Integrating Public Transport in Sustainable Last-Mile Delivery: Column Generation Approaches

Diego Delle Donne delledonne@essec.edu[1], Alberto Santini alberto.santini@upf.edu[1,2,3,4], and Claudia Archetti archetti@essec.edu[1]

[1] Department of Information Systems, Decision Sciences and Statistics, ESSEC Business School, 3 Avenue Bernard Hirsch, 95021, Cergy, France
[2] Institute of Advanced Studies, CY Cergy Paris Université, 1 Rue Descartes, 95000, Neuville-sur-Oise, France
[3] Department of Economics and Business, Universitat Pompeu Fabra, Carrer Ramon Trias-Fargas, 25–27, 08005, Barcelona, Spain
[4] Data Science Centre, Barcelona School of Economics, Carrer Ramon Trias-Fargas, 25–27, 08005, Barcelona, Spain

24th December 2024

Abstract

We tackle the problem of coordinating a three-echelon last-mile delivery system. In the first echelon, trucks transport parcels from distribution centres outside the city to public transport stops. In the second echelon, the parcels move on public transport and reach the city centre. In the third echelon, zero-emission vehicles pick up the parcels at public transport stops and deliver them to customers. We introduce two extended formulations for this problem. The first has two exponential sets of variables, while the second has one. We propose column generation algorithms and compare several methods to solve the pricing problems on specially constructed graphs. We also devise dual bounds, which we can compute even when the graphs are so large that not a single round of pricing completes within the time limit. Compared to previous formulations, our models find 17 new best known solutions out of an existing dataset of 24 from the literature.

Keywords:last-mile delivery, logistics, column generation, routing problems, resource-constrained shortest path problems

1 Introduction

The e-commerce boom and its growth projections (Alfonso et al. 2021) raise new challenges as retailers and couriers innovate their supply chains to keep up with demand. Last-mile delivery (LMD), the supply chain segment that starts at the last distribution centre and ends at the customer's doorstep, is particularly affected. Its nature changed when retailers stopped delivering to stores only and started delivering directly to consumers: couriers now handle many small parcels instead of fewer, larger shipments; they deliver during tight time windows when

Figure 1: Two pilot projects launched in 2021 in Germany and England to use spare capacity on public transport vehicles. On the left: operators loading parcels on a tram operated by AVG in Karlsruhe, Germany. In 2021, the Mobility Cluster of Karlsruhe started a pilot project named "RegioKargo" to transport light cargo on the region's tram and commuter train network. Photo by Michael Krauth, AVG, reproduced with permission. On the right: in 2021, Orion Railways launched a pilot project to deliver parcels on commuter trains to London Euston station, from where they are delivered using cargo bikes. Photo by Rails Ops Group reproduced with permission.

customers are at home; they deal in real-time with newly incoming orders while their fleet is already busy shipping other parcels.

The volume growth in LMD has also raised concerns, especially in dense urban environments where the externalities—traffic, emissions, noise, congestion—have become noticeable (see, e.g., the review by Viu-Roig and Alvarez-Palau (2020) and the recent papers by Wang, Rabinovich and Guda (2023), Majoral, Gasparín and Saurí (2021) and Caspersen (2021)). Several authors in Operational Research (OR), Environmental Engineering, Urban Planning, and Economics have proposed alternative LMD implementations to reduce externalities while guaranteeing a timely and cost-effective service. Solutions range from deliveries using autonomous drones, bicycles, or porters (Silva, Amaral and Fontes 2023) to congestion taxes and economic incentives. For an OR perspective on LMD services, we refer the reader to the two recent surveys by Archetti and Bertazzi (2021) and Boysen, Fedtke and Schwerdfeger (2021).

In the rest of this paper, we focus on a promising operational practice: integrating public transport within LMD to leverage the unused capacity of public transit vehicles and reduce the number of delivery vans in the city. This concept has emerged during the last years and is gaining traction, both academically (see Section 2) and in practice, as demonstrated by numerous pilot projects (Baron 2019; Sustainable Bus 2021; Der Spiegel 2020; Saito and Shimbun 2021; Clinnick 2020; Longhorn 2021; Antkowiak 2018), technical reports (Edrington et al. 2017; Deloison et al. 2020; Segura et al. 2020) and patents (Bhatt 2019).

In our study, we consider a three-echelon system. In the first echelon, transport trucks move parcels from distribution centres to public transport stops. In the second echelon, public transport vehicles transport the parcels towards the city centre and leave them at some of their scheduled stops. In the third echelon, zero-emission vehicles, such as cargo bikes, deliver the parcels to the customers' preferred locations.

This system reduces the number of vehicles on urban roads as parcels enter the city on public transport rather than on trucks. As mentioned above, pilot projects are exploring the feasibility of this approach using almost all types of transit vehicles: buses (Saito and Shimbun 2021), trams (Sustainable Bus 2021; Antkowiak 2018), commuter trains (Clinnick 2020; Longhorn 2021) and metro (Der Spiegel 2020). Most of these pilot projects started in 2020 or 2021 as a response to the increase in e-commerce which has followed the COVID-19 pandemic (see, e.g., Guthrie, Fosso-Wamba and Arnaud (2021), Villa and Monzón (2021b) and Beckers et al. (2021)). Figure 1 shows two examples of pilot projects launched in 2021, which use spare capacity on public transport vehicles to transport parcels. Mobility company AVG of Karlsruhe, Germany, uses trams, while Orion uses commuter trains in London, UK.

This paper proposes new formulations for the *Three-Tier Delivery Problem using Public Transportation* (3T-DPPT), a problem first introduced by Mandal and Archetti (2023). The main contributions of the present paper are the following:

- We introduce a 3T-DPPT extended formulation using two exponential sets of variables. We propose a column generation algorithm and solve separate pricing subproblems for each of the two exponential sets of variables. In one of the two subproblems, the resulting shortest path problem features time-dependent label costs. We propose two novel labelling algorithms to solve this subproblem: one explicitly exploits the time dependence, while the other operates on a transformed graph where costs no longer depend on time. We embed the column generation approaches in a restricted master heuristic, thus obtaining a heuristic algorithm for the 3T-DPPT.
- We perform a computational campaign showing that our algorithm outperforms, especially on larger instances, a compact formulation and a semi-compact formulation where one of the two sets of exponential variables mentioned above is replaced by a polynomial set of variables (with a different set of constraints). In addition, our heuristic outperforms the heuristics proposed by Mandal and Archetti (2023) by producing primal solutions with better objective values.

The paper is organised as follows. In Section 2, we survey the relevant literature. We define the problem in Section 3 and describe the extended formulations in Section 4. The latter section also introduces a semi-compact formulation containing only one set of exponentially many variables. The main characteristics of the column generation approach are illustrated in Section 5, while Section 6 describes additional speed-up techniques. We report computational results in Section 7 and conclusions in Section 8.

2 Related literature

This section positions our work in the growing literature on integrating public transport with LMD. Systems integrating parcel delivery and public transport are inherently multi-echelon. Indeed, public transit vehicles cannot be in charge of the parcels' first and last legs, i.e., picking them up at warehouses and delivering them to customers' homes. In most systems, parcels are transshipped twice; therefore, three echelons arise. From this point of view, the corresponding optimisation problems share some similarities with multi-echelon location-routing problems (Drexl and Schneider 2015). Although some authors have adapted hub location problems (Ji et al. 2020; Zhao et al. 2018) and location-routing problems (Gianessi et al. 2016) to the scenario of hybrid transit-freight LMDs, there are important differences between the two fields. Location-routing problems combine strategic location decisions with operational routing ones. This aspect is largely absent in the literature on integrating public transport and LMD, and the two decisions are usually made sequentially.

At the strategic level, Delle Donne, Alfandari et al. (2023) have studied long-term decisions arising specifically when planning a 3T-DPPT system. The decisions concern which public transport lines and stops to use in the 3T-DPPT, considering that the corresponding vehicles and stops will be equipped appropriately, at a cost. The objective is to maximise the number of parcels served through the system, subject to a maximum number of lines and stops to include. They propose a column generation approach and perform an extended computational campaign and a sensitivity analysis. They conclude that, given a limited budget, a planner should equip a limited number of vehicles and stops with high capacity rather than expanding the 3T-DPPT network to many lines and stops.

On the other hand, problems such as our 3T-DPPT are only concerned with operational decisions and assume that the public transport lines and the stops involved in the integrated network have already been fixed. In this line of work, we mention the contributions of Mandal and Archetti (2023), Ghilas, Demir and Van Woensel (2016b) and Ghilas, Demir and Van Woensel (2016a).

As mentioned in Section 1, Mandal and Archetti (2023) introduced the 3T-DPPT. In the problem name, the word "tier" is a synonym of "echelon", which is more popular in the logistics literature. Unlike our setting, in (Mandal and Archetti 2023), first-echelon transport trucks can park and wait indefinitely at public transport stops. Because such a practice would be hardly acceptable in real implementations, in our problem, we force trucks to unload parcels as soon as they arrive at a stop. The authors proposed a compact formulation, which they showed to be too large to be used in practice, and decomposition approaches. They split the problem into three subproblems corresponding to the three echelons of the delivery network and solve each part in sequence, using the solution obtained in a subproblem to constrain the decision space of the next one. They obtain three heuristic algorithms differing in the order in which the subproblems are solved. The authors test their approach on instances with 10–80 customers, which they identify as the largest instances for which they can obtain primal feasible solutions using the decomposition heuristic.

Like our work, Ghilas, Demir and Van Woensel (2016b) and Ghilas, Demir and Van Woensel (2016a) consider integrating generic scheduled public transport lines with LMD operations. The authors propose two variants of the Pickup and Delivery Problem with Time Windows (PDPTW), in which part of each request's journey can happen on a scheduled line. The first variant, studied in (Ghilas, Demir and Van Woensel 2016b), is the PDPTW with Scheduled Lines (PDPTWSL); the second, studied in (Ghilas, Demir and Van Woensel 2016a), is the PDPTWSL with Stochastic Demands (PDPTWSLSD). In both variants and different from our setting, the same trucks which bring the parcels to public transport stations can also perform the final delivery to customers. The authors also assume infinite capacity and no maximum waiting time at public transport stations and that parcel transhipments can only happen at end-of-line terminal stations. In the PDPTWSLSD, the amount of capacity that each parcel takes on the vehicles (i.e., the demand) is unknown—although distributed according to a known distribution—until the first truck picks it up, making the problem stochastic. If the realised demand for a parcel turns out to be too large for the capacity of the vehicle scheduled to pick it up, the authors assume the payment of a penalty due to outsourcing the shipment. In (Ghilas, Demir and Van Woensel 2016b), the authors propose an ALNS heuristic, and in (Ghilas, Demir and Van Woensel 2016a), they extend it, combining it with a Sample Average Approximation method to adapt it to the stochastic case. In both cases, they observe savings in operational costs compared to the truck-only scenario, but they do not analyse the potential environmental benefits.

Another difference with location-routing literature is that the parcels' itinerary is severely constrained both in space and time in at least one echelon. Indeed, public transit lines follow a fixed itinerary and a predetermined schedule. Some authors have thus focused on only optimising the echelon involving public transit. In such a case, a natural question is if the transit schedule can be improved to facilitate freight operations alongside passenger ones. In (Hörsting and Cleophas 2023), the authors study a system where a single tram line is used to move parcels, sharing capacity with passengers. Given a known passenger demand during the day, they aim to schedule parcel transport with a bi-objective model that minimises passenger inconvenience and delivery delays. Once this deterministic model is solved using a black-box integer solver, the authors validate the solution with a stochastic event-based simulation in which parcels are still deterministic, but passenger flows are stochastic.

Other approaches deal with the complexity of a three-echelon distribution system by applying some simplifying assumptions. Like our work, Masson et al. (2017) consider a three-echelon model with trucks moving parcels from distribution centres to bus stops. Buses bring the parcels—packed in roll containers—to designated stops, where tricycles perform deliveries to end customers. The authors make three assumptions. First, bus lines start at the distribution centre, reducing the number of echelons from three to two. Second, they assume that only one bus line is available. Finally, they assume that each tricycle can carry exactly one roll container. Therefore, no consolidation of parcels travelling on different buses but delivered by the same rider is possible; analogously, parcels carried on the same bus cannot be taken to different stops to be picked up by different riders. The authors minimise the number of tricycle riders needed to satisfy demand and, secondarily, their total travel time. They develop an Adaptive Large Neighbourhood Search (ALNS) heuristic, which they use to solve a case study in La Rochelle, France, involving one bus line with eight stops and up to 303 customers.

He and Yang (2018) studied the possibility of delivering parcels using urban buses in Dalian, China. In this case, the simplifying assumptions are that parcels can be loaded onto buses only at the start terminal of a line and unloaded only at the end terminal. Using an Ant Colony Algorithm, the authors minimise fixed and variable costs, compensation costs for excess carbon emissions, and late-delivery penalties. Their results show savings of up to 10% and a reduction of CO_2 emissions of up to 13%.

Not all works about integrating public transport with LMD use optimisation techniques to make the integrated system efficient. The contributions we describe in the following focus on case studies and are useful to assess the real-life impact of such systems on various metrics, such as costs, carbon emissions and passenger delays.

Kikuta et al. (2012) carried out a 2-week pilot project in Sapporo, Japan, where heavy snowfall poses significant challenges to parcel distribution in winter. The authors partnered with Sapporo's transport authority and a logistic operator to use the metro to carry hand-pushed carts with parcels during three off-peak times daily. The parcels moved from a suburban logistic centre to the inner city, where they went out for delivery. The authors concluded that this system reduced both congestion in surface roads and CO_2 emissions. Zhou and Zhang (2020) analysed the possibility of using a metro line for parcel delivery in three configurations: using trolley carts on a regular car, dedicating a car to freight cargo but within an otherwise passenger train, and using a freight-only train. They evaluate carbon emissions reductions of up to 50% for the scenario in which high parcel demand and commuters' off-hours justify using a dedicated train. Recently, Villa and Monzón (2021a) proposed using Madrid's metro system for parcel delivery. The authors devise a system in which packets travel on the metro and are stored at station lockers, where customers can pick them up at their convenience. They compare two modes of operation: shared trains, used by both parcels and commuters and dedicated freight-only trains. They quantify the economic, environmental and social costs required to implement such a system and find that, after an initial investment, operators can reduce logistic costs by 11–14%.

Bruzzone, Cavallaro and Nocera (2021) present two case studies of integrating LMD and public transport, both focused on sparsely populated areas. The first case concerns the farthest islands of the Venice lagoon in Italy, which are served by two water-bus lines; the second case includes the town of Velenje, Slovenia, and its neighbouring villages, which are collectively served by five bus lines. The authors assess the economic, environmental, and social advantages of carrying parcels on the (water-)buses and conclude that these advantages are more pronounced when delivery and pickup locations are limited, travel demand is inelastic, and parcel recipients are willing to pick their parcels up at public transport stops (thus eliminating the need for the third echelon to move parcels from stops to customers' houses). Through an analysis of both technical and legal literature, they conclude that regulatory aspects, more than technical challenges, hinder the wider adoption of such integrated systems.

3 Problem definition

In this section, we formally define the 3T-DPPT. The main notation used throughout the paper is reported in Table 1. Figure 2 provides a visual representation of the three-echelon system. Dashed arrows make up the first-echelon truck routes, coloured solid lines represent bus lines, and dotted arrows make up the third-echelon courier routes. Dots are public transport stops; the filled ones are the stops where buses pick up parcels and the empty ones are the stops where buses deliver them. The CDC is depicted as a rectangle, while customers are represented as small pentagons.

The objective of the problem is to deliver parcels to a set $\mathcal{C}$ of customers. Each customer $c \in \mathcal{C}$ must receive one parcel of size $q_c \geq 0$ (size here refers generically to the amount of capacity the parcel would use on a vehicle) during a delivery time window $\left[\underline{T}_c, \bar{T}_c\right]$ at a location of choice. We assume that all delivery requests are known in advance and that all parcels start their journey at a Consolidation and Distribution Centre (CDC), denoted with o.

In the first tier, trucks carry parcels from the CDC to public transport stops. We denote with $\mathcal{D}$ the set of trucks. We consider homogeneous vehicles with capacity Q^{D}, although our formulation can be naturally extended to a heterogeneous fleet. Let $\mathcal{S}^{\mathrm{in}}$ be the set of public transport stops that the trucks can use to unload a parcel so that a bus will later pick it up (in the following, we use the term "bus" for simplicity, although any form of public transport could be equally used). We refer to stops of $\mathcal{S}^{\mathrm{in}}$ as "in-stops". Unloading parcels at stop $s \in \mathcal{S}^{\mathrm{in}}$ incurs a service time T_s. Parcels can wait at the stop for a maximum time $W^{\max}$. The service time models the handling of the parcel: a parcel unloaded by the truck at time t will be ready for pick-up by a bus at time $t + T_s$. The maximum wait time can be used to ensure that bus stops are not used for long-term parcel storage.

The driving time between two locations $u, v \in \{o\} \cup \mathcal{S}^{\mathrm{in}}$ is denoted with l_{uv}. We assume that each truck route is elementary, i.e., it visits each in-stop at most once. This requirement, however, would be easy to drop given our method to handle the first tier (see Section 4) because it would correspond to dropping the elementarity requirement from a shortest-path problem. Finally, we impose that trucks depart from the in-stop as soon as they have unloaded the corresponding parcels. As mentioned in Section 2, this is the only respect in which our definition differs from that of Mandal and Archetti (2023). We introduce this requirement because a truck parked at a stop would unduly occupy the public road and hinder the operations of public transport vehicles. If, for some particular application, this requirement does not apply, it can be straightforwardly removed from our model.

Set	Description
$\mathcal{C}$	Set of customers.
$\mathcal{D}$	Set of first-echelon trucks.
$\mathcal{S}$	Set of bus stops.
$\mathcal{P}$	Set of buses.
$\mathcal{K}$	Set of couriers.
$\mathcal{S}^{\text{in}} \subset \mathcal{S}$	Stops where buses can pick parcels up (in-stops).
$\mathcal{S}^{\text{out}} \subset \mathcal{S}$	Stops where buses can deliver parcels (out-stops).
$\mathcal{S}^{\text{in}}_c \subset \mathcal{S}^{\text{in}}$	In-stops that can be used to deliver a parcel to customer $c \in \mathcal{C}$.
$\mathcal{S}^{\text{out}}_c \subset \mathcal{S}^{\text{out}}$	Out-stops from where customer $c \in \mathcal{C}$ can be served by a courier.
$\mathcal{S}_p \subseteq \mathcal{S}$	Sequence of bus stops served by bus $p \in \mathcal{P}$.
$\mathcal{S}^{\text{in}}_{pc} \subseteq \mathcal{S}^{\text{in}}_c$	In-stops served by a bus $p \in \mathcal{P}$ that can carry the parcel of customer $c \in \mathcal{C}$.
$\mathcal{R}^{\text{D}}$	Set of feasible truck routes.
$\mathcal{R}^{\text{D}}_c \subseteq \mathcal{R}^{\text{D}}$	Feasible truck routes which carry the parcel of customer $c \in \mathcal{C}$.
$\mathcal{R}^{\text{D}}_{spc} \subseteq \mathcal{R}^{\text{D}}_c$	Feasible routes that a truck can use to take the parcel of customer $c \in \mathcal{C}$ to in-stop $s \in \mathcal{S}^{\text{in}}_c$, at a time at which bus $p \in \mathcal{P}$ may pick it up.
$\mathcal{R}^{\text{F}}$	Set of feasible courier routes.
$\mathcal{R}^{\text{F}}_c \subseteq \mathcal{R}^{\text{F}}$	Feasible courier routes which deliver the parcel of customer $c \in \mathcal{C}$.
$\mathcal{R}^{\text{F}}_s \subseteq \mathcal{R}^{\text{F}}$	Feasible courier routes starting and ending at out-stop $s \in \mathcal{S}^{\text{out}}$.
$\mathcal{R}^{\text{F}}_{spc} \subseteq \mathcal{R}^{\text{F}}_c$	Set of feasible routes which a courier can use to deliver the parcel of customer $c \in \mathcal{C}$ after bus $p \in \mathcal{P}$ delivers it at out-stop $s \in \mathcal{S}^{\text{out}}_c$.
$\mathcal{P}^{\text{in}}_{sc} \subseteq \mathcal{P}$	Buses serving in-stop $s \in \mathcal{S}^{\text{in}}$ at a time compatible with carrying the parcel of customer $c \in \mathcal{C}$.
$\mathcal{P}^{\text{out}}_{sc} \subseteq \mathcal{P}$	Buses serving out-stop $s \in \mathcal{S}^{\text{out}}$ at a time compatible with carrying the parcel of customer $c \in C$.

Parameter	Description
$q_c \geq 0$	Size of the parcel of customer $c \in \mathcal{C}$.
$\underline{T}_c \geq 0$	Start of the delivery time window at customer $c \in \mathcal{C}$.
$\bar{T}_c \geq \underline{T}_c$	End of the delivery time window at customer $c \in C$.
$T_s \geq 0$	Service time at bus stop $s \in \mathcal{S}$.
$T_c \geq 0$	Service time at customer $c \in \mathcal{C}$.
$t^s_p \geq 0$	Scheduled arrival time of bus $p \in \mathcal{P}$ at stop $s \in \mathcal{S}_p$.
o	Location of the CDC.
$l_{uv} \geq 0$	Travel time between locations u and v. Defined for $u, v \in \{o\} \cup \mathcal{S}^{\text{in}}$ and $u, v \in \mathcal{S}^{\text{out}} \cup \mathcal{C}$.
$Q^{\text{D}} \geq 0$	Truck capacity.
$Q_p \geq 0$	Capacity of bus $p \in \mathcal{P}$.
$Q^{\text{F}} \geq 0$	Courier capacity.
$W^{\max} \geq 0$	Maximum time a parcel can wait at a stop.
$L^{\max} \geq 0$	Maximum duration of a courier route.
$s_k \in \mathcal{S}^{\text{out}}$	Origin out-stop of courier $k \in \mathcal{K}$.
$n_s \in \mathbb{N}$	Number of available couriers at out-stop $s \in \mathcal{S}^{\text{out}}$.
$c_R \geq 0$	Cost of route $r \in \mathcal{R}^{\text{D}} \cup \mathcal{R}^{\text{F}}$.
$G = (V, A)$	Graph used for time-dependent labelling.
$H = (W, B)$	Graph used for scalar-cost labelling.
$t_a \geq 0$	Travel time of arc $a \in A$ in the graph used for time-dependent labelling (Section 5.1.1).
$\nu_{sc}(t) \in \mathbb{R}$	Benefit of arriving at vertex $(s, c) \in V$ at time $t \geq 0$ when using time-dependent labelling.
I_{ps}	Time interval during which it is feasible for a parcel ready at bus stop $s \in \mathcal{S}^{\text{in}}$ to be loaded onto bus $p \in \mathcal{P}$.
$C_P(t) \in \mathbb{R}$	Time-dependent reduced cost function of a path P in graph G, when the truck leaves the CDC at time t.
$\Lambda_s \in \mathbb{N}^+$	Number of time intervals during which the dual values associated with a stop $s \in \mathcal{S}$ are constant in a given pricing problem iteration.
$w_{s\ell}$	The ℓ-th time interval during which the dual values associated with stop $s \in \mathcal{S}$ are constant.
$\text{mrc}_s(c) \in \mathbb{R}$	Highest reduced cost achievable by delivering the parcel of customer $c \in \mathcal{C}$ at in-stop $s \in \mathcal{S}^{\text{in}}_c$.
v^+_s and v^-_s	Checkpoint vertices added to the time-dependent labelling graph G.

Variable	Description
$x_r \in \{0, 1\}$	Takes value one if and only if route $r \in \mathcal{R}^{\text{D}}$ is used.
$y_r \in \{0, 1\}$	Takes value one if and only if route $r \in \mathcal{R}^{\text{F}}$ is used.
$z^{\text{in}}_{spc} \in \{0, 1\}$	Takes value one if and only if bus $p \in \mathcal{P}^{\text{in}}_{sc}$ picks up the parcel of customer $c \in \mathcal{C}$ at in-stop $s \in \mathcal{S}^{\text{in}}_c$.
$z^{\text{out}}_{spc} \in \{0, 1\}$	Takes value one if and only if bus $p \in \mathcal{P}^{\text{out}}_{sc}$ unloads the parcel of customer $c \in C$ at out-stop $s \in \mathcal{S}^{\text{out}}$.

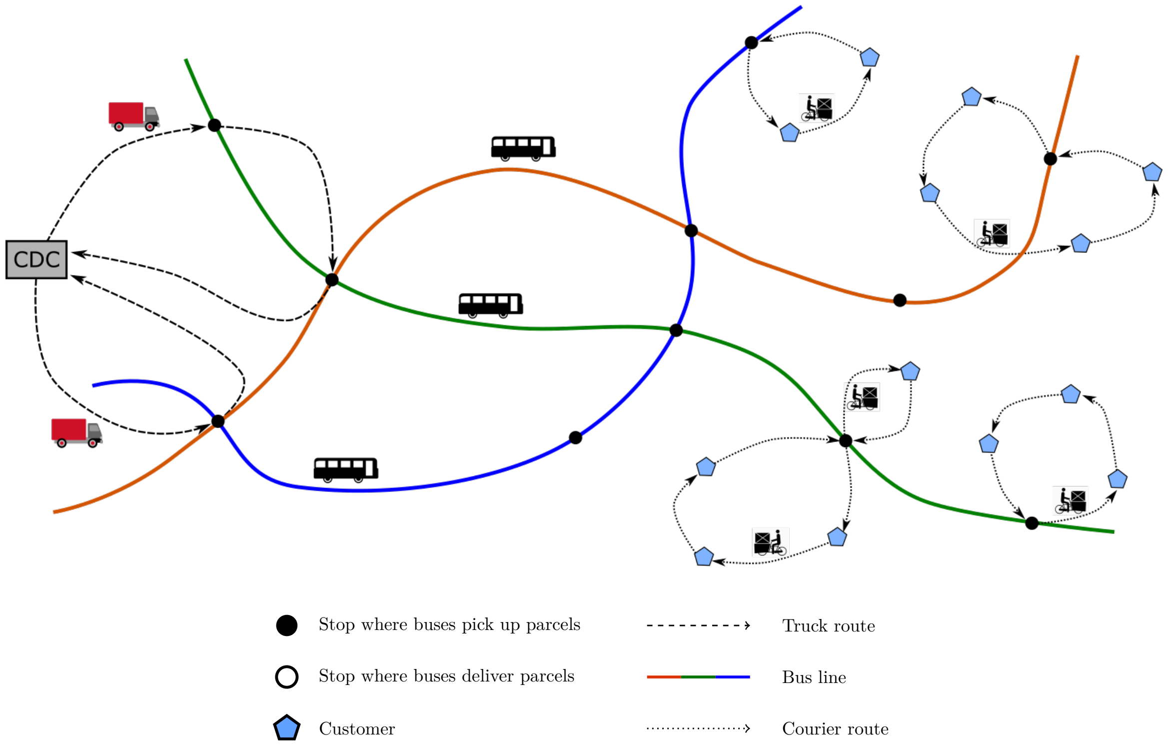

Figure 2: Illustrative example of a last-mile delivery system based on the 3T-DPPT.

We remark that we make no assumptions on how trucks continue their service. Because in-stops are usually geographically concentrated and available spare capacity on public transport is also concentrated during off-peak hours, trucks can be used for standard delivery service during the rest of the day. For example, if trucks should be used for home deliveries after visiting the in-stops, the planner can devise open routes that end at the last visited in-stop by setting $l_{uo} = 0 \; \forall u \in \mathcal{S}^{\text{in}}$.

In the second tier, each parcel travels between two bus stops aboard a bus. We denote the set of buses with $\mathcal{P}$. Each bus $p \in \mathcal{P}$ has capacity Q_p and serves a route $\mathcal{S}_p = (s_p^1, \dots, s_p^{|\mathcal{S}_p|})$ represented by the ordered list of stops the bus visits. To denote that a stop s is part of the route of bus p, we write simply $s \in \mathcal{S}_p$. The scheduled arrival time of p at stop $s \in \mathcal{S}_p$ is t_p^s. Because bus routes operate according to periodic timetables, set $\mathcal{P}$ contains multiple buses operating the same route at different times. Moreover, in real life, a single bus can operate a given line many times during the day. Such a bus would be represented by multiple entries in $\mathcal{P}$.

In the third tier, couriers move parcels between bus stops and customer locations. The set of bus stops enabled for courier pick up is denoted as $\mathcal{S}^{\text{out}}$; we refer to these as "out-stops". Because trucks do not enter the city centre and couriers do not go outside the city, each stop is either used for truck-to-bus transhipment (if outside the centre) or for bus-to-courier transhipment (if in the city centre), and thus $\mathcal{S}^{\text{in}} \cap \mathcal{S}^{\text{out}} = \emptyset$. For the same reason, in-stops precede out-stops in all bus routes. Each stop $s \in \mathcal{S}^{\text{out}}$ has a service time T_s and a maximum wait time $W^{\max}$, analogous to those of $\mathcal{S}^{\text{in}}$. We denote with $\mathcal{K}$ the set of couriers who can perform the deliveries to the customers. A courier $k \in \mathcal{K}$ has capacity Q^{F} and is assigned to a stop $s_k \in \mathcal{S}^{\text{out}}$. Each courier starts the route at s_k, visits a number of customers, and finally comes back to the same stop. Courier travel time between locations $u, v \in \mathcal{S}^{\text{out}} \cup \mathcal{C}$ is denoted as l_{uv} (because the set of locations which trucks and couriers can visit is disjoint, the notation is not ambiguous). Each courier route has a maximum duration of $L^{\max}$, which includes both the travel time and service times T_c incurred when delivering parcels to customers $c \in \mathcal{C}$. We assume a homogeneous fleet of couriers and denote with n_s the number of couriers which can start from stop $s \in \mathcal{S}^{\text{out}}$. To facilitate delivery operations, each customer c has an associated subset of stops $\mathcal{S}_c^{\text{out}} \subseteq \mathcal{S}^{\text{out}}$ from which they can be served. For example, stops $s \in \mathcal{S}^{\text{out}}$ such that $l_{sc} + T_c + l_{cs} > L^{\max}$ are excluded from $\mathcal{S}_c^{\text{out}}$ because they cannot be used to serve c while respecting the courier's maximum route duration time. Other real-life operational considerations can lead to excluding further stops, e.g., if a stop is too far from the customer or if it is not possible to reach the customer using a safe bike path.

The goal of the 3T-DPPT is to determine a distribution plan to deliver all parcels at minimum cost. The cost is the sum of the route costs of the first- and third-tier vehicles.

4 Formulation

To introduce our formulation for the 3T-DPPT, we must consider some additional notation. We denote the set of in-stops which can be used to deliver a parcel to customer c as $\mathcal{S}_c^{\text{in}}$. We build this set and the other sets introduced in this section in Appendix A. We denote with $\mathcal{R}_c^{\text{D}}$ the set of feasible truck routes which carry the parcel of customer c and with $\mathcal{R}^{\text{D}}$ the set of all feasible truck routes. A feasible route must visit each in-stop at most once, deliver each parcel on board the truck to exactly one in-stop, start and end at the CDC, respect the capacity limit, and have an assigned start time from the CDC.

With an analogous notation, $\mathcal{R}_c^{\text{F}}$ denotes the set of feasible courier routes which deliver customer c's parcel, while $\mathcal{R}^{\text{F}}$ is the set of all feasible courier routes. A feasible route must start and end at the same out-stop, visit each customer whose parcel is on-board exactly once and during the corresponding time window, and respect both the courier's capacity and the maximum route duration limit. We can also partition $\mathcal{R}^{\text{F}}$ based on the starting out-stop of the route; in this case, set $\mathcal{R}_s^{\text{F}}$ denotes all feasible courier routes starting and ending at $s \in \mathcal{S}^{\text{out}}$.

Finally, we introduce two more subsets of routes. The first, $\mathcal{R}_{spc}^{\text{D}} \subseteq \mathcal{R}_c^{\text{D}}$, contains all feasible routes that a truck can use to take customer c's parcel to in-stop $s \in \mathcal{S}_c^{\text{in}}$, at a time at which bus $p \in \mathcal{P}$ may pick it up. The second, $\mathcal{R}_{spc}^{\text{F}} \subseteq \mathcal{R}_c^{\text{F}} \cap \mathcal{R}_s^{\text{F}}$, contains all feasible routes which a courier can use to deliver customer c's parcel after bus $p \in \mathcal{P}$ delivers it at out-stop $s \in \mathcal{S}_c^{\text{out}}$. These two sets are characterised more precisely in Appendix A.

Both truck and courier routes have an associated cost $c_r > 0$ (for $r \in \mathcal{R}^{\text{D}} \cup \mathcal{R}^{\text{F}}$). Our formulation allows the cost to depend on the route characteristics in many ways to adapt to real-life circumstances, and the only assumption we make on route costs is that they are strictly positive. For example, in the case of trucks, the cost is often proportional to the distance travelled (fuel cost) or to the travel time (driver hourly compensation). In the case of couriers, the cost can be proportional to the travel time or a step function of the travelled distance (as is common in crowdsourcing platforms). For consistency with Mandal and Archetti (2023), we consider route costs proportional to the travelled distance, with different multipliers for drivers and couriers.

In the following formulation, we denote the sets of buses that serve in-stop (out-stop) s at a time compatible with carrying the parcel of customer c as $\mathcal{P}_{sc}^{\text{in}}$ ($\mathcal{P}_{sc}^{\text{out}}$), and the set of in-stops served by a bus p that can carry the parcel of customer c as $\mathcal{S}_{pc}^{\text{in}}$. These sets are defined more formally in Appendix A.

Our formulation uses the following sets of variables: $x_r \in \{0,1\}$ taking value 1 iff truck route $r \in \mathcal{R}^{\text{D}}$ is used; $y_r \in \{0,1\}$ taking value 1 iff courier route $r \in \mathcal{R}^{\text{F}}$ is used; $z_{spc}^{\text{in}} \in \{0,1\}$ with value 1 iff bus $p \in \mathcal{P}_{sc}^{\text{in}}$ picks up the parcel of customer $c \in \mathcal{C}$ at in-stop $s \in \mathcal{S}_c^{\text{in}}$; $z_{spc}^{\text{out}} \in \{0,1\}$ with value 1 iff bus $p \in \mathcal{P}_{sc}^{\text{out}}$ unloads the parcel of customer $c \in \mathcal{C}$ at out-stop $s \in \mathcal{S}_c^{\text{out}}$. The extended formulation of the 3T-DPPT then reads as follows:

$$
\begin{aligned}
\min \quad & \sum_{r \in \mathcal{R}^{\mathrm{D}}} c_r x_r + \sum_{r \in \mathcal{R}^{\mathrm{F}}} c_r y_r && && \text{(1a)} \\
\text{s.t.} \quad & \sum_{r \in \mathcal{R}^{\mathrm{D}}} x_r \leq |\mathcal{D}| && && \text{(1b)} \\
& \sum_{r \in \mathcal{R}^{\mathrm{F}}_s} y_r \leq n_s && \forall s \in \mathcal{S}^{\mathrm{out}} && \text{(1c)} \\
& \sum_{s \in \mathcal{S}^{\mathrm{in}}_c} \sum_{p \in \mathcal{P}^{\mathrm{in}}_{sc}} z^{\mathrm{in}}_{spc} = 1 && \forall c \in \mathcal{C} && \text{(1d)} \\
& \sum_{s \in \mathcal{S}^{\mathrm{in}}_{pc}} z^{\mathrm{in}}_{spc} = \sum_{s \in \mathcal{S}^{\mathrm{out}}_{pc}} z^{\mathrm{out}}_{spc} && \forall c \in \mathcal{C},\ \forall p \in \mathcal{P}_c && \text{(1e)} \\
& \sum_{c \in \mathcal{C}} q_c \sum_{s \in \mathcal{S}^{\mathrm{in}}_{pc}} z^{\mathrm{in}}_{spc} \leq Q_p && \forall p \in \mathcal{P} && \text{(1f)} \\
& z^{\mathrm{in}}_{spc} \leq \sum_{r \in \mathcal{R}^{\mathrm{D}}_{spc}} x_r && \forall c \in \mathcal{C},\ \forall p \in \mathcal{P}_c,\ \forall s \in \mathcal{S}^{\mathrm{in}}_{pc} && \text{(1g)} \\
& z^{\mathrm{out}}_{spc} \leq \sum_{r \in \mathcal{R}^{\mathrm{F}}_{spc}} y_r && \forall c \in C,\ \forall p \in \mathcal{P}_c,\ \forall s \in \mathcal{S}^{\mathrm{out}}_{pc} && \text{(1h)} \\
& \sum_{r \in \mathcal{R}^{\mathrm{D}}_c} x_r \geq 1 && \forall c \in \mathcal{C} && \text{(1i)} \\
& \sum_{r \in \mathcal{R}^{\mathrm{F}}_c} y_r \geq 1 && \forall c \in \mathcal{C} && \text{(1j)} \\
& x_r \in \{0,1\} && \forall r \in \mathcal{R}^{\mathrm{D}} && \text{(1k)} \\
& y_r \in \{0,1\} && \forall r \in \mathcal{R}^{\mathrm{F}} && \text{(1l)} \\
& z^{\mathrm{in}}_{spc} \in \{0,1\} && \forall c \in C,\ \forall s \in \mathcal{S}^{\mathrm{in}}_c,\ \forall p \in \mathcal{P}^{\mathrm{in}}_{sc} && \text{(1m)} \\
& z^{\mathrm{out}}_{spc} \in \{0,1\} && \forall c \in C,\ \forall s \in \mathcal{S}^{\mathrm{out}}_c,\ \forall p \in \mathcal{P}^{\mathrm{out}}_{sc}. && \text{(1n)}
\end{aligned}
$$

The objective function (1a) minimises the combined costs of routing trucks and couriers. Constraints (1b) and (1c) ensure, respectively, that the maximum number of available trucks and couriers (the latter at each out-stop) is not exceeded. Constraint (1d) asserts that each parcel is picked up by one bus, while constraint (1e) ensures that the same bus picks up and delivers each parcel. Constraint (1f) ensures that the bus capacity is respected. Constraints (1g) and (1h) link, respectively, variables z^{in} with x and variables z^{out} with y. Finally, constraints (1i) and (1j) state that the parcel of each customer must be included in at least one truck route and at least one courier route, respectively. These constraints are not required for a correct formulation of the problem. The following theorem, proved in the supplemental material's Section 1, shows that, despite their simplicity, (1i) and (1j) are not implied by the other inequalities in fractional solutions, and can considerably improve the quality of the linear relaxation of (1a)–(1n).

Theorem 1. *Denote with* (1^-) *formulation* (1) *without constraints* (1i) *and* (1j) *and with* $LP(\star)$ *the linear relaxation of a generic formulation* $\star$. *Then* $\mathrm{LP}(1^-)$ *can be arbitrarily bad, i.e.,* $\frac{\mathrm{LP}(1)}{\mathrm{LP}(1^-)} \to \infty$ *in the worst case.*

As we will discuss in Section 5.3, if we use formulation (1) within a column generation scheme, the pricing problem associated with variables x is extremely challenging because it involves finding shortest paths on a graph with time-dependent costs. Therefore, we also evaluate an alternative model in which we replace the truck-route variables x with three polynomial-sized sets of variables. We present the resulting *semi-compact formulation* in Appendix C.

5 Column generation

Formulation (1a)–(1n), which we denote as the Master Problem (MP), uses two exponential sets of routes: $\mathcal{R}^{\mathrm{D}}$ and $\mathcal{R}^{\mathrm{F}}$. We call the corresponding variables x and y, taken together, the *columns* of MP. Because sets $\mathcal{R}^{\mathrm{D}}$ and $\mathcal{R}^{\mathrm{F}}$ are too large to enumerate in practice, we use a column generation approach initially considering reduced sets of variables. We initialise these sets with a small number of columns, as explained in Section 4 of the supplemental material. The resulting formulation is the Reduced Master Problem (RMP). We then consider the continuous relaxation of RMP, known as the Reduced Relaxed Master Problem (RRMP), obtained by replacing integrality constraints (1k)–(1n) with non-negativity constraints.

In the remainder of this section, we explain how to solve the continuous relaxation of MP, denoted $\mathrm{MP}_{\textsc{cont}}$, by iteratively solving RRMP and generating new variables x and y which improve the objective value. At each iteration of the column generation algorithm, we perform three tasks:

1. We solve the RRMP and collect the dual values associated with the constraints involving variables x and y. We denote with $\lambda^{(n)}$ the dual variable associated with constraint (n) of the model. For example, $\lambda^{(1g)}_{spc}$ will refer to the dual variable associated with the inequality (1g) indexed by s, p and c. We consider RRMP in its canonical form and therefore obtain non-negative dual variables.

2. We solve a column-generation (pricing) subproblem SP_x to find x variables with negative reduced cost. Denoting with $\mathcal{C}_r$ the set of customers whose parcels are carried on a truck in route $r \in \mathcal{R}^{\mathrm{D}}$, the reduced cost of a variable x_r is

$$c_r + \lambda^{(1b)} - \sum_{c \in \mathcal{C}_r} \lambda^{(1i)}_c - \sum_{c \in \mathcal{C}_r} \sum_{p \in \mathcal{P}_c} \sum_{\substack{s \in \mathcal{S}^{\mathrm{out}}_{pc} \\ \text{s.t. } r \in \mathcal{R}^{\mathrm{D}}_{spc}}} \lambda^{(1g)}_{spc}. \tag{2}$$

   The reduced cost contains a fixed term (dual variable $\lambda^{(1b)}$) and dual prizes for each customer whose parcel is transported (dual variables $\lambda^{(1i)}_c$) and for each bus which can pick up the parcel at the in-stop (dual variables $\lambda^{(1g)}_{spc}$).

3. We solve a second pricing subproblem SP_y to find y variables with negative reduced cost. Using again notation $\mathcal{C}_r$ to represent the set of customers whose parcels are carried by a courier in route $r \in \mathcal{R}^{\mathrm{F}}$, the reduced cost of a variable y_r is

$$c_r + \lambda^{(1c)}_{s_r} - \sum_{c \in \mathcal{C}_r} \lambda^{(1j)}_c - \sum_{c \in \mathcal{C}_r} \sum_{\substack{p \in \mathcal{P}_c \\ \text{s.t. } r \in \mathcal{R}^{\mathrm{F}}_{s_r pc}}} \lambda^{(1h)}_{s_r pc}, \tag{3}$$

   where s_r indicates the starting out-stop of route r. The reduced cost contains a fixed term (dual variable $\lambda^{(1c)}$) and dual prizes for each customer whose parcel is transported (dual variables $\lambda^{(1j)}_c$) and for each bus which can deliver the parcel to the out-stop (dual variables $\lambda^{(1h)}_{s_r pc}$).

The algorithm terminates when there are no more negative reduced cost columns to generate.

We remark that subproblems SP_x and SP_y need not be solved in the given order; indeed, in our implementation, we solve SP_x only when SP_y does not produce any negative reduced

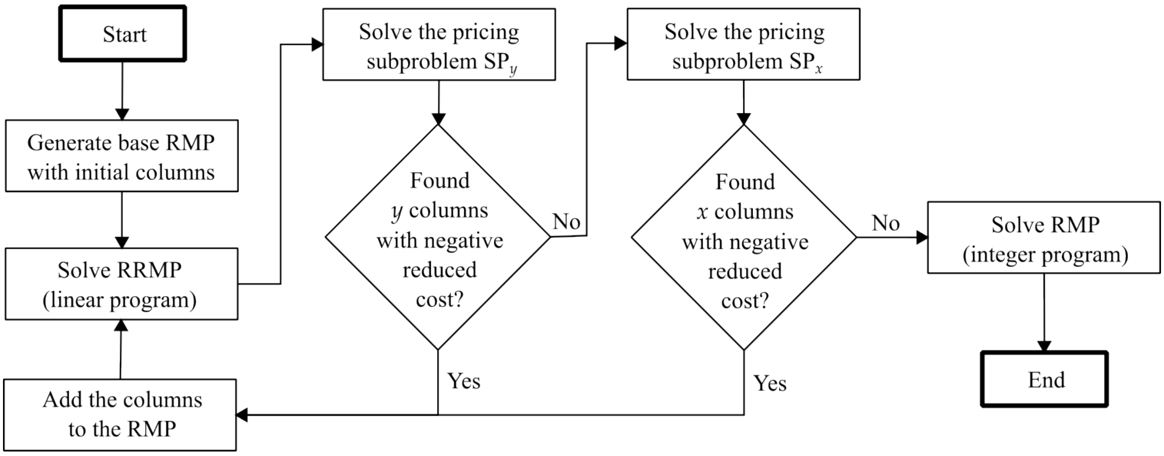

Figure 3: Flowchart illustrating the restricted master heuristic proposed in this work.

cost column. During each pricing step, obtaining one column with a negative reduced cost (regardless if it corresponds to a variable x or y) would be sufficient to improve the objective value of the RRMP. However, because solving SP_y is much faster, we first try to generate as many y columns as possible. When there are no more negative reduced cost columns y, we try solving SP_x. If we find a column by solving SP_x, we add this column to the RMP and we keep iterating the process, i.e., we will solve SP_y first in the next pricing iteration.

Although the proposed method can be used to find the optimal solution of $\mathrm{MP}_{\mathrm{CONT}}$, this task is computationally demanding. Therefore, in Section 6.1, we describe algorithms to get high-quality columns heuristically. Once we generate a sufficient number of columns, or when a time limit hits, we solve RMP with a black-box Mixed-Integer Programming (MIP) solver to obtain a primal solution for MP. The overall approach combining column generation and solving the RMP as a MIP is known in the literature as "price-and-branch" or "restricted master heuristic" (Sadykov et al. 2019). Figure 3 schematically resumes the main steps of the restricted master heuristics. To prove the quality of such a solution, in Section 5.3, we devise dual bounds for MP.

5.1 Column generation for truck routes

In this subsection, we explain how to solve the pricing subproblem SP_x to generate truck routes with negative reduced costs or prove that no such route exists. A truck route $r \in \mathcal{R}^{\mathrm{D}}$ is defined by three elements: (i) a sequence of in-stops, without repetitions; (ii) a set of parcels to deliver to each visited in-stop such that each parcel is delivered once and the total size of the parcels does not exceed the truck capacity; (iii) the route start time from the CDC. Because of these three elements, complete enumeration of truck routes is infeasible in practice even when they only visit a limited number of in-stops.

The route start time is particularly important: because trucks are not allowed to wait at stops, the start time determines the visit time of all in-stops. Moreover, the time at which the truck deposits parcels at a stop is crucial: a truck should not arrive too early or too late. If the truck arrives too early, the parcels will not be able to use some buses, which might be associated with large dual prizes (recall that parcels can stay a maximum of $W^{\max}$ units of time at an in-stop). Analogously, if the truck arrives too late, the parcels will miss some buses with a possibly high dual prize. Contrast this with, e.g., vehicle routing problems with time windows in which arriving late is forbidden or penalised, but vehicles can arrive early and wait at customer locations.

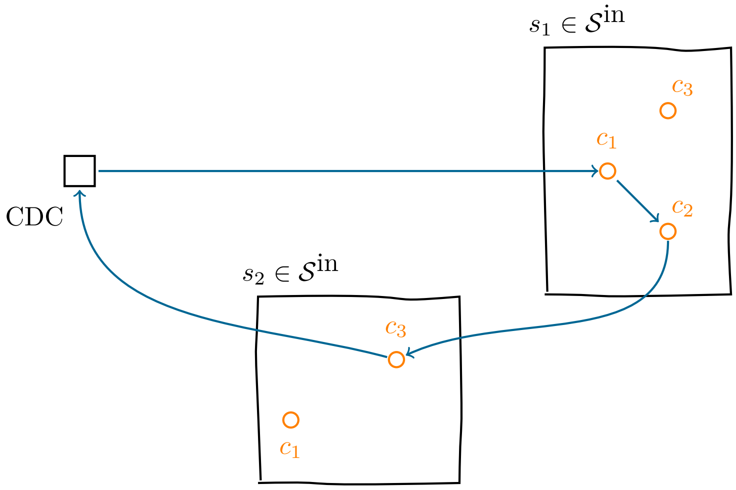

Figure 4: Small example of graph G with two in-stops and three customers. Blue arrows denote a possible truck route.

In what follows, we describe two different approaches to solve SP_x. They use an underlying graph and solve special types of shortest-path problems with a dynamic programming labelling algorithm. The main difference between them is in how they model time.

5.1.1 Labelling algorithm with a time-dependent cost function

We start by describing the directed graph $G = (V, A)$ used by the labelling algorithm. The vertex set is $V = \{o\} \cup \{(s, c) \ : \ c \in \mathcal{C}, \ s \in \mathcal{S}^{\text{in}}_c\}$ and it contains the CDC (o) and the set of all pairs of in-stops and customers (s, c), taking care to add only those pairs which are compatible (see Appendix A for a formal definition of $\mathcal{S}^{\text{in}}_c$). The arc set contains three types of arcs. First, arcs from o to all other vertices and from the other vertices to o. Second, arcs from vertices (s, c) to (s', c') such that $s \neq s'$ and $c \neq c'$. These arcs correspond to the truck moving from in-stop s, where it unloads parcel c, to in-stop s', where it unloads parcel c'. Third, within-stop arcs from vertices (s, c) to vertices (s, c'). These arcs indicate that the truck unloads both parcels c and c' at in-stop s. Because there is no inherent order in which parcels are unloaded, we can drastically reduce the number of arcs by defining an arbitrary ordering of the parcels (say $c_1, \dots, c_{|\mathcal{C}|}$) and only adding arcs from (s, c_i) to (s, c_j) if $j > i$. Figure 4 shows a small example in which rectangles correspond to in-stops and orange circles correspond to vertices of type (s, c). The square vertex is the CDC, while blue arrows represent a possible truck route which delivers parcels for c_1 and c_2 at stop s_1, and for c_3 at stop s_2.

A path in G can only define two of the three elements required to identify a truck route: the sequence of in-stops and the sets of parcels delivered to each in-stop. Therefore, a route $r \in \mathcal{R}^{\text{D}}$ is determined by both a suitable path in G and a start time from the CDC. To include the time dimension, we first associate each arc $a = \big((s, c), (s', c')\big) \in A$ $(s \neq s')$ with a traversing time t_a equal to the travel time from the origin in-stop to the destination in-stop, $l_{ss'}$, plus the service time $T_{s'}$. If origin and destination in-stops coincide $(s = s')$, then the traversing time is $t_a = 0$. The traversing time of arcs arriving at o only includes the travel time $(t_a = l_{so})$. We then define the *benefit* of arriving at a vertex $(s, c) \in V$ at time t as

$$\nu_{sc}(t) = \lambda^{(1\text{i})}_c + \sum_{\substack{p \in \mathcal{P}^{\text{in}}_{sc} \text{ s.t.} \\ t^s_p - W^{\max} \leq t \leq t^s_p}} \lambda^{(1\text{g})}_{cps}.$$

Interval $I_{ps} := \left[t^s_p - W^{\max}, t^s_p\right]$ defines all time instants when a parcel ready for pick-up at stop

s can be loaded onto bus p. Finally, we associate with a path $P = (o, (s_1, c_1), \dots, (s_k, c_k), o)$ of G its reduced cost function

$$C_P(t) = c_r + \lambda^{(1b)} - \sum_{i=1}^{k} \nu_{s_i c_i}(\hat{t}_i(t)), \tag{4}$$

where t is the start time from the CDC, c_r is the cost of the truck route associated with path P and $\hat{t}_i$ is the truck arrival time at in-stop s_i, defined as

$$\hat{t}_i(t) = t + t_{(o,(s_1,c_1))} + \sum_{j=1}^{i-1} t_{(s_j,c_j),(s_{j+1},c_{j+1})}.$$

$C_P(t)$ is a step-wise non-convex function and, as such, needs tailored dominance rules, as explained in the following.

Finding a truck route of negative reduced cost corresponds to finding a path P in G such that its corresponding route is feasible and that $C_P(t) < 0$ for at least one start time t. To accomplish this task, we use a labelling algorithm which associates a label L_P with each partial path P from the CDC to a vertex $(s, c) \in V$. The label has the following components: $v_P = (s, c)$ is the end vertex of the path; $\mathcal{S}_P \subseteq \mathcal{S}^{\text{in}}$ is the set of in-stops which can still be visited when departing from v_P; $\mathcal{C}_P \subseteq \mathcal{C}$ is the set of customers whose parcels can still be delivered when departing from v_P; $\tau_P \geq 0$ is the sum of traversing times of the arcs used in the path; $Q_P \geq 0$ is the spare capacity on the truck when departing from v_P; $C_P : \mathbb{R}_0^+ \to \mathbb{R}$ is the reduced cost function associated with P, obtained by truncating the sum in (4) to the end vertex v_P.

As shown in Section 1 of the supplemental material, to get a valid dominance rule, the cost function associated with each partial path has to be calculated with respect to the arrival time at the end vertex v_P. To this end, we introduce function $\bar{C}_P(t)$ representing the reduced cost of path P when the truck arrives at the end vertex v_P at time t and we define $\bar{C}_P(t) = \infty$ for all $t < \tau_P$. By contrast, $C_p(t)$ is a function of the start time from the CDC. Because the arrival time at v_P is completely determined by the start time of the route, one can always recover the value of $C_P(t)$ from that of $\bar{C}_P(t)$ and vice-versa.

When extending a path P to a new vertex $(s', c') \in V$ along arc $a \in A$, the label associated with the new path P' has the following components:

$$\begin{aligned} &v_{P'} = (s', c'), && \mathcal{S}_{P'} = \mathcal{S}_P \setminus \{s'\}, && \mathcal{C}_{P'} = \mathcal{C}_P \setminus \{c'\}, \\ &\tau_{P'} = \tau_P + t_a, && Q_{P'} = Q_P - q_{c'}, && \bar{C}_{P'}(t) = \nu_{s'c'}(t) + \bar{C}_P(t - t_a) + c_a, \end{aligned}$$

where c_a is the routing cost associated with arc a. Figure 5 shows an example of how cost function $\bar{C}_P$ gets updated to $\bar{C}_{P'}$ using benefit function $\nu_{s'c'}$. In the figure, $\bar{C}_{P'}$ and $\nu_{s'c'}$ are represented as functions of t, i.e., the arrival time at the new end vertex $v_{P'}$; $\bar{C}_P$ is represented as a function of $t - t_a$, i.e., the arrival time at the old end vertex v_P.

The following proposition, proven in supplemental material's Section 1, establishes a dominance relation allowing to discard labels which cannot possibly be extended to an optimal complete path, i.e., to a path with the lowest possible reduced cost.

Proposition 1. *Given two paths P_1 and P_2, and their corresponding labels L_{P_1} and L_{P_2}, ending at the same vertex $v_{P_1} = v_{P_2}$, L_{P_1} dominates L_{P_2} if: $\mathcal{S}_{P_2} \subseteq \mathcal{S}_{P_1}$, $\mathcal{C}_{P_2} \subseteq \mathcal{C}_{P_1}$, $Q_{P_2} \leq Q_{P_1}$, $\bar{C}_{P_1}(t) \leq \bar{C}_{P_2}(t)$ for all $t \geq \tau_{P_2}$, and at least one condition holds strictly. We remark that the condition on $\bar{C}$ implies that $\tau_{P_1} \leq \tau_{P_2}$.*

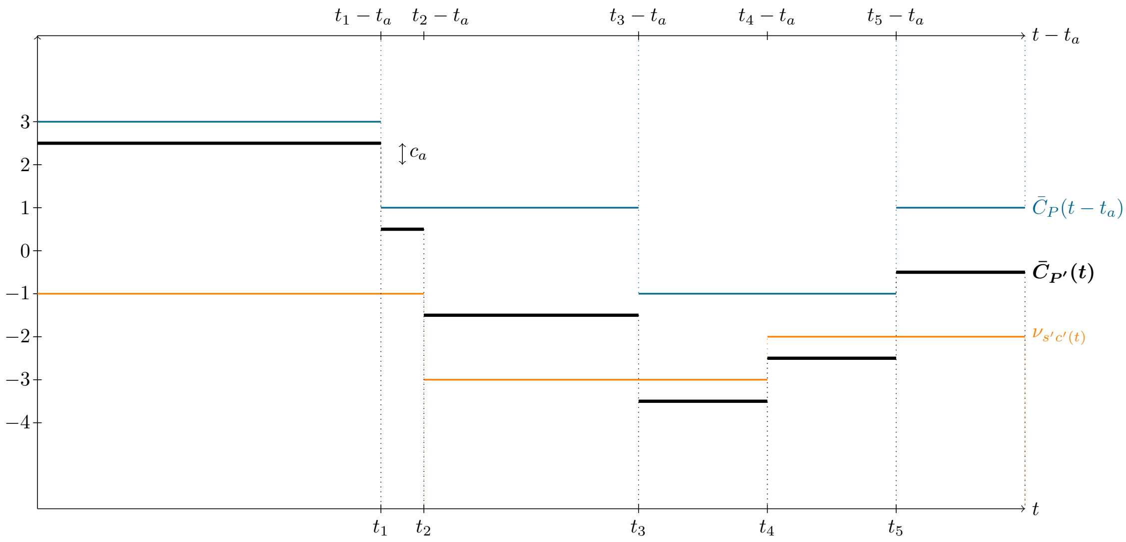

Figure 5: Example of updating cost function $\bar{C}_P$ to $\bar{C}_{P'}$ using the benefit function $\nu_{s'c'}$, during label extension. Time t at the end-point of P' is on the bottom horizontal axis. Time $t - t_a$ at the end-point of P is on the top horizontal axis.

The dominance rule presented in Proposition 1 implies a point-to-point pairwise comparison of the two labels. Indeed, contrary to standard scalar costs, the dominance should be checked for any value of the time-dependent cost function (4). A similar dominance rule was proposed by Tagmouti, Gendreau and Potvin (2007) for a time-dependent arc routing problem. However, the authors do not specify how they implement dominance checking and calculate the cost of a path with respect to the starting time from the depot, which leads to a non-valid dominance rule in our case, as discussed in Section 1 of the supplemental material. More recently, Baum et al. (2020) and Klein and Schiffer (2022) also proposed a labelling algorithm for a subproblem with a time-dependent piece-wise linear convex function. Their pairwise dominance rule is similar to the one proposed in Proposition 1. Moreover, Klein and Schiffer (2022) use a set-based dominance rule, where a label can be jointly dominated by a set of labels. Preliminary tests showed that set-based dominance did not improve the performance of our algorithm.

5.1.2 Labelling algorithm with a scalar cost

The dual prizes collected at the vertices of G only change, as a function of the arrival time at the in-stop, at the intersection of the time intervals I_{sp} defined in Section 5.1.1. In other words, the dual prizes change at time instants when a bus becomes available or is no longer available to pick up parcels. Therefore, for each in-stop, it is possible to partition the time horizon into intervals where the dual prizes are constant.

Consider, for example, the instance presented in Figure 4. Assume that buses p_1, p_2, p_3 serve in-stop s_1, and buses p_4, p_5 serve in-stop s_2. A possible time horizon partitioning for both in-stops is depicted in Figure 6; the left part of the figure refers to s_1 and the right one to s_2. The horizontal lines in the top part of the picture represent time intervals I_{sp}. The line at the bottom is the time horizon, partitioned into intervals $w_{s\ell}$, where $s \in \mathcal{S}^{\text{in}}$ is the in-stop and ℓ is the index of the interval ($\ell \in \{1, \ldots, \Lambda_s\}$).

As a consequence of the above observation, we propose an alternative approach to solving SP_x as a shortest path problem where, contrary to the problem described in Section 5.1.1, the cost is no longer a time-dependent function but a scalar. We create a directed graph $H = (W, B)$

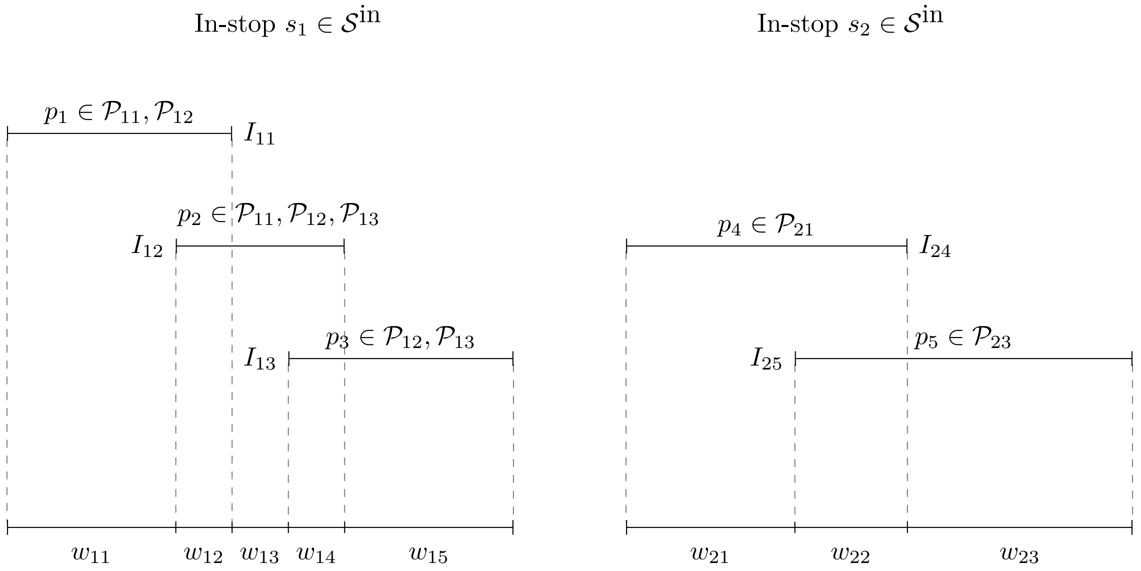

Figure 6: Time intervals $w_{s\ell}$ for the instance presented in Figure 4. Buses p_1, p_2, p_4 are compatible with customer c_1; p_1, p_2, p_3 with customer c_2; and p_2, p_3, p_5 with customer c_3.

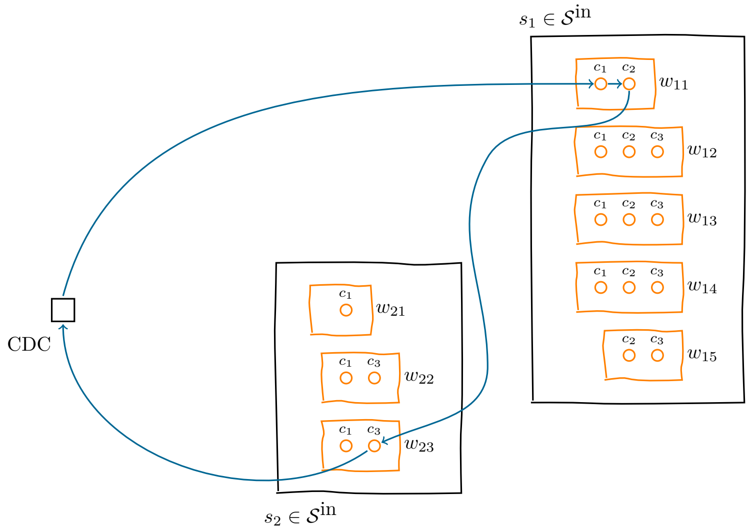

Figure 7: Small example of graph H with two in-stops and three customers. Buses p_1, p_2, p_3 serve in-stop s_1; buses p_4, p_5 serve in-stop s_2. Time intervals w are as in Figure 6. Blue arrows denote a possible truck route.

with vertex set:

$$W = \{o\} \cup \left\{(s, c, \ell) \;:\; c \in \mathcal{C},\; s \in \mathcal{S}_c^{\text{in}},\; \ell \in \{1, \dots, \Lambda_s\},\; \exists p \in \mathcal{P}_{cs} \text{ s.t. } I_{sp} \cap w_{s\ell} \neq \emptyset\right\}.$$

Each vertex (s, c) of V is copied multiple times in W: it gets one copy for each time interval $w_{s\ell}$ during which at least one bus can pick up the parcel of c at s. The arc set B contains three types of arcs. First, arcs from o to all other vertices, and from the other vertices to o. Second, arcs from vertices (s, c, ℓ) to vertices (s', c', ℓ') such that $s \neq s'$ and $c \neq c'$. Third, within-interval arcs from vertices (s, c, ℓ) to vertices (s, c', ℓ), indicating that the truck delivers both parcels for c and c' at s during interval ℓ. Similar to what was noted in Section 5.1.1, the number of such arcs can be drastically reduced through an arbitrary ordering of the customers. Figure 7 shows the same instance and route of Figure 4, but on graph H with time intervals defined as in Figure 6. The route delivers c_1's and c_2's parcels at in-stop s_1 during time interval w_{11}, and c_3's parcel at in-stop s_2 during time interval w_{23}. Therefore, the only possible continuation of the parcels' journeys is on bus p_1 for c_1 and c_2 and on bus p_5 for c_3.

The labelling algorithm used to find truck routes of negative reduced cost (or prove that none exists) on graph H is similar to classical algorithms to solve the resource-constrained shortest path problem. We give a detailed description of it in Appendix B.1. Here, we only note that using this approach, we no longer need to associate a cost function with each label. Because the vertices already encode the information about buses compatible with a given route, we can use scalar costs. The trade-off is that graph H is much larger than graph G and each vertex (s, c, ℓ) has an associated time window $w_{s\ell}$ during which the vertex can be visited. We will introduce speed-up techniques to overcome the challenges of solving SP_x both on G and H in Section 6.

5.2 Column generation for courier routes

We now explain how to solve the pricing subproblem SP_y. We first remark that SP_y is decomposable by out-stop. The lowest reduced cost of any courier route can be written as

$$\min_{s \in \mathcal{S}^{\text{out}}} \min_{r \in \mathcal{R}_s^{\text{F}}} \left\{ c_r + \lambda_s^{(1c)} - \sum_{c \in \mathcal{C}_r} \lambda_c^{(1j)} - \sum_{c \in \mathcal{C}_r} \sum_{\substack{p \in \mathcal{P}_c \\ \text{s.t. } r \in \mathcal{R}_{spc}^{\text{F}}}} \lambda_{spc}^{(1h)} \right\},$$

and all inner minimisation problems are independent of each other.

For a given out-stop $s \in \mathcal{S}^{\text{out}}$ and customer $c \in \mathcal{C}_s$, the dual prizes $\lambda_{spc}^{(1h)}$ which a courier route can collect only depend on the route start time from s because this time determines which buses can deliver the customer's parcel. Using the same key ideas from Section 5.1.2, we can partition the time horizon into intervals in which these dual prizes stay constant. Keeping the same notation, we denote these intervals with $w_{s\ell}$ ($w \in \{1, \dots, \Lambda_s\}$). We then decompose SP_y both by out-stop and by time interval, and we let $\text{SP}_y(s, \ell)$ be the subproblem associated with $s \in \mathcal{S}^{\text{out}}$ and $\ell \in \{1, \dots, \Lambda_s\}$, in which we use $w_{s\ell}$ as the out-stop time window limiting the possible start times of the courier routes. Considering customer time windows, the capacity of the courier, and that each customer must be visited at most once, we model $\text{SP}_y(s, \ell)$ as an elementary shortest-path problem with time windows. Appendix B.2 presents a labelling algorithm to solve $\text{SP}_y(s, \ell)$.

5.3 Dual bounds

A straightforward way to obtain a dual bound for MP is to solve MP_{CONT} to optimality, i.e., until there are no more negative reduced cost columns. In practice, this approach only works for small instances. For medium-size instances, the column generation procedures do not usually terminate within the time limit of one hour that we impose in computational tests. For large instances, even solving one iteration of the column generation algorithm may take more than one hour; in particular, solving a single pricing problem SP_x is computationally prohibitive. Pricing problem SP_y, on the other hand, is usually solved in less than a second. This is mainly due to the fact that we solve an independent subproblem $\mathrm{SP}_y(s, \ell)$ for each out-stop s and each time interval ℓ. Hence, the size of these subproblems remains tractable even for large instances. Additionally, courier capacity is usually limited, preventing sets $\mathcal{R}^{\mathrm{F}}$ from growing large.

We can speed up the solution of the pricing problems by approaching them heuristically (see Section 6.1). In this case, however, we can no longer guarantee that all remaining columns have a non-negative reduced cost and, thus, that we have solved $\mathrm{MP}_{\mathrm{CONT}}$. Therefore, in the following, we introduce an alternative dual bound for the 3T-DPPT, which can be computed off-line when it is intractable to solve SP_x or SP_y. In Section 3 of the supplemental material, we also describe a way to obtain an on-line Lagrangian bound while running the restricted master heuristic.

We obtain the off-line bound considering the relaxation of the 3T-DPPT in which we ignore the second echelon and solve the first- and third-echelon problems separately. In particular, we no longer specify which buses transport parcels from in-stop to out-stop but rather devise truck and courier routes independently. The resulting solution is likely unfeasible: for example, it can involve a truck delivering a parcel at in-stop s_1 and a courier delivering the same parcel starting from out-stop s_2, even when there is no bus connecting s_1 and s_2. Furthermore, we relax the in-stop elementarity and no-wait constraints from the first-echelon problem: we allow trucks to visit the same stop multiple times and to wait at a stop before delivering parcels. In the following, we describe the proposed solution approach for the first- and third-echelon problems.

With the above relaxations, the first-echelon problem becomes a variant of the Generalised Vehicle Routing Problem with Time Windows (GVRPTW). The GVRP is an extension of the classical Vehicle Routing Problem in which the customers are partitioned into clusters. Vehicles must only visit one customer in each cluster, and, in doing so, they deliver the entire cluster demand. In the GVRPTW, additionally, each customer can be visited only within a given time window.

Consider the graph G introduced in Section 5.1.1, in which each vertex other than the CDC corresponds to a pair of in-stop and customer. We partition these vertices grouping together those that correspond to the same customer. Because a customer's parcel must be delivered at most once, it is sufficient to visit only one vertex in each cluster; this vertex determines the in-stop used to deliver the parcel. We obtain time windows for each vertex (s, c) considering the delivery time window for customer c, the possible out-stops which can handle c's parcel, and the possible buses which can carry the parcel from s to these out-stops. The corresponding time window Θ_{sc} is defined in Appendix A. In a GVRPTW solution, the same vehicle can visit two vertices referring to the same in-stop at any point in its route. Thus, we cannot guarantee the elementarity of the in-stops. Section 3 of the supplemental material details the solution approach used to solve the GVRPTW.

The third-echelon problem, on the other hand, is a Multi-Depot Vehicle Routing Problem with Time Windows (MDVRPTW). The capacity constraints of the couriers considerably limit the number of customers served by the same route. Therefore, we solve the MDVRPTW using a straightforward extended formulation enumerating all feasible routes.

6 Other elements of the column generation algorithm

In this section, we describe the auxiliary components of the column generation algorithm. We will focus on subproblem SP_x because it is, in practice, much harder than SP_y. Section 6.1 describes how to speed up the solution of SP_x using graph sparsification heuristics. We obtain a further speed-up by considering the order in which we extend and dominate labels, as explained in Section 6.2. Finally, in Section 6.3, we explain how to post-process truck routes to exploit eventual residual capacity. Section 4 of the supplemental material describes the greedy algorithm we use to generate an initial set of columns.

6.1 Heuristic column generation

We speed up the generation of columns in subproblem SP_x, reducing the number of arcs in graphs G and H, respectively introduced in Sections 5.1.1 and 5.1.2. We start by describing the techniques used on graph G. Recall that, for a given in-stop s, we only generate the within-stop arc from (s, c_i) to (s, c_j) if $i < j$ in an arbitrary ordering of the customers $(c_1, \dots, c_{|\mathcal{C}|})$. In our heuristic arc reduction, we sort the customers within the same in-stop by increasing (negative) maximum reduced cost $\mathrm{mrc}_s(c)$, given by

$$\mathrm{mrc}_s(c) = \max_t \left\{\nu_{sc}(t)\right\} = \max_t \left\{ \lambda_c^{(1\mathrm{i})} + \sum_{\substack{p \in \mathcal{P}_{sc}^{\mathrm{in}} \text{ s.t.} \\ t_p^s - W^{\max} \le t \le t_p^s}} \lambda_{cps}^{(1\mathrm{g})} \right\}.$$

Then, for each node (s, c), we only keep the within-stop arc to (s, c'), where c' is the customer following c in the given order, over those customers not yet visited by the route being constructed. In practice, this technique forces a truck at an in-stop to deliver the parcel of the k-th best customer by reduced cost only if it also delivers the parcels of the best, second-best, …, $(k-1)^{\text{th}}$-best customers for the stop, either at the current or at previous stops. We denote this technique as PATH heuristic.

We use an analogous procedure to reduce the number of arcs in the graph H. Given an in-stop s and a time interval ℓ, we sort customers by increasing (negative) reduced cost, expressed in this case as

$$\lambda_{c'}^{(1\mathrm{i})} + \sum_{\substack{p \in \mathcal{P}_{s'c'} \text{ s.t.} \\ I_{s'p} \cap w_{s'\ell'} \neq \emptyset}} \lambda_{s'pc'}^{(1\mathrm{g})}.$$

We only generate arcs from (s, c, ℓ) to the (s, c', ℓ), with c' immediately following c in the given order.

6.2 Using checkpoints in truck columns labelling

Graph G of Section 5.1.1 contains $\mathcal{O}(|\mathcal{S}^{\mathrm{in}}|^2 |\mathcal{C}|^2)$ "travelling" arcs of type $\big((s, c), (s', c')\big)$ because each pair (s, c) of in-stop and customer is connected to each other pair (s', c') (when $s \neq s'$ and $c < c'$). In this section, we explain how to reduce these arcs to $\mathcal{O}(|\mathcal{S}^{\mathrm{in}}|^2 + |\mathcal{S}^{\mathrm{in}}| \cdot |\mathcal{C}|)$, at the expense of adding $\mathcal{O}(|\mathcal{S}^{\mathrm{in}}|)$ vertices. Because in practical applications $|\mathcal{S}^{\mathrm{in}}| \ll |\mathcal{C}|$, this considerably reduces the number of arcs in G and speeds up the labelling algorithm to solve SP_x. The main idea is to add to each in-stop s two "checkpoint" vertices denoted v_s^- (entry point) and v_s^+ (exit point). We remove from the graph the travelling arcs described above, and we replace them with: (i) $\mathcal{O}(|\mathcal{S}^{\mathrm{in}}|^2)$ new travelling arcs of type $(v_s^+, v_{s'}^-)$ for any pair of in-stops $s \neq s'$; (ii) $\mathcal{O}(|\mathcal{S}^{\mathrm{in}}| \cdot |\mathcal{C}|)$ entry-to-customer arcs from v_s^- to (s, c) for all $c \in \mathcal{C}_s$; (iii) $\mathcal{O}(|\mathcal{S}^{\mathrm{in}}| \cdot |\mathcal{C}|)$ customer-to-exit arcs from (s, c) to v_s^+ for all $c \in \mathcal{C}_s$. Figure 8 (left) shows the same instance

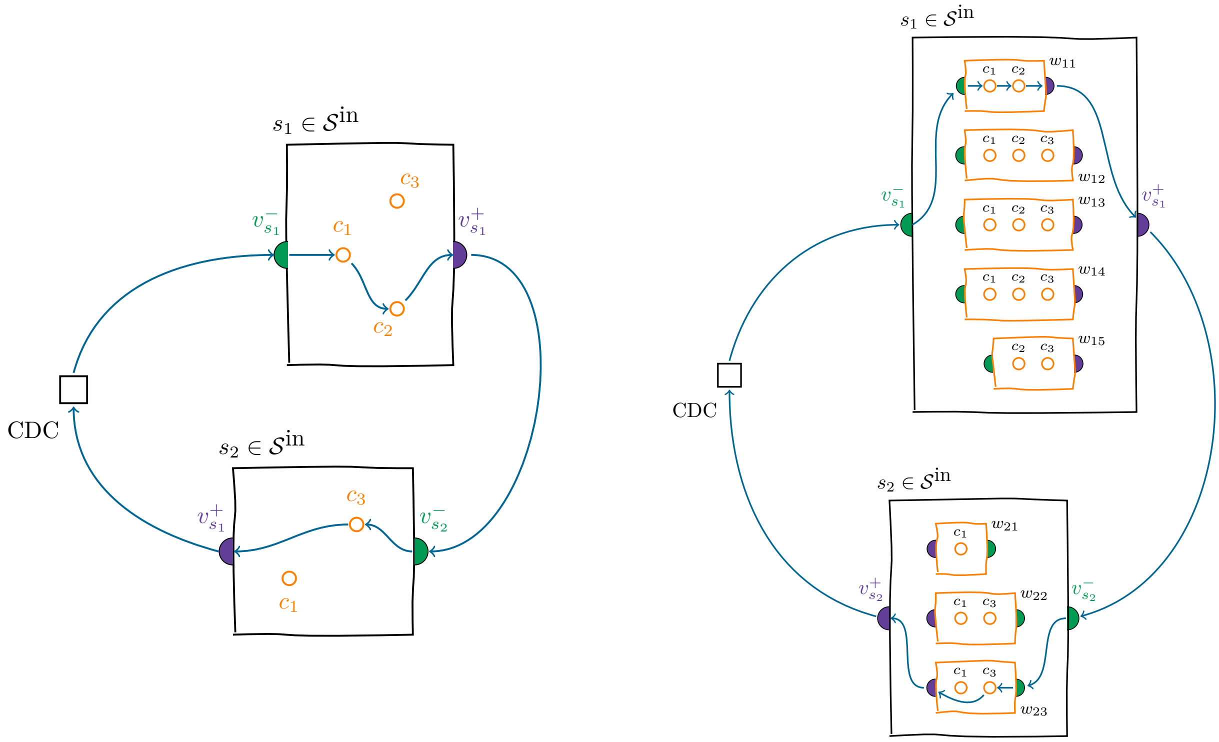

Figure 8: Introduction of checkpoints in graph G of Section 5.1.1 (left) and in graph H of Section 5.1.2 (right).

and route as Figure 4 after the introduction of checkpoints. The same idea can be applied to graph H introduced in Section 5.1.2, as depicted in Figure 8 (right). For this graph, in addition to checkpoints for each stop, we add analogous checkpoints for each interval $w_{s\ell}$ inside the stop.

6.3 Post-processing of columns

We post-process the truck routes generated when solving subproblem SP_x to exploit eventual spare capacity. In the following, we give a brief description of the post-processing algorithm for a route $r \in \mathcal{R}^{\text{D}}$ with its associated start time. The algorithm consists of two phases.

First, we order the set of customers not covered by r in ascending order of ratio $\lambda_c^{(1\text{i})}/q_c$. Then, we go through the in-stops visited along route r, and we deliver at each of them as many additional parcels as possible in the given order. The truck can deliver a parcel at a stop if there is at least one bus compatible with that parcel, if there is enough capacity on the truck, and if the parcel is not already included in r. Performing this post-processing step does not increase the route's cost but makes it cover more customers.

In the second phase, we use a procedure which might increase the route's cost. We first compute, for each in-stop s not visited by r, the best dual prize we could collect if we visit s and deliver parcels using the same criterion as in the first phase. Because the best dual prize depends on the position in which we insert s in the route, we consider all possible positions. For each stop s, giving dual prizes larger (in absolute value) than the detour cost, we create a copy of r in which we insert stop s. We repeat this procedure until no stop can be inserted.

7 Computational experiments

In this section, we present the results of a wide range of computational experiments. First, in Section 7.1, we assess the trade-offs associated with the different subproblem solution methods we presented. Second, in Section 7.2, we compare three possible formulations for the 3T-DPPT. Finally, in Section 7.3, we compare our heuristic based on column generation with the only other comparable approach from the literature, i.e., the decomposition heuristics of Mandal and Archetti (2023).

We ran our experiments on a cluster with Intel Xeon CPUs running at 2.4GHz. Each run was limited to using one core and 32GB of memory. The RRMP solver was Cplex version 20.1, used through its `C++` API with default parameters. We generated the instance set following the same procedure as Mandal and Archetti (2023), which uses the instance generator developed by Delle Donne, Alfandari et al. (2023). The dataset contains ten instances for each number of customers in $\{20, 25, 30, 40, 50\}$, for a total of 50 instances. Each instance contains between 21 and 32 stops and between 45 and 60 buses. All instances and results are available in a GitHub repository (Delle Donne, Santini and Archetti 2024).

Our default time limit is one hour for all algorithms, as this is an appropriate choice for an operational problem with a planning horizon of one day or less. When solving the extended and semi-compact formulations, we reserve a minimum time of 5 minutes for the final step which solves the RMP with the generated columns. Therefore, if the column generation phase is not over after 55 minutes, we stop it and start this final MIP. We use the one-hour time limit in all experiments described in Section 7.1 and Section 7.2. In Section 7.3, we compare our approach with the decomposition heuristic of Mandal and Archetti (2023) that uses a three-hour time limit. To make a fair comparison, we use the same time limit and reserve a minimum time of one hour for the final MIP.

7.1 Ablation study

In this section, we present the results of two ablation studies in which we disable each of the enhancements proposed in Section 6. We show that disabling any of these enhancements degrades the algorithm's performances, thus justifying their use. The pricing method used in the first study is the labelling algorithm with a time-dependent cost function introduced in Section 5.1.1 and denoted CostFunction. In the second study, we use the labelling algorithm with a scalar cost that we introduced in Section 5.1.2; we denote it ScalarCost.

We consider six configurations in each study. In four of them, we selectively disable one of the following enhancements:

1. The heuristic pricing strategy presented in Section 6.1.
2. The subproblem graph transformations using checkpoints, presented in Section 6.2.
3. The post-processing phase introduced in Section 6.3.
4. The first-echelon column pool initialisation strategy described in Section 4 of the supplemental material. This strategy uses a greedy algorithm to generate a set of initial first-echelon columns. When it is disabled, we instead initialise the column pool with dummy columns as explained in Section 4 of the supplemental material.

In the other two configurations we respectively enable and disable all the enhancements. Table 2 shows the ablation study's results. Each cell shows the average result over the set of 50 instances.

Columns "Count" report the number of instances for which the method obtained a feasible

	CostFunction			ScalarCost		
Configuration	Count	Gap	Time	Count	Gap	Time
All enhancements	49	20.0%	2756	49	19.7%	586
No heuristic pricing	48	20.9%	3565	48	21.2%	3564
No checkpoints	48	21.5%	3571	48	20.7%	130
No postprocessing	47	20.5%	2995	49	20.0%	667
No initial heuristic	44	20.2%	2487	44	19.7%	475
No enhancements	40	22.4%	3559	40	22.0%	3558

Table 2: Summary of the ablation study results.

solution. Columns "Gap" list the average gap between the feasible solution and the best known lower bound for the instance. The average gap is calculated over the subset of instances for which every configuration has found a feasible solution. For CostFunction, these are the 40 instances for which "No enhancements" finds a feasible solution. For ScalarCost, although each configuration finds a feasible solution for at least 40 instances, there are only 39 instances for which all configurations find a feasible solution. In particular, "No enhancement" solves an instance that the other configurations do not. Therefore, the ScalarCost gaps are computed on 39 instances. This allows a fair comparison among the different configurations, as in the cases in which no solution is found, no appropriate gap can be considered. Columns "Time" report the average computing time in seconds.

Table 2 shows that enabling all enhancements consistently produces the best results in terms of number of feasible solutions found and average gaps. The configuration in which all enhancements are enabled almost always finds a strictly larger number of feasible solutions compared to the other configurations. The only exception occurs for ScalarCost, in which the "All enhancements" and "No postprocessing" configurations are tied in terms of solutions found but, still, the "All enhancement" configurations gives a lower gap and a shorter runtime. A remarkable difference on the results can be observed when none of the enhancements is enabled. In both studies, the number of feasible solutions found decreases from 49 to 40, the average gaps increase, and the runtimes are considerably longer.

Selectively disabling some of the enhancements results in shorter runtimes. The most striking example is in the ScalarCost study, where disabling checkpoints reduces the average runtimes from 586 to 130 seconds. However, the number of feasible solutions found decreases and the average gap increases. Therefore, because our main objective is to provide feasible solutions of good quality, we decided to keep all enhancements enabled in the computational experiments presented in the following sections.

7.2 Formulation comparison

In this section we compare the three proposed formulations for the 3T-DPPT, namely:

1. The extended formulation with two exponential sets of variables presented in Section 4. We test two different algorithms for this formulation: the one using the time-dependent cost-function (Section 5.1.1) and the one using scalar costs (Section 5.1.2). As we did in Section 7.1, we denote these two algorithms CostFunction and ScalarCost, respectively.
2. The semi-compact formulation with one exponential set of variables described in Appendix C and denoted with "SCF".

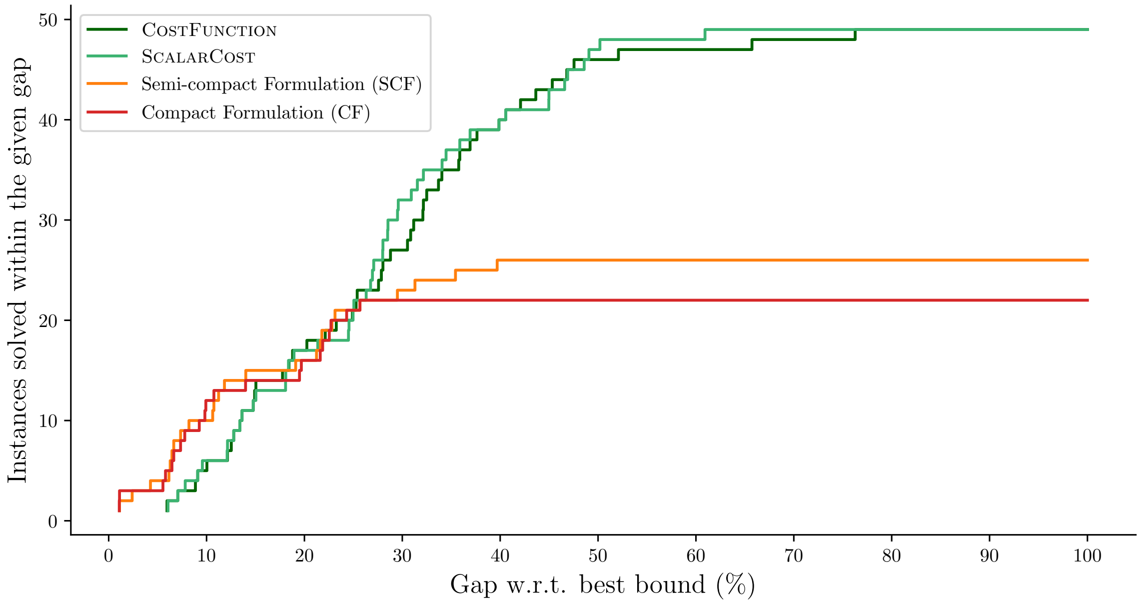

Figure 9: Performance profiles of the four tested algorithms.

3. A compact formulation initially introduced by Mandal and Archetti (2023). In our experiments, we amended this formulation to forbid trucks from waiting at stops indefinitely. The resulting model is presented in Section 6 of the supplemental material. This formulation is denoted “CF”.

Figure 9 presents the performance profiles of the four algorithms CostFunction, ScalarCost, SCF, and CF. The x-axis reports the gap with respect to the best dual bound, i.e., the tightest among all the dual bounds produced during the experimental campaign. These bounds are obtained by the following techniques: (i) running the column generation algorithm solving the pricing subproblem for the x variables using the MIP model described in Section 2 of the supplemental material (and the subproblem for the y variables to optimality) and applying the Lagrangian bound described in Section 3 of the supplemental material; (ii) computing the decomposition bound described in Section 5.3; (iii) solving the semi-compact or the compact formulation and taking the best dual bound reported when the solver terminates. For any given gap on the x-axis, the y-axis reports the number of instances solved within the gap. Each curve corresponds to one of the four algorithms (dark green for CostFunction, light green for ScalarCost, orange for SCF, and red for CF). The curves for SCF and CF do not reach the top-right corner of the figure because these algorithms find feasible solutions for 26 and 22 instances only. On the other hand, CostFunction and ScalarCost produce a feasible solution for all instances but one. On the bottom-left part of the chart, the curves relative to SCF and CF dominate those relative to CostFunction and ScalarCost. Indeed, for small instances, SCF and CF find optimal or almost optimal solutions. For larger instances, however, CostFunction and ScalarCost are mostly the only two methods yielding primal solutions at all. We finally note that, although no column generation method dominates the other completely, ScalarCost is often slightly better than CostFunction.

Figure 10 provides disaggregated results. The number of customers is reported on the x-axis, with the instances first sorted by the number of customers and then by the gap provided by the best method; the y-axis shows the percentage gap. The numbers above the chart count how many methods produced a feasible solution. This same information is also conveyed visually: a darker background corresponds to an instance for which fewer algorithms produce a feasible solution. Finally, Table 3 summarises the number of instances solved by each method. We do

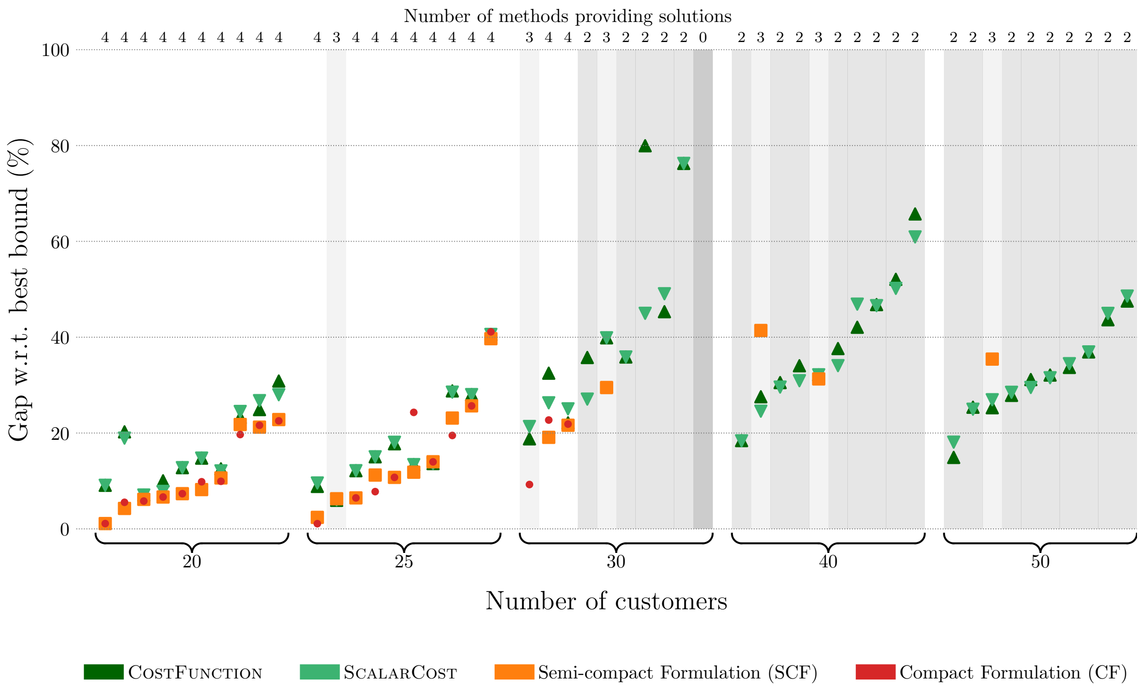

Figure 10: Disaggregated results for the four tested algorithms.

	# of customers					
Method	20	25	30	40	50	Total
CF	10	9	3	0	0	22
SCF	10	10	3	2	1	26
CostFunction	10	10	9	10	10	49
ScalarCost	10	10	9	10	10	49

Table 3: Number of instances in which each method finds a feasible solution. The instances are grouped by number of customers.

not report average solution times of SCF and CF because, except for a few exceptions (when the gap is zero in Figure 9), all other instances reach the 1-hour time limit. We recall that the average solution times of CostFunction and ScalarCost can be found in Table 2.

Overall, we highlight the fact that, despite the difficulty of the problem, column generation methods that use two exponential sets of variables can find feasible solutions for most instances. By contrast, the CF and SCF fail to do so for most instances with 30 or more customers. In particular, SCF finds a feasible solution for only three instances with 40 or more customers and, in two of these three cases, with much larger gaps than CostFunction and ScalarCost.

7.3 Comparison with a heuristic from the literature

In this section, we compare the performance of our column-generation-based methods against the only other comparable approaches in the literature, i.e., the compact formulation (CF) and the decomposition heuristic (DH) of Mandal and Archetti (2023). In their paper, the authors devise several heuristics, and their computational experiments show that none dominates the others. Therefore, we compare the best solution obtained by any of the heuristics in (Mandal and Archetti 2023) with the best solution obtained by our proposed methods. This detailed comparison is presented in Section 5 of the supplemental material.

As mentioned at the beginning of Section 7, in these experiments, we use the same time limit as Mandal and Archetti (2023), i.e., three hours. For the SCF, CostFunction, and ScalarCost methods, we set a maximum time of two hours for the column generation phase and, therefore, a minimum of one hour for the final MIP. We remark that the ratio of these time limits (two hours and one hour) is not the same as the one used in Section 7.1 (55 minutes and five minutes). However, our column generation methods generate a significant number of high-quality columns already during the first two hours. In preliminary experiments, we noticed that extending the column generation timeout gives diminishing returns in terms of column quality and increases the size of the final MIP. This increased size and the reduced MIP timeout often result in sub-optimal solutions for the final MIP and degrade the overall performance.

We recall that Mandal and Archetti (2023) allow trucks to wait at in-stops, and we do not. Therefore, the problems being solved are slightly different, and the solution space of Mandal and Archetti's formulation is larger than ours. This implies that the problem solved by the decomposition heuristics may admit optimal solutions with lower objective values than in our formulation. Both Mandal and Archetti's instances and the instances we used in Section 7.1 and Section 7.2 were produced using the same generator and are structurally similar. Indeed, the only differences are in whether trucks are allowed to wait at in-stops and in the size of the largest instances. Our instances contain up to 50 customers, while Mandal and Archetti use instances with up to 80 customers.

Table 4 (left) reports the results obtained on the 24 instances tested in Mandal and Archetti (2023). Column "Instance" reports the instance number. Column "CF" lists the percentage objective cost improvement of the best solution found by our column-generation-based methods compared with the compact formulation. Analogously, column "DH" reports the percentage improvement compared with the best of Mandal and Archetti's decomposition heuristics. An empty cell indicates that the CF or DH methods could not find a feasible solution for the instance. The summary table on the right of Table 4 reports the number of instances in which the best solution found by one of our column-generation-based methods has better, worse or the same cost as CF, DH and the best solution between CF and DH.

Compared with DH, our algorithm found better solutions in 21 out of the 24 instances. The average improvement in these 21 instances is 8.38%, reaching more than 16% in the best case. We tie on one instance (number 1), for which our method finds an optimal solution; optimality is proven because CF terminates within the time limit and finds the same solution. We find a worse solution in only two instances (13 and 17). However, by inspecting instance 13, we verified that DH's best solution includes a route in which the truck waits at an in-stop for a large part of the planning horizon. Therefore, DH's solution is infeasible for our formulation. Our solutions are never worse than those produced by CF. We get the same result on six instances and improve the solution for the other 18, including the 13 instances for which CF fails to find a feasible solution.

To summarise, when we compare our results against the best result between CF and DH for each instance (last column of the right table in Table 4), we improve in 16 out of 24 instances, tie in six and find worse solutions in two cases. In one of these (instance 13), DH's solution is considered infeasible to our methods and in the other (instance 17) the cost difference is less than 1%.

	Improvement%			Improvement%	
Inst	CF	DH	Inst	CF	DH
1	0.00	0.00	13	—	-5.39
2	0.00	5.43	14	—	6.37
3	0.00	8.37	15	—	13.96
4	0.00	5.13	16	—	9.04
5	3.94	8.68	17	—	-0.52
6	0.00	12.36	18	—	3.86
7	7.61	11.19	19	—	7.32
8	0.00	8.54	20	—	7.49
9	6.00	10.03	21	—	10.96
10	—	5.85	22	—	16.57
11	5.68	11.27	23	—	6.06
12	10.41	4.25	24	—	4.05

	Comparison vs.		
	CF	DH	Best
# Better cost:	18	21	16
# Worse cost:	0	2	2
# Same cost:	6	1	6

Table 4: Left: relative improvements of the solutions obtained by our methods on the 24 instances from Mandal and Archetti 2023, compared with CF and DH. Right: summary of the number of instances in which our methods find a solution with a better, worse, or with the same cost as CF, DH and the best between the two.

8 Conclusions

In this paper, we develop a decision-support tool for a last-mile delivery system that uses spare capacity on public transport to bring parcels from an outside distribution centre into the city. In particular, we proposed extended formulations for the Three-Tier Delivery Problem using Public Transportation (3T-DPPT), a problem already studied by Mandal and Archetti (2023). Two characteristics set the main extended formulation apart from most other models from the literature: first, the presence of two exponential sets of variables giving rise to two pricing subproblems; second, that one of these subproblems involves finding the shortest path on a graph with time-dependent costs. The complete price-and-branch algorithm involves several components. Apart from the algorithms to solve the pricing problems to optimality, we implemented an initial column generation heuristic, a column postprocessing algorithm, a sparsification technique to accelerate pricing, and dual bounding approaches based on problem decomposition and on the Lagrangian bound. Computational tests on benchmark instances show that column generation applied to the extended formulations is a viable approach to obtain good primal solutions in less than one hour of computing time.

While our method produces high-quality primal solutions, more work is needed to devise better dual bounds. In small instances, we observed that the continuous relaxation of the extended formulation provides tight bounds. In larger instances, however, solving the continuous relaxation to optimality is prohibitive because of the high computation time required by exact pricing. The other bounding techniques we propose (based on decomposition and the Lagrangian bound) do not provide equally good dual bounds.

The problem tackled in this work resembles classical 3-echelon vehicle routing problems. However, it has the particular characteristic that the transportation means used in the second echelon have fixed routes and schedules which cannot be changed. The problem is also connected with vehicle routing problems with *synchronisation*, in which the routes must meet at some common point in time and/or space. Following these connections with classical problems, adapting or generalising our method to these problems is an interesting future research direction.

Acknowledgements

We are extremely grateful to the authors of (Pessoa et al. 2023) for sharing with us their VrpSolver implementation of the GVRP. This work was partially funded by the CY Initiative of Excellence (grant "Investissements d'Avenir" ANR-16-IDEX-0008). The work of Alberto Santini has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska Curie grant agreement number 945380. We thank three anonymous reviewers whose comments helped improving a former version of the paper.

References

Alfonso, Viviana et al. (Jan. 2021). *E-commerce in the pandemic and beyond*. Bulletin 36. Bank for International Settlements. URL: `https://web.archive.org/web/20210126163639/https://www.bis.org/publ/bisbull36.htm` (visited on 26/01/2021).

Antkowiak, Tifany (Jan. 2018). 'Le projet de tram fret est arrêté à Saint-Étienne'. In: *France Bleu*. URL: `https://web.archive.org/web/20180217163752/https://www.francebleu.fr/infos/transports/le-projet-de-tram-fret-est-arrete-a-saint-etienne-1514458535` (visited on 17/02/2018).

Archetti, Claudia and Luca Bertazzi (2021). 'Recent challenges in Routing and Inventory Routing: E-commerce and last-mile delivery'. In: *Networks* 77.2, pp. 255–268. DOI: `10.1002/net.21995`.

Baron, Ethan (Feb. 2019). 'Amazon looks to turn public buses into mobile delivery stations'. In: *Star Tribune*. URL: `https://web.archive.org/web/20190811055124/http://www.startribune.com/amazon-looks-to-turn-public-buses-into-mobile-delivery-stations/505136612/` (visited on 11/08/2019).

Baum, Moritz et al. (2020). 'Modeling and engineering constrained shortest path algorithms for battery electric vehicles'. In: *Transportation Science* 54.6, pp. 1571–1600.

Beasley, John Edward and Nicos Christofides (1989). 'An algorithm for the resource constrained shortest path problem'. In: *Networks* 19 (4), pp. 379–394. DOI: `10.1002/net.3230190402`.

Beckers, Joris et al. (2021). 'COVID-19 and retail: The catalyst for e-commerce in Belgium?' In: *Journal of Retailing and Consumer Services* 62. DOI: `10.1016/j.jretconser.2021.102645`.

Bhatt, Kushal Mukesh (2019). *Mobile Pickup Locations*. US Patent n. US10192189B2, holder: Amazon Technologies, Inc.

Boysen, Nils, Stefan Fedtke and Stefan Schwerdfeger (2021). 'Last-mile delivery concepts: a survey from an operational research perspective'. In: *OR Spectrum* 43, pp. 1–58. DOI: `10.1007/s00291-020-00607-8`.

Bruzzone, Francesco, Federico Cavallaro and Silvio Nocera (2021). 'The integration of passenger and freight transport for first-last mile operations'. In: *Transport Policy* 100, pp. 31–48. DOI: `10.1016/j.tranpol.2020.10.009`.

Caspersen, Elise (2021). 'Freight trip generation and consumer preferences for reducing externalities from last mile deliveries'. PhD Dissertation. Norwegian University of Life Sciences.

Clinnick, Richard (Apr. 2020). 'GB Railfreight uses electric commuter unit for London freight trial'. In: *Rail Magazine*. URL: `https://web.archive.org/web/20201201151339/https://www.railmagazine.com/news/network/gbrf-runs-first-uk-freight-emu-trial` (visited on 01/12/2020).

Delle Donne, Diego, Laurent Alfandari et al. (2023). 'Freight-on-Transit for urban last-mile deliveries: A strategic planning approach'. In: *Transportation Research Part B: Methodological* 169, pp. 53–81. DOI: 10.1016/j.trb.2023.01.004.

Delle Donne, Diego, Alberto Santini and Claudia Archetti (2024). *Instances and Results for the Paper "Integrating Public Transport in Sustainable Last-mile Delivery: Column Generation Approaches"*. DOI: 10.5281/zenodo.10493847. URL: https://github.com/alberto-santini/public-transport-lmd.

Deloison, Thomas et al. (Jan. 2020). *The future of the last-mile ecosystem.* Tech. rep. World Economic Forum. URL: https://web.archive.org/web/20210327151647/https://www.weforum.org/reports/the-future-of-the-last-mile-ecosystem (visited on 27/03/2021).

Der Spiegel (Feb. 2020). 'Scheuer will Paketauslieferung per U-Bahn testen'. In: *Der Spiegel*. URL: https://web.archive.org/web/20201130023547/https://www.spiegel.de/wirtschaft/unternehmen/scheuer-will-paketauslieferung-per-u-bahn-testen-a-ac3bd289-92d9-4c31-a969-4dbd8b504712 (visited on 30/11/2020).

Drexl, Michael and Michael Schneider (2015). 'A survey of variants and extensions of the location-routing problem'. In: *European Journal of Operational Research* 241 (2), pp. 283–308. DOI: 10.1016/j.ejor.2014.08.030.

Edrington, Suzie et al. (Mar. 2017). *Using public transportation to facilitate last mile delivery package delivery.* Guidebook TxDOT-0-6891-P2. Texas A&M Transportation Institute and Texas Department of Transportation. URL: https://web.archive.org/web/20170910010040/https://static.tti.tamu.edu/tti.tamu.edu/documents/0-6891-P2.pdf (visited on 10/09/2017).

Ghilas, Veaceslav, Emrah Demir and Tom Van Woensel (2016a). 'A scenario-based planning for the pickup and delivery problem with time windows, scheduled lines and stochastic demands'. In: *Transportation Research Part B: Methodological* 91, pp. 34–51. DOI: 10.1016/j.trb.2016.04.015.

Ghilas, Veaceslav, Emrah Demir and Tom Van Woensel (2016b). 'An adaptive large neighborhood search heuristic for the Pickup and Delivery Problem with Time Windows and Scheduled Lines'. In: *Computers & Operations Research* 72, pp. 12–30. DOI: 10.1016/j.cor.2016.01.018.

Gianessi, Paolo et al. (2016). 'The Multicommodity-Ring Location Routing Problem'. In: *Transportation Science* 50.2, pp. 541–558. DOI: 10.1287/trsc.2015.0600.

Guthrie, Cameron, Samuel Fosso-Wamba and Jean Brice Arnaud (2021). 'Online consumer resilience during a pandemic: An exploratory study of e-commerce behavior before, during and after a COVID-19 lockdown'. In: *Journal of Retailing and Consumer Services* 61. DOI: 10.1016/j.jretconser.2021.102570.

He, Yunzhu and Zhongzhen Yang (2018). 'Parcel delivery by collaborative use of truck fleets and bus-transit vehicles'. In: *Transportation Journal* 57.4, pp. 399–428. DOI: 10.5325/transportationj.57.4.0399.

Hörsting, Lena and Catherine Cleophas (2023). 'Scheduling shared passenger and freight transport on a fixed infrastructure'. In: *European Journal of Operational Research* 306 (3), pp. 1158–1569. DOI: 10.1016/j.ejor.2022.07.043.

Irnich, Stefan and Guy Desaulniers (2005). 'Shortest Path Problems with Resource Constraints'. In: *Column Generation*. Ed. by Guy Desaulniers, Jacques Desrosiers and Marius Solomon. Springer, pp. 33–65. DOI: 10.1007/0-387-25486-2_2.

Ji, Yuxiong et al. (2020). 'A Multimodal Passenger-and-Package Sharing Network for Urban Logistics'. In: *Journal of Advanced Transportation*. Article ID 6039032. DOI: 10.1155/2020/6039032.

Kikuta, Jun et al. (2012). 'New subway-integrated city logistics system'. In: *Procedia Social and Behavioral Sciences* 39, pp. 476–489. DOI: 10.1016/j.sbspro.2012.03.123.

Klein, Patrick Sean and Maximilian Schiffer (2022). 'Electric vehicle charge scheduling with flexible service operations'. In: *arXiv preprint arXiv:2201.03972*.

Longhorn, Danny (July 2021). 'Passenger trains converted to deliver parcels to city centres'. In: *Rail Business Daily*. URL: https://web.archive.org/web/20210707152625/https://news.railbusinessdaily.com/passenger-trains-converted-to-deliver-parcels-to-city-centres/ (visited on 07/07/2021).

Majoral, Genís, Francesc Gasparín and Sergi Saurí (2021). 'Reucing e-commerce delivery externalities with taxation. Application to Barcelona.' In: *Proceedings of the Transportation Research Board 100th Annual Meeting, 05–29 January 2021, Washington DC, United States*. Transportation Research Board 100th Annual Meeting (5th–29th Jan. 2021). Washington DC, United States, pp. 642–655. DOI: 10.1177/03611981211012412.

Mandal, Minakshi Punam and Claudia Archetti (2023). *A Decomposition Approach to Last Mile Delivery Using Public Transportation Systems*. arXiv: 2306.04219 [math.OC].

Masson, Renaud et al. (2017). 'Optimization of a city logistics transportation system with mixed passengers and goods'. In: *EURO Journal on Transportation and Logistics* 6 (1), pp. 81–109. DOI: 10.1007/s13676-015-0085-5.

Pessoa, Artur et al. (2023). 'A unified exact approach for Clustered and Generalized Vehicle Routing Problems'. In: *Computers & Operations Research* 149. DOI: 10.1016/j.cor.2022.106040.

Sadykov, Ruslan et al. (2019). 'Primal heuristics for branch and price: The assets of diving methods'. In: *INFORMS Journal on Computing* 31.2, pp. 251–267. DOI: 10.1287/ijoc.2018.0822.

Saito, Shigeo and Yomiuri Shimbun (Feb. 2021). 'Next stop, package delivery: bus and courier link up to serve rural area'. In: *The Japan News*. URL: https://web.archive.org/web/20210219030006/https://the-japan-news.com/news/article/0007140779 (visited on 19/02/2021).

Segura, Vicente et al. (Feb. 2020). *Last mile logistics: challenges and solutions in Spain*. Tech. rep. Deloitte. URL: https://web.archive.org/web/20210802053540/https://www2.deloitte.com/content/dam/Deloitte/es/Documents/operaciones/deloitte-es-operations-last-mile.pdf (visited on 02/08/2021).

Silva, Vasco, António Amaral and Tânia Fontes (2023). 'Sustainable Urban Last-Mile Logistics: A Systematic Literature Review'. In: *Sustainability* 15.3. DOI: 10.3390/su15032285.

Sustainable Bus (July 2021). 'What if trams were also to deliver goods? RegioKargo project launched in Karlsruhe'. In: *Sustainable Bus*. URL: https://web.archive.org/web/20210708123229/https://www.sustainable-bus.com/news/trams-deliver-goods-regiokargo-karlsruhe-project/ (visited on 08/07/2021).

Tagmouti, Mariam, Michel Gendreau and Jean-Yves Potvin (2007). 'Arc routing problems with time-dependent service costs'. In: *European Journal of Operational Research* 181.1, pp. 30–39.

Villa, Rafael and Andrés Monzón (2021a). 'A Metro-Based System as Sustainable Alternative for Urban Logistics in the Era of E-Commerce'. In: *Sustainability* 13 (8). DOI: 10.3390/su13084479.

Villa, Rafael and Andrés Monzón (2021b). 'Mobility Restrictions and E-Commerce: Holistic Balance in Madrid Centre during COVID-19 Lockdown'. In: *Economies* 9.2 (57). DOI: 10.3390/economies9020057.

Viu-Roig, Marta and Eduard Alvarez-Palau (2020). 'The Impact of E-Commerce-Related Last-Mile Logistics on Cities: A Systematic Literature Review'. In: *Sustainability* 12.16. DOI: 10.3390/su12166492.

Wang, Lina, Elliot Rabinovich and Harish Guda (2023). 'An analysis of operating efficiency and policy implications in last-mile transportation following Amazon's integration'. In: *Journal of Operations Management* 69 (1), pp. 9–35. DOI: 10.1002/joom.1172.

Zhao, Laijun et al. (2018). 'Location selection of intra-city distribution hubs in the metro-integrated logistics system'. In: *Tunnelling and Underground Space Technology* 80, pp. 246–256. DOI: 10.1016/j.tust.2018.06.024.

Zhou, Fangting and Jin Zhang (2020). 'Freight Transport Mode Based on Public Transport: Taking Parcel Delivery by Subway as an Example'. In: *Proceedings of the 6th International Conference on Transportation Engineering, 20–22 September 2019, Chengdu, China.* 6th International Conference on Transportation Engineering (20th–22nd Sept. 2019). Chengdu, China, pp. 745–754.

A Additional notation

We first describe how we build set $\mathcal{S}^{\text{in}}_c$ introduced in Section 4, i.e., the set of in-stops that a truck can use to unload a parcel for customer $c \in \mathcal{C}$. Given two stops $s_1 \in \mathcal{S}^{\text{in}}$ and $s_2 \in \mathcal{S}^{\text{out}}$, let the indicator parameter $\beta_{s_1 s_2} \in \{0, 1\}$ take value 1 iff a bus route links the two stops. Then, for each customer $c \in \mathcal{C}$, we define the following.

For each out-stop $s_2 \in \mathcal{S}^{\text{out}}_c$, feasible arrival times of c's parcel at s_2 are constrained by the customer's time window, the maximum wait time at the stop and the maximum courier route time. We denote this set of feasible arrival times as $\Theta_{s_2 c}$:

$$\Theta_{s_2 c} = \left[\underline{T}_c - T_{s_2} - W^{\max} - (L^{\max} - l_{cs_2}), \bar{T}_c - T_{s_2} - l_{s_2 c}\right].$$

Feasible times $\Theta_{s_2 c}$ constrain the set of buses which can carry c's parcel, when the parcel is delivered starting from s_2. We denote with $\mathcal{P}^{\text{out}}_{s_2 c}$ such set:

$$\mathcal{P}^{\text{out}}_{s_2 c} = \left\{p \in \mathcal{P}_{s_2} \;:\; t^{s_2}_p \in \Theta_{s_2 c}\right\},$$

where $\mathcal{P}_{s_2}$ is the set of buses which serve stop s_2. Moving backwards to the first tier, we can then consider the feasible arrival times of a parcel at an in-stop $s_1 \in \mathcal{S}^{\text{in}}$ if the parcel must then travel on bus p. These times are constrained by the maximum wait time at the in-stop and the minimum time it takes to travel from the CDC to s_1:

$$\Theta_{s_1 p} = \left[\max\{t^{s_1}_p - T_{s_1} - W^{\max}, l_{os_1}\}, t^{s_1}_p - T_{s_1}\right].$$

The set of feasible arrival times for a parcel $c \in \mathcal{C}$ at in-stop $s_1 \in \mathcal{S}^{\text{in}}_c$ is similarly defined as

$$\Theta_{s_1 c} = \bigcup_{p \in \mathcal{P}_c} \Theta_{s_1 p},$$

where $\mathcal{P}_c$ is defined below. We can then denote with $\mathcal{S}^{\text{in}}_{s_2 c}$ the set of in-stops which a bus can use to unload a parcel which will be delivered to customer c starting from out-stop s_2:

$$\mathcal{S}^{\text{in}}_{s_2 c} = \left\{s_1 \in \mathcal{S}^{\text{in}} \;:\; \beta_{s_1 s_2} = 1 \text{ and } \exists p \in \mathcal{P}^{\text{out}}_{s_2 c} \text{ s.t. } \Theta_{s_1 p} \neq \emptyset\right\}.$$

Finally, the set of in-stops that a truck can use to unload a parcel bound to customer c is

$$\mathcal{S}^{\text{in}}_c = \bigcup_{s_2 \in \mathcal{S}^{\text{out}}_c} \mathcal{S}^{\text{in}}_{s_2 c}.$$

Next, we characterise set $\mathcal{R}^{\text{D}}_{s_1 pc}$ introduced in Section 4. We can first determine a subset of buses $\mathcal{P}_c \subseteq \mathcal{P}$, which can carry the parcel to customer c. Each bus in this set must reach an out-stop from where the parcel can reach c's location and must do so at a time which is compatible with c's time windows once the maximum wait time at the out-stop, the courier travel time and the courier maximum route duration have been taken into account. Therefore, we define

$$\begin{aligned}\mathcal{P}_c = \{p \in \mathcal{P} \mid \exists s_2 \in \mathcal{S}_p \cap \mathcal{S}_c^{\text{out}} \ : &\ t_p^{s_2} + T_{s_2} + W^{\max} + L^{\max} \geq \underline{T}_c \text{ and} \\ &\ t_p^{s_2} + T_{s_2} + l_{cs_2} \leq \bar{T}_c\}.\end{aligned}$$

We also consider the set of buses which can carry customer c's parcel when a truck unloads it at in-station $s_1 \in \mathcal{S}_c^{\text{in}}$. This set is simply $\mathcal{P}_{s_1c}^{\text{in}} = \mathcal{P}_{s_1} \cap \mathcal{P}_c$. We define $\mathcal{R}_{s_1pc}^{\text{D}}$ for each customer c, each in-stop $s_1 \in \mathcal{S}_c^{\text{in}}$ and each bus $p \in \mathcal{P}_{s_1c}^{\text{in}}$ as:

$$\mathcal{R}_{s_1pc}^{\text{D}} = \{r \in \mathcal{R}_c^{\text{D}} \mid t_p^{s_1} - W^{\max} \leq t_r^{s_1} \leq t_p^{s_1}\},$$

where $t_r^{s_1}$ indicates the arrival time of the truck arrives at in-stop s_1 in route r, plus the handling time T_{s_1}. The above definition ensures that the truck arrives at the stop before the bus (leaving enough time for parcel handling) but not so early that it violates the maximum parcel wait time.

We now characterise set $\mathcal{R}_{ps_2c}^{\text{F}}$ for each customer $c \in \mathcal{C}$, each out-stop $s_2 \in \mathcal{S}_c^{\text{out}}$, and each bus $p \in \mathcal{P}_{s_2c}^{\text{out}}$:

$$\mathcal{R}_{ps_2c}^{\text{F}} = \{r \in \mathcal{R}_{s_2}^{\text{F}} \cap \mathcal{R}_c^{\text{F}} \mid t_p^{s_2} \leq t_r^{s_2} \leq t_p^{s_2} + W^{\max}\},$$

where $t_r^{s_2}$ denotes the start time of the courier from out-stop s_2 in route r, plus the handling time T_{s_2}. The above definition ensures that the bus arrives at the out-stop before the courier leaves (leaving enough time for parcel handling) but not so early that it violates the maximum parcel wait time. Finally, we introduce the following convenient notation for the set of stops at which a parcel can be picked up and, respectively, delivered by a given bus:

$$\mathcal{S}_{pc}^{\text{in}} = \mathcal{S}_p \cap \mathcal{S}_c^{\text{in}}, \quad \mathcal{S}_{pc}^{\text{out}} = \mathcal{S}_p \cap \mathcal{S}_c^{\text{out}}.$$

B Details of the route generation subproblems

B.1 Labelling algorithm introduced in Section 5.1.2

As for the labelling algorithm on graph H, we associate a label L_P to each partial path P from the CDC to a vertex $(s, c, \ell) \in W$. The label has the following components:

- $v_P = (s, c, \ell)$ is the end vertex of the path.
- $\mathcal{S}_P \subseteq \mathcal{S}^{\text{in}}$ is the set of in-stops which can still be visited when departing from v_P.
- $\mathcal{C}_P \subseteq \mathcal{C}$ is the set of customers whose parcels can still be delivered when departing from v_P.
- $\tau_P \geq 0$ is the traversing time of the path, i.e., the sum of traversing times of the arcs used in the path.
- $Q_P \geq 0$ is the spare capacity on the truck when departing from v_P.
- $C_P \in \mathbb{R}$ is the reduced cost associated with the path.

The relevant differences with the algorithm presented in Section 5.1.1 are the presence of time windows on the vertices (s, c, ℓ), implicitly defined by the intervals $w_{s\ell}$, and the scalar cost C_P. Time windows are easily accounted for using a time resource and the corresponding resource windows (Beasley and Christofides 1989). Regarding the cost, when path P is extended to a new vertex $(s', c', \ell') \in W$ along arc $b \in B$, C_P is updated as follows:

$$C_{P'} = C_P + c_b + \lambda^{(1i)}_{c'} + \sum_{\substack{p \in \mathcal{P}_{s'c'} \text{ s.t.} \\ I_{s'p} \cap w_{s'\ell'} \neq \emptyset}} \lambda^{(1g)}_{s'pc'}.$$

B.2 Labelling algorithm introduced in Section 5.2

Given an out-stop $s \in \mathcal{S}^{\text{out}}$, and an interval $w_{s\ell}$, we first define the complete, simple, directed graph $G_{s\ell}$ used to solve the shortest-path problem mentioned in Section 5.2. The vertex set, denoted $V_{s\ell}$, consists of s and all customers which can be served from s starting at interval ℓ:

$$V_{s\ell} = \left\{ c \in \mathcal{C} \ : \ s \in \mathcal{S}^{\text{out}}_c, \ \Theta_{sc} \cap w_{s\ell} \neq \emptyset \right\},$$

where $\mathcal{S}^{\text{out}}_c$ and Θ_{sc} are defined, respectively, in Section 3 and Appendix A. A feasible route corresponds to a path in $G_{s\ell}$ which: (i) starts from s at a time contained in interval $w_{s\ell}$; (ii) ends in s; (iii) visits each other vertex of $V_{s\ell}$ at most once; (iv) respects the courier capacity Q^{F}; (v) has a maximum travel time, defined as the sum of the arc traversing times, of $L^{\max}$; (vi) visits each vertex c not later than $\bar{T}_c$. As is standard in the vehicle routing literature, we allow a courier to visit customer c before the beginning of c's time window (time $\underline{T}_c$) but, in that case, the courier must wait until $\underline{T}_c$ before performing the delivery. The cost of a path in $G_{s\ell}$ is equal to the sum of the costs of the used arcs, minus dual prizes

$$\lambda^{(1j)}_c + \sum_{\substack{p \in \mathcal{P}^{\text{out}}_{sc} \\ t^s_p + T_s \in w_{s\ell}}} \lambda^{(1h)}_{spc}$$

collected at each visited customer c, plus the constant dual price $\lambda^{(1c)}_s$.

As mentioned in Section 5.2, the problem of finding the shortest path in $G_{s\ell}$ corresponding to a feasible courier route is a resource-constrained shortest-path problem. As is common in the literature, elementarity is ensured by associating a binary resource with each customer. To respect capacity, time windows, and maximum duration constraints, we further introduce their respective continuous resources. Finally, we solve $\text{SP}_y(s, \ell)$ with a labelling algorithm with the usual dominance rules. We refer the reader to (Irnich and Desaulniers 2005) for a thorough introduction to labelling algorithms for resource-constrained shortest-path problems.

C Semi-compact formulation

We present the complete semi-compact formulation for 3T-DPPT introduced in Section 4. Using notation $\bar{\mathcal{S}}^{\text{in}} = \mathcal{S}^{\text{in}} \cup \{o\}$ and $\bar{\mathcal{S}}^{\text{in}}_c = \mathcal{S}^{\text{in}}_c \cup \{o\}$, the three sets of variables we use to replace variables x are as follows. First, $w_{dij} \in \{0, 1\}$, taking value one if and only if truck $d \in \mathcal{D}$ travels from i to j $(i, j \in \bar{\mathcal{S}}^{\text{in}}$. Second, $\pi_{ds} \in \mathbb{R}$, indicating the time when truck $d \in \mathcal{D}$ completes parcel delivery at stop $s \in \mathcal{S}^{\text{in}}$. (We also use variable π_{do} to indicate the return time at the CDC.) Finally, $\delta_{dcs} \in \{0, 1\}$, taking value one if and only if truck $d \in \mathcal{D}$ delivers parcel $c \in \mathcal{C}$ at stop $s \in \mathcal{S}^{\text{in}}$. In the following, c_{ij} is the travel cost of a truck driving from i to j and M_s (for $s \in \mathcal{S}^{\text{in}}$) is a big constant, e.g., the last bus arrival time at stop s.

$$\min \quad \sum_{d \in \mathcal{D}} \sum_{\substack{i,j \in \bar{\mathcal{S}}^{\text{in}}_c \\ i \neq j}} c_{ij} w_{dij} + \sum_{r \in \mathcal{R}^{\text{F}}} c_r y_r \tag{5a}$$

$$\sum_{s\in\bar{\mathcal{S}}^{\text{in}}} w_{dos} = 1 \qquad \forall d \in \mathcal{D} \tag{5b}$$

$$\sum_{i\in\bar{\mathcal{S}}^{\text{in}}\setminus\{s\}} w_{dis} = \sum_{i\in\bar{\mathcal{S}}^{\text{in}}\setminus\{s\}} w_{dsi} \qquad \forall d \in \mathcal{D},\ \forall s \in \mathcal{S}^{\text{in}} \tag{5c}$$

$$\sum_{s\in\bar{\mathcal{S}}^{\text{in}}} w_{dso} = 1 \qquad \forall d \in \mathcal{D} \tag{5d}$$

$$\pi_{di} + l_{ij} - M_j(1 - w_{dij}) \le \pi_{dj} - T_j \qquad \forall d \in \mathcal{D},\ \forall i \in \bar{\mathcal{S}}^{\text{in}},\ \forall j \in \mathcal{S}^{\text{in}} \setminus \{i\} \tag{5e}$$

$$\pi_{dj} - T_j \le \pi_{di} + l_{ij} + M_j(1 - w_{dij}) \qquad \forall d \in \mathcal{D},\ \forall i \in \bar{\mathcal{S}}^{\text{in}},\ \forall j \in \mathcal{S}^{\text{in}} \setminus \{i\} \tag{5f}$$

$$\sum_{p\in\mathcal{P}_c} t_p^s z_{pcs}^{\text{in}} - W^{\max} - M_s(1 - \delta_{dcs}) \le \pi_{ds} \qquad \forall d \in \mathcal{D},\ \forall c \in \mathcal{C},\ \forall s \in \mathcal{S}_c^{\text{in}} \tag{5g}$$

$$\pi_{ds} \le \sum_{p\in\mathcal{P}_c} t_p^s z_{pcs}^{\text{in}} + M_s(1 - \delta_{dcs}) \qquad \forall d \in \mathcal{D},\ \forall c \in \mathcal{C},\ \forall s \in \mathcal{S}_c^{\text{in}} \tag{5h}$$

$$\delta_{dcs} \le \sum_{i\in\mathcal{S}^{\text{in}}\cup\{o\},i\neq s} w_{dis} \qquad \forall d \in \mathcal{D},\ \forall c \in \mathcal{C},\ \forall s \in \mathcal{S}_c^{\text{in}} \tag{5i}$$

$$\sum_{s\in\mathcal{S}_c^{\text{in}}} \sum_{d\in\mathcal{D}} \delta_{dcs} = 1 \qquad \forall c \in \mathcal{C} \tag{5j}$$

$$\sum_{p\in\mathcal{P}_{cs}} z_{pcs}^{\text{in}} = \sum_{d\in\mathcal{D}} \delta_{dcs} \qquad \forall c \in \mathcal{C},\ \forall s \in \mathcal{S}_c^{\text{in}} \tag{5k}$$

$$\sum_{c\in\mathcal{C}} \sum_{s\in\mathcal{S}_c^{\text{in}}} q_c \delta_{dcs} \le Q_d \qquad \forall d \in \mathcal{D} \tag{5l}$$

$$\text{Constraints (1c)-(1f), (1h), (1j)} \tag{5m}$$

$$w_{dij} \in \{0,1\} \qquad \forall d \in \mathcal{D},\ \forall i,j \in \bar{\mathcal{S}}^{\text{in}} \tag{5n}$$

$$\pi_{ds} \in \mathbb{R}^+ \qquad \forall d \in \mathcal{D},\ \forall s \in \bar{\mathcal{S}}^{\text{in}} \tag{5o}$$

$$\delta_{dcs} \in \{0,1\} \qquad \forall d \in \mathcal{D},\ \forall c \in C,\ \forall s \in \mathcal{S}_c^{\text{in}} \tag{5p}$$

$$y_r \in \{0,1\} \qquad \forall r \in \mathcal{R}^{\text{F}} \tag{5q}$$

$$z_{spc}^{\text{in}} \in \{0,1\} \qquad \forall c \in C,\ \forall s \in \mathcal{S}_c^{\text{in}},\ \forall p \in \mathcal{P}_{sc}^{\text{in}} \tag{5r}$$

$$z_{spc}^{\text{out}} \in \{0,1\} \qquad \forall c \in C,\ \forall s \in \mathcal{S}_c^{\text{out}},\ \forall p \in \mathcal{P}_{sc}^{\text{out}}. \tag{5s}$$

Constraints (5b)–(5d) are flow conservation constraints. Constraints (5e) and (5f) set the arrival times π_{ds} of trucks to stops. Constraints (5g) and (5h) ensure that these times (and the corresponding delivered parcels) are compatible with the bus arrival times. Constraint (5i) forbids delivering a package unless the corresponding truck visits the stop. Constraint (5j) ensures that each package is delivered to one stop by one truck. Constraint (5k) ensures that each package is collected by a bus at the same stop where it was delivered. Finally, constraint (5l) asserts that the truck capacities are respected.

Supplemental Material
Integrating Public Transport in Sustainable Last Mile Delivery

Diego Delle Donne, Alberto Santini and Claudia Archetti

1 Proofs of the theorems

1.1 Proof of Theorem 1

We begin by showing that, without valid inequalities (1i) and (1j), the optimal solution of the continuous relaxation of the 3T-DPPT can be arbitrarily small. Consider an instance with a single customer, $\mathcal{C} = \{c\}$, a single in-stop $\mathcal{S}^{\text{in}} = \{s_1\}$ and a single out-stop $\mathcal{S}^{\text{out}} = \{s_2\}$. The instance consists of one truck and one courier. Assume that both the truck and the courier have enough capacity to carry c's parcel, that c's time window corresponds to the entire planning horizon, and that $W^{\max}$ and $L^{\max}$ are equal to the planning horizon's length. Consider a bus line running through s_1 and s_2, with n buses scheduled during the time horizon, i.e., $\mathcal{P} = \{p_1, \dots, p_n\}$. Because of the assumptions above, $\mathcal{P}^{\text{in}}_{s_1 c} = \mathcal{P}^{\text{out}}_{s_2 c} = \mathcal{P}$.

Then set $\mathcal{R}^{\text{D}}$ consists of routes of type (o, s_1, o), with one route for each feasible start time. Analogously, $\mathcal{R}^{\text{F}}$ consists of routes of type (s_2, c, s_2), with one route for each feasible start time. All the routes in $\mathcal{R}^{\text{D}}$ have the same cost c_{D}, and all the routes in $\mathcal{R}^{\text{F}}$ have the same cost c_{F}.

Consider a route $r_d \in \mathcal{R}^{\text{D}}$ such that $t_{r_d s_1} \leq t^{p_1}_{s_1}$ and a route $r_f \in \mathcal{R}^{\text{F}}$ such that $t_{r_f s_2} \geq t^{p_n}_{s_2}$. An optimal solution of LP(1^-) is: $x_{r_d} = y_{r_f} = \frac{1}{n}$ and $z^{\text{in}}_{s_1 pc} = z^{\text{out}}_{s_2 pc} = \frac{1}{n}$ for all $p \in \mathcal{P}$. This solution satisfies constraints (1b)–(1h), and results in an objective value of $\frac{1}{n}(c_{\text{D}} + c_{\text{F}})$. Considering an instance with a sufficiently large number of buses n, then, one can make the objective value as small as desired.

We now consider the optimal solution of LP(1) for the same instance, i.e., the optimal solution of formulation 1 where (1i) and (1j) are included. Whatever the value of variables x and y in the solution of the relaxation, because of constraints (1i) and (1j), their respective sum is equal to 1. Thus the solution value is $c_{\text{D}} + c_{\text{F}}$ and $\frac{\text{LP}(1)}{\text{LP}(1^-)} = \frac{c_{\text{D}} + c_{\text{F}}}{\frac{1}{n}(c_{\text{D}} + c_{\text{F}})} = n$ which tends to infinity for $n \to \infty$. $\square$

More in general, constraints (1i) and (1j) guarantee that the value of LP(1) is always positive with a lower bound corresponding to the sum of the least cost route in $\mathcal{R}^{\text{D}}$ and $\mathcal{R}^{\text{F}}$. This lower bound is invalid in case constraints (1i) and (1j) are removed.

1.2 Proof of Proposition 1

To prove that L_{P_1} dominates L_{P_2} we shall prove that: (i) L_{P_1} can be extended to any *complete* route (i.e., ending at the CDC) to which L_{P_2} can be extended, and (ii) the corresponding extension of L_{P_1} has a cost better or equal than the extension of L_{P_2}.

Assume that L_{P_2} is extended by a sequence of nodes $(s_1, c_1), \dots, (s_k, c_k)$. This implies that $s_i \in \mathcal{S}_{P_2} \subseteq \mathcal{S}_{P_1}$, $c_i \in \mathcal{C}_{P_2} \subseteq \mathcal{C}_{P_1}$, for $i = 1, \dots, k$, and $\sum_{i=1}^k q_{c_i} \leq Q_{P_2} \leq Q_{P_1}$. Therefore, L_{P_1} can also be extended by the same sequence, thus proving (i). Now, the cost function for the extension of L_{P_1} to a node $w = (s, c) \in V$ is

$$\bar{C}_{P_1+w}(t) = \nu_{sc}(t) + \bar{C}_{P_1}(t - t_{vw}) + c_{vw} \leq \nu_{sc}(t) + \bar{C}_{P_2}(t - t_{vw}) + c_{vw} = \bar{C}_{P_2+w}(t),$$

where $v = v_{P_1} = v_{P_2}$. Hence, any sequence of extensions applied to both labels L_{P1} and L_{P2} will give routes in which the cost of the former is always better or equal to the cost of the latter, thus proving (ii). $\square$

As mentioned in Section 5.1.1, a key factor for the correctness of the dominance rule presented above is that the reduced cost associated with the label is calculated with respect to the ending vertex of the path, i.e., $\bar{C}_P(t)$, and not with respect to the starting time from the CDC, $C_P(t)$, as done by Tagmouti, Gendreau and Potvin (2007). Indeed, consider the example depicted in Figure 1 where, for

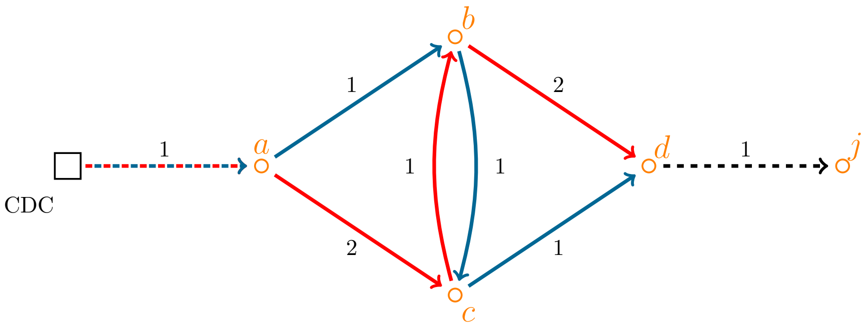

Figure 1: Example graph highlighting the importance of using $\bar{C}_P(t)$ over $C_P(t)$ in the dominance rule.

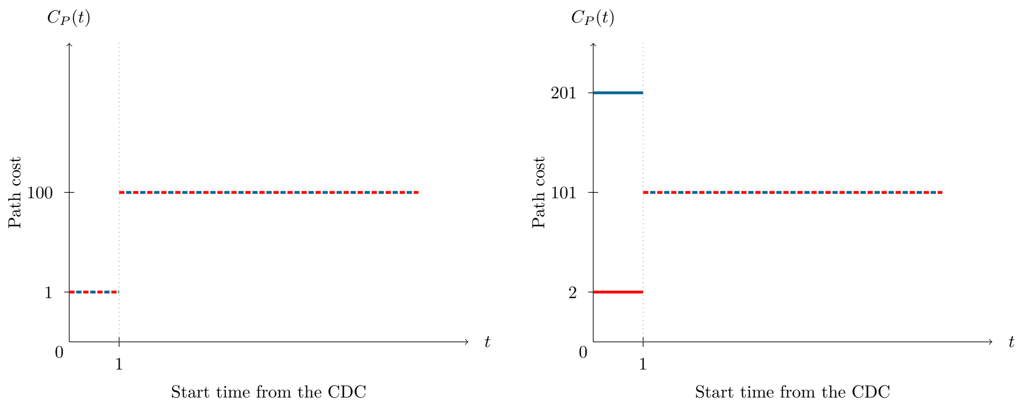

Figure 2: Cost functions of paths ending at vertex d (left) and paths extended to vertex j (right). Costs are in function to starting time from the CDC (as in Tagmouti, Gendreau and Potvin 2007).

ease of exposition, we do not consider the capacity constraint and where the numbers close to each arc are the corresponding travelling times.

The red and blue paths both end at vertex d. The total travel time is four for the blue path and six for the red one. We now extend the paths to vertex j. In our instance, the cost functions associated with each vertex of the graph are constant zero, except for vertices a and j where they are

$$C_a(t) = \begin{cases} 1 & \text{for } t \leq 2 \\ 100 & \text{for } t > 2 \end{cases} \qquad C_j(t) = \begin{cases} 200 & \text{for } t \leq 6 \\ 1 & \text{for } t > 6. \end{cases}$$

If we used the rule proposed by Tagmouti, Gendreau and Potvin 2007, calculating the costs with respect to the starting time from vertex the CDC, i.e., using $C_P(t)$, we would obtain the costs depicted in Figure 2 (left). The blue and red paths have the same cost for all values of t. Because the total travel time of the blue path is lower, it would dominate the red one. However, extending both labels to vertex j we get the costs depicted in Figure 2 (right). Here we see that the red path is no longer dominated because it has a lower cost for start times $t \leq 1$. In this example, in fact, it is preferable to arrive soon to node a but late to node d. Since trucks cannot wait at nodes, it is better to take a longer route between nodes a and d. Figure 3 shows the costs of the blue and red paths at vertex d, when computing costs using the arrival times at the nodes, i.e., using cost function $\bar{C}_P(t)$.

This example shows that we cannot apply the rule proposed by Tagmouti, Gendreau and Potvin (2007) to our case because trucks cannot wait along the route, and cost functions associated with nodes might be decreasing.

As a side note, we remark that there exists some ambiguity in the problem setting presented in Tagmouti, Gendreau and Potvin 2007. On one side, the compact problem formulation presented in

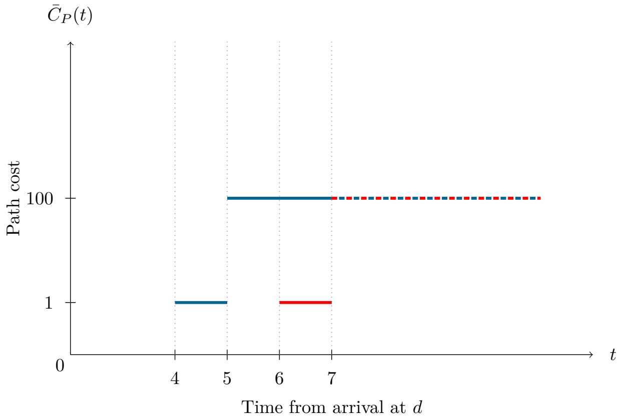

Figure 3: Cost functions of paths ending at vertex d, in function to arrival times (i.e., $\bar{C}_P(t)$).

the paper allows for waiting at customers. However, the problem description states that waiting is not allowed. Moreover, the way in which the cost function associated with each partial path is calculated is such that waiting at customers is not allowed.

2 A MIP approach to the pricing subproblem

The pricing problem SP_x can be tackled as a MIP and solved using a black-box solver. On small instances for which we can solve SP_x to optimality, the MIP formulation usually performs worse than the specialised approaches presented in Sections 5.1.1 and 5.1.2. Even when solving the pricing problem heuristically, the methods of Section 6.1 produce columns with lower reduced cost in a shorter time. On the other hand, the advantage of the MIP approach is that it promptly provides us with a lower (dual) bound on the reduced cost of a truck column. Such a bound can be exploited to obtain a dual bound for the entire 3T-DPPT, as in Section 3.1.

Keeping in mind the notation introduced in Table 1, consider variables $w_{ij} \in \{0,1\}$ defined for $i,j \in \{o\} \cup \mathcal{S}^{\text{in}}$ $(i \neq j)$ and taking value 1 iff j is visited immediately after i. Let $\pi_s \geq 0$ be a variable representing the departure time from $s \in \mathcal{S}^{\text{in}}$, if s is visited. Let $\gamma_{sc} \in \{0,1\}$ be a binary variable taking value 1 iff the route delivers the parcel of $c \in \mathcal{C}$ at in-stop $s \in \mathcal{S}^{\text{in}}$. Finally, let $\delta_{spc} \in \{0,1\}$ be a binary variable taking value 1 iff the route delivers the parcel of $c \in \mathcal{C}$ at in-stop $s \in \mathcal{S}^{\text{in}}$ at a time compatible with pick-up by bus $p \in \mathcal{P}^{\text{in}}_{sc}$. Then a MIP model to solve SP_x reads as follows:

$$
\begin{aligned}
\min \quad & \sum_{i,j\in\{o\}\cup\mathcal{S}^{\text{in}}} c_{ij} w_{ij} + \\
& \sum_{c\in\mathcal{C}} \sum_{s\in\mathcal{S}^{\text{in}}_c} \lambda^{(1\text{i})}_c \gamma_{sc} + \\
& \sum_{c\in\mathcal{C}} \sum_{s\in\mathcal{S}^{\text{in}}_c} \sum_{p\in\mathcal{P}^{\text{in}}_{cs}} \lambda^{(1\text{g})}_{spc} \delta_{spc} && && \text{(1a)} \\
\text{s.t.} \quad & \sum_{s\in\mathcal{S}^{\text{in}}} w_{os} = \sum_{s\in\mathcal{S}^{\text{in}}} w_{so} = 1 && && \text{(1b)} \\
& \sum_{\substack{s'\in\{o\}\cup\mathcal{S}^{\text{in}} \\ s'\neq s}} w_{ss'} = \sum_{\substack{s'\in\{o\}\cup\mathcal{S}^{\text{in}} \\ s'\neq s}} w_{s's} && \forall s \in \mathcal{S}^{\text{in}} && \text{(1c)} \\
& \pi_i + l_{ij} + T_j - M(1 - w_{ij}) \leq \pi_j && \forall i,j \in \mathcal{S}^{\text{in}} \cup \{0\},\ i \neq j, j \neq 0 && \text{(1d)} \\
& \pi_j \leq \pi_i + l_{ij} + T_j + M(1 - w_{ij}) && \forall i,j \in \mathcal{S}^{\text{in}} \cup \{0\},\ i \neq j, j \neq 0 && \text{(1e)} \\
& t^s_p - W^{\max} - M(1 - \delta_{spc}) \leq \pi_s && \forall c \in \mathcal{C},\ \forall s \in \mathcal{S}^{\text{in}}_c,\ \forall p \in \mathcal{P}^{\text{in}}_{sc} && \text{(1f)} \\
& \pi_s \leq t^s_p + M(1 - \delta_{spc}) && \forall c \in \mathcal{C},\ \forall s \in \mathcal{S}^{\text{in}}_c,\ \forall p \in \mathcal{P}^{\text{in}}_{sc} && \text{(1g)}
\end{aligned}
$$

Set	Description
$\mathcal{S}$	Set of bus stops.
$\mathcal{C}$	Set of customers.
$\mathcal{P}$	Set of buses.
$\mathcal{S}^{\text{in}} \subset \mathcal{S}$	Stops where buses can pick parcels up (in-stops).
$\mathcal{S}^{\text{in}}_p \subseteq \mathcal{S}^{\text{in}}$	In-stops served by bus $p \in \mathcal{P}$.
$\mathcal{S}^{\text{in}}_c \subset \mathcal{S}^{\text{in}}$	In-stops that can be used to deliver a parcel to customer $c \in \mathcal{C}$.
$\mathcal{P}^{\text{in}}_{sc} \subseteq \mathcal{P}$	Buses serving in-stop $s \in \mathcal{S}^{\text{in}}$ at a time compatible with carrying the parcel of customer $c \in \mathcal{C}$.
Parameter	**Description**
o	Location of the CDC.
$c_{ij} \geq 0$	Truck travel cost from location i to j; $i, j \in \{o\} \cup \mathcal{S}^{\text{in}}$.
$l_{ij} \geq 0$	Truck travel time from location i to j; $i, j \in \{o\} \cup \mathcal{S}^{\text{in}}$.
$T_s \geq 0$	Service time at bus stop $s \in \mathcal{S}^{\text{in}}$.
$t^s_p \geq 0$	Scheduled arrival time of bus $p \in \mathcal{P}$ at stop $s \in \mathcal{S}^{\text{in}}_p$.
$W^{\max} \geq 0$	Maximum time a parcel can wait at a stop.
$Q^{\text{D}} \geq 0$	Truck capacity.
$M > 0$	A sufficiently large number.
Variable	**Description**
$w_{ij} \in \{0, 1\}$	Takes value one if and only if the truck travels directly from location i to j; $i, j \in \{o\} \cup \mathcal{S}^{\text{in}}$.
$\pi_s \geq 0$	Truck departure time from in-stop $s \in \mathcal{S}^{\text{in}}$. Its value is unspecified if the truck does not visit in-stop s.
$\gamma_{sc} \in \{0, 1\}$	Takes value one if and only if the truck delivers the parcel of customer $c \in \mathcal{C}$ at in-stop $s \in \mathcal{S}^{\text{in}}$.
$\delta_{spc} \in \{0, 1\}$	Takes value one if and only if the truck delivers the parcel of customer $c \in \mathcal{C}$ at in-stop $s \in \mathcal{S}^{\text{in}}$ at a time compatible with pick-up by bus $p \in \mathcal{P}^{\text{in}}_{cs}$.

Table 1: Main notation used in Section 2.

$$\delta_{spc} \leq \gamma_{sc} \qquad \forall c \in \mathcal{C},\ \forall s \in \mathcal{S}^{\text{in}}_c,\ \forall p \in \mathcal{P}^{\text{in}}_{sc} \tag{1h}$$

$$\gamma_{sc} \leq \sum_{p \in \mathcal{P}^{\text{in}}_{sc}} \delta_{spc} \qquad \forall c \in \mathcal{C},\ \forall s \in \mathcal{S}^{\text{in}}_c \tag{1i}$$

$$\gamma_{sc} \leq \sum_{i \in \{o\} \cup \mathcal{S}^{\text{in}}} w_{is} \qquad \forall c \in \mathcal{C},\ \forall s \in \mathcal{S}^{\text{in}}_c \tag{1j}$$

$$w_{ij} \leq \sum_{c \in \mathcal{C}} \gamma_{jc} \qquad \forall i \in \{o\} \cup \mathcal{S}^{\text{in}},\ \forall j \in \mathcal{S}^{\text{in}} \setminus \{i\} \tag{1k}$$

$$\sum_{s \in \mathcal{S}^{\text{in}}_c} \gamma_{sc} \leq 1 \qquad \forall c \in \mathcal{C} \tag{1l}$$

$$\sum_{c \in \mathcal{C}} \sum_{s \in \mathcal{S}_c} q_c \gamma_{sc} \leq Q^{\text{D}} \tag{1m}$$

$$w_{ij} \in \{0, 1\} \qquad \forall i, j \in \{o\} \cup \mathcal{S}^{\text{in}},\ i \neq j \tag{1n}$$

$$\pi_s \geq 0 \qquad \forall s \in \mathcal{S}^{\text{in}} \tag{1o}$$

$$\gamma_{sc} \in \{0, 1\} \qquad \forall c \in \mathcal{C},\ \forall s \in \mathcal{S}^{\text{in}}_c \tag{1p}$$

$$\delta_{spc} \in \{0, 1\} \qquad \forall c \in \mathcal{C},\ \forall s \in \mathcal{S}^{\text{in}}_c,\ \forall p \in \mathcal{P}^{\text{in}}_{sc}, \tag{1q}$$

where $M > 0$ is a sufficiently large number. The objective function (1a) minimises the reduced cost of the route (but for constant term $\lambda^{(1b)}$). Constraints (1b) and (1c) are classical arc-based formulation constraints ensuring flow, elementarity, and starting and ending at the CDC. Constraints (1d) are MTZ-like constraints used to set the value of variables π_s while Constraints (1e) forbids the truck to delay its route at a stop. Variables π and δ are linked through constraints (1f) and (1g); γ and δ through constraints (1h) and (1i); w and γ through constraints (1j) and (1k). Finally, constraints (1l) ensure that each parcel is delivered at most once, and (1m) make sure that the truck's capacity is respected.

3 Details of the bounding techniques

3.1 Lagrangean bound

Consider an optimal solution to the RRMP when potentially not all negative reduced cost columns have been generated. The corresponding dual solution, which we denote with $\bar{\lambda}$, can be unfeasible for the dual of $\mathrm{MP}_{\mathrm{CONT}}$. This is because a missing column in RRMP may correspond to a violated constraint in the dual. Therefore, its objective value $\bar{Z}$ is not a valid dual bound for $\mathrm{MP}_{\mathrm{CONT}}$. The key idea behind the first bounding technique is to provide a way to restore the dual feasibility of $\bar{\lambda}$. In this way, we obtain a new dual-feasible solution $\tilde{\lambda}$ with dual objective value $\tilde{Z}$. By weak duality, $\tilde{Z}$ is a valid dual bound for $\mathrm{MP}_{\mathrm{CONT}}$ and, therefore, for the 3T-DPPT.

Let $\bar{Z}^{\mathrm{D}}$ be a lower bound for the minimum reduced cost of a truck route, and $\bar{Z}^{\mathrm{F}}_s$ be a lower bound for the minimum reduced cost of a courier route starting from $s \in \mathcal{S}^{\mathrm{out}}$. In other words, the following inequalities hold for $\bar{Z}^{\mathrm{D}}$ and $\bar{Z}^{\mathrm{F}}_s$:

$$\bar{Z}^{\mathrm{D}} \leq \min_{r \in \mathcal{R}^{\mathrm{D}}} \left\{ c_r + \bar{\lambda}^{(1\mathrm{b})} - \sum_{c \in \mathcal{C}_r} \bar{\lambda}^{(1\mathrm{i})}_c - \sum_{c \in \mathcal{C}_r} \sum_{p \in \mathcal{P}_c} \sum_{\substack{s \in \mathcal{S}^{\mathrm{out}}_{pc} \\ \text{s.t. } r \in \mathcal{R}^{\mathrm{D}}_{spc}}} \bar{\lambda}^{(1\mathrm{g})}_{spc} \right\} \tag{2}$$

$$\bar{Z}^{\mathrm{F}}_s \leq \min_{r \in \mathcal{R}^{\mathrm{F}}_s} \left\{ c_r + \bar{\lambda}^{(1\mathrm{c})}_s - \sum_{c \in \mathcal{C}_r} \bar{\lambda}^{(1\mathrm{j})}_c - \sum_{c \in \mathcal{C}_r} \sum_{\substack{p \in \mathcal{P}_c \\ \text{s.t. } r \in \mathcal{R}^{\mathrm{F}}_{spc}}} \bar{\lambda}^{(1\mathrm{h})}_{spc} \right\}. \tag{3}$$

Consider now the dual solution $\tilde{\lambda}$ obtained from $\bar{\lambda}$ modifying the following components:

$$\tilde{\lambda}^{(1\mathrm{b})} = \bar{\lambda}^{(1\mathrm{b})} - \bar{Z}^{\mathrm{D}} \tag{4}$$

$$\tilde{\lambda}^{(1\mathrm{c})}_s = \bar{\lambda}^{(1\mathrm{c})}_s - \bar{Z}^{\mathrm{F}}_s \quad \forall s \in \mathcal{S}^{\mathrm{out}}. \tag{5}$$

Theorem SM.1 establishes that $\tilde{\lambda}$ is, indeed, feasible for the dual of $\mathrm{MP}_{\mathrm{CONT}}$. The corresponding dual bound $\tilde{Z}$, takes value

$$\tilde{Z} = \bar{Z} - \left(\tilde{\lambda}^{(1\mathrm{b})} - \bar{\lambda}^{(1\mathrm{b})}\right) \cdot |D| - \sum_{s \in \mathcal{S}^{\mathrm{out}}} \left(\tilde{\lambda}^{(1\mathrm{c})}_s - \bar{\lambda}^{(1\mathrm{c})}_s\right) \cdot n_s.$$

Theorem SM.1. *Dual solution $\tilde{\lambda}$ obtained modifying $\bar{\lambda}$ according to* (4) *and* (5)*, and keeping all other components equal, is feasible for the dual of* $\mathrm{MP}_{\mathrm{CONT}}$.

Proof. Violated dual constraints correspond to missing primal variables x and y. There are two families of such constraints:

$$-\lambda^{(1\mathrm{b})} + \sum_{c \in \mathcal{C}_r} \lambda^{(1\mathrm{i})}_c + \sum_{c \in \mathcal{C}_r} \sum_{p \in \mathcal{P}_c} \sum_{\substack{s \in \mathcal{S}^{\mathrm{out}}_{pc} \\ \text{s.t. } r \in \mathcal{R}^{\mathrm{D}}_{spc}}} \lambda^{(1\mathrm{g})}_{spc} \leq c_r \qquad \forall r \in \mathcal{R}^{\mathrm{D}} \tag{6}$$

$$-\lambda^{(1\mathrm{c})}_s + \sum_{c \in \mathcal{C}_r} \lambda^{(1\mathrm{j})}_c + \sum_{c \in \mathcal{C}_r} \sum_{\substack{p \in \mathcal{P}_c \\ \text{s.t. } r \in \mathcal{R}^{\mathrm{F}}_{spc}}} \lambda^{(1\mathrm{h})}_{spc} \leq c_r \qquad \forall s \in \mathcal{S}^{\mathrm{out}},\ \forall r \in \mathcal{R}^{\mathrm{F}}_s. \tag{7}$$

Furthermore, $\lambda^{(1\mathrm{b})}$ only appears in (6) and $\lambda^{(1\mathrm{c})}$ only appears in (7). It follows that we shall prove that $\tilde{\lambda}^{(1\mathrm{b})}$ satisfies (6) and non-negativity constraints and that $\tilde{\lambda}^{(1\mathrm{c})}$ satisfies (7) and non-negativity constraints.

Indeed, (2) and (4) imply that (6) is satisfied. Analogously, (7) follows from (3) and (5). Finally, we remark that the RHS of (2) and (3) are non-positive if negative reduced cost columns are missing from the reduced sets, and thus the non-negativity of $\tilde{\lambda}^{(1\mathrm{b})}$ and $\tilde{\lambda}^{(1\mathrm{c})}$ follows. In case any of the two RHS is strictly positive, then one can always set the corresponding LHS ($\bar{Z}^{\mathrm{D}}$ or $\bar{Z}^{\mathrm{F}}_s$) equal to zero. In this case, the corresponding component of $\tilde{\lambda}$ is equal to the original component in $\bar{\lambda}$, and this component satisfies both the dual constraint and the non-negativity condition. □

Finally, we explain how to obtain lower bounds $\bar{Z}^{\mathrm{D}}$ and $\bar{Z}^{\mathrm{F}}_s$. Because, in practice, we can always solve SP_y to optimality in fractions of a second, we simply use the RHS of (3) as the value of $\bar{Z}^{\mathrm{F}}_s$. We then focus on $\bar{Z}^{\mathrm{D}}$. A straightforward approach to bound the reduced cost of a truck route involves relaxing some of the constraints of SP_x. In particular, one can relax the "hard" in-stop and customer elementarity constraints by applying the state-space relaxation technique (Christofides, Mingozzi and Toth 1981) to the labelling algorithms proposed in Sections 5.1.1 and 5.1.2. In practice, however, such relaxed subproblems are still time-consuming and produce loose bounds. Therefore, we decide to use a dual bound from the MIP model introduced in Section 2. We solve the model with a black-box solver and a short time limit and use the best dual bound returned by the solver.

3.2 Decomposition bound

We solve the GVRPTW introduced in Section 5.3 via branch-price-and-cut, adapting the algorithm presented by Pessoa et al. (2023). We make two modifications to their model. First, we add support for time windows, introducing a time resource and appropriate bounds in the pricing problem. Second, we change the VrpSolver (Pessoa et al. 2020) edge mapping to consider a directed graph and an asymmetric vehicle routing problem. The reason is that once time windows are introduced, the direction in which each route is traversed becomes essential.

4 Generating an initial set of first-tier columns

To initialise the column generation algorithm, we populate $\mathcal{R}^{\mathrm{D}}$ and $\mathcal{R}^{\mathrm{F}}$ with dummy columns which have: (i) a very high cost c_r; (ii) coefficient zero in all inequalities (1b) and (1c); (iii) coefficient one in all inequalities (1g), (1h), (1i) and (1j). Any solution of MP which selects a dummy column is infeasible for the 3T-DPPT.

Algorithm 1 Procedure to generate the initial columns of $\mathcal{R}^{\mathrm{D}}$.

$\bar{\mathcal{C}} \leftarrow \emptyset$ ▷ set of covered customers
$\bar{\mathcal{R}} \leftarrow \emptyset$ ▷ set of routes without start time
Phase 1: greedy creation
while $\bar{\mathcal{C}} \neq \mathcal{C}$ **do**
  $r \leftarrow (o)$ ▷ create an empty route
  $Q_r \leftarrow 0$ ▷ initialise truck used capacity
  $\underline{S} \leftarrow \mathcal{S}^{\mathrm{in}}$ ▷ available in-stops
  while $\underline{S} \neq \emptyset$ and $Q_r \leq Q^{\mathrm{D}}$ **do**
    Let s be the stop in $\underline{S}$ closest to the endpoint of r
    Append s to r
    $\underline{S} \leftarrow \underline{S} \setminus \{s\}$ ▷ update available in-stops
    for $c \in (\mathcal{C} \setminus \bar{\mathcal{C}}) \cap \mathcal{C}_s$ **do**
      if $Q_r + q_c \leq Q^{\mathrm{D}}$ **then**
        Route r will deliver c at s
        $Q_r \leftarrow Q_r + q_c$ ▷ update truck capacity
        $\bar{\mathcal{C}} \leftarrow \bar{\mathcal{C}} \cup \{c\}$ ▷ update covered customers
  $\bar{\mathcal{R}} \leftarrow \bar{\mathcal{R}} \cup \{r\}$ ▷ Add new route to initial set
Phase 2: greedy augmentation
for $r \in \bar{\mathcal{R}}$ **do**
  for in-stop s visited by r **do**
    for customer c not covered by r **do** ▷ c is covered by another route
      if $Q_r + q_c \leq Q^{\mathrm{D}}$ **then**
        Route r will deliver c at s
        $Q_r \leftarrow Q_r + q_c$
Phase 3: time assignment
$\bar{T} \leftarrow$ {possible start times}
for $r \in \bar{\mathcal{R}}$ **do**
  for start time $t \in \bar{T}$ **do**
    $r_t \leftarrow$ a copy of r with truck start time t
    for customer c covered by r_t **do**
      $s \leftarrow$ in-stop at which r_t delivers c's parcel
      $t' \leftarrow$ time at which c's parcel is ready for bus pick-up at s, according to r_t
      if $t' \notin \Theta_{sc}$ **then** ▷ no bus can pick up c's parcel
        Remove c from route r_t
    $\mathcal{R}^{\mathrm{D}} \leftarrow \mathcal{R}^{\mathrm{D}} \cup \{r_t\}$

To speed up the convergence of the pricing algorithm, in addition to the dummy column, we populate $\mathcal{R}^{\mathrm{D}}$ with feasible columns generated through Algorithm 1. In the first phase, the algorithm creates a set of truck routes without specifying their start time. It greedily adds new routes until all customers are served. In the second phase, for each route, it tries to fill the truck capacity greedily, adding more parcels to the route. This means that some parcels might be present in more than one route. This, however, is not a problem due to set-covering constraints (1i) and the fact that constraints (1g) ensure that exactly one parcel per client will be loaded onto a bus. In the last phase, each route is copied multiple times, changing its start time. When assigning times, it can happen that a route delivers c's parcel at an in-stop s at time t', but there is no bus which can pick up the parcel at a compatible time, i.e., $t' \notin \Theta_{sc}$, where Θ_{sc} is defined in A. In this case, such a route would lead to a worse continuous relaxation because it would cover row (1i) for customer c, but not row (1g) associated with c. Therefore, in the third phase, we remove from each route all parcels with no compatible bus.

5 Detailed numerical results

Table 2 shows the solutions obtained by our methods on the instances from Mandal and Archetti 2023, and a comparison of these with the results for the compact formulation (CF) and with the best results obtained by the decomposition heuristics (DH) proposed in Mandal and Archetti 2023. Column *Instance* is the instance number. The following three columns display the cost of the best solution found by CF, DH and our work, respectively. An empty cell indicates that the method could not find any feasible solution for the instance. The columns under the label *"Improvement of our work vs"* reports the percentage objective cost improvement of the best of our methods compared with: CF; the best of Mandal and Archetti's decomposition heuristics; and the best between the latter two, respectively. The summary table at the bottom reports the number of instances in which the best solution found by our method has better, worse or the same cost as the best solution from Mandal and Archetti 2023, either CF or DH.

6 The compact formulation of (Mandal and Archetti 2023) and our improvement

In the following, we detail the modifications required to forbid trucks from waiting at public transit stops to the formulation presented by Mandal and Archetti (2023). The notation used in that paper is different from the one used in our paper. The reader should refer to Mandal and Archetti's Table 1 for the list of sets and parameters and Table 2 for the list of decision variables. The formulation is presented in (3.1)–(3.35), in Section 1 of (Mandal and Archetti 2023). For the sake of completeness, we recall here that decision variable t^1_{ud} represents the time at which truck $d \in \mathcal{D}$ leaves stop u (or the CDC for $u = 0$) and binary decision variable w_{uvd} equals 1 if truck $d \in \mathcal{D}$ traverses arc (u, v). Also, parameters T^1_{uvd} and T'_v represent the time needed by for truck $d \in \mathcal{D}$ to traverse arc (u, v) and the service time at stop v, respectively. The following constraints are added to the formulation to prevent trucks from waiting at public transit stops:

$$t^1_{vd} \leq t^1_{ud} + T^1_{uvd} + T'_v + M(1 - w_{uvd}), \qquad \forall i, j \in \mathcal{S}^{\mathrm{in}} \cup \{0\},\ i \neq j,\ j \neq 0,\ \forall d \in \mathcal{D} \tag{8}$$

where $M > 0$ is a sufficiently large number.

References

Christofides, Nicos, Aristide Mingozzi and Paolo Toth (1981). 'State-space relaxation procedures for the computation of bounds to routing problems'. In: *Networks* 11.2, pp. 145–164. DOI: 10.1002/net.3230110207.

Mandal, Minakshi Punam and Claudia Archetti (2023). *A Decomposition Approach to Last Mile Delivery Using Public Transportation Systems*. arXiv: 2306.04219 [math.OC].

Pessoa, Artur et al. (2020). 'A generic exact solver for vehicle routing and related problems'. In: *Mathematical Programming* 183.1, pp. 483–523. DOI: 10.1007/s10107-020-01523-z.

		Best setting from		Improvement of our work vs		
Instance	CF	DH	This work	CF	DH	Best
1	2295.02	2295.02	2295.02	0.00 %	0.00 %	0.00 %
2	1460.23	1544.11	1460.23	0.00 %	5.43 %	0.00 %
3	1250.02	1364.22	1250.02	0.00 %	8.37 %	0.00 %
4	2182.70	2300.73	2182.70	0.00 %	5.13 %	0.00 %
5	1826.87	1921.52	1754.81	3.94 %	8.68 %	**3.94 %**
6	1929.18	2201.15	1929.18	0.00 %	12.36 %	0.00 %
7	2876.60	2992.42	2657.61	7.61 %	11.19 %	**7.61 %**
8	3402.97	3720.52	3402.97	0.00 %	8.54 %	0.00 %
9	2630.94	2748.94	2473.11	6.00 %	10.03 %	**6.00 %**
10	–	3303.32	3109.95	–	5.85 %	**5.85 %**
11	3294.88	3502.31	3107.72	5.68 %	11.27 %	**5.68 %**
12	4852.02	4540.10	4347.14	10.41 %	4.25 %	**4.25 %**
13	–	3957.66	4170.87	–	-5.39 %	-5.39 %
14	–	3532.50	3424.44	–	3.06 %	**3.06 %**
15	–	4113.17	3851.36	–	6.37 %	**6.37 %**
16	–	6491.98	5585.94	–	13.96 %	**13.96 %**
17	–	5690.91	5176.37	–	9.04 %	**9.04 %**
18	–	4817.65	4842.93	–	-0.52 %	-0.52 %
19	–	8102.66	7509.90	–	7.32 %	**7.32 %**
20	–	6941.22	6421.23	–	7.49 %	**7.49 %**
21	–	6315.74	5623.39	–	10.96 %	**10.96 %**
22	–	7184.62	5994.22	–	16.57 %	**16.57 %**
23	–	6388.33	6001.21	–	6.06 %	**6.06 %**
24	–	8350.63	8012.50	–	4.05 %	**4.05 %**
			# Better cost:	18	21	16
			# Worse cost:	0	2	2
			# Same cost:	6	1	6

Table 2: Solutions obtained by our methods on the instances from Mandal and Archetti 2023, compared with the results for the compact formulation (CF) and with the best results obtained by the decomposition heuristics (DH) proposed in the mentioned literature.

Pessoa, Artur et al. (2023). 'A unified exact approach for Clustered and Generalized Vehicle Routing Problems'. In: *Computers & Operations Research* 149. DOI: 10.1016/j.cor.2022.106040.

Tagmouti, Mariam, Michel Gendreau and Jean-Yves Potvin (2007). 'Arc routing problems with time-dependent service costs'. In: *European Journal of Operational Research* 181.1, pp. 30–39.