Seamless Multimodal Transportation Scheduling

Raghunathan, Arvind; Bergman, David; Hooker, John; Serra, Thiago; Kobori, Shingo

TR2023-076   July 04, 2023

Abstract

Ride-hailing services have expanded the role of shared mobility in passenger transportation systems, creating new markets and creative planning solutions for major urban centers. In this paper, we consider their use for the first-mile or last-mile passenger transportation in coordination with a mass transit service to provide a seamless multimodal transportation experience for the user. A system that provides passengers with predictable information on travel and waiting times in their commutes is immensely valuable. We envision that the passengers will inform the system of their desired travel and arrival windows so that the system can jointly optimize the schedules of passengers. The problem we study balances minimizing travel time and the number of trips taken by the last-mile vehicles, so that long-term planning, maintenance, and environmental impact are all taken into account. We focus on the case where the last-mile service aggregates passengers by destination. We show that this problem is NP-hard, and propose a decision diagram-based branch-and-price decomposition model that can solve instances of real-world size (10,000 passengers spread over an hour, 50 last-mile destinations, 600 last-mile vehicles) in computational time (about 1 minute) that is orders-of-magnitude faster than other methods appearing in the literature. Our experiments also indicate that aggregating passengers by destination on the last-mile service provides high-quality solutions to more general settings.

Informs Journal on Computing 2023

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Seamless Multimodal Transportation Scheduling

Arvind U. Raghunathan
Mitsubishi Electric Research Labs, Cambridge, MA, 02139, USA
raghunathan@merl.com

David Bergman
Operations and Information Management, University of Connecticut, Storrs, Connecticut 06260, USA
david.bergman@uconn.edu

John N. Hooker
Tepper School of Business, Carnegie Mellon University, Pittsburgh, PA, 15213, USA
jh38@andrew.cmu.edu

Thiago Serra
Freeman College of Management, Bucknell University, Lewisburg, PA, 17837, USA
thiago.serra@bucknell.edu

Shingo Kobori
Advanced Technology R&D center, Mitsubishi Electric Corporation , Hyogo, 661-8661, Japan
Kobori.Shingo@cj.MitsubishiElectric.co.jp

Ride-hailing services have expanded the role of shared mobility in passenger transportation systems, creating new markets and creative planning solutions for major urban centers. In this paper, we consider their use for the first-mile or last-mile passenger transportation in coordination with a mass transit service to provide a seamless multimodal transportation experience for the user. A system that provides passengers with predictable information on travel and waiting times in their commutes is immensely valuable. We envision that the passengers will inform the system of their desired travel and arrival windows so that the system can jointly optimize the schedules of passengers. The problem we study balances minimizing travel time and the number of trips taken by the last-mile vehicles, so that long-term planning, maintenance, and environmental impact are all taken into account. We focus on the case where the last-mile service aggregates passengers by destination. We show that this problem is NP-hard, and propose a decision diagram-based branch-and-price decomposition model that can solve instances of real-world size (10,000 passengers spread over an hour, 50 last-mile destinations, 600 last-mile vehicles) in computational time ($\sim$ 1 minute) that is orders-of-magnitude faster than other methods appearing in the literature. Our experiments also indicate that aggregating passengers by destination on the last-mile service provides high-quality solutions to more general settings.

Key words: last-mile; mass transit; scheduling; decision diagrams; branch and price

1. Introduction

Shared mobility is gradually changing how people live and interact in urban centers (Savelsbergh and Van Woensel 2016). According to McKinsey & Company, the shared mobility market for China, Europe, and the United States totalled almost $54 billion in 2016, and it is expected to grow at least 15% annually over the next 15 years (Grosse-Ophoff et al. 2017). There is wide interest in integrating these emerging modes of transportation with public transit systems, as is indicated by the U.S. Department of Transportation (McCoy et al. 2018), with interest expressed for collaboration by other key stakeholders such as companies and public sector entities signing the Shared Mobility Principles for Livable Cities (Chase 2017). In the future, autonomous vehicles may be combined with other modes of transportation to reduce traffic congestion and the need for car ownership.

In this paper, we consider the use of shared vehicles such as a bus, van, or taxi in the last-mile passenger transportation, a particular form of transportation on demand (Cordeau et al. 2007). Last-mile transportation is defined as a service that delivers people from a hub of mass transit service to each passenger's final destination. Mass transit services can comprise air, boat, bus, or train. The last-mile service can be facilitated by bike (Liu et al. 2012), car (Shaheen 2004, Thien 2013), autonomous pods (Shen et al. 2017), or personal rapid transit systems. Although last-mile may also refer to the movement of goods in supply chains, home-delivery systems, and telecommunications, we will restrict our attention in this paper exclusively to the transportation of people. A last-mile service expands the access of mass transit to an area wider than that defined as "walking distance" of a transportation hub. Interest in the design and operation of last-mile services has grown tremendously in the past decade. This has been driven primarily by three factors (Wang 2017): (i) governmental push to reduce congestion and air pollution; (ii) increasing aging population in cities; and (iii) providing mobility for the differently abled and school children.

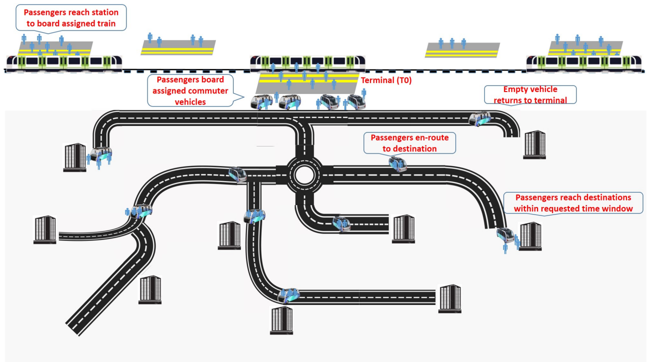
Passengers reach station to board assigned train
Passengers board assigned commuter vehicles
Terminal (T0)
Empty vehicle returns to terminal
Passengers en-route to destination
Passengers reach destinations within requested time window

Figure 1 Schematic of an integrated last-mile system (images licensed from shutterstock.com and alamy.com).

Figure 1 shows a typical scenario for the operation of mass transit in conjunction with a last-mile service. All passengers start their journey from mass transit stations served, for example, by a train, and request transportation to their destinations within a time-window. Destinations can represent offices in specific buildings or a location, such as a bus stop, that serves multiple buildings that are easily accessed on foot. These destinations can be accessed by paths that may be exclusive to last-mile vehicles or by means of existing road infrastructure. For convenience, we will refer to the vehicle providing last-mile service as a *commuter vehicle* (CV). The CVs are typically parked at a terminal ($\mathbf{T}_0$) at which passengers arrive from mass transit services and proceed to their respective destination buildings by sharing a ride in a CV. The last-mile service may represent the morning commute to the office or the evening commute back to residences. Once all passengers are delivered to their common destination, it is assumed that the CVs return back to the terminal for subsequent trips. Namely, we make the following assumptions: (1) all passengers of a trip share the same destination; and (2) CVs return to their depot after a trip.

The particular system that we describe is currently being planned for the city of Milton-Keynes in the United Kingdom. The project called UK Autodrive (`http://www.ukautodrive.com/`) trialled the use of autonomous pods for the passenger transportation within the city. In particular, the city officials in Milton-Keynes are looking to deploy this system for transport of passengers between the Milton-Keynes train station and the city limits[1].

Such an application is timely and well suited for integrated prescriptive analytics. This type of system is already realizable in practice by integrating with ride-hailing services such as conventional taxi services, Uber, and Lyft. Moreover, the peak use of the system is expected to coincide with morning and afternoon work commutes, which can be made predictable by design: work commuters often know in advance when they would like to arrive at work and would be willing to provide such information in advance for better service. Note that this assumption on operation mode does not preclude a real-time decision making process. One could adopt a rolling horizon approach, fixing decisions for previously scheduled passengers and re-optimizing for newly arrived requests as often as needed. Finally, note that the *first-mile* operation, wherein the passengers first ride on CVs to reach a hub of a mass transit service, can easily be accommodated in an analogous manner, as long as there are shared pickup locations that will serve as hubs.

[1] `http://www.ukautodrive.com/pods-provide-a-first-last-mile-solution-in-milton-keynes/`

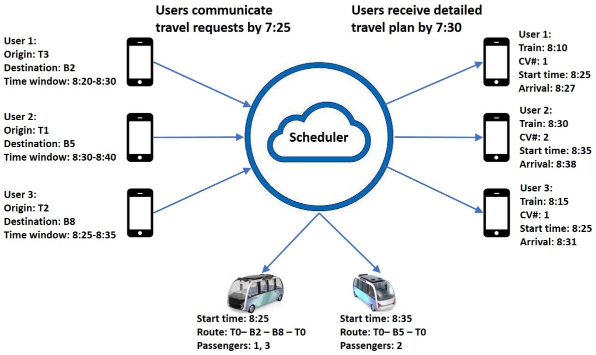

Figure 2 **Schematic representation of the interaction between the passengers in the system, the scheduler, and the CVs (images licensed from shutterstock.com and alamy.com)**

Therefore, we envision an operational scenario such as in Figure 2, where passengers indicate their station of origin, destination, and the desired time-window of arrival at the destination. This information is assumed to be available to the scheduler only slightly in advance of scheduling decisions. For instance, consider the situation where the destinations represent buildings with offices and the passengers enter requests through a smartphone app.Once all requests have been received for a certain time-period, the scheduler determines for each user: (i) the mass transit trip to board at their station of origin; (ii) the CV to board at $\mathbf{T}_0$; (iii) the time the CV will depart from $\mathbf{T}_0$; and (iv) the time of arrival at the destination. The scheduler also communicates to the different CVs the routes, start times, and list of passengers. The choice of route determines the times of arrival of passengers at their destination as well as total travel time.

This application characterizes what we denote as the Integrated Last-Mile Transportation Problem (ILMTP). The ILMTP is defined as the problem of scheduling passengers jointly on mass transit and last-mile services so that the passengers reach their destinations within specified time-windows. We propose to minimize a linear combination of the total transit time for all passengers and the number of trips required. The former quantity captures the quality of service provided; the latter addresses fuel consumption, long-run operational costs, and environmental considerations. Transit time includes the time spent traveling in both transportation modes and waiting between both services. The solution of the ILMTP also specifies the set of passengers that share a ride in a CV. The time spent by the passengers in the CV also depends on the co-passengers, where we focus on a variant simplified as discussed below.

1.1. Contributions of this paper

This paper proposes algorithms that are scalable to multimodal transportation systems consisting of a central depot connecting a mass transit system with multiple small-capacity vehicles. Our experimental evaluation on instances generated to emulate what might be realized in real-world settings indicates that the algorithms developed can obtain optimal solutions in one minute to problems with 10,000 passengers, 50 destinations, and 600 vehicles that commonly utilized techniques are not able to solve in reasonable time. We also obtain high-quality solutions to more general problems than the one we focus on, finding heuristic solutions in seconds to problem instances for which other approaches in the literature cannot find a feasible solution in three hours.

More specifically, we extend results first described in Raghunathan et al. (2018), where a single-destination-per-trip (SDPT) assumption is also imposed. We note that this assumption can be regarded as a technical constraint of the business model. First, users do not want to be delayed due to other passengers leaving the car in previous stops, especially if the distance between these stops is walkable. Second, the literature on last-mile transportation regards the use of existing destinations, such as bus stops, as an effective aggregator of individual destinations (Stiglic et al. 2015, Mahéo et al. 2018). We can observe some instances of this logic in practice, such as the new Uber Express POOL service (Uber 2018), which embodies these considerations by creating pooled pickup and drop-off locations.

Aligned with (Raghunathan et al. 2018), we make the following three assumptions on the operation of the system:

- The trains serves station along the line and have a deterministic travel between the stations.
- Each CV trip is a round trip from the terminal $\mathbf{T}_0$ to one of the buildings in $\mathcal{D}$.
- The passengers have identical time window on arrival at destination.

The above assumptions imply that the ILMTP is modeled and solved for each terminal station independently when applied to a realistic transportation network.

The main contributions of the paper can be summarized as follows:

- **Computational Complexity:** We show that the ILMTP with the SDPT assumption, which, for simplicity, we refer to simply as the ILMTP-SD unless otherwise noted, is *NP-Hard.* This shows that the simplifying assumption does not render the problem computationally easier to solve and developing an efficient algorithm is necessary.

- **Structure of Optimal Solutions:** The optimal solutions to the ILMTP-SD are shown to satisfy an ordering property. This result is a restatement of Raghunathan et al. (2018)[Theorem 1], included for completeness. We also show that the number of solutions to the ILMTP-SD grows exponentially with the number of passengers.
- **Decision Diagram (DD)-based Algorithm:** Based on the structure of optimal solutions, we describe a novel optimization algorithm based on a DD representation of the space of solutions to the ILMTP-SD. Our approach builds on a state-space decompositions (Bergman and Cire 2016, 2018), which is an outgrowth of the growing body of literature on DD-based optimization (Bergman et al. 2011, 2016).
- **Branch-and-price DD decomposition:** We improve the performance of the proposed DD algorithm by developing a branch-and-price scheme (Barnhart et al. 1998) through which columns are generated from paths in the DDs. This scheme could be easily adapted to other problems in which such a DD-based algorithm is devised.
- **Numerical Evaluation**: An experimental evaluation indicates that the proposed model is orders-of-magnitude faster than existing techniques. We also report the performance of the proposed approach on instances where the assumptions on the ordering property are violated, namely (i) mass transit service also includes express trains and (ii) CVs can make multiple stops in the last-mile. We show that even in these settings the potential loss of optimality incurred by the SDPT assumption is insignificant for all practical purposes, in that existing models are often unable to find a single feasible solution in hours while the algorithm we propose prove optimality for the restricted version in seconds.Thus, existing approaches are inappropriate for real-time use in the ILMTP, especially when requests are received without much advance notice. The proposed method is capable of optimizing this problem and is appropriate as a real-time heuristic.

1.2. Related work

The ILMTP can be broadly viewed as an instance of routing and scheduling with time-windows. We survey the relevant literature and describe the key differences with this paper.

The literature on last-mile transportation is mostly focused on last-mile service, without much consideration to the mass transit system. Seminal work in this area dates back to the 1960s, focused mostly on freight transportation (see Wang (2017) for a discussion). Wang (2017) is the first work to consider routing and scheduling in the last-mile, where the minimization of total travel time is considered and the author proposes a heuristic approach. The ILMTP is a strict generalization of Wang (2017), in that we consider time-windows for arrival and scheduling on the mass transit service. An alternative approach to exact last-mile integration consists of simulation-based approaches (Pinto et al. 2018, Ömer Verbas et al. 2016) as well as last-mile dispatching policies (van Heeswijk et al. 2017, Agussurja et al. 2018). This is complemented by another stream of work on uncertainty of mass transit service (Boyan et al. 2001, Häme and Hakula 2013). More recently, Mahéo et al. (2018) approached the design of a public transit system that includes multiple modes of transportation, however the authors did not consider scheduling aspects. This paper expands this literature with an optimal approach to scheduling passengers in the last-mile~~.~~, and complements the work of Mahéo et al. (2018) by focusing exclusively on scheduling.

Personal Rapid Transit (PRT) has similarities to the last-mile problem. Research has been conducted on PRT system control frameworks (Anderson 1998), financial assessments (Bly and Teychenne 2005, Berger et al. 2011), performance approximations (Lees-Miller et al. 2009, 2010), and case studies (Mueller and Sgouridis 2011). However, none of these papers have addressed last-mile operational issues.

On the other hand, a large body of research has been devoted to Demand Responsive Transit (DRT), which is another type of on-demand service. Some papers focus on DRT concept discussions, practical implementation, and assessment of simulations in case studies (Brake et al. 2004, Horn 2002b, Mageean and Nelson 2003, Palmer et al. 2004, Quadrifoglio et al. 2008). Models have been developed to assist in system design and regulation (e.g., Daganzo (1978), Diana et al. (2006), Wilson and Hendrickson (1980)). Routing options in specific contexts have also been considered (Chevrier et al. 2012, Horn 2002a). The ILMTP can be viewed as a specific variant of a broadly defined DRT concept—namely, a DRT system that addresses last-mile service requests with batch passenger demand and a shared passenger origin. The same applies to ride-sharing models more generally (Agatz et al. 2012). Unlike most papers in the DRT literature, we focus on scheduling the mass transit service and the last-mile optimization from an operational perspective.

A much broader stream of related work consists of vehicle routing problems (VRPs), which have long been studied and comprise a large body of literature. The VRP with time windows (VRPTW) has been the subject of intensive study, with many heuristic and exact optimization approaches suggested in the literature (Toth and Vigo 2014). The dial-a-ride problem (DARP) and related variations have also been extensively investigated (Cordeau and Laporte 2007, Jaw et al. 1986, Lei et al. 2012). As argued by Wang (2017), the VRPTW focuses on reducing operating costs while the ILMTP aims to improve the level-of-service by minimizing total passenger transit time, especially with single-destination routes. The typical size of the problems that can be solved to optimality for the VRPTW and DARP are on the order of hundreds of requests and dozens of vehicles. This is far smaller than the size of the instances that we solve to optimality in this paper. Although solving the VRPTW and DARP optimally is difficult for large-size instances, good heuristics exist for these problems, which include the Savings algorithm (Clarke and Wright 1964), its variants, and insertion heuristics (Vigo 1996, Salhi and Nagy 1999, Campbell and Savelsbergh 2004). In a recent work, Riley et al. (2019) propose a column generation method for scheduling passengers on ride-sharing vehicles which scales to 30,000 requests per hour, but does not consider the aspect of mass transit service and time-windows.

Finally, the computational approach introduced in this paper calls for modeling problems using disjoint and connected DDs. DD-based optimization is an emerging field within computational optimization (Andersen et al. 2007, Bergman et al. 2011, Gange et al. 2011, Cire and van Hoeve 2013, Bergman et al. 2016, Perez and Régin 2017). The idea used in this paper is to model a problem with a set of DDs, from which a solution is obtained by a network flow reformulation based on such DDs (Bergman and Cire 2016, 2018). This paper proposes a path-based model for solving the resulting network-flow reformulation, which is similar to the approach by Morrison et al. (2016) for the vertex coloring problem. To the best of our knowledge, this paper is the first application to multi-valued DDs.

In summary, the ILMTP has the following features that distinguishes it from previous studies in the literature:

- joint scheduling of passengers on mass transit systems and last-mile services;
- consideration of time-windows on arrival at destination;
- common last-mile origin (which is also the vehicle depot); and
- weighted minimization of the total passenger transit time and number of CV trips.

The ILMTP models real-world transportation systems that are prevalent across the globe, and this paper provides a mechanism for optimal operational decisions.

2. Problem Description and Mathematical Formulation

In this section we provide a formulation of the ILMTP-SD. Prior to that, we describe the different elements in the ILMTP-SD such as the mass transit system, last-mile vehicles, destinations, passenger requests and associated parameters such as the travel time associated with the transportation services, and time windows for arrival.

Mass transit system: For the sake of convenience, we will assume that the mass transit is a train system. Let $\mathbf{T}_0$ be the *terminal station* that links a mass transit system with a last-mile service system. The mass transit system is described by a set of *trips*, denoted by $\mathcal{C}$. Each trip originates at a station in set $\mathcal{S}$ and ends at $\mathbf{T}_0$. The trips are regular in the sense that the train stops at all stations in $\mathcal{S}$ sequentially, with $\mathbf{T}_0$ as the last stop of each trip. The time a trip c leaves station $s \in \mathcal{S}$ is $\tilde{t}(c,s)$ and the time it arrives to the terminal is $\tilde{t}(c)$. This paper is only concerned with the moving portion of the mass transit commute and so it assumes that each passenger arrives at the station of origin at the time that the mass transit trip is departing that location. The time that a passenger waits in such stations is not of concern in our objective or constraints.

Destinations: Let $\mathcal{D}$ be the set of destinations where the CVs make stops,where we assume $\mathbf{T}_0 \in \mathcal{D}$. For each destination $d \in \mathcal{D}$, let $\tau(d)$ be the total time it takes a CV to depart $\mathbf{T}_0$, travel to d (denoted by $\tau^1(d)$), stop at d for passengers to disembark (denoted by $\tau^2(d)$), and return to $\mathbf{T}_0$ (denoted by $\tau^3(d)$). Therefore, $\tau(d) = \tau^1(d) + \tau^2(d) + \tau^3(d)$. Let $\mathcal{T} := \{1, \ldots, t^{\max}\}$ be an index set of the operation times of both systems. We assume that the time required to board passengers into the CVs is incorporated in $\tau^1(d)$. For simplicity, the boarding time is independent of the number of passengers. A passenger arrives to a destination $\tau^1(d)$ time units after departing from the terminal.

Last-mile system: Let V be the set of CVs.Let v^{cap} denote the maximum number of passengers that can be assigned to a single CV trip. Each CV trip consists of a set of passengers boarding the CV, traveling from $\mathbf{T}_0$ to a destination $d \in \mathcal{D}$, and then returning back to $\mathbf{T}_0$. Passengers on a common CV trip must request transportation to a common building. We also assume that each CV must be back at the terminal by time $t^{\max}$.

Passengers: Let $\mathcal{J}$ be the set of passengers. Each passenger $j \in \mathcal{J}$ requests train from a station $s(j) \in \mathcal{S}$ to $\mathbf{T}_0$, and then by CV from $\mathbf{T}_0$ to destination $d(j) \in \mathcal{D}$, to arrive at time $t^r(j)$. The set of passengers that request service to destination d is denoted by $\mathcal{J}(d)$. Let

$n := |\mathcal{J}|$ and $n_d := |\mathcal{J}(d)|$. Each passenger $j \in \mathcal{J}$ must arrive to $d(j)$ between $t^r(j) - T_w$ and $t^r(j) + T_w$.

Problem Statement: The ILMTP-SD is the problem of assigning train trips and CVs to each passenger so that the total transit time and the number of CV trips utilized is minimized. A solution therefore consists of a partition $\mathsf{g} = \{g_1, \ldots, g_\gamma\}$ of $\mathcal{J}$, with each group g_l associated with a departure time t_l^{g}, for $l = 1, \ldots, \gamma$, which indicates the time the CV carrying the passengers in g_l departs $\mathbf{T}_0$, satisfying all request time and operational constraints. For any passenger $j \in \mathcal{J}$, let $\mathsf{g}(j)$ be the group in g that j belongs to.

To balance the potentially conflicting objectives, the objective function we consider is a convex combination of objective terms, defined by α, with $0 \le \alpha \le 1$. Hence, we balance these objectives by using α times the waiting time plus $(1-\alpha)$ times the number of CVs, which is therefore used as our objective function, represented as $f(\alpha)$.

2.1. IP Model

In this section we present an improved IP model for the ILMTP-SD, which is based on the one from Raghunathan et al. (2018). Let $\mathcal{T}_j := [t^r(j) - T_w - \tau^1(d(j)), t^r(j) + T_w - \tau^1(d(j))]$ represent the possible times at which a passenger can depart on a CV from $\mathbf{T}_0$. The total travel time for each passenger j that travels on a CV that departs from $\mathbf{T}_0$ at time $t \in \mathcal{T}_j$ is

$$\mathrm{tt}_{j,t} := \left(t + \tau^1(d(j)) - \max_{c \in \mathcal{C}: \tilde{t}(c) \le t} \tilde{t}(c, s(j)) \right).$$

which is composed of travel time on the train, waiting time at $\mathbf{T}_0$, and CV travel time. Note that the passenger is assigned to the latest train that allows to reach $\mathbf{T}_0$ prior to departure on the CV. The variables in our model are as follows:

- $z_{j,t}$: indicator if passenger $j \in \mathcal{J}$ leaves $\mathbf{T}_0$ at time $t \in \mathcal{T}_j$
- n_t: number of CVs parked in $\mathbf{T}_0$ at time $t \in \mathcal{T}$
- $n_{d,t}$: number of CVs assigned to destination $d \in \mathcal{D}$ to depart $\mathbf{T}_0$ at time $t \in \mathcal{T}$

An optimization model for ILMTP-SD is as follows:

$$\min \quad f(\alpha) = \alpha \cdot \sum_{j \in \mathcal{J}} \sum_{t \in \mathcal{T}_j} \mathrm{tt}_{j,t} \cdot z_{j,t} + (1-\alpha) \cdot \sum_{d \in \mathcal{D}} \sum_{t \in \mathcal{T}} n_{d,t} \tag{IP.1}$$

$$\text{s.t.} \quad \sum_{t \in \mathcal{T}_j} z_{j,t} = 1, \qquad \forall j \in \mathcal{J} \tag{IP.2}$$

$$n_t = n_{t-1} + \sum_{d \in \mathcal{D}} n_{d,t-\tau(d)} - \sum_{d \in \mathcal{D}} n_{d,t} \qquad \forall t \in \mathcal{T} \tag{IP.3}$$

$$v^{\text{cap}} \cdot (n_{d,t} - 1) + 1 \leq \sum_{j \in \mathcal{J}(d)} z_{j,t} \leq v^{\text{cap}} \cdot n_{d,t} \qquad \forall d \in \mathcal{D}, \forall t \in \mathcal{T} \tag{IP.4}$$

$$z_{j,t} \in \{0,1\}, \qquad \forall j \in \mathcal{J}, \forall t \in \mathcal{T} \tag{IP.5}$$

$$n_t \geq 0, \qquad \forall t \in \mathcal{T} \tag{IP.6}$$

$$n_{d,t} \geq 0, \qquad \forall d \in \mathcal{D}, \forall t \in \mathcal{T} \tag{IP.7}$$

$$n_0 = |V|. \tag{IP.8}$$

The objective function, parametrized by $\alpha \in [0,1]$, balances the sum of the total travel time of all passengers with the number of CV trips that take place over the planning horizon. This objective function generalizes the one in Raghunathan et al. (2018) by also including the number of CV trips as an element of the objective function. The smaller α is, the more emphasis is placed on minimizing the number of trips a CV takes. Fewer CV trips results in fewer maintenance tasks as well as lower emissions, which is critical for long-term planning and for minimizing environmental impact. In the case of the ILMTP-SD with regular train trips, only the waiting time at the terminal varies across solutions taking the shortest route to each destination. Nevertheless, we use total travel time for consistency in the section on experiments where we relax SDPT and the regular train times assumption.

The first set of constraints in(IP.2) ensure each passenger is assigned to oneone CV trip. Constraints (IP.3) through (IP.4) bookmark the number of CVs in use at any given time. In contrast to previous formulations, the constant value 1 on constraint (IP.4) prevents empty CVs in feasible solutions. Constraints (IP.5) through (IP.7) enforce bounds, binary restrictions, and initial conditions, as necessary. Note that (IP) can be extended to handle time-dependent travel times on the CV by replacing the occurrence of $\tau^1(d)$, $\tau(d)$ in (IP) with CV travel times that are dependent on the time of departure from $\mathbf{T}_0$.

3. Complexity

It is known that generalizations of the ILMTP are NP-hard (Raghunathan et al. 2018). We show in this section that the ILMTP-SD is at least as hard, and in fact a much simpler version of the ILMTP-SD with a single mass transit service and CVs of unitary capacity is sufficient to define an NP-hard problem, with proof in Appendix Section A.

Theorem 1 *Deciding feasibility of an instances of the ILMTP-SD is NP-complete.*

Corollary 1 *ILMTP-SD is NP-hard.*

4. Structure of Optimal Solution to ILMTP-SD

In this section we restate a result from Raghunathan et al. (2018) uncovering a structural property of optimal solutions to the ILMTP-SD. This structural result enables us to construct a recursive model for solutions to the ILMTP-SD as will be discussed in § 5.1. We also prove a exponential lower bound on the number of solutions to the ILMTP-SD in Theorem 3. Note that the assumptions stated in Raghunathan et al. (2018) hold in the present context (see § 2).

Theorem 2 *For all $d \in \mathcal{D}$, let $\mathcal{J}(d) = \left\{j_1^d, \dots, j_{n_d}^d\right\}$ represent a partitioning of $\mathcal{J}$ by destination and let $\mathcal{J}(d)$ be ordered so that, for $1 \le i \le n_d - 1$, $t^r(j_i^d) \le t^r(j_{i+1}^d)$. There exists an optimal solution for which there are no triples of passengers $j_{i_1}^d, j_{i_2}^d, j_{i_3}^d$ with $i_1 < i_2 < i_3$ for which $j_{i_1}^d$ and $j_{i_3}^d$ share a common CV trip without $j_{i_2}^d$.*

Theorem 2 indicates that one need only search for solutions which group passengers in CV trips sequentially by order of requested arrival times. We will exploit this result in order to create compact DDs for each destination. The proof of Theorem 2 follows from the proof of Theorem 1 from Raghunathan et al. (2018), where we note that in the exchange argument, no CV trips are added or deleted. We provide the proof in Appendix B.

4.1. Exponential Lower Bound on Number of Solutions

By Theorem 2, the search for optimal solutions can be restricted to groups of passengers that are consecutive when ordered by deadlines. Theorem 1 shows that the problem remains NP-hard even over these solutions. Theorem 3 relates that number of partitions of passengers going to the same destination to the Fibonacci series establishing a exponential lower bound on the number of feasible solutions even in this restricted set.

Theorem 3 *Let $\phi(n)$ be the number of partitions of n passengers into groupings, each containing passengers with consecutive deadlines and going to a common destination. If the time windows and requested arrivals times are such that every pair of consecutive passengers can travel with one another on a common CV, then the number of partitions of passengers into groups is bounded below by the $(n+1)$st Fibonacci number $\mathsf{F}(n)$, and is hence exponential in n ($\phi(n) \sim O(1.6^n)$).*

Proof. Consider the case when $v^{\text{cap}} = 2$. For $n \ge 3$, we have ~~htat~~that $\phi(n) = \phi(n-1) + \phi(n-2)$. This can be shown by conditioning on whether or not the last passenger travels alone or with the penultimate passenger. In the first case, of traveling alone, the number of partitions of the remaining set of passengers is $\phi(n-1)$. In the second case, of traveling with another passenger, the number of partitions of the other passengers is $\phi(n-2)$. Since $\phi(1)=1$ and $\phi(2)=2$, the result follows. For larger v^{cap}, the recursion is written

$$\phi(n)=\phi(n-1)+\phi(n-2)+\cdots+\phi(n-v^{\text{cap}}),$$

which is bounded from below by $\phi(n-1)+\phi(n-2)$. □

5. State-Space Decomposition

In this section we discuss the modeling of the ILMTP-SD through *decision diagram (DD) decomposition* (Bergman and Cire 2016, 2018), which relates to *state-space decompositions* using dynamic programming (Bertsekas 1999, 2012). In particular, we show how one can model every possible single-destination CV trip through a compact DD. We then describe how the DDs can be concurrently optimized over through a network-flow reformulation with channeling constraints, which provides a novel and computationally advantageous remodeling of the ILMTP-SD relying on the structural property in Theorem 2. Section 5.1 describes how such a collection of DDs can be efficiently constructed, and Section 5.2 shows how to jointly use the constructed DDs to find an optimal solution for the problem.

5.1. Single destination BDD

For each destination $d \in \mathcal{D}$ we construct a DD that encodes every possible partition of $\mathcal{J}(d)$ into CV trips through paths. A DD is an acyclic digraph with nodes that are split among layers and arcs that connect nodes in different layers. DDs have close connection to the stage-state representation in dynamic programming (Bertsekas 2012). The layers in the DD can be identified with stages, the nodes in each layer represent different states and the arcs in the DD represent feasible transitions between states in successive stages.

Theorem 1 shows that it is sufficient to only consider groups of passengers with consecutive deadlines. Let the passengers in $\mathcal{J}(d)$ be ordered in nondecreasing order of $t^r(j)$ as $j_1^d,\ldots,j_{n_d}^d$. In order to encode the set of all partitions of $\mathcal{J}(d)$ into CV trips, the key decision for a pair of consecutive passengers j_i^d, j_{i+1}^d is whether they ride together in a CV trip. The decision of j_i^d sharing a ride in a CV with j_{i+1}^d is predicated on (a) prior decisions (other passengers $j_{i-1}^d, j_{i-2}^d, ..., j_{i-v^{\text{cap}}+1}^d$ already sharing a vehicle with j_i^d) since this affects the availability of a seat for j_{i+1}^d and (b) the time windows of the passengers should have non-empty intersection indicating that there exists a common time of arrival.

Accordingly, the layer in the DD (stage in DP) will denote the index of the passengers whose decisions have already been considered, the nodes in the DD (states in DP) will denote the number of occupants in the CV, and finally the arcs between the layers in the DD will represent the decision of a passenger represented by the layer to ride together with the next passenger in $\mathcal{J}(d)$. The addition of the arcs will satisfy the stated conditions in (a) and (b). Further, we associate additional information on start time and the objective contribution from CV trip on the arcs. Additionally, each path establishes the departure time of each CV and the total contribution to the objective function of the passengers in $\mathcal{J}(d)$ given the partition prescribed by the path. We make these precise in the following.

Formally, we construct, for every destination $d \in \mathcal{D}$, a ~~DD~~ diagram D^d , which is a layer-acyclic digraph $\mathsf{D}^d = (\mathsf{N}^d, \mathsf{A}^d)$. N^d is partitioned into $n_d + 1$ ordered layers $\mathsf{L}^d_1, \mathsf{L}^d_2, \dots, \mathsf{L}^d_{n_d+1}$ where $n_d = |\mathcal{J}(d)|$. Layer $\mathsf{L}^d_1 = \{\mathbf{r}^d\}$ and layer $\mathsf{L}^d_{n_d+1} = \{\mathbf{t}^d\}$ consist of one node each; the *root* and *terminal*, respectively. The layers $i = 1, ..., n_d$ correspond to the passenger j^d_i. The *layer* of node $\mathsf{u} \in \mathsf{L}^d_i$ is defined as $\ell(\mathsf{u}) = i$. Each arc $a \in \mathsf{A}^d$ is directed from its *arc-root* $\psi(a)$ (a node in layer i) to its *arc-terminal* $\omega(a)$ (a node in layer $i+1$), *i.e.* $\ell(\psi(a)) = \ell(\omega(a)) - 1$. We denote the *arc-layer* of a as $\ell(a) := \ell(\psi(a))$. It is assumed that every maximal arc-directed path connects $\mathbf{r}^d$ to $\mathbf{t}^d$. Each node u is associated with a *state* $\mathsf{s}(\mathsf{u})$ that defines the passengers aboard a CV trip. Since the capacity of the CV is v^{cap} the possible states are $0, \dots, (v^{\text{cap}} - 1)$. An arc between the $\mathsf{u} \in \mathsf{L}^d_i$ and $\mathsf{u}' \in \mathsf{L}^d_{i+1}$ exists if (i) $\mathsf{s}(\mathsf{u}') = \mathsf{s}(\mathsf{u}) + 1$ or $\mathsf{s}(\mathsf{u}') = 0$ and (ii) if $\mathsf{s}(\mathsf{u}') \neq 0$, $\cap_{k \in \{0,\dots,\mathsf{s}(\mathsf{u}')\}} [t^r(j^d_{i+1-k}) - T_w, t^r(j^d_{i+1-k}) + T_w] \neq \emptyset$. It is easy to verify that (i) and (ii) imply the satisfaction of (a) and (b) stated earlier in the section. We distinguish between the arcs that connect to the node with state 0 in the next layer and the remaining arcs. The former set of arcs are called *one-arcs* and the rest of the arcs are called the *zero-arcs*, indicated by one-arcs, $\phi(a) = 1$ and zero-arcs, $\phi(a) = 0$. In fact, we duplicate the one-arcs as many times as there are time instances in the set $\cap_{k \in \{0,\dots,\mathsf{s}(\mathsf{u})\}} [t^r(j^d_{i-k}) - T_w, t^r(j^d_{i-k}) + T_w]$.

Every one-arc a corresponds to a group $g(a) = \left\{ j^d_{\ell(a)-\mathsf{s}(\psi(a))}, j^d_{\ell(a)-\mathsf{s}(\psi(a))+1}, \dots, j^d_{\ell(a)} \right\}$, i.e, the set of contiguously indexed $\mathsf{s}(\psi(a)) + 1$ passengers ending in index $\ell(a)$. A one-arc stores an *arc-cost* $\eta(a)$ and an *arc-start-time* $t^0(a)$. The arc-cost of an arc corresponds to the total objective function cost incurred by group $g(a)$, and the arc-start-time indicates the time at which group $g(a)$ depart on a CV. These attributes are irrelevant in zero-arcs. A zero-arc indicates when a passenger shares the CV with the next passenger, i.e. is not the last passenger in the group. This thereby necessitates that any path containing a zero-arc will be followed by a one-arc dictating a group containing the associated passenger.

- Arc-start-time: As mentioned earlier the one-arcs are duplicated for each time instant $t \in \cap_{j\in g(a)}[t^r(j) - T_w, t^r(j) + T_w]$. For each such duplicated one-arc we have $t^0(a) = t - \tau^1(d)$, which is a feasible start time from the terminal $\mathbf{T}_0$ for the group $g(a)$.
- Arc-cost: In order to encode the objective function on the arcs, we set

$$\eta(a) := \alpha \cdot \sum_{j\in g(a)} \left(t^0(a) + \tau^1(d(j)) - \max_{c\in\mathcal{C}:\tilde{t}(c)\le t^0(a)} \tilde{t}(c, s(j)) \right) + (1-\alpha). \tag{1}$$

The first term is scaled by α and multiplies the total travel time in the objective function, which is composed of CV travel time and wating time at $\mathbf{T}_0$.

Let $\mathcal{P}^d$ be the set of arc-specified $\mathbf{r}^d$-to-$\mathbf{t}^d$ paths in D^d. For any path $p \in \mathcal{P}^d$, the groups $\mathsf{g}\,(p)$ composing the partition defined by p is $\mathsf{g}\,(p) := \bigcup_{a\in A^d:\phi(a)=1} g(a)$. The DDs are constructed in such a way that for every arc-specified $\mathbf{r}^d$-to-$\mathbf{t}^d$ path, each passenger $j \in \mathcal{J}(d)$ is in exactly one $g \in \mathsf{g}\,(p)$. The paths also entail the times that each group departs $\mathbf{T}_0$ and the impact on the objective function of selecting an arc. Time $t^0(a)$ indicates that the passenger departs to destination d, so that $g(a)$ occupies the CV assigned to it from time $t^0(a)$ until $t^0(a) + \tau(d)$. We note here that the construction of the DD ensures that the arrival time to destination d for each group is feasible with respect to requested arrival times. In particular, for each passenger $j \in g(a)$ we have $t^r(j) - T_w \le t^0(a) + \tau^1(d) \le t^r(j) + T_w$.

In the notation of (IP), this will correspond to $\sum_{j\in g(a)} \mathrm{tt}_j$ if $z_{j,t^0(a)} = 1$ for $j \in g(a)$. The second term in the objective function, $1-\alpha$, scales the indicator that this represents a CV trip. The cost of a path $\eta(p)$ is the sum of the arc-costs of the one-arcs in p.

Consider the following two properties; $\forall d \in \mathcal{D}, \forall p \in \mathcal{P}^d, \forall j \in \mathcal{J}(d)$:

(DD-1)*:* there is exactly one group $g \in \mathsf{g}\,(p)$ for which $j \in g$ (denote by $a^p(g)$ the one-arc selecting group g); and

(DD-2)*:* for such a group $g \in \mathsf{g}\,(p)$ with $j \in g$, $t^r(j) - T_w \le t^0(a^p(g)) + \tau^1(d) \le t^r(j) + T_w$.

We will denote by $g(j) \in \mathsf{g}\,(p)$ the unique group to which j belongs in the partition p. If properties **(DD-1)** and **(DD-2)** are satisfied, then any collection $\mathcal{Q}$ of $|\mathcal{D}|$ ~~$(:= |\mathcal{D}|)$~~ paths $\{p^1, \dots, p^{|\mathcal{D}|}\}$, where, for $d \in \mathcal{D}$, p^d is a $\mathbf{r}^d$-to-$\mathbf{t}^d$ in D^d, partitions all of $\mathcal{J}$ into $\bigcup_{d\in\mathcal{D}} \{\mathsf{g}\,(p)\}$ groups and the objective function of such a partition is $\sum_{d\in\mathcal{D}} \sum_{p\in\mathcal{P}^d} c(p) = f(\alpha)$.

Let $\mathcal{G}^d$ be every possible partition of $\mathcal{J}(d)$ into contiguous subsets for which each subset of passengers can board a common CV. By Proposition 2, we can consider only these partitions in seeking optimal solutions. Consider the following property as well:

(DD-3)*:* For every partition $\mathsf{g} \in \mathcal{G}^d$, there exists a path $p \in \mathcal{P}^d$ for which $\mathsf{g}\,(p) = \mathsf{g}$ and for every time for which the passengers in groups specified by $\mathsf{g}\,(p)$ can depart together.

If property **(DD-3)** is satisfied in D^d, then the paths in $\mathcal{P}$ list all possible partitions and departure times from $\mathbf{T}_0$, and therefore defines the feasible region.

Finally, consider the following property defined over paths $\mathcal{Q} = \{p^1, \dots, p^{|\mathcal{D}|}\}$:

(DD-4)*:* $\forall t \in \mathcal{T}, \left| \bigcup_{d \in \mathcal{D}} \{a \in p^d : t^0(a) \leq t \leq t^0(a) + \tau(d)\} \right| \leq |V|.$

If $\mathcal{Q}$ satisfies **(DD-4)**, then assigning unique CV trips to each group $g \in \bigcup_{d \in \mathcal{D}} \{\mathsf{g}\,(d)\}$ and leaving $\mathbf{T}_0$ at time $t^0(g)$ defines a feasible solution to ILMTP-SD that has objective function value $\sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{P}^d} c(p)$. Therefore, building a set of DDs satisfying properties **(DD-1)**, **(DD-2)** and **(DD-3)**, and finding a collection of paths satisfying condition **(DD-4)** which is of minimum total length outlines another approach for solving ILMTP-SD.

Algorithm 1 constructs a DD that satisfies properties **(DD-1)** to **(DD-3)** for each destination. The algorithm proceeds as follows. Algorithm 1 starts by computing (Line 1) for each passenger j_i^d, the earliest $(t^e(j_i^d))$ and latest $(t^l(j_i^d))$ possible departure time from $\mathbf{T}_0$ if she were to ride alone using (1). Line 2 creates the root node $\mathbf{r}^d$, which is also referred to as u_1^0 for ease of notation. Each iteration of the loop in line 3 creates the arcs from layer i to layer $i+1$ and the corresponding nodes for every possible state. For ease of notation, we also denote a node u in layer $\ell(u) = i$ and state $s(u) = k$ by u_i^k. Note that for $i = n_d$ the node $u_{n_d+1}^0$ represents the terminal node $\mathbf{t}^d$ of the DD D^d. Line 4 creates a single node to add to the $(i+1)$-th layer. Arcs drawn from the nodes u_i^k in layer i to the node u_{i+1}^0 are *one-arcs* and represent the grouping of passengers $\{j_{i-k}^d, \dots, j_k^d\}$. The creation of the one-arcs as well as the assignment of CV start times and costs are executed in the loop defined by Line 11 of the algorithm. The loop in line 5 iterates over the nodes in layer i and adds *zero-arcs* between the layers i and $i+1$. In particular, the loop creates nodes u_{i+1}^k in layer $i+1$ for a positive state k, which entails grouping passengers $j_{i+1-k}^d, \dots, j_{i+1}^d$ together through a zero-arc, if the conditions in line 6 holds: (i) the number of passengers does not exceed v^{cap}; (ii) there exists such a passenger j_{i+1}^d; and (iii) the time windows of passengers j_{i+1-k}^d and j_{i+1}^d overlap. Since the passengers are ordered by increasing deadlines, satisfaction of (iii) implies that the passengers in $\{j_{i+1-k}^d, \dots, j_{i+1}^d\}$ have at least one CV starting time at $\mathbf{T}_0$ for which they arrive at their destination within their time windows. Note that the cost of arcs is set to zero since this is a zero-arc. Finally, the loop in line 11 creates multiple one-arcs for each node, which entails that passenger j_i^d is the last in a group, by iterating over all possible departure times shared by passengers j_{i-k}^d to j_i^d and computing the corresponding cost of such a grouping according to the departure time.

Algorithm 1 Construction of D^d for passengers $j_1^d, \dots, j_{n_d}^d$ with destination d

```
Compute for each passenger $j_i^d$ the earliest and latest times of departure from the $\mathbf{T}_0$:
    $t^e(j_i^d) \leftarrow \max\left\{ t^r(j_i^d) - \tau^1(d), \min_{c\in\mathcal{C}} \{\tilde{t}(c)\} \right\}$
    $t^l(j_i^d) \leftarrow \min\left\{ t^r(j_i^d) + \tau^1(d) - 1, \max_{c\in\mathcal{C}} \{\tilde{t}(c)\} \right\}$
Add new node $u_1^0$ to $\mathsf{L}_1^d$ such that $s(u_1^0) = 0$ and $\ell(u_1^0) = 1$        ▷ Same as root node $\mathbf{r}^d$
for $i \leftarrow 1, \dots, n_d$ do        ▷ Determines transitions after passenger $j_i^d$
    Add new node $u_{i+1}^0$ to $\mathsf{L}_{i+1}^d$ such that $s(u_{i+1}^0) = 0$ and $\ell(u_{i+1}^0) = 1$
    for $k \leftarrow 0, \dots, |\mathsf{L}_i^d| - 1$ do        ▷ Number of unassigned passengers up to $j_i^d$
        if $k < v^{\text{cap}} - 1$ and $i < n_d$ and $t^e(j_{i+1}^d) \le t^l(j_{i-k}^d)$ then        ▷ Checks if $j_{i+1}^d$ can join them
            Add new node $u_{i+1}^{k+1}$ to $\mathsf{L}_{i+1}^d$ such that $s(u_{i+1}^{k+1}) = k+1$ and $\ell(u_{i+1}^{k+1}) = i+1$
            Add new arc $a$ to $\mathsf{A}^d$ such that $\psi(a) = u_i^k,\ \omega(a) = u_{i+1}^{k+1}$, and $\phi(a) = 0$
            $\eta(a) \leftarrow 0$        ▷ Group cost deferred
        end if
        for $t \leftarrow t^e(j_i^d), t^e(j_i^d) + 1, \dots, t^l(j_{i-k}^d)$ do        ▷ Departure times for group $\{j_{i-k}^d, \dots, j_i^d\}$
            Add new arc $a$ to $\mathsf{A}^d$ such that $\psi(a) = u_i^j,\ \omega(a) = u_{i+1}^0,\ \phi(a) = 1$, and $t^0(a) = t$
            $\eta(a) \leftarrow \alpha \sum_{j\in g(a)} \left( t + \tau^1(d) - \max_{c\in\mathcal{C}:\tilde{t}(c)\le t^0(a)} \{\tilde{t}(c, s(j))\} \right) + (1-\alpha)$        ▷ Group cost incurred
        end for
    end for
end for
```

Theorem 4 below shows that Algorithm 1 constructs DDs that are polynomial in the size of the input satisfying properties **(DD-1)** to **(DD-3)**. Section 5.2 discusses how to identify the optimal collection of paths satisfying property **(DD-4)**.

Theorem 4 *For every $d \in \mathcal{D}$, Algorithm 1 constructs DDs D^d with $O(n_d \cdot v^{\text{cap}})$ nodes and $O(n_d \cdot v^{\text{cap}} \cdot T_w)$ arcs satisfying properties* **(DD-1)**, **(DD-2)**, *and* **(DD-3)** *that can be constructed in time $O(n \cdot v^{\text{cap}} \cdot T_w)$.*

The proof of this result is deferred to Appendix C.

Example 1. Consider the following ILMTP instance. All passengers request transportation to a single destination d. Let $\tau(d) = 4, (\tau^1(d) = 2, \tau^2(d) = 0, \tau^3(d) = 2)$, $T_w = 1$, $|V| = 2$, and $v^{\text{cap}} = 3$. There are 5 passengers, requesting arrival time to d at 5, 6, 6, 7, 9 for passengers $j_1^d, \dots, j_5^d$, respectively. Mass transit trips arrive to $\mathbf{T}_0$ at time 2 and 6 ($c = 1$ and $c = 2$). All passengers originate from the same s and $\tilde{t}(c, s) = 0, 4$ for $c = 1, 2$ respectively.

Figure 3 depicts a DD that satisfies properties **(DD-1)**, **(DD-2)**, and **(DD-3)**. The layers are drawn in ascending order from top to bottom. One-arcs are depicted as solid lines, and zero-arcs as dashed lines, interpreted to be pointing downwards. Each one-arc ~~a~~ has two labels depicted in parentheses, the first label is the arc-start-time $t^0(a)$ and the second label is the total travel time of passengers in $g(a)$. Each arc-cost (for the one-arcs) is the second argument times α plus $(1-\alpha)$.

Consider the red-colored path p', which traverses arcs connecting $\mathbf{r}^b$ to $\mathbf{t}^b$ through the node-specified path $\mathbf{r}^b - 0 - 0 - 0 - 0 - \mathbf{t}^b$ along arcs labeled $(2,4)-(3,5)-(3,5)-(6,4)-(6,4)$. Each arc emanates from a node with label 0, which indicates that the passengers travel alone in a CV. The arcs on this path dictates that the passengers leave $\mathbf{T}_0$ at times $t = 2,3,3,6,6$ and have total travel times of $4,5,5,4,4$ time units, respectively. To achieve these travel times, passengers j_1^d, j_2^d, j_3^d arrive to $\mathbf{T}_0$ on the mass transit trip $c = 1$, and passengers j_4^d, j_5^d arrive on trip $c = 2$. This path satisfies properties **(DD-1)** and **(DD-2)**, as do all paths in the diagram. However, this path does violate property **(DD-4)**—consider for example $t = 4$. The first three passengers are each assigned CVs that will be en-route at $t = 4$ which violates the restriction that $|V| = 2$.

Consider now the green-colored path p'', which traverses arcs connecting $\mathbf{r}^d$ to $\mathbf{t}^d$ through the node-specified path $\mathbf{r}^d - 1 - 0 - 1 - 0 - \mathbf{t}^d$ with one-arcs labeled $(3,10)-(5,14)-(7,5)$ on layers 2, 4, and 5, respectively. This path has three one-arcs that specify groups $\left\{j_1^d, j_2^d\right\}, \left\{j_3^d, j_4^d\right\}, \left\{j_5^d\right\}$ to leave $\mathbf{T}_0$ at times $t = 3,5,7$, respectively. The total travel times on each CV trip are 10 (5+5), 14 (7+7), 5 (5) achieved by passengers $j_1^d, j_2^d, j_3^d, j_4^d$ arriving on mass transit trip $c = 1$, and passenger j_5^d arriving on trip $c = 2$. For $t = 0,1,\ldots,10$, the number of active CVs is 0,0,0,1,1,2,2,2,2,1,0, respectively, upon which all CVs have returned to $\mathbf{T}_0$, thereby never violating the constraints on the number of CVs. This path therefore satisfies property **(DD-4)** and corresponds to a feasible solution. The evaluation of the objective function corresponding to this solution depends on α, and is evaluated as $\eta(p'') = \alpha \cdot (10+14+5) + (1-\alpha)\cdot 3$.

There are 492 paths in the depicted DD, corresponding to $|\mathcal{G}^d|$. This example suggests the advantages of a DD-based approach, in that an exponentially sized set of solutions can be represented, compactly, in a small-sized diagram.

An additional note is in order. Consider the set of arcs directed between the nodes labeled 0 in the penultimate layers of the DD. Each arc represents group $\left\{j_4^d\right\}$, the singleton

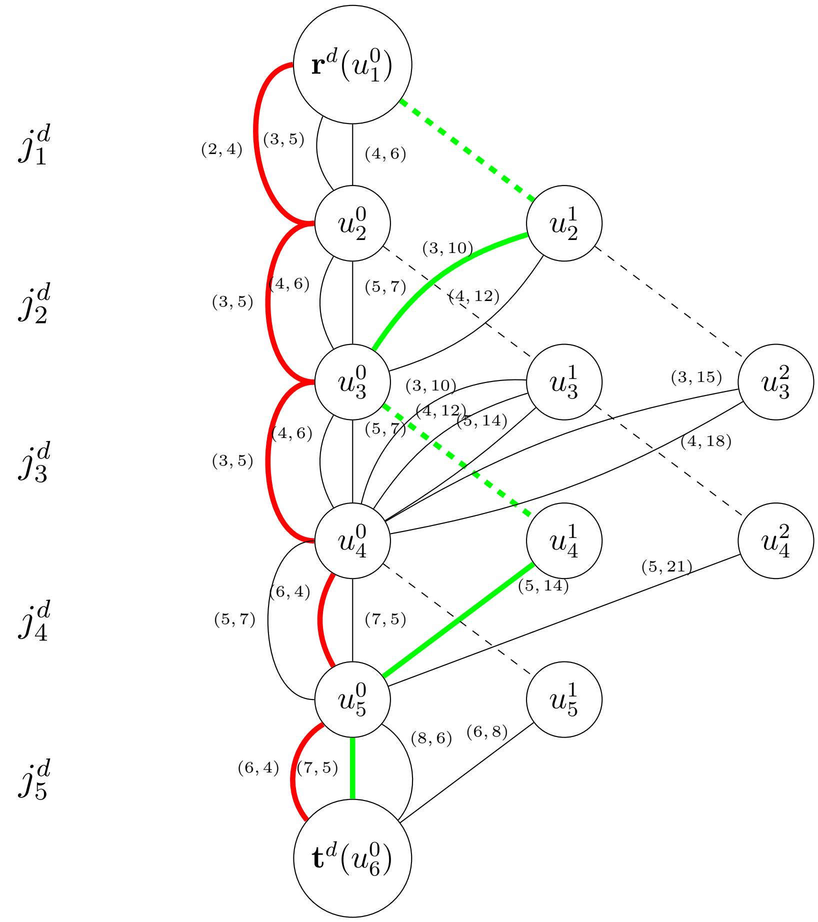

Figure 3 **Decision diagram for Example 1**

passenger traveling alone. There are three ways this can happen—the passenger arrives to $\mathbf{T}_0$ on the mass transit trip $c = 1$, waits for 3 time units, and then boards a CV, or the passenger arrives to $\mathbf{T}_0$ on trip $c = 2$, waits 0 or 1 time units, and then boards a CV. Should this group be selected as part of the solution, the selection of arc (and, therefore, mass transit trip and waiting time) will depend on the availability of CVs that can be restricted based on CV trips to destination d and to other destinations in the system.

5.2. Network-flow reformulation

Given, for each destination $d \in \mathcal{D}$, a DD D^d satisfying properties **(DD-1)**, **(DD-2)**, and **(DD-3)**, we can reformulate the ILMTP-SD optimization problem as a *consistent path problem*. In each DD, we must select a path so that at any time t no more than $|V|$ CVs are assigned to the groups. More formally, we want to select, for every $d \in \mathcal{D}$, a path $p^d \in D^d$ such that, for every t, the number of one-arcs with $t^0(a) \leq t \leq t^0(a) + \tau(d)$ is less than or equal to $|V|$. Let $\chi(a,t) \in \{0,1\}$ indicate that a CV would be active at time t (i.e.

$t^0(a) \leq t \leq t^0(a) + \tau(d))$ if arc a is chosen. One can formulate this by assigning a variable y_a to each arc a and solving the following optimization problem:

$$\begin{aligned}
\min \quad & \sum_{d \in \mathcal{D}} \sum_{a \in \mathsf{A}^d} \eta(a) y_a \\
\text{s.t.} \quad & \sum_{a:\psi(a)=\mathbf{r}^d} y_a = 1, \quad \sum_{a:\omega(a)=\mathbf{t}^d} y_a = 1, && \forall d \in \mathcal{D} \\
& \sum_{a:\psi(a)=\mathsf{u}} y_a - \sum_{a:\omega(a)=\mathsf{u}} y_a = 0, && \forall d \in \mathcal{D}, \forall \mathsf{u} \in \mathsf{L}_2^d \cup \ldots \cup \mathsf{L}_{n_d}^d \\
& \sum_{d \in \mathcal{D}} \sum_{a \in \mathsf{A}^d : \chi(a,t)=1} y_a \leq |V|, && \forall t \in \mathcal{T} \\
& y_a \in \{0,1\} && \forall d \in \mathcal{D}, \forall a \in \mathsf{A}^d
\end{aligned} \tag{NF}$$

Model (NF) directly models each DD as a network-flow problem, where we seek to send one unit of flow from $\mathbf{r}^d$ to $\mathbf{t}^d$. The sum of the arc weights are minimized, subject to the singular linking constraint, that enforces the restriction on the number of CVs. Model (NF) therefore identifies a collection of paths $\mathcal{Q}$ satisfying property **(DD-4)** of minimum total cost, therefore providing a valid formulation for the ILMTP-SD.

5.3. A Column Generation Approach

An alternative solution scheme for the ILMTP-SD given a collection of DDs satisfying properties **(DD-1)**, **(DD-2)**, and **(DD-3)** is to use a branch-and-price scheme (Barnhart et al. 1998) by associating a binary variable z_p to every $\mathbf{r}^d$-to-$\mathbf{t}^d$ path in the collection of DDs, where we let $k(p,t)$ be the number of one-arcs in p for which $\chi(a,t) = 1$ (which we refer to as the *master problem*):

$$\begin{aligned}
\min \quad & \sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{P}^d} \eta(p) z_p \\
\text{s.t.} \quad & \sum_{p \in \mathcal{P}^d} z_p = 1, && \forall d \in \mathcal{D} \\
& \sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{P}^d} k(p,t) z_p \leq |V|, && \forall t \in \mathcal{T} \\
& z_p \in \{0,1\}, && \forall d \in \mathcal{D}, \forall p \in \mathcal{P}^d.
\end{aligned} \tag{MP}$$

Since there is an exponential number of variables corresponding to those paths, we propose to solve this model by column generation which can be turned into an exact approach via branch-and-price.

The procedure begins by defining an initial search-tree node with no branching decisions, and, for all $d \in \mathcal{D}$, a subset of the paths $\tilde{\mathcal{P}}^d \subseteq \mathcal{P}^d$. Let $\mathcal{P} = \cup_{d\in\mathcal{D}}\mathcal{P}^d$ and $\tilde{\mathcal{P}} = \cup_{d\in\mathcal{D}}\tilde{\mathcal{P}}^d$. The *restricted master problem* (RMP($\tilde{\mathcal{P}}$)) is (MP) restricted to only those variables in $\tilde{\mathcal{P}}$. $\tilde{\mathcal{P}}$ should contain at least one feasible solution, which we address in Section 5.4.

We solve the LP relaxation of (MP) by *column generation*, where we add paths $p \in \mathcal{P}\backslash\tilde{\mathcal{P}}$ to $\tilde{\mathcal{P}}$ if the associated variable in (MP) has a reduced cost that is negative at the solution corresponding to the optimal LP relaxation of (RMP($\tilde{\mathcal{P}}$)). Since we do not assume an enumeration of $\mathcal{P}$, we identify if such a path exists by solving a *pricing problem* (PP), as described in the following proposition.

Proposition 1 *For all $d \in \mathcal{D}$, let μ_d be the dual variable associating constraint $\sum_{p\in\tilde{\mathcal{P}}^d} z_p = 1$ and, for all $t \in \mathcal{T}$, let λ_t be the dual variable associated with constraint $\sum_{d\in\mathcal{D}} \sum_{p\in\tilde{\mathcal{P}}^d} k(p,t)z_p \leq |V|$ at an optimal solution to the LP relaxation of (RMP($\tilde{\mathcal{P}}$)). For all $d \in \mathcal{D}, a \in A^b$, and $t \in \mathcal{T}$, let $\chi(a,t)$ indicate if arc a is a one-arc and $t \in \{t^0(a), \ldots, t^0(a) + \tau(d)\}$, i.e., that taking the one-arc requires an active CV at time t.*

Let us define, for all $a \in \cup_{d\in\mathcal{D}}A^d$, a binary variable y_a and, for all $d \in \mathcal{D}$, a binary variable ζ_d. If the optimal value of

$$\begin{aligned}
\min & \sum_{d\in\mathcal{D}}\sum_{a\in\mathsf{A}^d} \eta(a)y_a - \sum_{d\in\mathcal{D}} \mu_d\zeta_d + \sum_{d\in\mathcal{D}}\sum_{a\in\mathsf{A}^d}\sum_{t\in\mathcal{T}} \lambda_t\chi(a,t)y_a & \\
\text{s.t.} & \sum_{a:\psi(a)=\mathbf{r}^d} y_a = \zeta_d, \qquad \sum_{a:\omega(a)=\mathbf{t}^d} y_a = \zeta_d, & \forall d \in \mathcal{D} \\
& \sum_{a:\psi(a)=\mathsf{u}} y_a - \sum_{a:\omega(a)=\mathsf{u}} y_a = 0, & \forall d \in \mathcal{D}, \forall \mathsf{u} \in \mathsf{N}^d \text{ with } \mathsf{u} \notin \left\{\mathbf{r}^d, \mathbf{t}^d\right\} \\
& y_a \in \{0,1\} & \forall d \in \mathcal{D}, \forall a \in \mathsf{A}^d
\end{aligned} \tag{PP}$$

is non-negative, then the optimal LP solution of (RMP($\tilde{\mathcal{P}}$)) is an optimal LP solution of (MP). Otherwise the set of arcs for which $y_a = 1$ defines a path in the diagram D^d for which $\zeta_d = 1$ with negative reduced cost.

Proof. Follows immediately from the definition of reduced cost. □

Subproblem (PP) decomposes into separate shortest path problems. For each $d \in \mathcal{D}$, let $p^{d,*}$ be the shortest path and $f^{d,*}$ be the shortest path length in D^d, where each arc has length $\eta(a) - \sum_{t\in\mathcal{T}} \lambda_t\chi(a,t)$. The variable $z_{p^{d,*}}$ associated with the path that achieves the minimum value $f^{d,*} - \mu_d$ in the pricing problem will be the variable in the exponential model (MP) that has the lowest reduced cost. Since this can be done separately for each destination, and since each DD is directed and acyclic, the pricing problem is solved in linear time in the size of the DDs. Note that the IP solution to (RMP($\tilde{\mathcal{P}}$)) will always be a feasible solution to the ILMTP-SD instance. This equips us with a mechanism for generating feasible solution and upper bounds.

A branch-and-bound search can be conducted to complete a branch-and-price implementation. A queue of search-tree nodes Γ is defined and initialized as a singleton γ'. At any point in the execution of the algorithm, each search node $\gamma \in \Gamma$ is defined by a set of branching decisions $\text{out}(\gamma), \text{in}(\gamma) \in \tilde{\mathcal{P}}$. We also maintain the best-known solution z^* and its objective value f^*.

While $\Gamma \neq \emptyset$, a search node γ is selected to explore. The LP relaxation of (RMP($\tilde{\mathcal{P}}$)), with additional constraints requiring $z_p = 0, 1$ for those paths $p \in \text{out}(\gamma), \text{in}(\gamma)$, respectively, is solved via column generation. This can be implemented as follows. When $z_p = 1$, the associated DD can be simplified to a single path. When $z_p = 0$, the associated DD can be modified to delete only the path p and retain all other paths, noting that removing one path from a DD takes linear time and increases the width of the DD by at most one (Ciré and Hooker 2014). One can store the collection of DDs resulting from the branching decisions at each search tree node, or remove the paths with $z_p = 0$ each time, starting from the original DDs. Suppose $\{p^{d,1}, \ldots, p^{d,l}\} \subset \text{out}(\gamma)$ be an arbitrary ordering of all paths in $\text{out}(\gamma)$ that are in D^d and are to be removed, i.e. $z_p = 0$ for all $p \in \{p^{d,1}, \ldots, p^{d,l}\}$. Note that l can vary by d. The procedure for removing the said paths from D^d is iterative. Starting with $\mathsf{D}^{d,0} = \mathsf{D}^d$, at each iteration $k = 1, \ldots, l$ the path $p^{d,k}$ is removed from $\mathsf{D}^{d,k-1}$ as described by (Ciré and Hooker 2014) and the resulting diagram is denoted as $\mathsf{D}^{d,k}$. The diagram $\mathsf{D}^{d,l}$ is the decision diagram for d that is used in the relaxation of node γ. If the optimal value of the LP relaxation of (RMP($\tilde{\mathcal{P}}$)) is greater than or equal to f^*, the node is pruned, and search continues by selecting another node in Γ. Otherwise, the IP (RMP($\tilde{\mathcal{P}}$)) is solved and if the optimal value f' is less than f^*, this solution replaces z^* and f^* is updated with f'. Note that enforcing a path is the same as removing a DD from the problem and fixing this decision, and since there is a one-to-one mapping between passenger-groups traveling to a building and paths in a DD, this will not cycle and is complete. We note here that our experiments did not require branching since the root node gap is shown experimentally to be very small.

We also describe another approach to identifying a feasible solution in Section 5.5 since the solution of the IP $(\text{RMP}(\tilde{\mathcal{P}}))$ can be computationally prohibitive. In general, a branching scheme must be devised in order to close any gap at the root node. As discussed in the experimental results, the optimality gap at the root node is very small in this application, and so we need not explore branching rules.

5.4. Finding an initial feasible solution

We can generate an initial feasible solution to (MP) by appending an extra path to each DD that represents not assigning any passenger. Specifically, for each $d \in \mathcal{D}$, create nodes $\mathsf{u}_2^d, \ldots, \mathsf{u}_i^d$ with state $\mathsf{s}(\mathsf{u}_i^d) = \emptyset, \forall i = 2, \ldots, n$. Add one-arcs from $\mathbf{r}^d$ to u_2^d, from u_n^d to $\mathbf{t}^d$, and, for $i = 2, \ldots, n-1$, from u_i^d to u_{n-1}^d. For each new arc a set $\eta(a) = -\infty$ and $t^0(a) = 0$. For the new path p_0^d in each DD set $z_{p_0^d} = 1$ and all other variables equal to 0. Since for all $t \in \mathcal{T}, k(p_0^d, t) = 0$, this will be an initial feasible solution to (MP).

5.5. Identifying a feasible solution

Suppose that z^* is a solution to the LP relaxation of $(\text{RMP}(\tilde{\mathcal{P}}))$ and that z^* is not integral. Denote by $p^d \in \tilde{\mathcal{P}}^d$ for each destination $d \in \mathcal{D}$ as the path satisfying $p^d = \arg\max_{p \in \tilde{\mathcal{P}}^d} z_p^*$. In other words, p^d is path for destination d with the largest fractional value in the solution of the LP relaxation. As defined in Section 5.1, let $\mathsf{g}\left(p^d\right)$ represent the partition of $\mathcal{J}(d)$ defined by p^d. The path p^d encodes the particular start times on the CV for the groups in $\mathsf{g}\left(p^d\right)$. In the following, we describe an IP that fixes the groups in $\mathsf{g}\left(p^d\right)$ but attempts to assign starting times for the groups such that the resulting CV trips are feasible for the ILMTP-SD. Note that by assigning different start times we are implicitly enumerating other paths in the DD D^d with the same groupings. The IP formulation is a simplified version of the one presented in Raghunathan et al. (2018), where route assignments were also considered in the IP.

Prior to describing the model we introduce relevant notation. Define earliest and latest CV start times for the group $g \in \mathsf{g}\left(p^d\right)$, for all $d \in \mathcal{D}$, as $t^e(g) = \max_{j \in g} t^e(j)$ and $t^l(g) = \min_{j \in g} t^l(j)$ where $t^e(j), t^l(j)$ are as defined in (1). Define the objective value associated with the particular start time $t \in [t^e(g), t^l(g)]$ as

$$\eta(g,t) = \alpha \sum_{j \in g} \left(t + \tau^1(d) - \max_{c \in \mathcal{C}: \left\{\tilde{t}(c)\right\} \leq t} \left\{\tilde{t}(c, s(j))\right\} \right) + (1-\alpha).$$

Further, let $\chi(g,t,t') \in \{0,1\}$ indicate the times t' (i.e. $t \le t' \le t + \tau(d)$) at which a CV serving group g leaving terminal at time t would be active. The decision variable in the IP formulation is $x_{g,t} \in \{0,1\}$ for $t \in [t^e(g), t^l(g)]$ indicating the choice of start time of t on the CV for group g. The IP formulation is:

$$\min \sum_{g\in \bigcup\limits_{d\in\mathcal{D}} \mathtt{g}(d)} \eta(g,t)x_{g,t} \tag{2a}$$

$$\text{s.t.} \sum_{t=t^{\mathrm{e}}(g)}^{t^{\mathrm{l}}(g)} x_{g,t} = 1 \qquad \forall\, g \in \bigcup_{d\in\mathcal{D}} \mathtt{g}\,(d) \tag{2b}$$

$$\sum_{g\in\cup_{d\in\mathcal{D}}\mathtt{g}(d)} \sum_{t=t^e(g)}^{t^l(g)} \chi_{g,t,t'}x_{g,t} \le |V| \qquad \forall\, t' \in \mathcal{T}. \tag{2c}$$

Constraint (2b) imposes that each group is assigned to exactly one CV route and start time. Constraint (2c) ensures that the number simultaneous trips on the CVs does not exceed the number of CVs. A feasible solution to (2) is easily seen to be a feasible solution ILMTP-SD. The above IP typically solves at the root node in fractions of a second.

6. Numerical Evaluation

This section details an experimental evaluation conducted to evaluate the efficiency of the algorithms presented, sensitivity analysis on the quality of the solutions obtained, and how the solutions to the ILMTD-SD can be used in more general settings. Section 6.1 describes the instances, all of which are available at `ftp://ftp.merl.com/pub/raghunathan/LastMile-TestSet/`. Due to company policy, we are unable to make the data and code available at the INFORMS Journal of Computing repository, but we do provide a sample instance with the structure as those used in the experiments (`https://github.com/INFORMSJoC`). Section 6.2 provides an experimental evaluation of the formulations and algorithms. We analyze the trade-offs on different objectives and the effect of the time window length on the quality of the solutions obtained in Sections 6.3 and 6.4, respectively. Finally, in Section 6.6 we study the quality of the solution to the ILMTP-SD for more general variants of the problem. All experiments were run on a machine with an Intel(R) Core(TM) i7-4770 CPU @ 3.40GHz and 16 GB RAM. All algorithms were implemented in `Python 2.7.6` and the ILPs are solved using `Gurobi 8.0.1`.

6.1. Instance generation

We generate instances with number of destinations $|\mathcal{D}| \in \{10, 25, 50\}$. In order to test how well the algorithms scale, we specify the number of passengers per destination as $\frac{n}{|\mathcal{D}|} \in \{100, 150, 200\}$, and use the corresponding value for n (i.e., if $|\mathcal{D}| = 10$ and $\frac{n}{|\mathcal{D}|} = 100$, we use $n = 1000$) so that our instances have up to 10,000 passengers. We set the number of CVs $|V| = \mathsf{round}(0.06 * n)$ where $\mathsf{round}(\cdot)$ rounds to the nearest integer and CV capacity $v^{\text{cap}} = 5$. We generate 5 instances per configuration. The number of stations where passengers board the mass transportation system is 4 and so $\mathcal{S} = \{\mathbf{T}_0, 1, 2, 3, 4\}$. The station of origin for each passenger is generated independently and uniformly at random from $\{1, 2, 3, 4\}$. We discretize time in 2 units per minute and we consider serving passengers with requests within a full hour period, hence totalling 120 time units of requests, which span between 90 and 210. Hence, the requested arrival time $t^r(\cdot)$ is generated independently and uniformly at random from the set $\{90, 91, \ldots, 210\}$. The trips $\mathcal{C}$ consist of trains that depart the farthest station 4 at times $\tilde{t}(c, 4) \in \{0, 30, 60, \ldots, 210\}$. The travel time between stations is 10 and so $\tilde{t}(c, s) = \tilde{t}(c, 4) + (4 - s) \cdot 10$ for $s = 1, 2, 3$ and $\tilde{t}(c) = \tilde{t}(c, 4) + 40$ for all $c \in \mathcal{C}$. The travel time from $\mathbf{T}_0$ to destinations d is chosen independently and uniformly at random between $\{10, 11, \ldots, 20\}$, so if travel time to a destination is t^d then $\tau^1(d) = t^d + 1$ where 1 is the time to board, $\tau^2(d) = 1$, which is the time to deboard, and $\tau^3(d) = t^d$.

We will at certain points of the discussion evaluate the impact of choosing different values of $\alpha \in [0, 1]$. Since the two objectives are on a different scale, we multiple the coefficient of the number of trips by 100 in order to put the objective on a relatively equivalent scale.

6.2. Algorithm comparison

We test the efficiency of our three algorithms. **IP** refers to solving (IP). **NF** refers to solving (NF). **CG** refers to solving (MP) through the column generation algorithm in Section 5.3. For comparison with heuristic approaches, **BAS** refers to the break-and-shift heuristic introduced by Raghunathan et al. (2018). **BAS** first sorts the passengers for each destination by their arrival times, creates groups sequentially from such ordering by adding passengers to each new group while possible, then iteratively breaks groups in which one passenger delays another and move passengers to consecutive groups when no travel time delay is incurred. The four algorithms are applied to each generated instance and for each setting of T_w and α, resulting in 450 runs each. Since the transportation systems will in practice be repeatedly optimized, the ILMTP-SD requires efficient solution methodologies. Accordingly, we set a time limit of 10 minutes.

IP does not solve any of the instances tested to optimality within 10 minutes of computational time, and only finds a feasible solution to 2 instances within the time limit without closing the optimality gap. Even when supplied with a warm-start solution provided by the heuristic from Raghunathan et al. (2018) and adding ordering constraints based on Theorem 2, **IP** fails to solve any instance tested to optimality within 10 minutes. All other algorithms solve all instances in 10 minutes. This provides clear indication of the superiority of the DD-based algorithm, and so for the remainder of this section we provide a comparison only of **NF** and **CG**. We note that the root-node optimality gap ($\frac{UB-LB}{LB} \times 100$ where LB is the lower bound and UB is the upper bound) for the DD-based model is below 0.5%, demonstrating the quality of the LP relaxation. We therefore only report results for solving the root node as branching is not necessary, but can be used if closing the gap is critical.

Figure 4 depicts a cumulative distribution plot of performance over all runs for **NF**, **CG**, and **BAS**. Each line corresponds to an algorithm, and each point on a line is composed of coordinates (t, s) which is the number of instances solved s by time t by the algorithm. The figure demonstrates that **CG** solves more problems than **NF** for any time limit. In addition, **CG** finds some optimal solutions faster than the **BAS** runtime.

A more detailed pairwise comparison of **CG** with **NF** and **BAS** appears in the plots of Figure 5. The coordinates are the runtime of **NF** and the runtime of **CG** and **BAS**, respectively, in the left and right plot, in seconds in log-scale. The size of each point correspond to n (increasing in size as n increases) and the color of each point corresponds to α. This plot more readily reveals the advantage of **CG**. In only 17 of the smallest instances is the the runtime of **NF** smaller than that of **CG**. Additionally, the relative superiority of **CG** grows as α decreases and the objective is more scaled to the number of CV trips. A similar comparison of runtime exists for **CG** versus **BAS**, except that for smaller instances **BAS** is more efficient, while for larger instances **CG**, an exact approach, can be more efficient than the heuristic **BAS**.

It is also noteworthy to compare the relative quality of the solutions obtained by **BAS** to the exact approach **CG**. The plot in Figure 6 depicts a scatter plot with the same coordinates as those in the right plot in Figure 6 comparing the runtimes of **CG** and

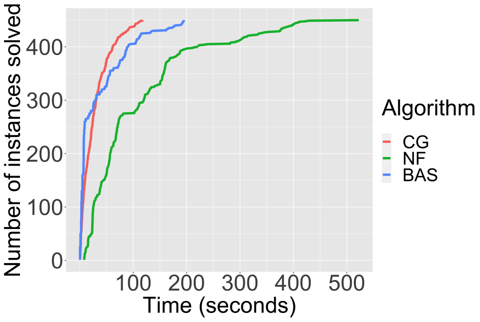

Figure 4 **Cumulative distribution plot of performance, comparing CG and NF.**

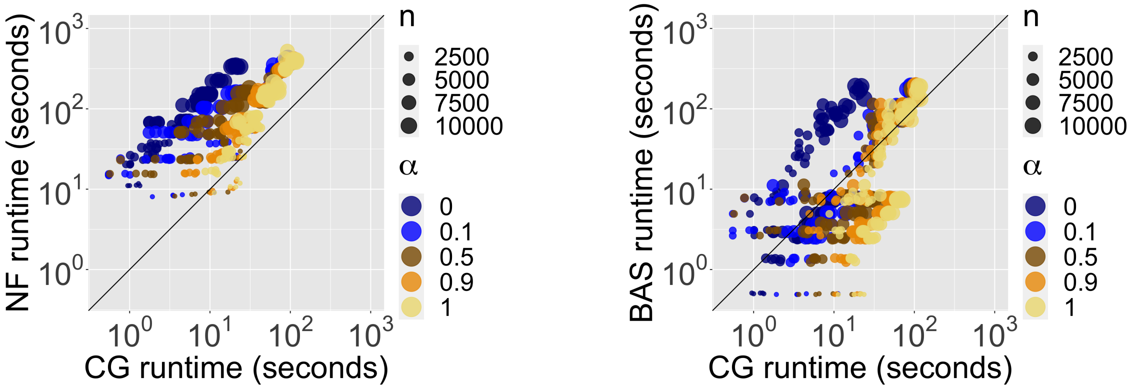

Figure 5 **Scatter plots, comparing CG with NF (left) and CG with BAS (right).**

BAS, but this time colored by the percent improvement in objective function value of the solution obtained by **CG** over those obtained by BAS, which is always non-negative because **CG** is an exact approach and always finds optimal solutions with the 10 minute timeout. The interesting observation is that for small α, **CG** and **BAS** obtain nearly the same solution quality, but **CG** is more efficient. For large α, **CG** takes slightly longer than **BAS** but obtains higher quality solutions. We therefore observe that for any setting of α there are gains that can be realized by utilizing the exact approach.

6.3. Objective trade-off analysis

It is critical to understand how the two objectives, total passenger wait time and number of CV trips, affect the solutions obtained, so that proper operation can be determined. As **CG** is shown to be the best algorithm among those tested, we use solutions obtained by **CG** for the remainder of this section. The plot in Figure 7 depicts the data through a line plot. Each point corresponds to the total passenger travel time and the number of

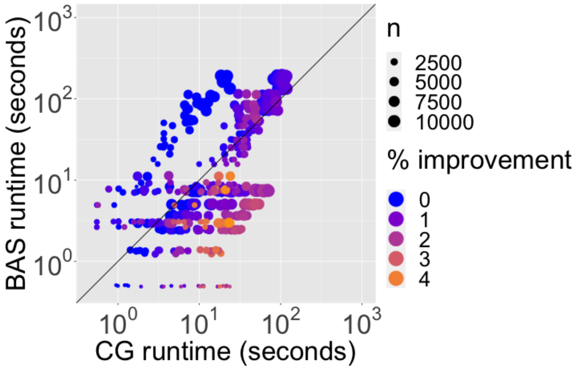

Figure 6 **Scatter plot of comparing CG and BAS, but colored by percent improvement in solution quality obtained by `CG` over that obtained by `BAS`.**

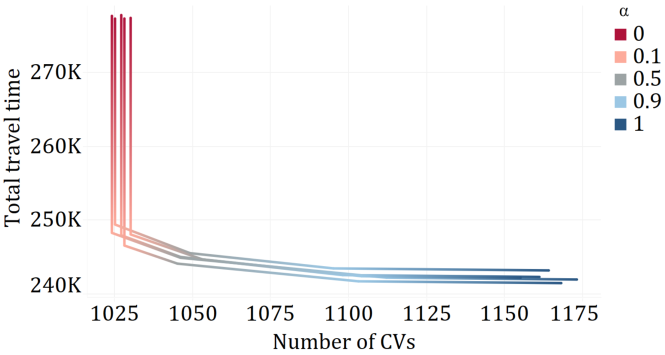

Figure 7 **Line plot depicting total passenger travel time and number of CVs for different settings of α.**

CVs trips in the solutions obtained by **CG**, for those instances with $T_w = 5, n = 5000$ and $|\mathcal{D}| = 50$, where we scaled the number of trips by 100 to ensure that the values of the two objectives are of the same order of magnitude. The color corresponds to α.

This plot highlights the power of considering a balanced objective. First, when $\alpha \in \{0, 1\}$, we see that considering only one objective can result in great solution for that metric alone, but can be very bad for the other, ignored objective. Changing α only slightly away from the boundary (i.e., to $\alpha \in \{0.1, 0.9\}$) sacrifices only a little on the main objective.

The plot reveals that setting $\alpha = 0.1$ leads, in general, to the most balanced solutions and so operators of systems should consider weighting the objectives in this region. In particular, over all instances tested, there is no difference in the number of CV trips in the optimal solutions obtained when changing $\alpha = 0.0$ to $\alpha = 0.1$. The same change of α

results in a reduction of total travel time from 236,693 second to 199,341, on average over all instances, a decrease of 15%. We also see a significant drop in number of CV trips as we change alpha from 1.0 to 0.9. On average, the number of CV trips decreases from 928 to 862 (a 7.1% decrease), while only resulting in an increase of average total travel time from 197,273 to 199,341 (a 1.0% increase). This will therefore lead to a significant decrease in operations costs and environmental impact with only slightly longer total passenger travel time, and so should be employed in operational decision making.

6.4. Effect of time-window variation on solutions

This subsection provides an analysis of how varying the time window affects the solutions obtained. In particular, we consider the solution time and quality of solutions obtained over all instances tested, for $T_w = 5$ and $T_w = 10$. In general, the quality of the solutions obtained are only marginally different when we widen the time windows. Averaged over all instances, the number of CV trips is 888 and 877 and the total travel time is 213,018 and 198,535, for $T_w = 5$ and $T_w = 10$, respectively. This represents a decrease in number of CV trips of 1.21% and a decrease in total travel time of 6.80%. The increase in solution time is much more substantial, increasing from 14.44 to 40.05 seconds, on average a 177.41% increase. This indicates that allowing more flexibility in arrival time constraints makes the problem significantly harder, but results in slightly better operational decisions and so an operator might try to solve the problem with relatively large time windows, but then decrease this flexibility should solutions need to be obtained expediently.

6.5. Effect of increasing CV capacity

We also evaluated the solution quality and the solution time of **CG** on instances with increasing vehicle capacity. We selected instances with medium sized attributes, namely $|\mathcal{D}| = 25, T_w = 10, n = 3,750$ (so that $\frac{n}{|\mathcal{D}|} = 150$), and hence $|V| = 225$, using the 5 generated instances per configuration. We tested with $v^{\text{cap}} \in \{5, 6, 7, 8, 9, 10\}$.

The results are summarized in Table 1. For a fixed α, the solution times are generally longer with lower vehicle capacity. The solution changes significantly as v^{cap} grows for small α, whereas the differences in the optimal solution for different v^{cap} are marginal when α is larger. This can be attributed to the fact that, as α grows, the emphasis in the objective shifts to total wait time, and CVs will depart the terminal as quickly as possible — even if that means not filing capacity. Note that the problems are harder to solve at lower capacity.

α	v^{cap}	Average travel time	Average number of CV trips	Average solution time
0	5	209,030.2	759.2	3.724
0	6	208,846.0	635.6	1.706
0	7	208,928.6	547.6	1.986
0	8	209,162.0	481.4	2.178
0	9	209,575.4	431.2	2.624
0	10	209,864.6	392.4	2.796
0.1	5	186,014.8	759.2	6.426
0.1	6	186,897.8	635.6	1.772
0.1	7	188,177.8	547.6	2.034
0.1	8	189,933.8	481.4	2.392
0.1	9	191,461.0	431.2	2.514
0.1	10	192,911.2	392.4	2.626
0.5	5	184,019.6	769.6	11.664
0.5	6	184,155.4	651.6	1.814
0.5	7	184,815.8	565.6	1.950
0.5	8	185,433.4	503.4	2.128
0.5	9	185,808.2	459.4	2.312
0.5	10	186,391.4	421.0	2.584
0.9	5	183,060.8	792.4	22.826
0.9	6	181,419.2	742.8	3.668
0.9	7	181,665.6	671.4	1.886
0.9	8	181,720.4	638.4	2.210
0.9	9	181,817.4	610.8	2.314
0.9	10	181,861.4	591.6	2.590
1	5	182,952.0	822.2	29.004
1	6	181,029.0	819	18.448
1	7	180,719.2	821.4	18.182
1	8	180,573.4	822.4	18.942
1	9	180,491.6	823.4	20.844
1	10	180,430.0	822.8	21.520

Table 1 Comparison of solution quality and runtime of CG for increasing CV capacity.

6.6. Problem generalizations

There are various assumptions we place on the transportation system in order to ensure the optimality of the solutions obtained by **CG**. In particular, we require the structural results from § 4. Although the proof of optimality remains valid when we extend to time-dependent travel times, there are other dimensions of the problem that we can expand which results in **CG** returning only heuristic solutions. As discussed in Raghunathan et al. (2018), the solutions obtained can still be close to optimal even in more general settings. This subsection provides an analysis of two main problem generalization, and uncovers that the solutions we identify through **CG** are often superior to what can be found by other techniques, even in the more general settings where the solutions may not be optimal.

6.6.1. Express trains The structural result from § 4 which proves that there exists an optimal solution where passengers going to a common destination are partitioned in the order of their arrival times fails to remain true when there are more complex train

n	$\lvert\mathcal{D}\rvert$	α	$n_{\mathbf{CG}}$	$\hat{t}_{\mathbf{CG}}$	$\hat{z}_{\mathbf{CG}}$	$n_{\mathbf{IP}}$	$\hat{t}_{\mathbf{IP}}$	$\hat{z}_{\mathbf{IP}}$
500	10	0	5	0.1	11760.0	0	601.5	11760.0
500	10	1	5	1.6	18691.4	5	33.7	18535.2
750	10	0	5	0.2	15780.0	0	603.8	15880.0
750	10	1	5	2.4	27095.8	5	14.5	26820.6
1250	25	0	5	0.3	29480.0	0	609.8	29680.0
1250	25	1	5	3.6	43983.0	5	144.3	43688.2
1875	25	0	5	0.4	39740.0	0	618.7	40260.0
1875	25	1	5	5.6	68745.8	5	239.3	68088.0

Table 2 Comparison of runtime and solution quality between CG and IP on instances with express trains.

systems that are integrated with last-mile transportation. For example, if trains do not arrive sequentially at regular intervals, stopping at each and every station, this result no longer holds. This is a common characteristic of real-world train systems, where express trains increase the frequency that high-traffic train stops have train service.

In order to test how well **CG** performs as a heuristic in this setting, we generate instances with express trains as follows. We generated instances with $|\mathcal{D}| \in \{10, 25\}$ and specify the number of passengers per destination $\frac{n}{|\mathcal{D}|} \in \{50, 75\}$. We set $|V| = \mathsf{round}(0.1 * n)$ which is a larger number of CVs as a fraction of passengers than the used in the previous tests. This was done primarily to provide the MIP formulation the benefit of solving more of the instances. We generate 5 instances per configuration. The set of passengers are generated independently at random as described in the beginning of the section. The trips for the trains consist of express trips that stop only at stations 4 and 2 before reaching the $\mathbf{T}_0$. These trips occur every 30 time units and can be specified using the notation of the paper as: $\tilde{t}(c,4) \in \{0, 30, \ldots, 180\}$, $\tilde{t}(c,2) = \tilde{t}(c,4) + 10$, $\tilde{t}(c) = \tilde{t}(c,4) + 20$ and $\tilde{t}(c,s) = -\infty$ for $s = 1, 3$. Another set of express trips stop only at stations 3 and 1 before reaching the $\mathbf{T}_0$. These trips also occur every 30 time units and can be specified as: $\tilde{t}(c,4) \in \{20, 50, \ldots, 200\}$, $\tilde{t}(c,1) = \tilde{t}(c,3) + 10$, $\tilde{t}(c) = \tilde{t}(c,4) + 15$ and $\tilde{t}(c,s) = -\infty$ for $s = 2, 4$. We refer to **IP** as the using an exact model for this problem setting (Raghunathan et al. 2018).

Table 2 reports the results obtained. The first three columns report the instance configuration. The next three columns, $n_{\mathbf{CG}}$, $\hat{t}_{\mathbf{CG}}$, and $\hat{z}_{\mathbf{CG}}$, report the number of instances solved, the average solution time, and the average objective function value for **CG**, respectively, over the five instances in that configuration. The final three columns, $n_{\mathbf{IP}}$, $\hat{t}_{\mathbf{IP}}$, $\hat{z}_{\mathbf{IP}}$, report the same statistics, but for **IP**. All instances were solved by **CG** in reasonable time, i.e., under 6 seconds, whereupon **IP** is only able to solve those instances with $\alpha = 1$ and even for these instances takes at least an order-of-magnitude more time, on average. Since **CG** is solving a more constrained version of the problem, its solutions can be worse than that of **IP**, but this improvement is marginal, at most $\sim 5\%$ worse. Alternatively, when **IP** is unable to find the provably optimal solution, **CG** is able to find a better solution, up to 2% better than the solution obtained by **IP** in 10 minutes. Since the solutions time for **CG** are also often an order-of-magnitude faster than the solution times for **IP**, if an operator needs high-quality solutions quickly, the solution obtained by **CG** could suffice.

6.6.2. Multiple destinations per CV trip Another restrictive assumption inherent in the BDD-based approach is that the CVs visit one-and-only-one destination per trip. Raghunathan et al. (2018) provided an analysis of how far from optimal the solutions obtained by single-destination-per-CV trip solutions are from those obtained by IP allowing passengers going to non-common destinations for small-scale problems. We extend that analysis here and find even more encouraging results.

We provide this comparison on the following instances. For this case, we used the same problem setting as in our conference paper Raghunathan et al. (2018). For sake of brevity, we refer the interested reader to the section on Experiments in Raghunathan et al. (2018) for a description on the routes for the commuter vehicles. In this case, the number of destinations is 10. The number of passengers is chosen as $n \in \{500, 750\}$. The train trips are retained as described in the beginning of the section. Again, we report results for $\alpha = 0, 1$.

For $\alpha = 0$, all instances were solved by **CG** in under one second, and none were solved by **IP** in over three hours. **IP** only found a feasible solution to two instances in the three-hour time limit, with **CG** finding solutions at least as good as **IP** for both of those instances. For $\alpha = 1$, **IP** ran out of memory, and **CG** solved all instances within 2.09 seconds. Despite lacking optimality guarantees, **CG** can be used as a heuristic where **IP** fails to identify any feasible solution. A significant drawback with the IP formulation is that the number of variables in the optimization problem scales with the number of possible route choices. The number of possible route choices when passengers with different destinations share a CV grows exponentially as, ${}^{K}P_1 + {}^{K}P_2 + \cdots + {}^{K}P_{v^{\text{cap}}}$. As a consequence, the loading of the **IP** model in memory consumes a significant amount of computational time ($\sim$1 hour) and solution of the linear relaxation at the root node also takes a comparable amount of time. This emphasizes the need for developing alternative methods for this problem generalization. Until such an algorithm is developed, **CG** offers a computationally inexpensive and scalable approach to obtaining high-quality solutions.

7. Conclusion and future work

In this paper we introduce an optimization algorithm for solving the problem of scheduling passengers on multiple legs of a last-mile transportation system. We study a version of the problem where in the last leg of the transportation system, passengers are transported via small-capacity commuter vehicles (CVs) to a limited set of destinations. In particular, we study a variant of the problem where each CV trip carries passengers to a common destination, showing that this simplified version remains NP-hard. The optimization framework relies on a decomposition of the problem into a collection of small-sized decision diagrams that can be mutually optimized over. This algorithm is shown to dramatically outperform existing techniques. Our experimental evaluation indicates that the algorithm developed can scale to problems of practical size and that considering conflicting objectives of minimizing passenger wait times with the total number of CV trips simultaneously does not significantly hinder the performance on either objective taken individually.

The potential for expansion of this work is vast, as automated transportation networks becomes a reality. The variant studied in this paper is a simplified version of real-world systems, where CVs can stop in multiple destinations. The work of Raghunathan et al. (2018) indicates that the gap between the optimal solutions obtained by limiting CVs to stop at only one destination per trip might not be far from the optimal solutions in the more general version of the problem. In fact, the solutions obtained by the decision diagram-based model can be used as a heuristic to more general variants, and the numerical evaluation in this paper suggests that the solutions can be found very quickly and are of high quality. Given the substantial savings in the long-run for any such improvement, adopting the methods developed in this paper to the more general case might be an interesting research direction. Incorporating more real-world features like time-dependent travel times on the CVs, dynamic scheduling and response to traffic are additional dimensions that could potentially be added to the models developed in this paper. This work gives a critical and substantial step towards understanding how to solve challenging automated scheduling problems in the context of automated commuter systems.

Appendix A: Proof of Theorem 1

Proof. We first show that the feasibility of ILMTP-SD is in NP. If we are given a solution consisting of the CV boarded by each passenger and the boarding time, then we can easily verify the feasibility of the solution. First, we check if, for each passenger, that there is a mass transit service that could bring the passenger to the terminal before the boarding time. In the worst case, this is proportional to the number of passengers times the number of mass transit trips. Second, for each CV we define a vector of tuples, each of which consists of the boarding time and destination of a passenger using that CV. After sorting each of those vectors by the boarding times, we check with a linear pass on the vectors if (i) passengers boarding the CV at the same time have the same destination; and (ii) the time between consecutive trips is sufficient for the CV to return to the terminal.

Next, we show that a decision version of the *bin packing* problem can be reduced in polynomial time and space to the feasibility of the ILMTP-SD. The bin packing problem can be stated as: Given a set of bins $B_1, B_2, \ldots$ with identical capacity V and a list of n items with sizes $a_1, \ldots, a_n$, does there exist a packing using at most M bins?

The Karp reduction (Karp 1972) is as follows. We define an ILMTP-SD instance with n passengers and M CVs, where passenger p_i corresponds to item i of the bin packing problem. Each of those passengers has a distinct destination d_i and the round trip time is $\tau(d_i) = a_i$. We assume that there is a single mass transit trip that can bring these passengers to $\mathbf{T}_0$ and that it arrives to $\mathbf{T}_0$ at time t_0, whereas the CVs must return to $\mathbf{T}_0$ by time $t^{\max} = t_0 + V$. Finally, the origin of each passenger is irrelevant, the capacity of each CV is $v^{\text{cap}} = 1$, we assume a time window $T_w = +\infty$, and the objective coefficient $\alpha = 1$.

If there is a feasible solution to the ILMTP-SD problem above, then the bin packing problem has an affirmative answer. Namely, let $\mathcal{P}^i$ be the set of all passengers that board CV i in the solution, which are on different trips since the CV capacity is 1. Assign the items corresponding to those passengers to bin i. Since the first passenger in $\mathcal{P}^i$ to board CV i left $\mathbf{T}_0$ after t_0 and the CV returned back to $\mathbf{T}_0$ before $t^{\max} = V$, the sum of the duration of the trips for the passengers in i must not exceed V. Therefore the associated items fit into bin i. Hence, each CV corresponds to a bin and all passengers boarding a given CV are assigned to that bin, using at most M bins.

Conversely, if there is no feasible solution the ILMTP-SD problem, then the bin packing problem has a negative answer. If the bin packing problem has a solution using at most M bins, then we can construct a solution for the corresponding ILMTP-SD instance through the same transformation.

Therefore, the feasibility of the ILMTP-SD is as hard as the bin packing decision problem, which is known to be NP-complete (Garey and Johnson 1979). □

Appendix B: Proof of Theorem 2

Before proceeding with the proof of Theorem 2, we present a lemma.

Lemma 1 *Consider any feasible solution* $\mathsf{g} = \{g_1, \ldots, g_\gamma\}$ *to the ILMTP-SD , with associated departure time from* $\mathbf{T}_0$ *of* t_l^{g} *, for* $l = 1, \ldots, \gamma$*. For any two passengers* j', j'' *for which* $d(j') = d(j'')$ *and* $\mathsf{g}(j') \neq \mathsf{g}(j'')$*, if the solution* $\mathsf{g}' = \{g'_1, \ldots, g'_\gamma\}$ *defined by*

$$g'_l = \begin{cases} g_l & : g_l \notin \{\mathsf{g}(j'), \mathsf{g}(j'')\} \\ g_l \setminus \{j'\} \cup \{j''\} & : g'_l = \mathsf{g}(j') \\ g_l \setminus \{j''\} \cup \{j'\} & : g'_l = \mathsf{g}(j''), \end{cases} \qquad \forall l = 1, \ldots, \gamma$$

with $t_l^{\mathsf{g}'} = t_l^{\mathsf{g}}$*, is a feasible solution, then the objective function value of both solutions is the same.*

Proof of Lemma 1. Switching the two passengers does not add or delete any CV trips, and hence, $(1-\alpha)$ times the number of CV trips remains unchanged. We need only ensure that the total wait time at $\mathbf{T}_0$ remains unchanged.

Let l', l'' be the indices of the groups that passengers j', j'' are assigned to in $\mathtt{g}$, respectively. Furthermore, let τ', τ'' be the time that passengers j', j'' arrive to $\mathbf{T}_0$ on their mass transit trips, respectively. The wait time at $\mathbf{T}_0$ for passenger j' is changed from $t^{\mathtt{g}}_{l'} - \tau'$ to $t^{\mathtt{g}}_{l''} - \tau'$, a net change of $t^{\mathtt{g}}_{l''} - t^{\mathtt{g}}_{l'}$. By the same argument, the net change in wait time for passenger j'' is $t^{\mathtt{g}}_{l'} - t^{\mathtt{g}}_{l''}$. The net changes in j', j'' cancel out. Since wait time of other passengers are not affected, the result holds.

Equipped with Lemma 1, we can now prove Theorem 2.

Proof of Theorem 2. By way of contradiction, suppose there exists an instance for which there is no optimal solution satisfying the condition of the theorem for a $d^* \in \mathcal{D}$. Consider the optimal solution for which the smallest indexed passenger j that violates this condition is maximized. Let j^* be the smallest index in this solution for which there exists a k with $\mathtt{g}(j^*) = \mathtt{g}(j^*+k)$ and $\mathtt{g}(j^*+1) \neq \mathtt{g}(j^*)$. Let k^* be such an index k, and let $g_\ell = \mathtt{g}(j^*)$ and $g_{\ell'} = \mathtt{g}(j^*+1)$. We first show that $j \geq j^*+1$ for all $\{j \,|\, j \in g_{\ell'}\}$. Suppose not; let $\hat{j} = \arg\max\{j \,|\, j \in g_{\ell'}, j < j^*+1\}$ (which will be non-empty by assumption). Then passenger j', $j'+k'$ with $j' = \hat{j}, k' = j^*+1-\hat{j}$ satisfy $j', j'+k' \in \mathtt{g}_{\ell'}$ and $j'+1 \notin \mathtt{g}_{\ell'}$ are a set of indices violating the claim of the theorem with $j' < j^*$, contradicting the minimality of j^*. Hence, (j^*+1) is the minimum index among all $j \in g_{\ell'}$.

In the remainder of the proof, we construct another solution in which the index j^* does not violate the claims of the theorem. This contradicts the maximality of j^* among all optimal solutions, thereby establishing the result.

Conditioning on the relative values of the departure times of the CVs for g_ℓ and $g_{\ell'}$, first consider the case where $t^{\mathtt{g}}_\ell \leq t^{\mathtt{g}}_{\ell'}$. We claim that exchanging the group assignment of j^*+1 and j^*+k^* and holding all else equal results in another feasible solution. For all passengers j, let $\mathtt{rel}(j), \mathtt{ded}(j)$ be $t^r(j) - T_w, t^r(j) + T_w$, respectively. We need only show that (a) $\mathtt{rel}(j^*+1) \leq t^{\mathtt{g}}_\ell + \tau^1(d^*) \leq \mathtt{ded}(j^*+1)$ and (b) $\mathtt{rel}(j^*+k^*) \leq t^{\mathtt{g}}_{\ell'} + \tau^1(d^*) \leq \mathtt{ded}(j^*+k^*)$. (a) follows because any passenger can be placed in a group with fewer than v^{cap} passengers without breaking feasibility if the passenger's request time is before or after at least one other passenger. The first inequality in (b) follows because $\mathtt{rel}(j^*+k^*) \leq t^{\mathtt{g}}_\ell + \tau^1(d^*)$, by the feasibility of the original solution, and $t^{\mathtt{g}}_\ell + \tau^1(d^*) \leq t^{\mathtt{g}}_{\ell'} + \tau^1(d^*)$, by assumption. The second inequality in (b) holds because $\mathtt{ded}(j^*+1) \leq t^{\mathtt{g}}_{\ell'} + \tau^1(d^*)$, by the feasibility of the original solution and $\mathtt{ded}(j^*+1) \leq \mathtt{ded}(j^*+k^*)$, by the ordering of passengers. Furthermore, by Lemma 1 the objective function remains unchanged by this exchange, and is therefore optimal. If the resulting solution satisfies the claim of this theorem, then the claim holds. If not, then the claim of this theorem is violated for another $j > j^*$; contradicting the maximality of j^*.

We now consider the alternative case where $t^{\mathtt{g}}_\ell > t^{\mathtt{g}}_{\ell'}$. The exchange from the previous case may not work because putting passenger j^*+k^* into group $g_{\ell'}$ may not be feasible. Additionally, if there exist $k' > 0$ passengers in g_ℓ with indices lower than j^* then these passengers must be $j^*-k', \ldots, j^*-1$. If not, it would contradict the assumption of j^* as the smallest index in the optimal solution violating the claim of this theorem. Define k' so that j^*-k' is the minimum indexed passenger in group g_ℓ, which is 0 if j^* is the minimum indexed passenger.

Consider the following two-step exchange—for $i=0,\ldots,k'$, move each passenger j^*-i from g_ℓ into $g_{\ell'}$. Then, move the $k'+1$ passengers with the highest indices in the resulting $g_{\ell'}$ into g_ℓ. The resulting groups have the same cardinality as they originally had, and so by Lemma 1 the objective values remain the same.

We now show that the resulting solution is valid and then show that the choice of optimal solution contradicts the maximality among optimal solutions assumption on the selection of j^*, which concludes the proof. Any passenger $j^*-i\in g_\ell$ for $i=0,...,k'$ can be moved to $g_{\ell'}$ without violating (j^*-i)'s arrival time window because $\texttt{rel}(j^*-i)\le\texttt{rel}(j^*+1)\le t^{\mathsf{g}}_{\ell'}+\tau^1(d^*)<t^{\mathsf{g}}_{\ell}+\tau^1(d^*)$ and $\texttt{ded}(j^*-i)\ge t^{\mathsf{g}}_{\ell}+\tau^1(d^*)>t^{\mathsf{g}}_{\ell'}+\tau^1(d^*)$. Additionally, any passenger $j\in g_{\ell'}$ can be moved to g_ℓ without violating the arrival time windows because $\texttt{rel}(j)\le t^{\mathsf{g}}_{\ell'}+\tau^1(d^*)<t^{\mathsf{g}}_{\ell}+\tau^1(d^*)$ and $\texttt{ded}(j)\ge\texttt{ded}(j^*+1)\ge\texttt{ded}(j^*)\ge t^{\mathsf{g}}_{\ell}+\tau^1(d^*)$ where first inequality follows from the result that (j^*+1) is the minimum index for all $j\in g_{\ell'}$. Hence, the resulting solution is also optimal. Finally, if the resulting solution violates the claim of this theorem, then the smallest index must be larger than j^*. This again contradicts the maximality of j^* among all optimal solutions, as assumed.

Appendix C: Proof of Theorem 4

Proof. Constructivelly from Algorithm 1, the number of nodes in the DD is at most $(n_d+1)\cdot v^{\text{cap}}=O(n_d\cdot v^{\text{cap}})$. Each node has at most $T_w\cdot 2+1$ arcs, so that the maximum number of arcs is $(n_d+1)\cdot v^{\text{cap}}\cdot(T_w*2+1)=O(n_d\cdot v^{\text{cap}}\cdot T_w)$, as desired. Since it takes constant time to create each arc, the time bound follows.

For property **(DD-1)**, fix an arbitrary destination d, and consider any path $p\in\mathcal{P}^d$ and a passenger j^d_i. Consider the first one-arc in p below layer i (including that layer). Note that since the last layer only contains one-arcs, such an arc exists. Suppose this one-arc a' is directed out of node u in layer $L^d_{i'}$, with $i'\ge i$. By construction, the state of u must be greater than or equal to $i'-i$. $g(a)$ therefore contains j^d_i, and so j^d_i is in at least one group in $\mathsf{g}(p)$. Furthermore, any one-arc below layer $L^d_{i'}$ that connects $\omega(a')$ to $\mathbf{t}^b$ can only select passengers $j^d_{i''}$, with $i''\ge i'$, and so j^d_i appears in exactly one $\mathsf{g}(p)$.

Property **(DD-2)** is satisfied by construction—each one-arc a' directed out of node u satisfies that every passenger in $g(a)$ will arrive to d within the desired range of arrival times.

What remains to be shown is that property **(DD-3)** is satisfied. Fix an arbitrary $\mathsf{g}'\in\mathcal{G}^d$. Let $g_1,\ldots,g_f$ be the f groups in g', ordered by the index of the passengers in the groups. Let $j^d_{i_h}$ be the highest indexed passenger in each group h, $h=1,\ldots,f$. We proceed by induction on h to show that there is a sequence of zero-arcs from the node on layer $\mathsf{L}^d_{i_h}$ to layer $\mathsf{L}^d_{i_{h+1}-1}$ to a node u with state $i_{h+1}-1-i_h$, and then a one-arc from u to the node on layer $\mathsf{L}^d_{i_{h+1}}$ with state 0 for every time that the passengers can mutually depart from $\mathbf{T}_0$.

We first establish the base case. Starting from $\mathbf{r}^d$, follow i_1-1 zero-arcs until layer $\mathsf{L}^d_{i_1-1}$. Because g_1 is in g', each $t\in\left\{t^e(j^d_{i_1}),\ldots,t^l(j^d_1)\right\}$ is a feasible departure time from $\mathbf{T}_0$ for g_1, and so a one-arc directed out of the node on this layer with state i_1-1 will be added to D^d with the arc-start-time t.

By induction on h, consider the node u on layer $\mathsf{L}^d_{i_h}$ with state 0. Follow $i_{h+1}-1-i_h$ zero-arcs. This will end at a node u' on layer $\mathsf{L}^d_{i_{h+1}-1}$ with state $i_{h+1}-1-i_h$. Since $g_h \in \mathsf{g}'$, each time $t \in \left\{t^e\left(j^d_{i_{h+1}}\right), \ldots, t^l\left(j^d_{i_h}+1\right)\right\}$ is a feasible departure time from $\mathbf{T}_0$ for all passengers in g_j, and so a one-arc from u' to the node on layer $\mathsf{L}^d_{i_{h+1}}$ with state 0 will be added to D^d with this arc-start-time.

References

Agatz N, Erera A, Savelsbergh M, Wang X (2012) Optimization for dynamic ride-sharing: A review. *European Journal of Operational Research* 223(2):295 – 303.

Agussurja L, Cheng SF, Lau HC (2018) A state aggregation approach for stochastic multi-period last-mile ride-sharing problems. *Transportation Science* .

Andersen HR, Hadzic T, Hooker JN, Tiedemann P (2007) A constraint store based on multivalued decision diagrams. *Proceedings of the 13th International Conference on Principles and Practice of Constraint Programming*, 118–132, CP'07 (Berlin, Heidelberg: Springer-Verlag), ISBN 978-3-540-74969-1, URL `http://dl.acm.org/citation.cfm?id=1771668.1771682`.

Anderson JE (1998) Control of personal rapid transit systems. *J. Adv. Transportation* 32(1):57–74.

Barnhart C, Johnson EL, Nemhauser GL, Savelsbergh MWP, Vance PH (1998) Branch-and-price: Column generation for solving huge integer programs. *Operations Research* 46(3):316–329.

Berger T, Sallez Y, Raileanu S, Tahon C, Trentesaux D, Borangiu T (2011) Personal rapid transit in an open-control framework. *Comput. Indust. Engrg.* 61(2):300–312.

Bergman D, Cire A, van Hoeve W, Hooker J (2016) *Decision diagrams for optimization* (Springer).

Bergman D, Cire AA (2016) Decomposition based on decision diagrams. Quimper CG, ed., *Integration of AI and OR Techniques in Constraint Programming*, 45–54 (Cham: Springer International Publishing), ISBN 978-3-319-33954-2.

Bergman D, Cire AA (2018) Discrete nonlinear optimization by state-space decompositions. *Management Science* 0(0):null, URL `http://dx.doi.org/10.1287/mnsc.2017.2849`.

Bergman D, van Hoeve WJ, Hooker JN (2011) Manipulating mdd relaxations for combinatorial optimization. Achterberg T, Beck JC, eds., *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, 20–35 (Berlin, Heidelberg: Springer Berlin Heidelberg), ISBN 978-3-642-21311-3.

Bertsekas DP (1999) *Nonlinear Programming* (Belmont, MA: Athena Scientific).

Bertsekas DP (2012) *Dynamic Programming and Optimal Control*, volume 1 (Athena Scientific), 4th edition.

Bly PH, Teychenne R (2005) Three financial and socio-economic assessments of a personal rapid transit system. *Proc. 10th Internat. Conf. Automated People Movers*, 1–16.

Boyan J, Mitzenmacher M, Mitzenmacher M (2001) Improved results for route planning in stochastic transportation. *Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms*, 895–902, SODA '01.

Brake J, Nelson JD, Wright S (2004) Demand responsive transport: Towards the emergence of a new market segment. *J. Transport Geography* 12(4):323–337.

Campbell AM, Savelsbergh M (2004) Efficient insertion heuristics for vehicle routing and scheduling problems. *Transportation Science* 38((3):369–378.

Chase R (2017) Shared mobility principles for livable cities. `https://www.sharedmobilityprinciples.org/`, accessed: 2018-07-20.

Chevrier R, Jourdan ALL, Dhaenens C (2012) Solving a dial-a-ride problem with a hybrid evolutionary multi-objective approach: Application to demand responsive transport. *Appl. Soft Comput.* 12(4):1247–1258.

Ciré AA, Hooker JN (2014) The separation problem for binary decision diagrams. *International Symposium on Artificial Intelligence and Mathematics (ISAIM).*

Cire AA, van Hoeve WJ (2013) Multivalued decision diagrams for sequencing problems. *Operations Research* 61(6):1411–1428, URL `http://dx.doi.org/10.1287/opre.2013.1221`.

Clarke G, Wright JW (1964) Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research* 12(4):568–581.

Cordeau JF, Laporte G (2007) The dial-a-ride problem: Models and algorithms. *Ann. Oper. Res.* 153(1):29–46.

Cordeau JF, Laporte G, Potvin JY, Savelsbergh MW (2007) Transportation on demand. Barnhart C, Laporte G, eds., *Transportation*, volume 14 of *Handbooks in Operations Research and Management Science*, 429 – 466 (Elsevier).

Daganzo CF (1978) An approximate analytic model of many-to-many demand responsive transportation systems. *Transportation Res.* 12(5):325–333.

Diana M, Dessouky MM, Xia N (2006) A model for the fleet sizing of demand responsive transportation services with time windows. *Transportation Res. Part B: Methodological* 40(8):651–666.

Gange G, Stuckey PJ, Szymanek R (2011) Mdd propagators with explanation. *Constraints* 16(4):407, ISSN 1572-9354, URL `http://dx.doi.org/10.1007/s10601-011-9111-x`.

Garey M, Johnson D (1979) *Computers and Intractability: A Guide to the Theory of NP-Completeness* (W. H. Freeman).

Grosse-Ophoff A, Hausler S, Heineke K, Möller T (2017) How shared mobility will change the automotive industry. URL `https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/how-shared-mobility-will-change-the-automotive-industry`.

Horn ME (2002a) Fleet scheduling and dispatching for demand responsive passenger services. *Transportation Res. Part C: Emerging Tech.* 10(1):35–63.

Horn ME (2002b) Multi-modal and demand-responsive passenger transport systems: A modelling framework with embedded control systems. *Transportation Res. Part A: Policy Practice* 36(2):167–188.

Häme L, Hakula H (2013) Dynamic journeying under uncertainty. *European Journal of Operational Research* 225(3):455 – 471.

Jaw JJ, Odoni AR, Psaraftis HN, Wilson NH (1986) A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows. *Transportation Res. Part B: Methodological* 20(3):243–257.

Karp R (1972) Reducibility among combinatorial problems. Miller R, Thatcher J, eds., *Complexity of Computer Computations*, 85–103 (Springer).

Lees-Miller JD, Hammersley JC, Davenport N (2009) Ride sharing in personal rapid transit capacity planning. *12th Internat. Conf. Automated People Movers*, 321–332.

Lees-Miller JD, Hammersley JC, Wilson RE (2010) Theoretical maximum capacity as benchmark for empty vehicle redistribution in personal rapid transit. *Transportation Res. Record: J. Transportation Res. Board* 2146(1):76–83.

Lei H, Laporte G, Guo B (2012) Districting for routing with stochastic customers. *Eur. J. Transportation Logist.* 1(1?2):67–85.

Liu Z, Jiang X, Cheng W (2012) Solving in the last mile problem: Ensure the success of public bicycle system in beijing. *Procedia Soc. Behav. Sci.* 43:73–78.

Mageean J, Nelson JD (2003) The evaluation of demand responsive transport services in europe. *J.Transport Geography* 11(4):255–270.

Mahéo A, Kilby P, Hentenryck PV (2018) Benders decomposition for the design of a hub and shuttle public transit system. *Transportation Science* .

McCoy K, Andrew J, Glynn R, Lyons W (2018) Integrating shared mobility into multimodal transportation planning: Improving regional performance to meet public goals. Technical report, Office of the Assistant Secretary of Transportation for Research and Technology, U.S. Department of Transportation.

Morrison DR, Sewell EC, Jacobson SH (2016) Solving the pricing problem in a branch-and-price algorithm for graph coloring using zero-suppressed binary decision diagrams. *INFORMS Journal on Computing* 28(1):67–82, URL `http://dx.doi.org/10.1287/ijoc.2015.0667`.

Mueller K, Sgouridis SP (2011) Simulation-based analysis of personal rapid transit systems: Service and energy performance assessment of the masdar city prt case. *J. Adv. Transportation* 45(4):252–270.

Palmer K, Dessouky M, Abdelmaguid T (2004) Impacts of management practices and advanced technologies on demand responsive transit systems. *Transportation Res. Part A: Policy Practice* 38(7):495–509.

Perez G, Régin J (2017) Soft and cost MDD propagators. Singh SP, Markovitch S, eds., *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, 3922–3928 (AAAI Press), URL `http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14544`.

Pinto HKRdF, Hyland M, Verbas IO, Mahmassani HS (2018) Integrated mode choice and dynamic traveler assignment-simulation framework to assess the impact of a suburban first-mile shared autonomous vehicle fleet service on transit demand. *Transportation Research Board 97th Annual Meeting*, number 18-06707 in TRB Annual Meeting.

Quadrifoglio L, Dessouky MM, Ordóñez F (2008) A simulation study of demand responsive transit system design. *Transportation Res. Part A: Policy Practice* 42(4):718?737.

Raghunathan AU, Bergman D, Hooker J, Serra T, Kobori S (2018) The Integrated Last-Mile Transportation Problem (ILMTP). *International Conference on Automated Planning and Scheduling*, 388–397.

Riley C, Legrain A, Hentenryck PV (2019) Column generation for real-time ride-sharing operations. *Proceedings of the 16th International Conference on the Integration of Constraint Programming, Artificial Intelligence and Operations Research*, 472–487.

Salhi S, Nagy G (1999) A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling. *Journal of the Operational Research Society* 50:1034–1042.

Savelsbergh M, Van Woensel T (2016) City logistics: Challenges and opportunities. *Transportation Science* 50(2):579–590.

Shaheen S (2004) U.S. carsharing & station car policy considerations: Monitoring growth, trends & overall impacts. Technical report, Institute of transportation studies, Working paper series, Institute of Transportation Studies, UC Davis.

Shen Y, Zhang H, Zhao J (2017) Simulating the First Mile Service to Access Train Stations by Shared Autonomous Vehicle. *Transportation Research Board 96th Annual Meeting.*

Stiglic M, Agatz N, Savelsbergh M, Gradisar M (2015) The benefits of meeting points in ride-sharing systems. *Transportation Research Part B: Methodological* 82:36 – 53.

Thien ND (2013) *Fair cost sharing auction mechanisms in last mile ridesharing.* Ph.D. thesis, Singapore Management University.

Toth P, Vigo D (2014) *Vehicle Routing: Problems, Methods and Applications* (SIAM), second edition.

Uber (2018) Express POOL: A more affordable shared ride. `https://www.uber.com/ride/express-pool/`, accessed: 2018-07-20.

van Heeswijk WJA, Mes MRK, Schutten JMJ (2017) The delivery dispatching problem with time windows for urban consolidation centers. *Transportation Science* .

Vigo D (1996) A heuristic algorithm for the asymmetric capacitated vehicle routing problem. *European Journal of Operational Research* 89(1):108–126.

Wang H (2017) Routing and Scheduling for a Last-Mile Transportation Problem. *Transportation Science* 1–17 (forthcoming), URL `http://dx.doi.org/https://doi.org/10.1287/trsc.2017.0753`.

Wilson NH, Hendrickson C (1980) Performance models of flexibly routed transportation services. *Transportation Res. Part B: Methodological* 14(1):67–78.

Ömer Verbas, Mahmassani HS, Hyland MF (2016) Gap-based transit assignment algorithm with vehicle capacity constraints: Simulation-based implementation and large-scale application. *Transportation Research Part B: Methodological* 93:1 – 16.