,789.6**	**55.9**	**80.9**
			Instance I5			
$\eta/20$	182,980.7	<0.01	63	17,610.5	61.7	85.1
$\eta/10$	183,104.3	<0.01	69	16,815.7	61.4	84.9
$\eta/5$	183,295.8	<0.01	72	16,664.1	60.9	84.6
$\eta/2$	183,636.3	0.01	102	17,545.9	56.0	82.6
η	183,993.2	0.01	89	18,419.1	57.5	86.9
2η	184,441.1	<0.01	88	16,282.0	57.4	81.7
4η	185,086.6	0.01	73	15,250.5	61.7	81.1
8η	186,046.9	0.01	93	15,672.9	58.7	80.2
10η	186,480.7	0.01	85	15,549.5	61.3	80.7
20η	188,499.8	<0.01	61	15,425.0	63.4	81.5
avg.	**184,756.5**	**0.01**	**80**	**16,523.5**	**60.0**	**82.9**

Table 7
Operational costs versus the three reliability metrics of the R-MDVSP-STT solutions.

β	Va. cost inc.(%)	Reliability					
		1st case			2nd case		
		γ	ψ	θ	γ	ψ	θ
				Instance I1			
$\eta/20$	0.06	0.47	0.05	0.31	2.12	0.22	0.98
$\eta/10$	0.23	0.39	0.05	0.23	1.86	0.21	0.91
$\eta/5$	0.58	0.33	0.04	0.20	1.60	0.19	0.84
$\eta/2$	1.57	0.22	0.02	0.13	1.00	0.13	0.46
η	2.95	0.16	0.02	0.07	0.68	0.08	0.30
2η	5.46	0.10	0.01	0.04	0.42	0.04	0.18
4η	8.06	0.08	0.01	0.03	0.29	0.03	0.11
8η	11.60	0.06	0.01	0.01	0.21	0.02	0.04
10η	13.02	0.06	0.01	0.01	0.19	0.02	0.03
20η	19.50	0.05	0.01	0.01	0.14	0.02	0.03
				Instance I2			
$\eta/20$	0.06	1.31	0.16	0.60	8.65	0.53	1.98
$\eta/10$	0.55	1.11	0.13	0.49	7.18	0.52	2.50
$\eta/5$	1.99	0.76	0.10	0.35	5.24	0.48	1.88
$\eta/2$	4.94	0.45	0.06	0.23	3.03	0.34	1.14
η	8.84	0.27	0.03	0.15	1.44	0.19	0.65
2η	13.02	0.18	0.02	0.09	0.88	0.11	0.34
4η	18.67	0.12	0.01	0.05	0.45	0.05	0.17
8η	23.71	0.09	0.01	0.03	0.32	0.04	0.09
10η	25.54	0.08	0.01	0.03	0.29	0.03	0.09
20η	30.24	0.07	0.01	0.03	0.25	0.03	0.07
				Instance I3			
$\eta/20$	0.08	1.65	0.20	0.71	10.34	0.63	1.27
$\eta/10$	0.30	1.53	0.19	0.63	9.48	0.62	1.34
$\eta/5$	1.37	1.20	0.15	0.54	7.95	0.60	1.35
$\eta/2$	5.02	0.71	0.09	0.36	5.05	0.50	1.48
η	8.80	0.50	0.06	0.25	3.07	0.37	1.09
2η	14.48	0.33	0.04	0.14	1.57	0.20	0.59
4η	20.30	0.25	0.03	0.10	1.02	0.13	0.41
8η	27.46	0.20	0.03	0.10	0.75	0.10	0.34
10η	29.49	0.19	0.02	0.09	0.70	0.09	0.31
20η	35.66	0.18	0.02	0.09	0.62	0.08	0.26
				Instance I4			
$\eta/20$	0.06	1.14	0.14	0.54	7.62	0.57	1.98
$\eta/10$	0.18	1.06	0.13	0.47	7.25	0.55	1.99
$\eta/5$	0.70	0.88	0.11	0.39	6.08	0.52	1.85
$\eta/2$	2.86	0.55	0.07	0.24	3.70	0.41	1.18
η	5.62	0.35	0.04	0.16	2.34	0.29	0.92
2η	10.44	0.19	0.02	0.08	1.01	0.12	0.37
4η	15.23	0.11	0.01	0.04	0.50	0.05	0.18
8η	22.30	0.06	0.00	0.01	0.22	0.02	0.05
10η	24.14	0.05	0.00	0.01	0.18	0.02	0.05
20η	29.40	0.03	0.00	0.01	0.12	0.01	0.03
				Instance I5			
$\eta/20$	0.64	2.65	0.20	0.40	9.64	0.68	1.60
$\eta/10$	1.32	2.00	0.17	0.36	8.20	0.65	1.55
$\eta/5$	2.88	1.35	0.13	0.29	6.52	0.58	1.33
$\eta/2$	5.78	0.76	0.08	0.19	3.70	0.43	0.89
η	9.33	0.47	0.04	0.11	2.00	0.24	0.50
2η	13.23	0.31	0.02	0.07	1.03	0.11	0.25
4η	17.30	0.23	0.02	0.03	0.63	0.05	0.12
8η	21.43	0.19	0.01	0.01	0.43	0.03	0.05
10η	23.12	0.19	0.01	0.02	0.39	0.03	0.04
20η	26.47	0.18	0.01	0.01	0.34	0.03	0.03

we set the numerical values of this penalty to 0.2, 0.4, 0.8, 1.2, and 2.4 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 min);
- 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.

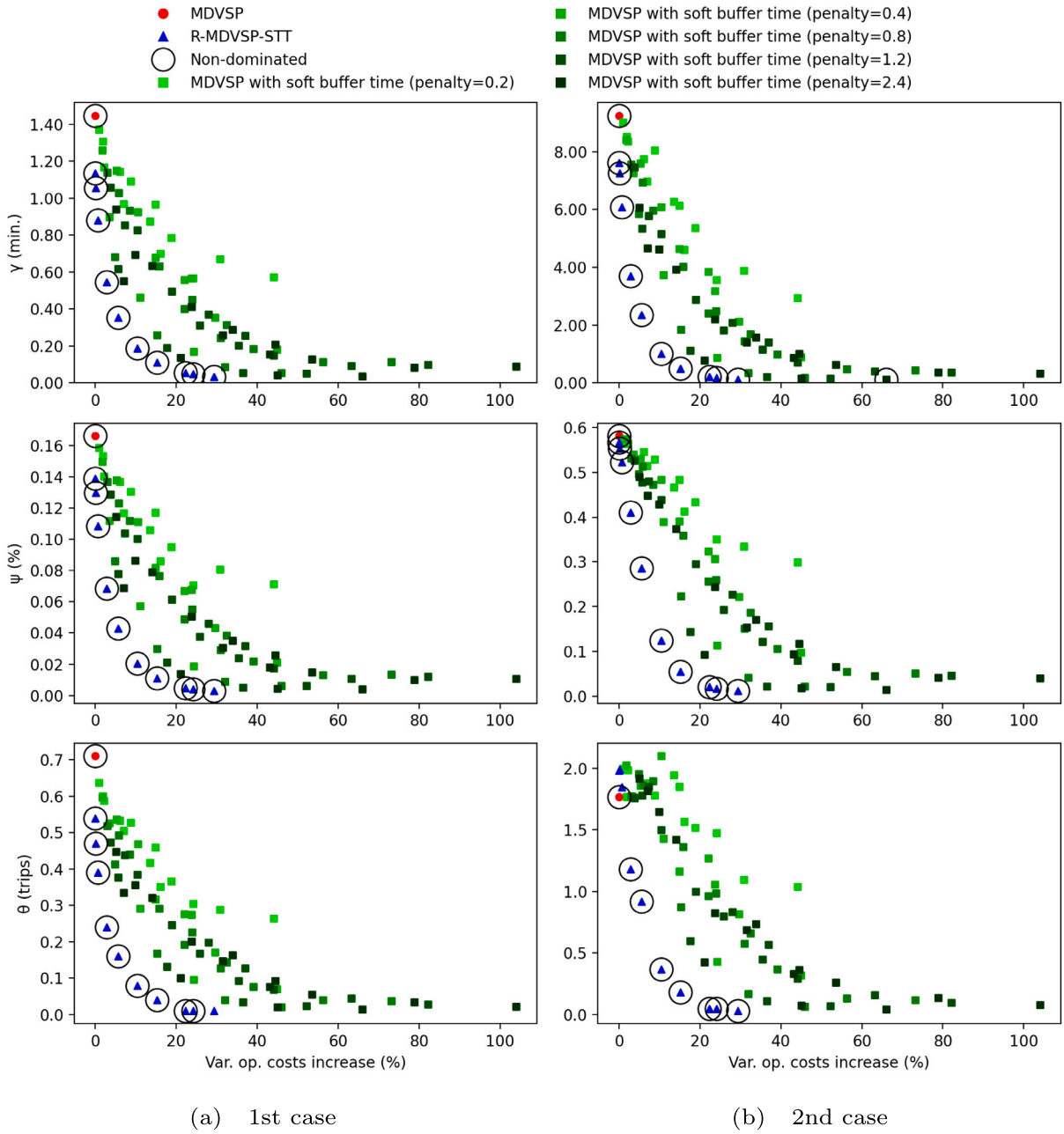


Fig. 4. Reliability metrics — instance I4.

Figs. 4 and 5 show the relationship between the values 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 for the two largest instances I4 and I5, respectively. 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.

Under normal and extraordinary circumstances, the large majority of R-MDVSP-STT solutions for instances I4 and I5 are non-dominated by those of the MDVSP and the MDVSP with soft minimum buffer time. Conversely, a large portion of the solutions of the MDVSP with soft minimum buffer time are dominated by those of our approach. Additionally, for instance I5, many of the minimum buffer time rules exhibit a large disparity in terms of their effectiveness in improving reliability compared to our approach. The minimum buffer time rule that delivers the best results corresponds to the third rule, which involves setting buffer times to cover the primary delay of each timetabled trips x% of the time. As this rule is the only one to use information on the probability distributions of the travel time, it suggests that taking into account the stochasticity of travel time is also relevant in deterministic approaches.

The proportions of all non-dominated solutions that are R-MDVSP-STT solutions for all combinations of instance, scenario, and metric are shown in Table 8. Each entry of this table represents the number of non-dominated R-MDVSP-STT solutions (where the maximum possible value is 10) on the total number of non-dominated solutions (including MDVSP, MDVSP with soft minimum buffer time, and S-MDVSP-STT). The table reports that the number of non-dominated R-MDVSP-STT solutions for instances I1–I3 is similar to the number for instances I4 and

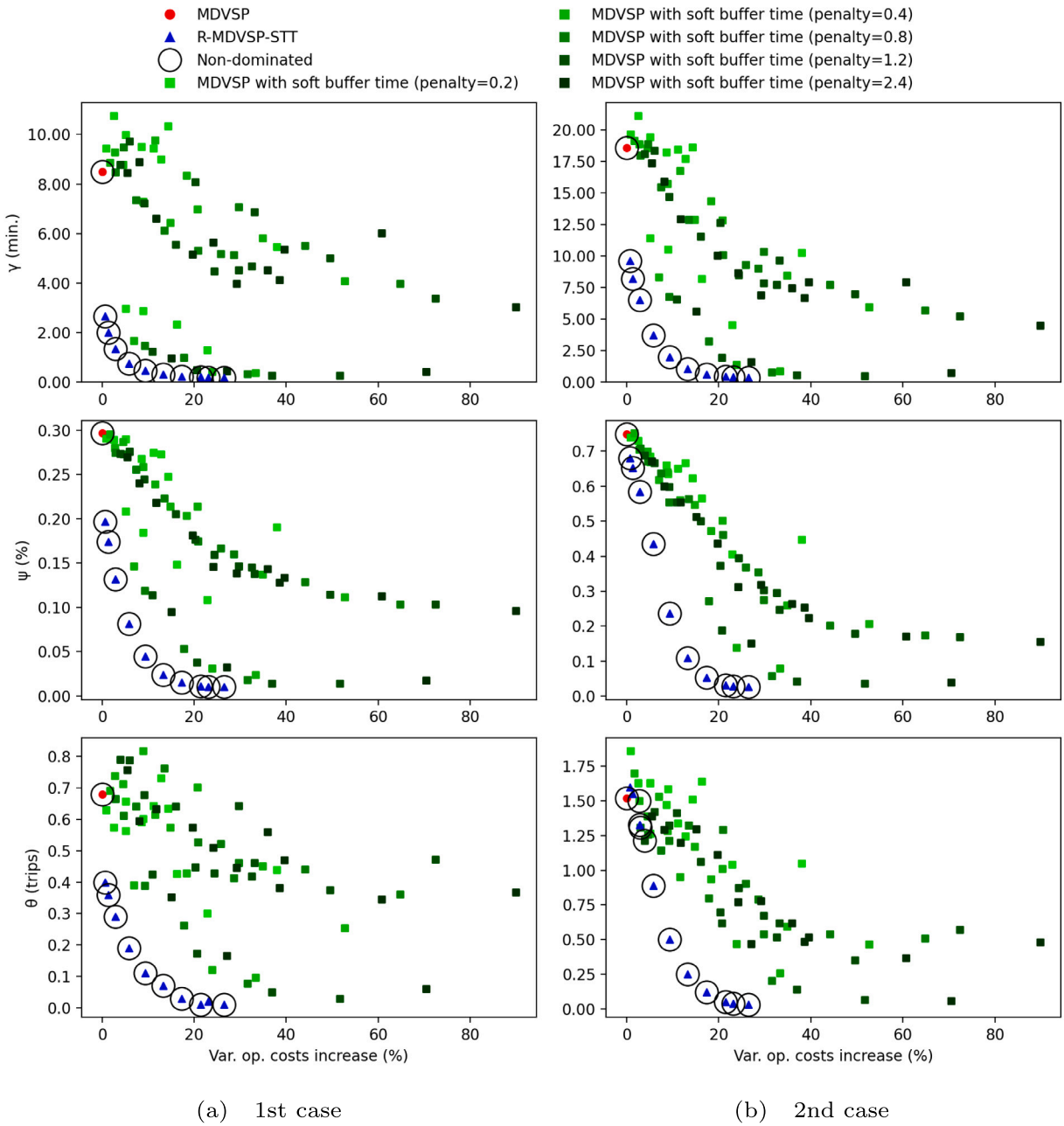


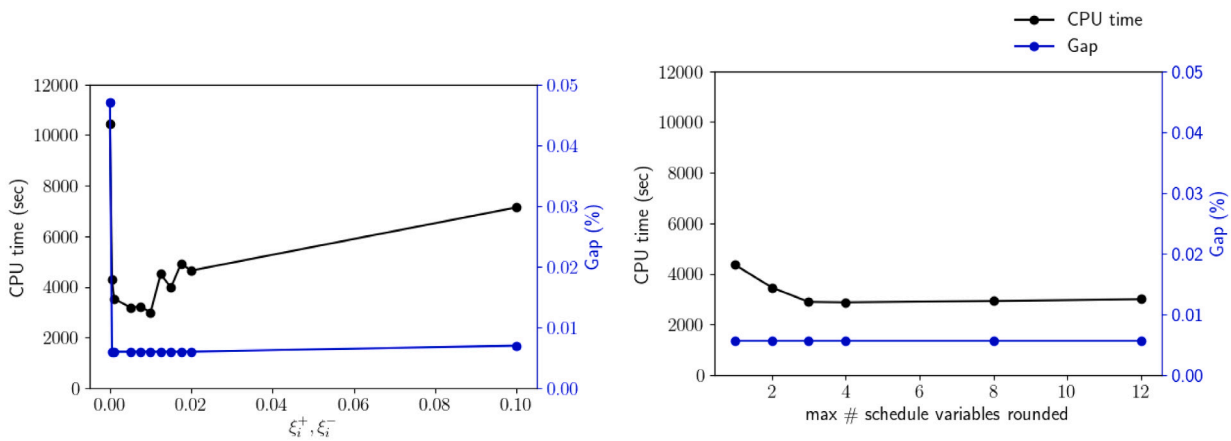
Fig. 5. Reliability metrics — instance I5.

I5. Specifically, at least 90% of the R-MDVSP-STT solutions are non-dominated for the first and second reliability metrics in both simulation cases and the third reliability metric in the first simulation case. Also, at least 60% of the R-MDVSP-STT solutions are non-dominated for the third metric under extraordinary factors.

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. Note that the shape of the travel time probability distributions, such as their skewness, can impact the form of the approximate Pareto frontier. Specifically, it may cause the frontier to shift either downward or upward, and it can also influence the curvature, either compressing it or spreading it out.

Table 8
Ratio of R-MDVSP-STT solutions among all non-dominated solutions.

Instance	1st case			2nd case		
	γ	ψ	θ	γ	ψ	θ
I1	10/11	10/11	9/10	10/11	10/11	10/11
I2	10/11	9/10	9/10	10/11	10/11	7/9
I3	10/11	10/11	10/11	10/11	10/12	6/9
I4	10/11	10/11	9/10	10/12	10/11	7/8
I5	10/11	10/11	9/10	10/11	10/11	8/12



(a) Upper bound values ξ_i^+, ξ_i^-

(b) Maximum number of schedule variables rounded at each branching

Fig. 6. Parameter setting tests — instance I1.

6.4. Analysis of the algorithm performance and parameters selection

The algorithmic parameters were set in order to achieve reasonable computing times and ensure high-quality solutions. The parameter setting tests for the upper bound values of the perturbation method (ξ_i^+, ξ_i^-) and the maximum number of schedule variables rounded at each branching are illustrated in Fig. 6. These tests were carried out on instance I1 by varying one algorithmic parameter at a time and keeping all the other ones constant.

On the one hand, the values of ξ_i^+ and ξ_i^- are set to 0.01. We observe that this value results in the shortest computing time and the best gap between the UB and the LB. For these parameters, we find that when values are less than 0.0005 or greater than 0.01, both the computing time and the gap between the UB and the LB increase. On the other hand, we set the value of the maximum number of schedule variables rounded at each branching to 3. The variation of this algorithmic parameter does not affect the gap between the UB and the LB, while it does have an impact on the computing time.

7. Discussion on robustness and effectiveness

To evaluate how effective our approach is compared to the traditional MDVSP, we start by examining edge cases where our approach and its benchmark (i.e., the MDVSP) produce identical solutions. Then, we determine the minimum prerequisites needed for our approach to offer solutions with lower expected secondary delay per passenger than the MDVSP. Finally, we present additional results on relatively small instances to demonstrate the impact of key factors on our approach's relevance and robustness.

In the R-MDVSP-STT, we achieve more reliable solutions by managing the allocation of buffer times in vehicle schedules. When a timetabled trip might be delayed, we allocate buffer time to absorb such delays. If the MDVSP solution already includes sufficient buffer time after each timetabled trip, our approach yields the same solution. This is because the part of our objective function related to expected secondary delay per passenger becomes zero, regardless of its weight. An example of this edge case is shown in Fig. 7. For this instance, our approach with $\beta \in \{\eta/20, \eta/10, \eta/5, \eta/2, \eta, 2\eta, 4\eta, 8\eta, 10\eta, 20\eta\}$ and the MDVSP (without considering reliability) provide the same solution with an expected secondary delay per passenger of 0 min. The solution consists of two vehicle schedules, the first containing five timetabled trips highlighted in green and the second containing two timetabled trips highlighted in yellow.

Building upon this edge case, we aim to identify the minimum requirements for our approach to offer solutions with lower expected secondary delay per passenger. We find that at least one tight connection in the MDVSP solution is necessary for this. Tight connections occur when buffer time is insufficient to accommodate all potential travel time deviations. To show this, we have modified the edge case presented in Fig. 7 by shifting only the last timetabled trip. The solution obtained by the MDVSP to this modified edge case (see Fig. 8(a)) has one tight connection, operational costs of 2,042, and an expected secondary delay per passenger of 0.32 min. For this instance, our approach with $\beta \in \{\eta/20, \eta/10, \eta/5, \eta/2, \eta, 2\eta, 4\eta, 8\eta, 10\eta, 20\eta\}$ finds another solution, that we have illustrated in Fig. 8(b). The operational costs of this solution are slightly higher, at 2,054, but the expected secondary delay per passenger is down to 0 min.

We have observed that our approach becomes effective when the MDVSP solution contains at least one tight connection. Such instances typically involve dense vehicle schedules, where there are many timetabled trips per bus. We confirm this through tests on smaller, single-line, and single-depot instances denoted S1–S6, each containing 197, 100, 394, 354, 264, and 288 timetabled trips, respectively. In Fig. 9, we compare the MDVSP solution and the R-MDVSP-STT solutions for the complete instances S1–S6 with the solutions of the corresponding reduced versions.[1] These reduced instances are created by randomly eliminating half of the timetabled trips from instances S1–S6. We notice that by reducing the size of instances S1–S6, the average density of vehicle schedules in the MDVSP solutions decreases. For example, the average density of the MDVSP solution for the full S6 instance is 18.0 trips per vehicle schedule, and drops to 11.1 trips per schedule for the reduced S6 instance. Comparing solutions between complete and reduced instances allows us to examine how schedule density affects the effectiveness of our method.

Fig. 9 shows that when the density of vehicle schedules decreases (i.e., when instance size is reduced), the expected secondary delay per

[1] All instances are solved under 774.6 s, with an average computing time of 94.2 s. Furthermore, the average relative difference in percentage between the UB and the LB (gap) is always less than 0.03% and the average gap is 0.01%. The parameters used are the same as the ones presented in Section 6.

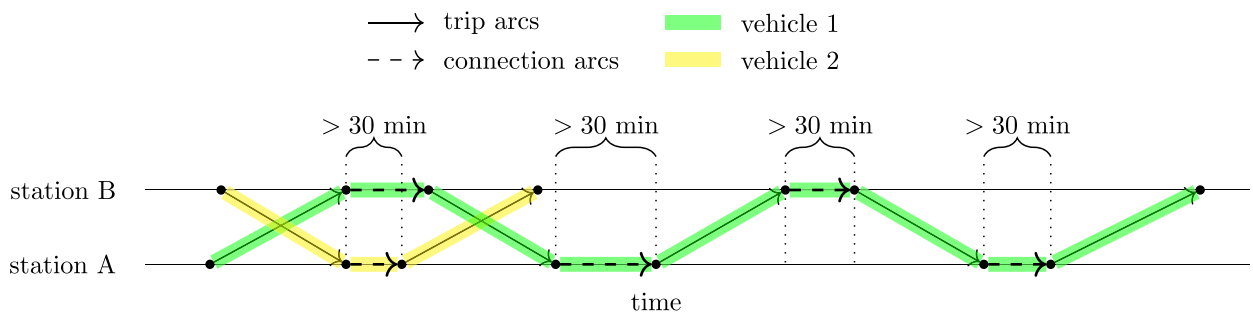


Fig. 7. Optimal solution of an edge case (for $\beta \geq 0$).

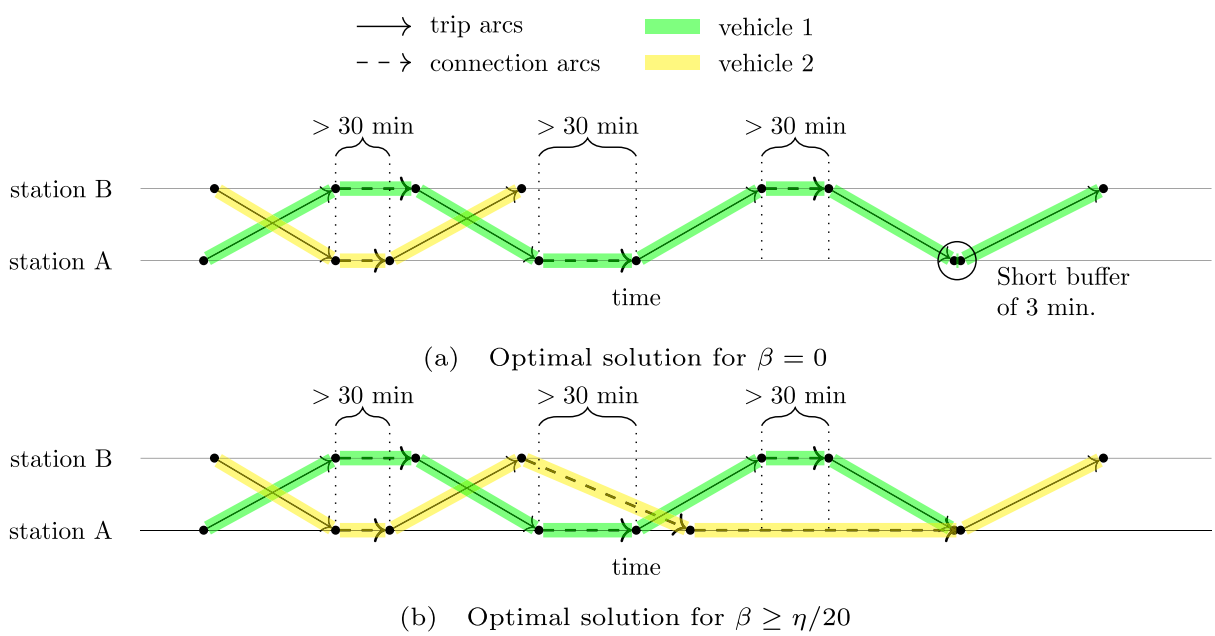


Fig. 8. Modified edge case with one timetabled trip shifted.

passenger (γ) obtained by the MDVSP also decreases. In such cases, our approach may offer fewer improvements compared to the MDVSP. Indeed, the approximate Pareto frontiers in red (associated with the reduced instances) generally present less advantageous gains in terms of expected secondary delay per passenger compared to the blue ones (associated with the full instances). However, in larger, real-life instances where vehicle schedule density is usually high, our approach provides a good balance between operational costs and expected secondary delay per passenger (see Section 6 for more details).

8. 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 five real-world instances of 1,175 to 2,341 timetabled trips and 2 to 4 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,

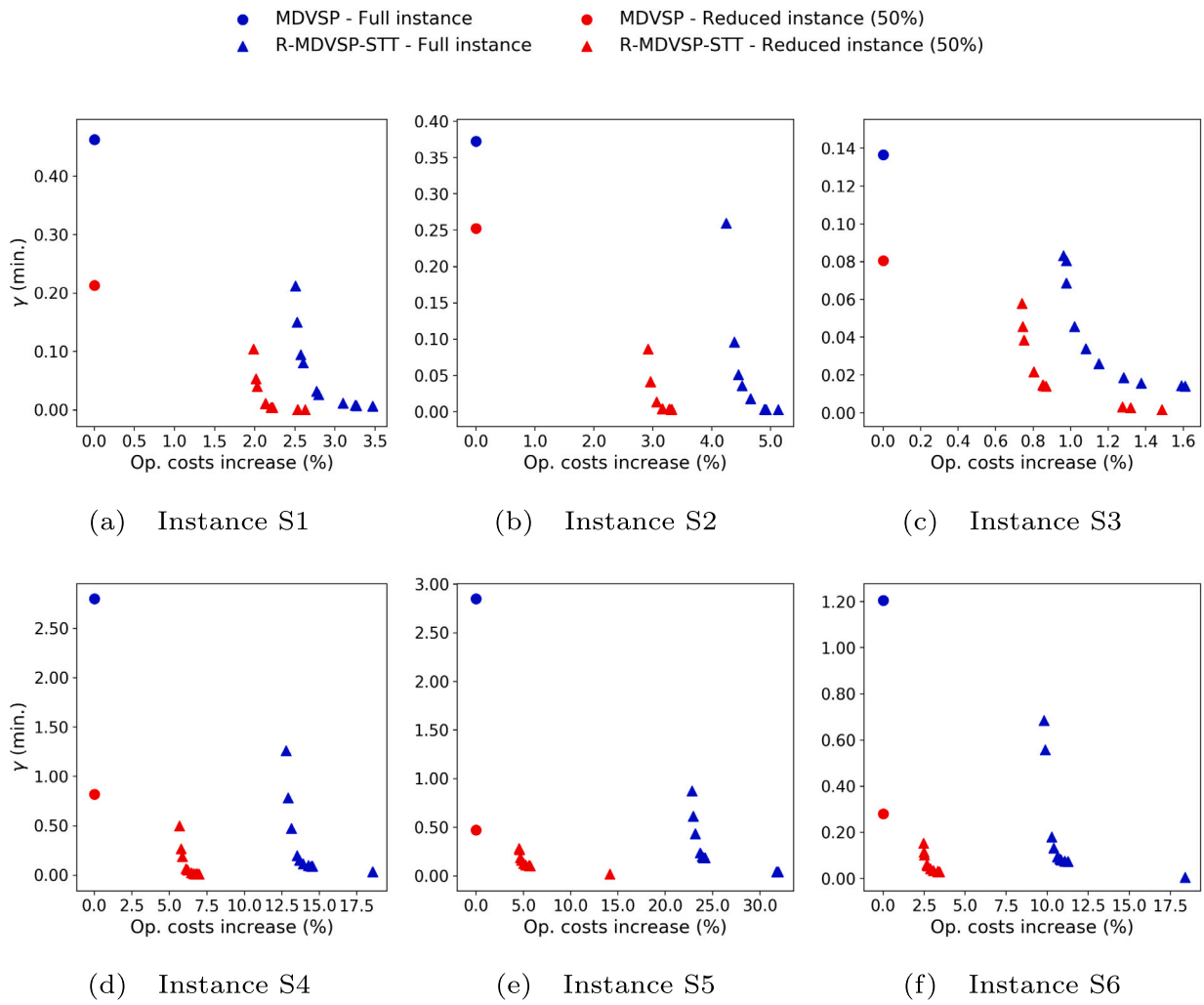


Fig. 9. Additional results on small, single-line, single-depot instances.

the R-MDVSP-STT solutions are more reliable in terms of the three reliability metrics than the corresponding solutions of the MDVSP. A large majority 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 both simulation cases. 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.

CRediT authorship contribution statement

Léa Ricard: Conceptualization, Formal analysis, Methodology, Software, Writing – original draft. **Guy Desaulniers:** Supervision, Writing – review & editing. **Andrea Lodi:** Supervision, Writing – review & editing. **Louis-Martin Rousseau:** Supervision, Writing – review & editing.

Declaration of competing interest

None for all authors.

Data availability

The authors do not have permission to share data.

Acknowledgments

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 work 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. The work of the first and third authors was supported by the Canada Excellence Research Chair in "Data Science for Real-Time Decision-Making".

Appendix A. Overview of the notation of the R-MDVSP-STT

Notation	Description
Set	
$\mathcal{V}$	set of n timetabled trips
$\mathcal{D}$	set of depots
$\mathcal{S}$	set of feasible vehicle schedules
$\mathcal{S}^d$	set of feasible vehicle schedules starting and ending at depot $d \in \mathcal{D}$
V_d	node set of network G^d for depot $d \in \mathcal{D}$
A_d	arc set of network G^d for depot $d \in \mathcal{D}$
Parameter	
b_d	capacity of depot $d \in \mathcal{D}$
β	weighting factor of the two objectives of the R-MDVSP-STT
$\kappa_{i,j}$	deadhead travel time between the end and starting locations of trips $i, j \in \mathcal{V}$, respectively
$\kappa_{d,i}$	deadhead travel time between the depot $d \in \mathcal{D}$ and the starting location of trip $i \in \mathcal{V}$
$\kappa_{i,d}$	deadhead travel time between the end location of trip $i \in \mathcal{V}$ and the depot $d \in \mathcal{D}$
τ	minimum layover time between two trips
d_i^0	departure time of trip $i \in \mathcal{V}$
d_i^1	arrival time of trip $i \in \mathcal{V}$
a_{is}	binary parameter equal to 1 if schedule $s \in \mathcal{S}$ covers trip $i \in \mathcal{V}$
c_s	cost of vehicle schedule $s \in \mathcal{S}$
c_{ij}^s	cost of arc (i, j) when covered by schedule $s \in \mathcal{S}$
η	cost per vehicle used
r^T	cost per minute of travel (either deadhead or pull-in/pull-out trips)
r^W	cost per minute of waiting outside depot
q_i^s	cost associated with potential delays in trip $i \in \mathcal{V}$ covered by schedule $s \in \mathcal{S}$
Δ	maximum waiting time between two trips in $\mathcal{V}$
α_i	relative passenger flow on trip $i \in \mathcal{V}$
$\tilde{c}_s$	reduced cost of schedule $s \in \mathcal{S}$
$\tilde{c}_{ij}^s$	reduced cost of arc (i, j) when covered by schedule $s \in \mathcal{S}$
C_i^p	accumulated reduced cost of path p ending at node $i \in V$
η_i^+, η_i^-	penalties in the objective function for under- and over-covering trip $i \in \mathcal{V}$
ξ_i^+, ξ_i^-	upper bounds of η_i^+ and η_i^-
Decision variable	
y_s	binary variable equal to 1 if schedule $s \in \mathcal{S}$ is selected
$(u_i)_{i\in\mathcal{V}}$	dual variables associated with constraints (2)
$(\pi_d)_{d\in\mathcal{D}}$	dual variables associated with constraints (3)
η_i^+, η_i^-	perturbation variables that allow limited under- and over-covering of trip $i \in \mathcal{V}$
Random variable	
X_i^s	random variable of the secondary delay of trip $i \in \mathcal{V}$ covered by schedule $s \in \mathcal{S}$
$\dot{T}_i$	random variable of the actual travel time of trip $i \in \mathcal{V}$
T_i	random variable with discrete and finite supports of the actual travel time of trip $i \in \mathcal{V}$
Y_i^s	random variable of the actual departure time of trip $i \in \mathcal{V}$ covered by schedule $s \in \mathcal{S}$
Function	
$\dot{h}_i(t)$, $h_i(t)$	PDF of $\dot{T}_i$ and T_i
$f_i^s(y)$	PDF of Y_i^s
$F_i^s(z)$	CDF of Y_i^s
$g_i^s(x)$	PDF of X_i^s

Appendix B. Derivation of the cumulative distribution functions of the actual departure time

Let a path p containing trips i and j, such that trip i precedes trip j. The equation for $F_j^p(z)$, the cumulative distribution function of Y_j^p, can be derived from Eq. (7) as

$$F_j^p(z) = \sum_{y=d_j^0}^{z} f_j^p(y) \tag{27}$$

$$= \left[\sum_{k\in\Phi_i} h_i(k) \times \sum_{y'=d_i^0}^{d_j^0-\kappa_{i,j}-\tau-k} f_i^{p'}(y')\right] + \sum_{y=d_j^0+1}^{z}\left[\sum_{k\in\Phi_i} h_i(k) \times f_i^{p'}(y-\kappa_{i,j}-\tau-k)\right] \tag{28}$$

$$= \sum_{k\in\Phi_i} h_i(k)\left[\sum_{y'=d_i^0}^{d_j^0-\kappa_{i,j}-\tau-k} f_i^{p'}(y') + \sum_{y=d_j^0+1}^{z} f_i^{p'}(y-\kappa_{i,j}-\tau-k)\right] \tag{29}$$

$$= \sum_{k\in\Phi_i} h_i(k) \sum_{y'=d_i^0}^{z-\kappa_{i,j}-\tau-k} f_i^{p'}(y') \tag{30}$$

$$= \sum_{k\in\Phi_i} h_i(k) \times F_i^{p'}(z-\kappa_{i,j}-\tau-k). \tag{31}$$

The term on the right-hand side of (29) reduces to $\sum_{y'=d_i^0}^{z-\kappa_{i,j}-\tau-k} f_i^{p'}(y')$ because

$$\sum_{y'=d_i^0}^{d_j^0-\kappa_{i,j}-\tau-k} f_i^{p'}(y') = f_i^{p'}(d_i^0) + \cdots + f_i^{p'}(d_j^0-\kappa_{i,j}-\tau-k), \tag{32}$$

and

$$\sum_{y=d_j^0+1}^{z} f_i^{p'}(y-\kappa_{i,j}-\tau-k) = f_i^{p'}(d_j^0+1-\kappa_{i,j}-\tau-k) + \cdots + f_i^{p'}(z-\kappa_{i,j}-\tau-k). \tag{33}$$

Appendix C. Additional results

Fig. 10 compares the results for the MDVSP, R-MDVSP-STT and the MDVSP with hard minimum buffer time that were obtained for instances I4 and I5 under the first simulation case.

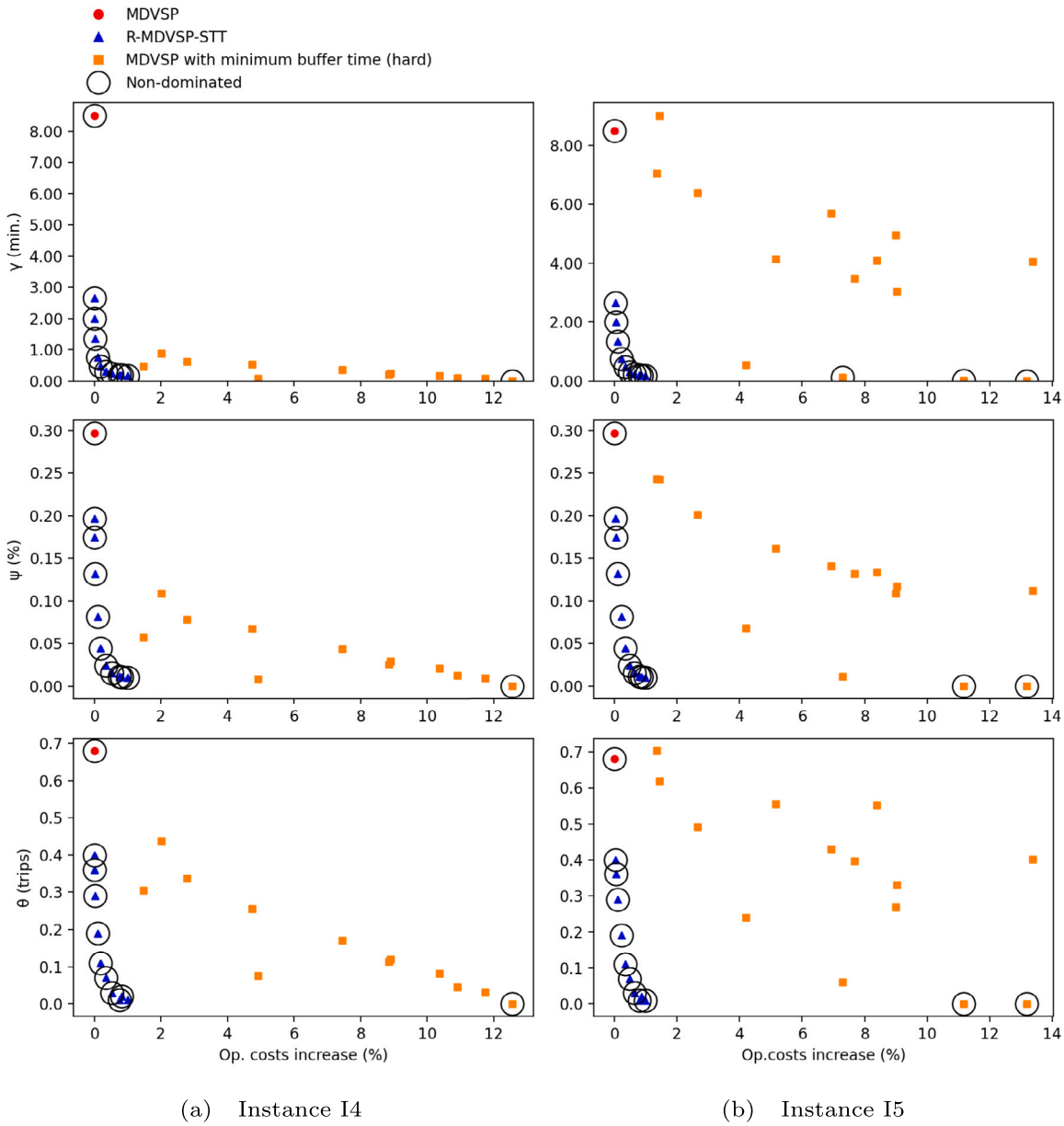


Fig. 10. Reliability metrics for the first simulation case.

References

[1] Desaulniers G, Hickman MD. Public transit. In: Barnhart C, Laporte G, editors. Transportation. Handbooks in operations research and management science, vol. 14, Elsevier; 2007, p. 69–127. http://dx.doi.org/10.1016/S0927-0507(06)14002-5.

[2] Ibarra-Rojas O, Delgado F, Giesen R, Muñoz J. Planning, operation, and control of bus transport systems: A literature review. Transp Res B 2015;77:38–75. http://dx.doi.org/10.1016/j.trb.2015.03.002.

[3] Bertossi AA, Carraresi P, Gallo G. On some matching problems arising in vehicle scheduling models. Networks 1987;17(3):271–81. http://dx.doi.org/10.1002/net.3230170303.

[4] Amberg B, Amberg B, Kliewer N. Increasing delay-tolerance of vehicle and crew schedules in public transport by sequential, partial-integrated and integrated approaches. Procedia - Soc Behav Sci 2011;20:292–301. http://dx.doi.org/10.1016/j.sbspro.2011.08.035.

[5] Naumann M, Suhl L, Kramkowski S. A stochastic programming approach for robust vehicle scheduling in public bus transport. Procedia - Soc Behav Sci 2011;20:826–35. http://dx.doi.org/10.1016/j.sbspro.2011.08.091.

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

[7] Kramkowski S, Kliewer N, Meier C. Heuristic methods for increasing delay-tolerance of vehicle schedules in public bus transport. In: VIII metaheuristics international conference. 2009.

[8] Amberg B, Amberg B, Kliewer N. Robust efficiency in urban public transportation: Minimizing delay propagation in cost-efficient bus and driver schedules. Transp Sci 2019;53(1):89–112. http://dx.doi.org/10.1287/trsc.2017.0757.

[9] Ricard L, Desaulniers G, Lodi A, Rousseau L-M. Predicting the probability distribution of bus travel time to measure the reliability of public transport services. Transp Res C 2022;138:103619. http://dx.doi.org/10.1016/j.trc.2022.103619.

[10] Dell'Amico M, Fischetti M, Toth P. Heuristic algorithms for the multiple depot vehicle scheduling problem. Manage Sci 1993;39(1):115–25.

[11] Carpaneto G, Dell'Amico M, Fischetti M, Toth P. A branch and bound algorithm for the multiple depot vehicle scheduling problem. Networks 1989;19(5):531–48. http://dx.doi.org/10.1002/net.3230190505.

[12] Ribeiro CC, Soumis F. A column generation approach to the multiple-depot vehicle scheduling problem. Oper Res 1994;42(1):41–52.

[13] Löbel A. Vehicle scheduling in public transit and Lagrangean pricing. Manage Sci 1998;44(12):1637–49.

[14] Fischetti M, Lodi A, Toth P. A branch-and-cut algorithm for the multiple depot vehicle scheduling problem. 1999, Dipartimento di Elittronica e Informatica, Università di Padova, Italy.

[15] Hadjar A, Marcotte O, Soumis F. A branch-and-cut algorithm for the multiple depot vehicle scheduling problem. Oper Res 2006;54(1):130–49. http://dx.doi.org/10.1287/opre.1050.0240.
[16] Kliewer N, Mellouli T, Suhl L. A time–space network based exact optimization model for multi-depot bus scheduling. European J Oper Res 2006;175(3):1616–27. http://dx.doi.org/10.1016/j.ejor.2005.02.030.
[17] Desaulniers G, Lavigne J, Soumis F. Multi-depot vehicle scheduling problems with time windows and waiting costs. European J Oper Res 1998;111(3):479–94. http://dx.doi.org/10.1016/S0377-2217(97)00363-9.
[18] Li L, Lo HK, Xiao F. Mixed bus fleet scheduling under range and refueling constraints. Transp Res C 2019;104:443–62. http://dx.doi.org/10.1016/j.trc.2019.05.009.
[19] Rinaldi M, Picarelli E, D'Ariano A, Viti F. Mixed-fleet single-terminal bus scheduling problem: Modelling, solution scheme and potential applications. Omega 2020;96:102070. http://dx.doi.org/10.1016/j.omega.2019.05.006.
[20] Zhou G-J, Xie D-F, Zhao X-M, Lu C. Collaborative optimization of vehicle and charging scheduling for a bus fleet mixed with electric and traditional buses. IEEE Access 2020;8:8056–72. http://dx.doi.org/10.1109/ACCESS.2020.2964391.
[21] Bunte S, Kliewer N. An overview on vehicle scheduling models. Public Transp : Plan Oper 2010;1(4):299–317. http://dx.doi.org/10.1007/s12469-010-0018-5.
[22] Pillac V, Gendreau M, Guéret C, Medaglia AL. A review of dynamic vehicle routing problems. European J Oper Res 2013;225(1):1–11. http://dx.doi.org/10.1016/j.ejor.2012.08.015.
[23] Huisman D, Freling R, Wagelmans APM. A robust solution approach to the dynamic vehicle scheduling problem. Transp Sci 2004;38(4):447–58. http://dx.doi.org/10.1287/trsc.1030.0069.
[24] He F, Yang J, Li M. Vehicle scheduling under stochastic trip times: An approximate dynamic programming approach. Transp Res C 2018;96:144–59. http://dx.doi.org/10.1016/j.trc.2018.09.010.
[25] Shen Y, Xu J, Li J. A probabilistic model for vehicle scheduling based on stochastic trip times. Transp Res B 2016;85:19–31. http://dx.doi.org/10.1016/j.trb.2015.12.016.
[26] van Kooten Niekerk ME. Optimizing for reliable and sustainable public transport [Ph.D. thesis], Netherlands: Univeristeit Utrecht; 2018.
[27] Hwang C-L, Masud ASM. Multiple objective decision making — methods and applications. Heidelberg: Springer; 1979, http://dx.doi.org/10.1007/978-3-642-45511-7.
[28] Yetiskul E, Senbil M. Public bus transit travel-time variability in Ankara (Turkey). Transp Policy 2012;23:50–9. http://dx.doi.org/10.1016/j.tranpol.2012.05.008.
[29] Mazloumi E, Currie G, Rose G. Using GPS data to gain insight into public transport travel time variability. J Transp Eng 2010;136(7):623–31. http://dx.doi.org/10.1061/(ASCE)TE.1943-5436.0000126.
[30] Errico F, Desaulniers G, Gendreau M, Rei W, Rousseau L-M. The vehicle routing problem with hard time windows and stochastic service times. EURO J Transp Logist 2018;7(3):223–51. http://dx.doi.org/10.1007/s13676-016-0101-4.
[31] Irnich S, Desaulniers G. Shortest path problems with resource constraints. In: Desaulniers G, Desrosiers J, Solomon MM, editors. Column generation. Springer; 2005, p. 33–65. http://dx.doi.org/10.1007/0-387-25486-2_2.
[32] Lübbecke ME, Desrosiers J. Selected topics in column generation. Oper Res 2005;53(6):1007–23. http://dx.doi.org/10.1287/opre.1050.0234.
[33] Barnhart C, Johnson EL, Nemhauser GL, Savelsbergh MWP, Vance PH. Branch-and-price: Column generation for solving huge integer programs. Oper Res 1998;46(3):316–29. http://dx.doi.org/10.1287/opre.46.3.316.
[34] Wellman MP, Ford M, Larson K. Path planning under time-dependent uncertainty. 2013, arXiv:1302.4987.
[35] Boland N, Dickson S, Savelsbergh M, Smilowitz K. Dominance in pricing problems with stochasticity. 2015, [Accessed 22 February 2022].
[36] Ahuja RK, Magnanti TL, Orlin JB. Network flows: theory, algorithms, and applications. Prentice Hall; 1993.
[37] Charnes A. Optimality and degeneracy in linear programming. Econom Soc 1952;20(2):160–70.
[38] Thomopoulos N. Essentials of Monte Carlo simulation: Statistical methods for building simulation models. New York: Springer; 2012.
[39] Stevens A. Monte-Carlo simulation: An introduction for engineers and scientists. CRC Press; 2022.
[40] Chen X, Yu L, Zhang Y, Guo J. Analyzing urban bus service reliability at the stop, route, and network levels. Transp Res A 2009;43(8):722–34. http://dx.doi.org/10.1016/j.tra.2009.07.006.
[41] Ma Z, Ferreira L, Mesbah M. Measuring service reliability using automatic vehicle location data. Math Probl Eng 2014;2014:1–12. http://dx.doi.org/10.1155/2014/468563.
[42] van Oort N. Incorporating service reliability in public transport design and performance requirements: International survey results and recommendations. Res Transp Economics 2014;48:92–100. http://dx.doi.org/10.1016/j.retrec.2014.09.036.
[43] Currie G, Douglas N, Kearns I. An assessment of alternative bus reliability indicators. In: Australasian transport research forum, ATRF 2012 - proceedings. 2014.
[44] Cham L. Understanding bus service reliability : A practical framework using AVL/APC data. 2006.
[45] van Oort N. Service reliability and urban public transport design [Ph.D. thesis], Netherlands: Delft University of Technology; 2011.