

VU Research Portal

The vehicle routing problem with partial outsourcing

Baller, Annelieke C.; Dabia, Said; Dullaert, Wout E.H.; Vigo, Daniele

published in

Transportation Science
2020

DOI (link to publisher)

[10.1287/TRSC.2019.0940](https://doi.org/10.1287/TRSC.2019.0940)

document version

Publisher's PDF, also known as Version of record

document license

Article 25fa Dutch Copyright Act

[Link to publication in VU Research Portal](#)

citation for published version (APA)

Baller, A. C., Dabia, S., Dullaert, W. E. H., & Vigo, D. (2020). The vehicle routing problem with partial outsourcing. *Transportation Science*, 54(4), 1034-1052. <https://doi.org/10.1287/TRSC.2019.0940>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

E-mail address:

vuresearchportal.ub@vu.nl



Transportation Science

Publication details, including instructions for authors and subscription information:
<http://pubsonline.informs.org>

The Vehicle Routing Problem with Partial Outsourcing

Annelieke C. Baller, Said Dabia, Wout E. H. Dullaert, Daniele Vigo

To cite this article:

Annelieke C. Baller, Said Dabia, Wout E. H. Dullaert, Daniele Vigo (2020) The Vehicle Routing Problem with Partial Outsourcing. *Transportation Science* 54(4):1034-1052. <https://doi.org/10.1287/trsc.2019.0940>

Full terms and conditions of use: <https://pubsonline.informs.org/Publications/Librarians-Portal/PubsOnLine-Terms-and-Conditions>

This article may be used only for the purposes of research, teaching, and/or private study. Commercial use or systematic downloading (by robots or other automatic processes) is prohibited without explicit Publisher approval, unless otherwise noted. For more information, contact permissions@informs.org.

The Publisher does not warrant or guarantee the article's accuracy, completeness, merchantability, fitness for a particular purpose, or non-infringement. Descriptions of, or references to, products or publications, or inclusion of an advertisement in this article, neither constitutes nor implies a guarantee, endorsement, or support of claims made of that product, publication, or service.

Copyright © 2020, INFORMS

Please scroll down for article—it is on subsequent pages







With 12,500 members from nearly 90 countries, INFORMS is the largest international association of operations research (O.R.) and analytics professionals and students. INFORMS provides unique networking and learning opportunities for individual professionals, and organizations of all types and sizes, to better understand and use O.R. and analytics tools and methods to transform strategic visions and achieve better outcomes.

For more information on INFORMS, its publications, membership, or meetings visit <http://www.informs.org>

The Vehicle Routing Problem with Partial Outsourcing

Annelieke C. Baller,^a Said Dabia,^a Wout E. H. Dullaert,^a Daniele Vigo^{a,b}

^aDepartment of Supply Chain Analytics, Vrije Universiteit Amsterdam, 1081 HV Amsterdam, Netherlands; ^bDepartment of Electrical, Electronic and Information Engineering “Guglielmo Marconi”, Alma Mater Università di Bologna, 04136 Bologna, Italy

Contact: a.c.baller@vu.nl,  <https://orcid.org/0000-0002-3826-0057> (ACB); s.dabia@vu.nl,  <https://orcid.org/0000-0003-0259-0684> (SD); w.e.h.dullaert@vu.nl,  <https://orcid.org/0000-0002-9416-211X> (WEHD); daniele.vigo@unibo.it,  <https://orcid.org/0000-0002-1499-8452> (DV)

Received: January 18, 2019

Revised: June 15, 2019

Accepted: August 7, 2019

Published Online in Articles in Advance:
June 17, 2020

<https://doi.org/10.1287/trsc.2019.0940>

Copyright: © 2020 INFORMS

Abstract. This paper introduces the vehicle routing problem with partial outsourcing (VRPPO) in which a customer can be served by a single private vehicle, by a common carrier, or by both a single private vehicle and a common carrier. As such, it is a variant of the vehicle routing problem with private fleet and common carrier (VRPPC). The objective of the VRPPO is to minimize fixed and variable costs of the private fleet plus the outsourcing cost. We propose two different path-based formulations for the VRPPO and solve these with a branch-and-price-and-cut solution method. For each path-based formulation, two different pricing procedures are designed and used when solving the linear relaxations by column generation. To assess the quality of the solution methods and gain insight in potential cost improvements compared with the VRPPC, we perform tests on two instance sets with up to 100 customers from the literature.

Funding: This work was supported by Nederlandse Organisatie voor Wetenschappelijk Onderzoek [Project Number 407-13-050].

Supplemental Material: The e-companion is available at <https://doi.org/10.1287/trsc.2019.0940>.

Keywords: vehicle routing problem • split delivery • private fleet and common carrier • outsourcing • exact algorithms

1. Introduction

To improve the efficiency of distribution logistics, academic literature has developed efficient solution algorithms and has explored novel distribution strategies. Fundamental optimization problems such as the capacitated vehicle routing problem and the vehicle routing problem with time windows (VRPTW), which underpin several distribution problems occurring in practice, have been the subject of intensive research efforts in search of more (cost) efficient solutions (Toth and Vigo 2014). Novel distribution strategies such as outsourcing and split deliveries contribute in a different but no less effective way to lowering distribution costs.

Outsourcing refers to the fact that the service of some customers can be entrusted to a third-party logistics service provider. In routing problems, outsourcing can be applied in case demand exceeds available transportation capacity or if it is more economical to do so. The two options for servicing customers—by a private vehicle fleet or a common carrier—open up additional opportunities for reducing distribution costs. Applications in the literature date back to Chu (2005), and a recent overview of the literature on the vehicle routing problem with private fleet and common carrier (VRPPC) can be found in Dabia, Lai, and Vigo (2019). Split deliveries imply that the demand of a customer is not necessarily delivered by a single

vehicle. Split deliveries are a necessity if demand exceeds vehicle capacity but can also prove cost efficient if, for example, demand of several customers is slightly higher than half of the vehicle capacity (Archetti, Savelsbergh, and Speranza 2008). Chen, Golden, and Wasil (2007) review several applications of split deliveries in VRPs.

Combining outsourcing and split delivery features has received increasing attention in the literature in recent years. To the best of our knowledge, only application-based studies have been reported in which heuristic methods are proposed, as discussed in Section 2.3. Our study is intended to contribute to the literature by formally describing the vehicle routing problem in which a customer can be served by a single private vehicle, by a common carrier, or by both a single private vehicle and a common carrier. We do not allow for multiple private vehicles to serve the same customer, as serving customers by multiple private vehicles and a common carrier may lead to customer inconvenience (see, for example, Bianchessi, Drexler, and Irnich (2019)). It should also be noted that allowing multiple private vehicle visits to a customer would lead to a more complicated optimization problem. We refer to the defined problem as the vehicle routing problem with partial outsourcing (VRPPO). We assume that customers impose time windows,

there is a heterogeneous limited fleet, and the outsourcing cost is a fixed fee per unit.

In this paper, we propose two path-based formulations for the VRPPO and solve these with a branch-and-price-and-cut approach because this method has proven to be an effective method for various (related) routing problems. The first formulation explicitly models the quantity delivered by a private vehicle to each customer, whereas the second formulation defers modeling this quantity and the resulting outsourced units to the pricing problem. For both problem formulations, we design two specialized pricing procedures to generate additional columns during the solution process. The first pricing procedure remains close to current literature and is closely related to the pricing algorithm proposed for the split delivery vehicle routing problem (SDVRP) by Desaulniers (2010). The second pricing procedure is closer to the algorithm proposed by Luo et al. (2016) and exploits some specific properties of the VRPPO.

A solution to the VRPPO consists of a set of routes for the private vehicles and a set of customers for which (part of) the demand's delivery is outsourced, and for both corresponding quantities. The objective is to minimize the total routing and outsourcing cost. The purpose of this paper is to investigate the effectiveness of the two formulations and corresponding solution methods and to compare the costs with the VRPPC to assess the gain that can be achieved by embracing both distribution strategies.

The paper is organized as follows. In Section 2, we review the literature related to the VRPPO. In Section 3, we describe the problem more formally and present the problem formulations and solution methods. Section 4 presents some implementation features, and Section 5 describes the computational results. Conclusions and suggestions for future research follow in Section 6.

2. Literature Review

In this section, the main literature related to the VRPPO is reviewed. First, in Section 2.1, we discuss literature on the SDVRP because the solution methods are relevant for solving the VRPPO. Second, the literature on the VRPPC is examined in Section 2.2 because a generalization of the VRPPC is studied. Finally, in Section 2.3, we review some work on combinations of the SDVRP and VRPPC.

2.1. Literature on the SDVRP

The vehicle routing problem with split deliveries and a private fleet is extensively studied. Archetti and Speranza (2012) provide a thorough overview of properties and solution methods, both exact and heuristic, for this problem and its variants, including the variant with time windows (SDVRPTW).

Desaulniers (2010) studies the SDVRPTW and proposes a branch-and-price-and-cut solution method. The columns in the master problem represent a route and corresponding "route delivery pattern," which indicates the quantities delivered to each customer. The pricing problem is a variant of the elementary shortest path problem with resource constraints (ESPPRC) and is solved with a labeling algorithm. In order to generate relevant delivery patterns, a label is extended to the next customer at most three times for a delivery of zero units, a delivery equal to the demand, and a delivery quantity strictly between zero and the demand. Archetti, Bianchessi, and Speranza (2011) propose an improved version of the algorithm by Desaulniers (2010) mainly by introducing a tabu search heuristic for the pricing problem and by applying new valid inequalities and separation procedures. Luo et al. (2016) propose another improvement over the algorithm by Archetti, Bianchessi, and Speranza (2011) and include a linear weight-related cost function. Instead of extending each label up to three times, they observe that the delivery quantities can be determined in a greedy way by fully serving the customers with the highest delivery quantity-related dual values. Therefore, during the labeling algorithm, Luo et al. (2016) do not keep track of the load of the vehicle and determine the delivery quantities afterward.

Archetti, Bianchessi, and Speranza (2011) propose a different solution method for the SDVRP with both limited and unlimited fleets. The master problem is comparable to the one of Desaulniers (2010), but the pricing problem is approached completely different. The pricing problem is also modeled as an ESPPRC, but each customer is represented by multiple nodes, one for each possible delivery quantity. As a result, the number of nodes in the expanded network increases rapidly, but the approach requires less complex dominance rules than Desaulniers (2010). Similarly, Salani and Vacca (2011) and Archetti, Bianchessi, and Speranza (2015) use expanded networks to solve variants of the SDVRP.

Several variants of the SDVRP have been proposed in the literature, and some of these variants attempt to prevent inconvenient situations for SDVRP customers. Gulczynski, Golden, and Wasil (2010) and Han and Chu (2016) consider the case with minimum delivery amounts, which reflect the fact that each delivery can be costly to both the distributor and the customer; hence, a delivery should be significant in terms of goods or value delivered. Ozbaygin, Karasan, and Yaman (2018) introduce for the SDVRP customer inconvenience constraints that set a maximum on the number of vehicles serving each customer, with the reasoning that handling multiple deliveries is not desirable in practice. Bianchessi, Drexler, and Irnich (2019)

apply these constraints to the SDVRPTW and, moreover, consider two other types of customer inconvenience constraints. They consider a maximum total number of visits to all customers together and temporal synchronization in which multiple visits can only be a certain number of time units apart from each other.

2.2. Literature on the VRPPC

The VRPPO is an extension of the VRPPC in which the service of a customer may be split between the two delivery types. The VRPPC was first proposed by Chu (2005) as a single depot routing problem with outsourcing options for which a simple heuristic based on a modified savings algorithm was developed. Later, some metaheuristics were introduced, and these achieved very good results on a large set of test instances considering both homogeneous and heterogeneous private fleets. In particular, Bolduc et al. (2008) proposed a perturbation-based procedure, Côté and Potvin (2009) defined a tabu search approach, and Potvin and Naud (2011) used an ejection-chain neighborhood within another tabu search algorithm. More recently, Stenger et al. (2013) introduced a multiple-depot version of the problem, called MDVRPPC, for which they developed a variable neighborhood search algorithm incorporating an innovative adaptive shaking mechanism to select routes and customers involved in the shaking step.

To the best of our knowledge, only two exact methods currently exist for this problem. Dabia, Lai, and Vigo (2019) developed a branch-cut-and-price algorithm for a variant of the VRPPC with heterogeneous fleet, which included time windows and quantity discount on the outsourced deliveries. Goeke, Gschwind, and Schneider (2019) propose a similar solution method for the VRPPC with customer-dependent, fixed fees as the outsourcing cost. For more recent literature on the VRPPC, we refer to the literature reviews by Gahm, Brabänder, and Tuma (2017) and Dabia, Lai, and Vigo (2019).

2.3. Literature on Combined Split Delivery and Outsourcing

Several studies consider the option to split deliveries over several shipment types, including private and common vehicles. Bolduc et al. (2010) study the so-called SDVRP with production and demand calendars, which is a multiperiod, inventory routing-like problem in which a delivery to a customer can be split over multiple, both private and common, vehicles. The authors propose a tabu search heuristic to solve the problem. A compact formulation for the studied problem, which is discarded in this paper because of conciseness, would be very similar to a single-period formulation of the problem by Bolduc et al. (2010) except for the splitting over private vehicles.

A ship routing problem with pickup and deliveries is studied by Lee and Kim (2015). It is possible to split the deliveries over multiple private vehicles, and outsourcing (part of) the delivery to a so-called tramp ship is also possible. The problem is formulated as a mixed-integer linear program in which the outsourcing costs are modeled as customer dependent and proportional to the outsourced quantity. Having pickup and deliveries is the main difference with the VRPPO. An adaptive large neighborhood search heuristic is proposed to solve the problem.

Keskin et al. (2014) consider a practical application in which outbound shipment of products needs to be optimized. Three transportation modes are considered simultaneously, with all vehicles belonging to outsourced carriers. Split deliveries between the different transportation modes is possible. For the so-called truckload mode, the routes also need to be determined because these are fully controlled by the company. The problem is split into an assignment and a routing problem, which are both solved using CPLEX.

Yan et al. (2015) study a multitrip SDVRP problem with soft time windows in which not all customers necessarily have to be fully served. For each undelivered unit of demand, a large penalty is incurred. This can also be seen as if delivery of these units is outsourced to a common carrier. The tests performed by Yan et al. (2015) are very limited; the authors speak about numerous test instances to assess the quality of the proposed heuristic, but they only report detailed results on one instance based on real-life data. In the solution of the real-life instance, all units of demand are delivered and the “outsourcing” option is not used, probably caused by the huge penalty on not satisfying some demand (10,000 New Taiwan Dollar (TWD) per unit compared with costs of 12.53TWD per unit distance traveling cost). The two-step solution approach based on time-space networks could, however, be used to model (a variant of) the VRPPO by choosing the appropriate values for the parameters.

In summary, for problems that contain both split delivery and outsourcing features, to our knowledge, only heuristic solution methods have been proposed in the literature. The problem formulation by Bolduc et al. (2010) is closest to a formulation for our problem. For the SDVRP, problem variants have been proposed that, for example, limit the number of visits to a customer, with the motivation to limit customer inconvenience. In the VRPPO, we do not allow multiple private vehicles to service the same customer. This restriction limits customer inconvenience by preventing a customer being served by, for example, two private vehicles and a common carrier. Moreover, this also substantially reduces the solution space, leading to more efficient solution methods. Some solution

methods developed for the SDVRP provide a starting point for our solution method for the VRPPO.

3. Problem Description and Formulation

The VRPPO is defined on a graph, $G = (V, A)$, in which V is the set of nodes containing a depot 0 and a set of customers $V_0 = V \setminus \{0\}$, and A is the set of arcs. Each customer has a certain demand, d_i , that must be fulfilled and a time window $[e_i, l_i]$ in which service must take place. There is a set of private, heterogeneous vehicles available to serve the customers, each with a capacity $Q_k, k \in K$, with K the set of vehicle types. There are m_k vehicles available of vehicle type $k \in K$. Next to the private vehicles, there is an option to outsource a delivery to a common carrier. In the VRPPO, it is possible to have a customer served by both a private vehicle and the common carrier. To limit the customer inconvenience, it is not possible to have multiple private vehicles serve the same customer. This results in three service options for each customer: full private delivery, full outsourcing, or a split delivery between one private vehicle and the common carrier.

Each arc (i, j) in graph G has an associated cost c_{ij} and a travel time t_{ij} . We assume that both the arc costs and travel times satisfy the triangle inequality. For each vehicle, there is a set up cost f_k depending on the type of vehicle $k \in K$. For outsourcing, a fixed fee of v is charged per unit of outsourced demand. Using a fixed fee per unit implies that there is no benefit from outsourcing more units than strictly necessary because of flat rate or discount structures. Hence, for a split customer, as many units as possible are delivered by the private vehicle. Moreover, because the fee is customer independent, there is no difference between outsourcing the delivery to different customers. The objective is to minimize total routing and outsourcing costs while respecting time windows, vehicle capacity, and vehicle fleet limitations.

We propose two path-based master problem formulations for this problem in Sections 3.2 and Section 3.3, respectively, which are solved with a branch-and-price-and-cut solution approach. Both formulations potentially have an exponential number of columns; therefore, we start with a limited subset of initial columns and iteratively generate more columns by solving a pricing problem. The solution of the pricing problem is a set of columns with negative reduced costs. If this set is empty, either branching is necessary or an integer solution is found. The procedure results in the optimal solution if all branch-and-bound nodes have been explored. To strengthen the linear relaxation of the master problem, subset-row (SR) inequalities are applied. These inequalities are introduced by Jepsen et al. (2008) for the VRPTW. The SR inequalities can immediately be applied to the first master problem; for the second master problem, we use the

generalized version as introduced by Dabia, Lai, and Vigo (2019) for a rich vehicle routing problem with private fleet and common carrier.

3.1. Properties

For the SDVRP, some properties have been established in the literature (Archetti and Speranza 2012); however, these mainly have to do with interactions among routes that visit the same customers and are not applicable here because, in the VRPPO, the routes of private vehicles do not visit the same customers. In the pricing problem, we do make use of the property in Lemma 1.

Lemma 1. *If the arc costs satisfy the triangle inequality, a route from the starting to the end depot contains at most one customer whose demand is split between the private vehicle and common carrier.*

The lemma can be proved by a simple exchange argument. Suppose that a route of a private vehicle contains a split delivery for two different customers, i and j , and that the vehicle's total load is equal to the vehicle capacity. By delivering more units to customer i with the private vehicle and outsourcing fewer units for customer j , and vice versa for customer j , and continuing this exchange until no units are delivered to customer j by the private vehicle, this exchange process results in a zero delivery by the private vehicle to customer j . Because of the triangle inequality, it is more expensive to visit customer j with the private vehicle (with a zero delivery) than not to visit customer j . Hence, such a route, with one split delivery, is never more expensive than the same route that contains two split deliveries. Therefore, this lemma implies that during execution of the labeling algorithm, in each partial path, there has to be at most one customer that does not receive its full demand by the private vehicle.

3.2. Master Problem 1 (MP1)

In the first master problem (MP1), a column represents a route visiting a set of customers and the corresponding delivery quantities delivered by the private vehicle. Let Ω_k be the set of routes p and associated delivery quantities for vehicle type $k \in K$, and let $\Omega = \bigcup_{k \in K} \Omega_k$. Let c_p be the routing cost of route $p \in \Omega$. Associate δ_i^p and a_{ip} with a route $p \in \Omega$ representing the delivery quantity for customer $i \in V$ and the number of times customer $i \in V$ is in the route, respectively. Define binary decision variables y_p , which indicate whether route $p \in \Omega$ is in the solution of MP1, and continuous decision variables β_i being the demand of customer $i \in V$ that is outsourced. The VRPPO can be formulated as follows:

$$\min \sum_{p \in \Omega} c_p y_p + \sum_{i \in V} v \beta_i, \quad (1a)$$

$$\text{s.t. } \sum_{p \in \Omega} \delta_i^p y_p + \beta_i \geq d_i, \forall i \in V, \quad (1b)$$

$$\sum_{p \in \Omega} a_{ip} y_p \leq 1, \quad \forall i \in V, \quad (1c)$$

$$\sum_{p \in \Omega_k} y_p \leq m_k, \quad \forall k \in K, \quad (1d)$$

$$\beta_i \geq 0, \quad \forall i \in V, \quad (1e)$$

$$y_p \in \{0, 1\}, \quad \forall p \in \Omega. \quad (1f)$$

The objective function (1a) minimizes the routing and outsourcing costs. Constraints (1b) make sure that the demand of each customer is satisfied by a delivery of a private vehicle, via outsourcing, or a combination of these. We do not allow for multiple visits by a private vehicle to a customer—that is, no private split delivery, which is enforced by constraints (1c). Constraints (1d) limit the number of vehicles used per type, and the domains of the decision variables are defined in constraints (1e) and (1f).

The pricing problem should generate columns defining a route and delivery quantities, in which at most one customer does not receive its full demand. Associate dual variables $\pi_i^{1b} \geq 0$, $\pi_i^{1c} \leq 0$, $\pi_k^{1d} \leq 0$ with constraints (1b)–(1d), respectively. Define $\pi_0^{1b} = 0$ and $\pi_0^{1c} = 0$ for the depot. Let δ^p be the vector of delivery quantities for the customers in route p . The reduced cost of a column for route p associated with vehicle type $k \in K$ can be expressed as follows:

$$\bar{c}_p(\delta^p) = f_k + \sum_{(i,j) \in A} (c_{ij} - \pi_i^{1c}) x_{ijp} - \sum_{i \in V^p} \delta_i^p \pi_i^{1b} - \pi_k^{1d}, \quad (2)$$

in which x_{ijp} is an integer variable counting the number of times arc (i, j) is traversed in route p , and V^p is the set of nodes in the route. Note that the reduced cost of a column depends on the delivery quantity to each customer in the route. For ease of notation, define the indicator function $\mathbb{1}\{event\} = 1$ if *event* is true, 0 otherwise, and define

$$\bar{c}_{ij} = c_{ij} - \pi_i^{1c} + \mathbb{1}\{i = 0\}(f_k - \pi_k^{1d}), \quad (3)$$

which gives reduced arc costs in which the vehicle setup and dual costs are accounted for in the outgoing arcs of the depot.

3.2.1. Pricing Algorithm 1 for MP1 (MP1-PA1). The pricing problem is a variant of the ESPPRC, which we solve with a labeling algorithm (see, e.g., Feillet et al. (2004), Righini and Salani (2006), and Tilk et al. (2017)). Because the pricing problem for the VRPPO is similar to the one for the SDVRPTW, we first adjust the labeling algorithm proposed for the SDVRPTW by Desaulniers (2010) to our problem (MP1-PA1). Desaulniers (2010) proposes creating up to three labels for each extension of a partial path in which the

delivery quantity to the next customer is either zero, full (equal to the demand), or partial (strictly between zero and the demand). Subsequently, in the master problem, any delivery quantity pattern for a route is created by taking convex combinations of the columns. To create any combination of delivery quantities, the zero deliveries are necessary. Desaulniers (2010) observes that only so-called extreme delivery patterns, which contain at most one partial delivery, are needed. In the SDVRPTW, multiple private vehicles can visit the same customer; therefore, routes are interdependent, and the actual delivery quantity cannot be determined in the pricing problem. On the contrary, in the VRPPO, this interdependency is not present; the delivery quantities can, therefore, be determined in the pricing problem. Consequently, we do not take convex combinations of columns in MP1, and hence, in the pricing algorithm for the VRPPO, the zero delivery option is not needed. By Lemma 1, only routes with at most one partial delivery need to be created.

Therefore, for the VRPPO, when extending a partial path to a customer j , up to two labels can be created; one label in which customer j is fully delivered by the private vehicle, and one label in which the demand of customer j will be partially delivered by the private vehicle and partially outsourced. Note that it is never optimal to fully outsource the demand of a visited customer, because in that case it is more efficient to not visit the customer at all. The outsourced part of the demand of the split customer is determined by the vehicle capacity and the demand of the other customers in the route, which means that we can only determine the delivery quantity when the route is complete. Because the delivery quantity δ_i^p for the split customer i is not known during the labeling algorithm, the reduced cost of a partial path p as in equation (3) cannot contain the contribution of this split delivery until reaching the end node. Therefore, during the labeling algorithm, we keep track of the maximum reduced cost of the partial path. This is equal to the case in which no units are delivered to the split customer by the private vehicle.

Let a label L correspond to a partial path $p(L)$ in the graph G starting at the depot. For a type- k vehicle, associate the following attributes with a label L :

- $i(L)$ Last node visited in partial path $p(L)$,
- $c(L)$ Maximum reduced cost of partial path $p(L)$ (i.e., no units to split customer),
- $q(L)$ Load of full deliveries in partial path $p(L)$,
- $t(L)$ Ready time at node $i(L)$ when reached through partial path $p(L)$,
- $r(L)$ Customer with split in path $p(L)$, -1 if no split in $p(L)$,

$\phi(L)$ The maximum quantity delivered in the split delivery by the private vehicle in partial path $p(L)$, if any ($\phi(L) = 0$ if $r(L) = -1$),
 $V(L)$ Set of visited nodes along path $p(L)$.

Furthermore, let $\bar{V}(L)$ denote the set of visited and unreachable nodes. Nodes are unreachable if they cannot be visited by extending the path $p(L)$ because of time windows. Also, if a split delivery is already in the path, and visiting a customer $j \in V_0$ would violate vehicle capacity, even by setting the split delivery to zero, then customer j is unreachable. Note that this differs from what can be assumed for the SDVRP (e.g., Desaulniers (2010)), because for the VRPPO, we do not have to consider extreme delivery patterns that contain zero deliveries.

Suppose we extend a label L' along arc $(i(L'), j)$ to node $j \in V \setminus \bar{V}(L')$ to generate a new label L . The resources for the new label L are established as follows:

$$\begin{aligned} i(L) &= j, \\ c(L) &= \begin{cases} c(L') + \bar{c}_{ij} - d_j \pi_j^{1b} & \text{if no split delivery at } j, \\ c(L') + \bar{c}_{ij} & \text{if a split delivery at } j, \end{cases} \\ q(L) &= \begin{cases} q(L') + d_j & \text{if no split delivery at } j, \\ q(L') & \text{if a split delivery at } j, \end{cases} \\ t(L) &= \max\{t(L') + t_{ij}, e_j\}, \\ r(L) &= \begin{cases} r(L') & \text{if no split delivery at } j, \\ j & \text{if a split delivery at } j, \end{cases} \\ \phi(L) &= \begin{cases} \min\{\phi(L'), Q - q(L)\} & \text{if no split delivery at } j, \\ \min\{d_j, Q - q(L)\} & \text{if a split delivery at } j, \end{cases} \\ V(L) &= V(L') \cup \{j\}. \end{aligned}$$

An extension from label L' to label L with $i(L) = j$ is feasible if $j \notin \bar{V}(L')$, $q(L) \leq Q$, $e_j \leq t(L) \leq l_j$, $\phi(L) \geq 0$. Note that a label with a split delivery for customer j cannot be created if $r(L') > -1$ because a split is already in the path.

The potential number of labels is huge. Therefore, to discard labels during the algorithm, sufficient dominance conditions are formulated. If $i \in V$ is the split customer, then the reduced cost of a label is a linear function in δ_i . Therefore, to compare two labels, the dominance criteria need to compare two linear functions. Because the linear functions have a limited domain, two line segments have to be considered to compare the reduced costs of two labels. Specifically, the dominance conditions must be able to handle the comparison of line segments which are restricted in domain by the quantity delivered to the split customer (between zero and $\phi(L)$) and in range by $c(L)$ and the minimum reduced cost that can be reached given $\phi(L)$, which is $c(L) - \phi(L)\pi^{1b}$. Desaulniers (2010) encounters the same issue for the SDVRPTW, and the author proposes the following sufficient dominance

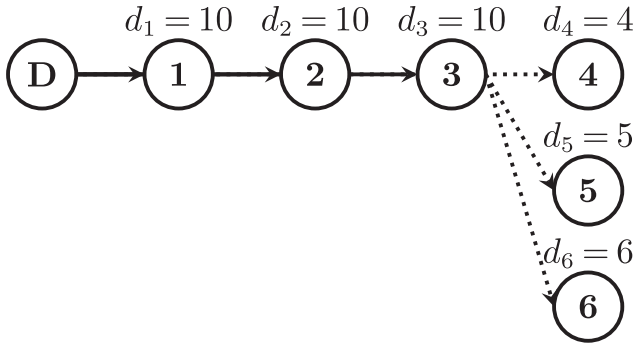
conditions to establish whether label L_1 dominates label L_2 associated with the same node:

- A1. $t(L_1) \leq t(L_2)$;
- A2. $q(L_1) \leq q(L_2)$;
- A3. $\mathbb{1}\{r(L_1) > -1\} \leq \mathbb{1}\{r(L_2) > -1\}$;
- A4. $\bar{V}(L_1) \subseteq \bar{V}(L_2)$;
- A5. $c(L_1) - \phi(L_1)\pi_{r(L_1)}^{1b} \leq c(L_2) - \phi(L_2)\pi_{r(L_2)}^{1b}$;
- A6. $c(L_1) - (q(L_2) - q(L_1))\pi_{r(L_1)}^{1b} \leq c(L_2)$;
- A7. $c(L_1) - (q(L_2) + \phi(L_2) - q(L_1))\pi_{r(L_1)}^{1b} \leq c(L_2) - \phi(L_2)\pi_{r(L_2)}^{1b}$;

in which condition A5 compares the minimums of both segments, and conditions A6 and A7 compare the costs of both paths at the lowest and highest loads of the path in label L_2 , respectively. These conditions prevent comparing crossing line segments; for details, we refer to Desaulniers (2010).

3.2.2. Pricing Algorithm 2 for MP1 (MP1-PA2). In MP1-PA1 (Section 3.2.1) in each label extension, a label with and a label without a split are explicitly created with at most one split per path. Hence, when a label with a split is created, the split customer is immediately determined for the resulting route. However, knowing the split customer is not necessary, it is possible instead to decide which customer to split when extending a path to the end node. Then, only a path with a total delivery quantity larger than the vehicle capacity will have a split, and any customer with a sufficiently high demand can be the split customer. The demand of a customer is sufficiently high if the demand is higher than the vehicle capacity shortage given the total demand in the path. For example, if $Q = 25$ and the total demand is 30, the capacity shortage is 5 units, and splitting a customer with demand 3 does not give a feasible path. Luo et al. (2016) come to a similar insight for the SDVRPTW. However, as explained in Section 3.2.1, zero deliveries are also necessary, and therefore, Luo et al. (2016) do not keep track of the vehicle capacity and, hence, are not able to exclude paths because vehicle capacity is violated. This leads to many very long routes, whereas by keeping track of the load, we can prevent many inefficiently long paths that would have many zero deliveries or, in our case, many outsourced units.

This leads us to a different solution method for the pricing problem (MP1-PA2). Instead of creating the split explicitly at a node during the label extension, a partial path is extended until it exceeds the vehicle capacity, after which a split is definitely needed. After exceeding the vehicle capacity, only customers with a small demand that respect an additional constraint can still be added to the path. To explain the additional constraint, consider Figure 1. Suppose the vehicle capacity is $Q = 25$, and there is a partial path depot-1-2-3,

Figure 1. Example of Possible Label Extensions

which exceeds the vehicle capacity with the extension to customer 3. The demand of each customer is indicated in the figure. The vehicle capacity is already exceeded by 5 units which have to be outsourced. Consider three possible extensions to customers 4, 5, and 6, respectively, with the indicated corresponding demands.

Suppose an extension is made to customer 6; then, the total load will be 36 units, and at least 11 units need to be outsourced. This implies that one customer's demand will be fully outsourced as well as one unit of another customer; hence, this path is not efficient. The same holds for an extension to customer 5, because exactly one customer's demand will be fully outsourced, in which case it is better not to visit the customer at all in this path. On the contrary, the extension to customer 4 is potentially efficient because the total load becomes 34 and, therefore, nine units have to be outsourced. Note that by Lemma 1 and the maximum demand in the path of 10, at most nine units can be outsourced. Hence, we make use of the customer with the highest demand currently in the path to determine whether an extension is still possible. Concluding, the additional constraint is that the total demand after adding a customer may not exceed vehicle capacity plus the highest demand currently in the path. Moreover, the quantity dedicated to this customer that can still be delivered by the private vehicle can be computed by vehicle capacity – current load + highest demand = 25 – 34 + 10 = 1.

Therefore, instead of explicitly creating labels with a split delivery, we propose an alternative labeling algorithm, in which the split customer and corresponding delivery quantity are determined in a post-processing step such that reduced costs are minimized. Note that if vehicle capacity is not exceeded, all units are delivered by the private vehicle. In the post-processing step, we know how many units need to be outsourced (the shortage in vehicle capacity). Then, it remains to decide which customer's demand to split. Jin, Liu, and Eksioglu (2008) have a similar issue for a variant of the SDVRP in which the number of vehicles to use is fixed up front, and they note that it is best to split the demand of the customer with the smallest

dual variable. Which customer to split in a path follows from Lemma 2, which can be proved by a simple interchange argument.

Lemma 2. *In an optimal solution to the pricing problem, each visited customer receives its full demand except for, at most, one customer with the smallest dual variable and demand higher than the shortage in vehicle capacity.*

Let a label L correspond to a partial path $p(L)$ in the graph G starting at the depot. For a type- k vehicle, associate the following attributes with a label L :

- $i(L)$ Last node visited in partial path $p(L)$,
- $c(L)$ Minimum reduced cost of partial path $p(L)$,
- $q(L)$ Load in partial path $p(L)$,
- $t(L)$ Ready time at node $i(L)$ when reached through partial path $p(L)$,
- $s(L)$ Indicates whether or not a split delivery is necessary in partial path $p(L)$,
if any ($\phi(L) = 0$ if $s(L) = 0$),
- $d_{\max}(L)$ Maximum demand over the customers in path $p(L)$, updated until vehicle capacity is exceeded,
- $V(L)$ Set of visited nodes along path $p(L)$.

For $d_{\max}(L)$, it is important to note that this value is no longer updated when vehicle capacity is already exceeded. Furthermore, let $\bar{V}(L)$ again denote the set of visited and unreachable nodes. Nodes can be unreachable because of time windows or vehicle capacity; an unreachable node j has one of the following properties: $t(L) + t_{i(L),j} > l_j$ or $s(L) = 1$ and $d_j > Q + d_{\max}(L) - q(L)$. If necessary, the customer with the lowest dual value that has a sufficiently high demand is split. Hence, the best candidate customer changes during the labeling algorithm, and the actual reduced cost of a path is not known during the labeling algorithm. Therefore, in the reduced cost $c(L)$, for all visited customers we account for the full demand d_i and subtract $d_i\pi_i^{1b}$. This results in keeping track of the minimum possible reduced cost of path $p(L)$. Note that in the case of a split delivery, this value $c(L)$ is lower than the actual reduced cost because the dual value is subtracted for too many units of demand. This needs to be accounted for when applying dominance rules. Also, at the end of the labeling algorithm, the correct reduced cost has to be computed for a column to decide whether adding it is actually efficient.

Suppose we extend a label L' along arc $(i(L'), j)$ to node $j \in V \setminus \bar{V}(L')$ to generate a new label L . The resources for the new label L are established as follows:

$$\begin{aligned} i(L) &= j, \\ c(L) &= c(L') + \bar{c}_{ij} - d_j\pi_j^{1b}, \\ q(L) &= q(L') + d_j, \\ t(L) &= \max\{t(L') + t_{ij}, e_j\}, \end{aligned}$$

$$\begin{aligned}
 s(L) &= s(L') + 1, \text{ if } q(L') < Q \wedge q(L) > Q, \\
 d_{\max}(L) &= \begin{cases} \max\{d_{\max}(L), d_j\} & \text{if } s(L') = 0, \\ d_{\max}(L') & \text{if } s(L') = 1, \end{cases} \\
 V(L) &= V(L') \cup \{j\}.
 \end{aligned}$$

An extension from label L' to label L with $i(L) = j$ is feasible if $j \notin \bar{V}(L')$ and $e_j \leq t(L) \leq l_j$. To discard labels, sufficient dominance conditions are formulated. If a path needs a split delivery, the reduced cost is a function of the quantity delivered to the split customer, with a multiplication factor equal to the corresponding dual variable. On the one hand, when deciding which customer to split, in a greedy way customers with the highest dual values will be served by the private vehicle because this results in the lowest reduced cost. On the other hand, the split customer must have sufficiently high demand (higher than the shortage in vehicle capacity) to obtain a feasible route. The dominance rules, again, must compare segments of reduced cost functions, but in this case it is not immediately clear what the slope and ranges of these segments are. Therefore, to decide whether a label L_1 dominates L_2 , we consider the “worst” case in terms of cost and range for L_1 and the “best” case for L_2 , which corresponds to the situation where L_1 is least likely to dominate L_2 . For the worst case for L_1 , determine the highest dual value over the customers in the path that have a sufficiently high demand (i.e., $\hat{\pi}_1 = \max_{i \in V(L_1): d_i > q(L_1) - Q} \pi_i^{1b}$). Similarly, for L_2 , determine the lowest dual value (i.e., $\tilde{\pi}_2 = \min_{i \in V(L_2): d_i > q(L_2) - Q} \pi_i^{1b}$). Define $\hat{\pi}_1 = 0$ and $\tilde{\pi}_2 = 0$ if a split is not necessary in the path. Note that if an extension of L_2 contains a customer with a lower dual value, this value cannot be lower than the lowest dual value of the extension of L_1 , in which case the slopes of both reduced cost functions are the same. Moreover, for both labels, the demand of the split customer is set to the highest demand of a customer in each path ($d_{\max}(L)$).

Define $\hat{q}(L) = q(L) - \mathbb{1}\{s(L) = 1\}d_{\max}(L)$, $\hat{c}(L_1) = c(L_1) + \mathbb{1}\{s(L_1) = 1\}d_{\max}(L_1)\hat{\pi}_1$, and $\hat{c}(L_2) = c(L_2) + \mathbb{1}\{s(L_2) = 1\}d_{\max}(L_2)\tilde{\pi}_2$. Also, for both labels, compute the maximum quantity delivered by the private vehicle of the maximum demand (i.e., $\phi(L) = Q - q(L) + d_{\max}(L)$). The line segment of the reduced cost function extends from a zero delivery to a delivery of $\phi(L)$ units to the split customer (i.e., has a domain of zero to $\phi(L)$). See Figure 2 for an example of segments for labels L_1 and L_2 .

Consider conditions B1–B7 to dominate label L_2 by label L_1 which correspond to the same customer:

- B1. $t(L_1) \leq t(L_2)$;
- B2. $\hat{q}(L_1) \leq \hat{q}(L_2)$;
- B3. $s(L_1) \leq s(L_2)$;
- B4. $\bar{V}(L_1) \subseteq \bar{V}(L_2)$;
- B5. $\hat{c}(L_1) - \phi(L_1)\hat{\pi}_1 \leq \hat{c}(L_2) - \phi(L_2)\tilde{\pi}_2$;

- B6. $\hat{c}(L_1) - (\hat{q}(L_2) - \hat{q}(L_1))\hat{\pi}_1 \leq \hat{c}(L_2)$;
- B7. $\hat{c}(L_1) - (\hat{q}(L_2) + \phi(L_2) - \hat{q}(L_1))\hat{\pi}_1 \leq \hat{c}(L_2) - \phi(L_2)\tilde{\pi}_2$.

Proposition 1. *Conditions B1–B7 are sufficient conditions to dominate label L_2 by label L_1 .*

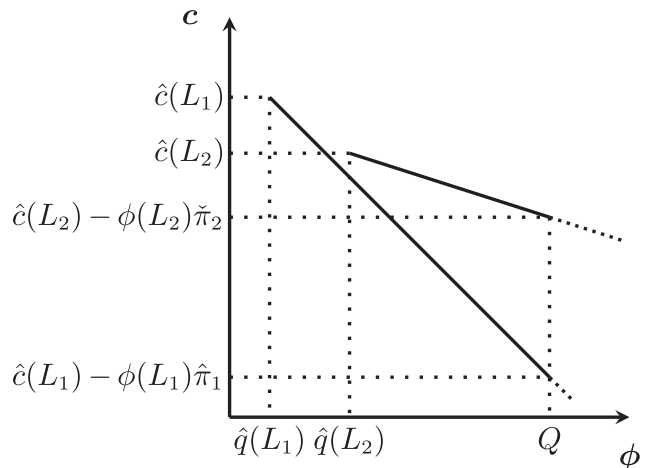
Proof. Any extension of label L_2 is feasible for label L_1 by conditions B1–B4, in which B1, B2, and B4 are the same as in Desaulniers (2010). Note that in the case of a split delivery in both labels L_1 and L_2 , condition B2 considers the load without the maximum demand, which allows for splitting the customer with the highest demand in extensions of both paths.

If neither label has a split delivery, the dominance conditions are the same as in Desaulniers (2010). If both labels have a split delivery, for label L_1 , $d_{\max}(L_1)$ units are outsourced maximally. If the corresponding customer would have dual value $\hat{\pi}_1$, the reduced cost would be highest and the chance of dominating label L_2 is lowest. Therefore, we create the artificial segment for label L_1 from $(\hat{q}(L_1), \hat{c}(L_1))$ to $(Q = \hat{q}(L_1) + \phi(L_1), \hat{c}(L_1) - \phi(L_1)\hat{\pi}_1)$. Note that the choice of customer demand to create the artificial segment does not impact the lowest point of the slope which is used in condition B5.

For label L_2 , with a similar reasoning except for using the lowest possible reduced cost, we create an artificial segment from $(\hat{q}(L_2), \hat{c}(L_2))$ to $(Q = \hat{q}(L_2) + \phi(L_2), \hat{c}(L_2) - \phi(L_2)\tilde{\pi}_2)$. The dominance conditions by Desaulniers (2010) are applied to these artificial segments (B5–B7). By construction, using these artificial segments makes it least likely that label L_1 will dominate L_2 , and hence, if dominance is established by conditions B1–B7, label L_2 will not result in a better path, which completes the proof. \square

Note that each resulting path in the labeling algorithm now results in, at most, one route (column) with negative reduced cost for MP1 because, in the end, we only split the customer with the lowest dual

Figure 2. Reduced Cost as a Function of the Delivered Quantity



value. One could create multiple columns by splitting different customers if the corresponding dual variables still lead to a negative reduced cost column. We do not generate multiple columns because the cost of outsourcing is the same for each customer; therefore, the objective value of MP1 remains the same no matter which customer receives a split delivery.

3.2.3. Subset-Row Inequalities. SR inequalities are introduced by Jepsen et al. (2008) for the VRPTW and are Chvátal-Gomory rank 1 cuts. We can apply these valid inequalities to MP1. Remember that a_{ip} is the number of times customer i is visited in $p \in \Omega$. The SR inequalities for a subset of nodes $S \subseteq V$ and an integer $0 < \kappa \leq |S|$ can be formulated as follows:

$$\sum_{p \in \Omega} \left[\frac{1}{\kappa} \sum_{i \in S} a_{ip} \right] y_p \leq \left\lfloor \frac{|S|}{\kappa} \right\rfloor. \quad (4)$$

Separation of these valid inequalities is NP-complete. As suggested by, for example, Jepsen et al. (2008), we enumerate all inequalities for subsets of customers of size three ($|S| = 3$) with $\kappa = 2$. As clearly stated by Spliet and Desaulniers (2015), these valid inequalities ensure that for every triplet of customers, at most one route can be selected that contains more than one of these customers. Because these inequalities are defined on the master problem variables, they change the structure of the pricing problem. Let $\xi_I < 0$ be the dual variable corresponding to a valid inequality of type (4) for subset $S_I \subseteq V$. If a column is generated that contributes to the valid inequality, i.e., for every κ customers in S_I visited in the path, ξ_I is subtracted from the reduced cost of the path. However, only when ending a path in the pricing problem (extend to the end node), one knows exactly what the contribution of the valid inequalities to the reduced cost is. As described by Jepsen et al. (2008), the contribution of the SR inequalities can be accounted for in the costs to handle the SR inequalities in the dominance conditions.

3.3. Master problem 2 (MP2)

In MP1, the quantity delivered by the private vehicle to each visited customer in a route is decided upon in the pricing problem. The outsourced quantities are handled in the master problem by the β variables. However, given a route with a split delivery retrieved from the pricing problem, the number of units delivered by the private vehicle to each customer is already known and, hence, the number of units that are outsourced and the corresponding costs. Therefore, we introduce MP2 in which the (outsourced) delivery quantities are not modeled explicitly via decision variables. For each customer, there are two options: either the customer is visited by a route, and from the pricing problem we know both the private

and outsourced delivery quantities; or the customer is not visited at all and demand is fully outsourced. The latter case should be explicitly modeled in MP2.

Let Λ_k be the set of routes and associated delivery quantities for vehicle type $k \in K$ and outsourcing quantities, and let $\Lambda = \bigcup_{k \in K} \Lambda_k$. Let $p \in \Lambda$ represent a route visiting a set of customers, the corresponding delivery quantities by the private vehicle, and the outsourced quantity ζ_p . Note that for a column p , knowing to which customer ζ_p belongs is not required. Let c_p , $p \in \Lambda$, be the total routing and outsourcing cost ($\zeta_p v$). Associate parameter a_{ip} with $p \in \Lambda$, representing the number of visits to customer $i \in V$ in the route. Define binary decision variables z_p which indicate whether $p \in \Lambda$ is in the solution of MP2, and define nonnegative decision variables w_i indicating whether a customer $i \in V$ is fully outsourced or not. MP2 is formulated as follows:

$$\min \sum_{p \in \Lambda} c_p z_p + \sum_{i \in V} w_i d_i v \quad (5a)$$

$$\text{s.t.} \sum_{p \in \Lambda} a_{ip} z_p + w_i = 1, \quad \forall i \in V, \quad (5b)$$

$$\sum_{p \in \Lambda_k} z_p \leq m_k, \quad \forall k \in K, \quad (5c)$$

$$w_i \geq 0, \quad \forall i \in V_0, \quad (5d)$$

$$z_p \in \{0, 1\}, \quad \forall p \in \Lambda. \quad (5e)$$

The objective function (5a) minimizes total routing and outsourcing costs, in which the outsourcing costs consist of the costs for split customers given by the pricing problem and costs for customers for which demand is fully outsourced. Constraints (5b) make sure that either a customer is visited by a private vehicle, potentially with a split delivery, or the customer's demand is fully outsourced. Constraints (5c) limit the number of vehicles used per type, and the domains of the decision variables are restricted in constraints (5d) and (5e). Note that the w variables represent a binary decision but can be added to MP2 as nonnegative variables because, if the z variables are integer, the w variables must be integer as well.

Associate dual variables $\mu_i^{5b} \in \mathbb{R}$ and $\mu_k^{5c} \leq 0$ with constraints (5b) and (5c), respectively. The reduced cost of a column $p \in \Lambda$ for a vehicle of type $k \in K$ can be expressed as follows:

$$\bar{c}_p = f_k + \zeta_p v + \sum_{(i,j) \in A} (c_{ij} - \mu_i^{5b}) x_{ijp} - \mu_k^{5c}, \quad (6)$$

in which ζ_p is a nonnegative variable being the quantity outsourced, and x_{ijp} are integer variables counting the number of times arc (i, j) is traversed in the route of $p \in \Lambda$. For ease of notation, define

$$\bar{c}_{ij} = c_{ij} - \mu_i^{5b} + \mathbb{1}\{i = 0\}(f_k - \mu_k^{5c}), \quad (7)$$

in which $\mu_0^{5b} = 0$ and the vehicle costs are accounted for in the outgoing arcs of the depot.

3.3.1. Pricing Algorithm 1 for MP2 (MP2-PA1). The first pricing problem for MP2 (MP2-PA1) is based on the same reasoning as MP1-PA1 as described in Section 3.2.1; therefore, we only highlight the differences here. Note that there is a significant difference in the reduced cost functions between the two master problems. For MP1, the reduced cost is a function of the quantities delivered by the private vehicle, whereas for MP2, the reduced cost is a function of the outsourced quantity. Accordingly, for MP1-PA1, the dual variable π_i^{1b} should be *deducted* from the reduced cost for each privately delivered unit, but this number of units is unknown for the split customer during the labeling algorithm. Conversely, for MP2-PA1, the outsourcing cost v should be *added* to the reduced cost for each outsourced unit, but the outsourced quantity is unknown during the labeling algorithm.

MP2-PA1 uses resources $i(L)$, $q(L)$, $t(L)$, $r(L)$, $\phi(L)$, $V(L)$, and $\bar{V}(L)$ as defined for MP1-PA1 in Section 3.2.1. Only $c(L)$, the reduced cost, is redefined. The number of outsourced units cannot be established until the path is finished. Moreover, the reduced cost is lowest if all units are delivered by the private vehicle (no outsourcing costs). Therefore, during this labeling algorithm, we keep track of the minimum possible reduced cost. This gives the following definition of the resource:

$c(L)$ Minimum reduced cost of partial path $p(L)$
 (i.e., all units delivered by private vehicle),

which is updated as follows for an extension of a label L' along arc $(i(L'), j)$ to a node $j \in V \setminus V(L')$ to generate a new label L :

$$c(L) = c(L') + \bar{c}_{ij}.$$

To reduce the number of labels, dominance rules can be applied to discard labels. The reduced cost of a path depends on the outsourced quantity or, equivalently, on the quantity delivered by the private vehicle to the split customer. Then we can express the reduced cost of a path in label L as a function of $\phi(L)$ as follows:

$$\bar{c}_p = f_k + (d_{r(L)} - \phi(L))v + \sum_{(i,j) \in A} (c_{ij} - \mu_j^{5b})x_{ijp} - \mu_k^{5c}, \quad (8)$$

in which x_{ijp} again indicates the number of times arc (i, j) is in the path, and recall that $r(L)$ is the split customer. The functional form of the rewritten reduced cost function is the same as for MP1. Therefore, to apply dominance rules to compare labels, again we have to compare segments of reduced cost functions. Similarly to Desaulniers (2010) and MP1, sufficient

dominance conditions for dominance of label L_1 over L_2 associated with the same node are given by

- C1. $t(L_1) \leq t(L_2)$;
- C2. $q(L_1) \leq q(L_2)$;
- C3. $s(L_1) \leq s(L_2)$;
- C4. $\bar{V}(L_1) \subseteq \bar{V}(L_2)$;
- C5. $c(L_1) + d_{s(L_1)}v - \phi(L_1)v \leq c(L_2) + d_{r(L_2)}v - \phi(L_2)v$;
- C6. $c(L_1) + d_{s(L_1)}v - (q(L_2) - q(L_1))v \leq c(L_2) + d_{r(L_2)}v$;
- C7. $c(L_1) + d_{s(L_1)}v - (q(L_2) + \phi(L_2) - q(L_1))v \leq c(L_2) + d_{r(L_2)}v - \phi(L_2)v$,

in which $d_r(L) = 0$ and $\phi(L) = 0$ if there is no split delivery in the path at label L . In conditions C5, C6, and C7, the minimum reduced cost is increased with the maximum outsourcing cost, and subsequently, the unit outsourcing cost is deducted for the units that are not outsourced. Additionally, note that the slopes of the compared segments are equal (v); therefore, we can discard condition C7 because it is redundant with conditions C2, C5, and C6.

3.3.2. Pricing Algorithm 2 for MP2 (MP2-PA2). As for MP1-PA2 in Section 3.2.2, it can also be observed for MP2 that knowing which customer to split is not necessary during the labeling algorithm. Moreover, for MP2, even in the master problem this information is not necessary because only information on which customers are visited is needed. Therefore, MP1-PA2 can be adjusted for MP2 (MP2-PA2), and again, we only highlight the differences here.

MP2-PA2 uses resources $i(L)$, $c(L)$, $q(L)$, $t(L)$, $s(L)$, $d_{\max}(L)$, $V(L)$, and $\bar{V}(L)$ as defined for MP1-PA2 in Section 3.2.2. Additionally, $\phi(L)$ is defined as follows:

$\phi(L)$ The maximum quantity delivered to a split customer by the private vehicle in partial path $p(L)$, assuming that the customer with the highest demand is split.

Although the definition of $c(L)$ is the same, the computation is different because of the different reduced cost function. Therefore, for an extension of a label L' along arc $(i(L'), j)$ to node $j \in V \setminus V(L')$ to generate a new label L , resources $c(L)$ and $\phi(L)$ are updated as follows:

$$\begin{aligned} \zeta(L) &= c(L') + \bar{c}_{ij}, \\ \phi(L) &= Q - q(L) + d_{\max}(L). \end{aligned}$$

To apply dominance, a reduced cost function similar to that for MP2-PA1 can be used in which $d_{s(L)}$ is replaced by d_{\max} . Again, segments of reduced cost functions have to be compared. Define $\hat{q}(L) = q(L) - \mathbb{1}\{s(L) = 1\}d_{\max}(L)$ and $\hat{c}(L) = c(L) + \mathbb{1}\{s(L) = 1\}d_{\max}(L)v$. Sufficient dominance conditions for dominance of label L_1 over L_2 associated with the same node are given by

- D1. $t(L_1) \leq t(L_2)$;
- D2. $\hat{q}(L_1) \leq \hat{q}(L_2)$;

- D3. $s(L_1) \leq s(L_2)$;
 D4. $\bar{V}(L_1) \subseteq \bar{V}(L_2)$;
 D5. $\hat{c}(L_1) - \phi(L_1)v \leq \hat{c}(L_2) - \phi(L_2)v$;
 D6. $\hat{c}(L_1) - (\hat{q}(L_2) - \hat{q}(L_1))v \leq \hat{c}(L_2)$;
 D7. $\hat{c}(L_1) - (\hat{q}(L_2) + \phi(L_2) - \hat{q}(L_1))v \leq \hat{c}(L_2) - \phi(L_2)v$.

These conditions are similar to the conditions for MP1-PA2. Because the slope of the segments is equal for both labels, condition D7 is redundant, analogous to MP2-PA1.

3.3.3. Generalized Subset-Row Inequalities. As in Dabia, Lai, and Vigo (2019), for MP2 we can still apply the SR inequalities as described in Section 3.2.3; however, these cuts do not make use of the fact that part of the demand can be outsourced to the common carrier. Dabia, Lai, and Vigo (2019) propose so-called generalized SR inequalities for this variant of the problem. These can be applied to the MP2 as well because constraint (5b) coincides with the first constraint of the second formulation in Dabia, Lai, and Vigo (2019).

Let $d(S) = \sum_{i \in S} d_i$ be the sum of the demand for subset of customers $S \subseteq V_0$. Dabia, Lai, and Vigo (2019) observe that splitting $d(S)$ in packages of size κ will result in a number of privately delivered packages in an integer solution that is less than the total number of packages minus the outsourced packages of the customers in S . For a subset of customers $S \subseteq V_0$ and $0 < \kappa < d(S)$, they propose the following valid inequalities:

$$\sum_{p \in \Omega} \left\lfloor \frac{1}{\kappa} \sum_{i \in S} d_i a_{ip} \right\rfloor z_p + \sum_{i \in S} \left\lfloor \frac{d_i}{\kappa} \right\rfloor w_i \leq \left\lfloor \frac{d(S)}{\kappa} \right\rfloor. \quad (9)$$

Dabia, Lai, and Vigo (2019) give two propositions to establish dominance rules that take the generalized SR inequalities into account, in which the second proposition contains strengthened dominance rules compared with the first proposition. If the generalized SR cuts are combined with ng-paths (see Section 4.2), it is not possible to use the strengthened version because it makes use of the demand of the set of reachable customers, which is changing during the algorithm when ng-paths are used. Therefore, similar dominance conditions as in the first proposition in Dabia, Lai, and Vigo (2019) are used for the VRPPO.

3.4. Model Extensions

In the problem described in Section 3, we assumed that the cost of outsourcing is a customer-independent fixed fee per unit v . Although we designed the four solution algorithms to solve the VRPPO for a fixed fee per unit cost structure, several of our algorithms can be easily adjusted for handling alternative cost structures. In particular, for MP1-PA1, MP1-PA2, and MP2-PA1, a customer-dependent cost structure, in which the fixed fee per unit can be different per

customer ($v_i \neq v_j$), can be easily integrated. In MP1, the outsourcing cost is only present in the master problem; hence, this can be easily changed to a customer-dependent fee without algorithmic changes in MP1-PA1 and MP1-PA2. However, in MP1-PA2, currently one column at most is generated per iteration because creating multiple columns would result in columns with the same costs. In case of customer-dependent outsourcing costs, it can be beneficial to generate multiple columns per pricing problem iteration. The reduced cost for MP2 does depend on the outsourcing cost. In MP2-PA1, the split customer, and hence its corresponding outsourcing cost, is decided during the labeling algorithm and, therefore, the customer-dependent outsourcing cost can easily be accounted for. On the contrary, for MP2-PA2, it is not straightforward to include customer-dependent outsourcing costs. This is caused by the fact that the reduced cost function is dependent on both the outsourced quantity and the outsourcing cost per unit. Hence, with customer-dependent outsourcing costs, in the dominance conditions both the range and the slope of the segments are unknown. Dominance conditions, therefore, have to be customized for MP2-PA2 to include customer-dependent outsourcing costs.

4. Implementation

4.1. Branching

Let x^* be the current fractional solution expressed in the arc flow variables in which x_{ij} is the arc flow variable of arc $(i, j) \in A$ of the underlying compact formulation. To result in a feasible solution, first, the algorithm branches on the total number of vehicles, $\sum_{k \in K} \sum_{i \in V_0} x_{0i}^k$. In the results section, we also consider discarding this first branching option, and start with the second branching strategy immediately. Second, the algorithm branches on the number of vehicles per vehicle type, $\sum_{i \in V_0} x_{0i}^k$. If, for all vehicle types, the number used is integer, branch on the edge variables $x_{ij}^k + x_{ji}^k$ for some vehicle type $k \in K$. The algorithm looks for i, j pairs, such that $x_{ij}^k + x_{ji}^k$ is close to 0.5 and imposes the branches $x_{ij}^k + x_{ji}^k \leq 0$ and $x_{ij}^k + x_{ji}^k \geq 1$. Finally, branching on one fractional arc x_{ij}^k is performed for some arc $(i, j) \in A$ and vehicle type $k \in K$. As in Dabia, Lai, and Vigo (2019), we apply strong branching, which means that potential branches are evaluated quickly to decide which one to continue with first. In this case, we solve the linear relaxation with only the columns already generated in the column-generation algorithm. This way, for each branching candidate, we estimate a lower bound in the two child nodes. The algorithm chooses the branch that maximizes the lower bound in the weakest of the two child nodes. In the first 15 nodes of the branch-and-bound tree, we consider 30 branch candidates, and 15 branch candidates in the remaining nodes.

4.2. Acceleration Techniques

To speed up the labeling algorithm, we implement bidirectional labeling. This means that paths are created both forward from the starting node and backward from the ending node up to some bound in one of the resources which are, afterward, merged to feasible paths. The dominance rules are also applied to backward labels. We apply an advanced version of bidirectional labeling in which the halfway point of the resource (up to which labels are extended) is determined dynamically during the labeling algorithm (Tilk et al. 2017). As the boundary resource for the VRPPO, we use the time.

We implemented the ng-path relaxation introduced by Baldacci, Mingozzi, and Roberti (2011), which allows for cycles in the labeling algorithm instead of finding only elementary routes. For each customer $i \in V_0$, define a neighborhood NG_i which contains node i itself and, at most, $b \in V_0$ other nodes that are closest to i . An ng-path can visit a customer i more than once only if at least one node $j \notin NG_i$ is visited between two visits to i . Allowing for such paths to be generated and added to the master problem may yield weaker lower bounds. However, the pricing problem may become easier to solve if b is sufficiently small. In the labeling algorithm, the number of times a customer is visited should now be counted instead of whether a customer is visited, and different label extensions are now possible with the ng-path relaxation. The labeling algorithms are adjusted accordingly.

The exact labeling algorithms as proposed in Sections 3.2.1, 3.2.2, 3.3.1, and 3.3.2 may be time consuming to fully execute. Therefore, before calling an exact pricing algorithm, we first apply a heuristic labeling algorithm to more quickly generate negative reduced cost columns. The exact labeling algorithm is only called when the heuristic does not find any negative reduced cost paths. The heuristic performs the labeling algorithm on a reduced graph that keeps for each node, at most, the k outgoing arcs with the smallest reduced cost. The number of kept arcs is increased to $2k$, then to $4k$ until some bound (in our case, set to 20) is reached.

5. Results

In the following sections, we will first compare, for both master problems, the two pricing algorithms and determine which algorithm performs best. Second, for the chosen algorithm, the performance and the cost improvement of the VRPPO over the VRPPC on two sets of instances will be investigated. The first set of instances is derived from the instances used by Dabia, Lai, and Vigo (2019) for the VRPPC (referred to as instance set \mathcal{A}). The second set of instances was constructed for the SDVRPTW and used by, for example, Desaulniers (2010) (referred to as instance set

\mathcal{B}). We use two sets of instances because the benefit of outsourcing part of a customer's demand may differ for instances originally designed to examine either the impact of outsourcing in the absence of split delivery (set \mathcal{A}) or the reverse (set \mathcal{B}).

The instances in set \mathcal{A} were originally constructed by Liu and Shen (1999) from the Solomon (1987) instances by adding information on heterogeneous vehicles. There are six types of instances, based on topology (R for randomly dispersed customers, C for clustered customers, and RC for a combination) and time window size (type 1 for tight time windows and type 2 for wide time windows). The instances contain heterogeneous vehicles, and there are three vehicles per vehicle type. The algorithms are tested on instances with 25, 50, and 100 customers. As in Dabia, Lai, and Vigo (2019) and Liu and Shen (1999), three different vehicle cost levels are considered. Types a, b , and c have high, medium, and low vehicle costs, respectively. We refer to Liu and Shen (1999) for more details on the vehicle compositions and vehicle fixed costs. The outsourcing cost is derived from Dabia, Lai, and Vigo (2019). We do not consider all unit discounts for outsourcing, as in Dabia, Lai, and Vigo (2019), but rather a fixed fee per unit outsourced as argued in Section 3. Therefore, we consider the highest cost and the lowest cost from Dabia, Lai, and Vigo (2019) in these experiments to examine the impact of different outsourcing cost levels. This means we set $v = 5.00, 3.50$ for R instances; $v = 2.00, 0.50$ for C instances; and $v = 3.50, 2.00$ for RC instances. There are 56 different instances, each with three vehicle costs, for the three different numbers of customers and for two outsourcing cost levels; this gives 1,008 instances in total.

The instances in set \mathcal{B} are also derived from the Solomon (1987) instances by allowing split deliveries. These instances contain homogeneous vehicles without fixed vehicle costs. Because the original vehicle capacity in the Solomon instances is relatively high, the vehicle capacity for the SDVRP is set to $Q = 30, 50, 100$, respectively. Based on preliminary experiments, for the VRPPO we only use $Q = 30, 50$ because results for $Q = 50$ and $Q = 100$ are comparable in terms of cost improvements. Desaulniers (2010) points out that because demand is randomly generated between 1 and 50, split deliveries can be necessary in the SDVRP for $Q = 30$. This implies that outsourcing can be necessary for the VRPPO and the VRPPC. The considered outsourcing costs are the same as for instance set \mathcal{A} as described in the previous paragraph. Again, there are 56 different instances, 2 vehicle capacities, 3 different numbers of customers, and 2 outsourcing cost levels, resulting in 672 instances in total.

For all instances, the master problems are initialized with a column that represents the solution in which delivery of all demand is fully outsourced. The

cost of this column provides a valid upper bound on the solution. The branch-price-and-cut algorithms are implemented using Java and Gurobi 8.0. All tests are performed on a desktop computer running Windows 10, using a single core from an eight core Intel(R) Core(TM) i7-6700K processor clocked at 4.00GHz with 24 GB RAM. The maximum computation time is one hour per instance.

The outline of the computational experiments is as follows. First, in Section 5.1, we compare the two master problems for each of the two pricing algorithms on a subset of the instances in set \mathcal{A} to evaluate their performance. Second, Section 5.2 presents aggregated results of extensive tests on both instance sets \mathcal{A} and \mathcal{B} for the selected algorithm from Section 5.1. Next, in Section 5.3, we compare for both instance sets the results of the VRPPO and the VRPPC to gain insight into the potential cost improvements of allowing a part of the demand to be outsourced. Finally, Section 5.4 presents some figures to gain insight into the structure of the solutions of the VRPPO. The results per instance are given in Online Appendix 6.

5.1. Comparing the Algorithms

For both MP1 (Section 3.2) and MP2 (Section 3.3), two algorithms have been proposed for the pricing problems (PA1 and PA2). Because the pricing algorithms differ in several aspects, it is hard to assess their performance on theoretical grounds. The first pricing algorithm can create multiple labels per node extension, which results in many labels, but vehicle capacity cannot be exceeded. On the contrary, the second pricing algorithm creates at most one label per node extension but can result in longer paths because vehicle capacity can be exceeded. Furthermore, because artificial segments for the reduced cost function are necessary to apply dominance in the second pricing algorithm the dominance criteria cannot be compared between the pricing algorithms. Moreover, for the second pricing algorithm a postprocessing step is required. Therefore, it cannot be stated up front which algorithm

will perform best in terms of running time and number of instances that can be solved. Hence, we test all four algorithms (MP1-PA1, MP1-PA2, MP2-PA1, and MP2-PA2) for multiple parameter settings on a subset of the instances in set \mathcal{A} .

Preliminary experiments showed that the differences between algorithms and parameter settings are quite substantial. To find the best performing algorithm (master problem, pricing algorithm and parameter setting), the R, C, and RC instances with time window type 1 (tight time windows) were selected, with 25 customers, for vehicle cost a , and with a high outsourcing cost to test on. This resulted in a set of 29 instances. Next, we ran the algorithms on 11 additional instances with time window type 2 (25 customers, vehicle cost a , high outsourcing cost) that are easier to solve compared with other instances with type 2 time windows.

In these experiments, three parameters that are likely to have an impact on the performance of the algorithms are evaluated. These are indicators signaling if branching on the total number of vehicles is used, the maximum number of active (generalized) SR inequalities at any point during the execution of the algorithm (value 30 or 40), and the size of the neighborhood of the ng-paths (value 7, 8, or 9). Column “S” in Table 1 indicates the results for eight different scenarios. The values of the parameters are indicated in the columns “Branch,” “#SR,” and “NG,” respectively. For each algorithm, per parameter setting, the average CPU time for solving the instances with time window type 1 to optimality (T1(s)), the number of solved instances out of the 40 instances (#Opt.), and the average CPU time for solving the 40 instances to optimality (T(s)) are given.

The first obvious conclusion is that PA2 performs much better than PA1 for both master problems. The running time with PA1 for time window type 1 instances is much higher than for PA2 (column T1(s)), and PA1 solves, at most, 37 out of 40 instances (column #Opt.). Comparing MP1-PA2 and MP2-PA2

Table 1. Comparison Algorithms for Different Parameter Settings

| S | Branch | #SR | NG | Master problem 1 | | | | | | Master problem 2 | | | | | |
|---|--------|-----|----|------------------|-------|------|-----------|-----------|------------|------------------|-------|------|-----------|-----------|------------|
| | | | | PA1 | | | PA2 | | | PA1 | | | PA2 | | |
| | | | | T1(s) | #Opt. | T(s) | T1(s) | #Opt. | T(s) | T1(s) | #Opt. | T(s) | T1(s) | #Opt. | T(s) |
| 1 | 1 | 40 | 7 | 117 | 37 | 297 | 31 | 40 | 195 | 103 | 37 | 290 | 32 | 40 | 185 |
| 2 | 1 | 40 | 8 | 110 | 37 | 304 | 29 | 39 | 139 | 84 | 37 | 297 | 22 | 39 | 86 |
| 3 | 1 | 40 | 9 | 110 | 37 | 320 | 31 | 39 | 138 | 99 | 36 | 235 | 21 | 39 | 87 |
| 4 | 0 | 40 | 7 | 95 | 37 | 273 | 24 | 39 | 115 | 94 | 37 | 281 | 22 | 40 | 186 |
| 5 | 0 | 40 | 8 | 82 | 37 | 283 | 25 | 38 | 56 | 71 | 37 | 282 | 20 | 39 | 100 |
| 6 | 0 | 30 | 7 | 122 | 37 | 295 | 30 | 40 | 173 | 146 | 37 | 321 | 28 | 40 | 189 |
| 7 | 0 | 30 | 8 | 98 | 37 | 293 | 25 | 39 | 123 | 98 | 37 | 302 | 25 | 39 | 94 |
| 8 | 1 | 30 | 7 | 175 | 37 | 336 | 35 | 40 | 174 | 120 | 37 | 301 | 36 | 40 | 167 |

suggests that the performance of these algorithms is quite similar, as for both MP1 and MP2 all 40 instances are solved in three and four scenarios, respectively (indicated in bold). Because Scenario 4 for MP2-PA2 clearly gives the lowest running time for the type 1 instances and does not perform worst on all 40 instances, it is selected for the remaining experiments.

5.2. Aggregated Results for VRPPO

Table 2 reports on results on the performance of MP2-PA2 for instance set \mathcal{A} , aggregated over time window type and vehicle costs. Table 3 gives the results for instance set \mathcal{B} , aggregated over time window type and vehicle capacity. Presented in both tables for each topology R, C, and RC (“Topology,” number of instances between brackets) and per instance size (N) are the number of instances solved to optimality (#Opt.), the number of solved instances in which the solution consists of outsourcing all demand (#All out.), the average computation time in seconds ($T(s)$), the average size of the branch-and-bound tree of the instances solved to optimality (Tree), and the average integrality gap (Gap(%)). The integrality gap indicates the percentage difference between the optimal integer solution (IP) and the solution of the linear relaxation at the root node (LB_{root}), which is calculated by $(IP - LB_{root})/IP$. Remember that if, in a solution, all demand is outsourced, this solution is the initial solution of the master problem.

5.2.1. Instance Set \mathcal{A} . Instances in set \mathcal{A} are easier to solve for low outsourcing costs than for high outsourcing costs as shown by the number of instances solved, the computation times, and the size of the branch-and-bound tree indicated in Table 2. As can be expected, the number of solved instances decreases if the number of customers increases. The number of solved instances is approximately the same for high and low outsourcing costs for the R instances. For the C and RC instances, lower outsourcing costs allow for solving more instances to optimality. This observation can be explained by the fact that, for low outsourcing

costs, the optimal solution is to outsource regularly all demand, which is a relatively easy solution to find because it is the initial solution. On average, all instances are solved within 17 minutes, with more than half of the instances being solved within 10 seconds; however, for some instances, the optimal solution is only found very close to the time limit (see Online Appendix 6 for the results per instance). For all solved instances, the integrality gap is low with all averages below 1.5% and a maximum of 4.40%.

5.2.2. Instance Set \mathcal{B} . Table 3 gives the performance results on the VRPPO for instance set \mathcal{B} aggregated on vehicle capacity and time window type. The time to solve these instances to optimality is smaller than those in set \mathcal{A} , probably because the instances in set \mathcal{B} do not contain heterogeneous vehicles and, therefore, less pricing problems need to be solved. As a result, all 25 and 50 customer instances are solved to optimality. The difference in solution time between the instances with high and low outsourcing costs is smaller for these instances than those in set \mathcal{A} . Again, instances with low outsourcing costs regularly result in solutions in which all demand is outsourced, for both C and RC instances. For the instances in set \mathcal{B} , the RC instances are not harder to solve than the R and C instances as opposed to set \mathcal{A} . On average, the instances in set \mathcal{B} are solved within 6 minutes, with more than three-quarters of them being solved within 10 seconds (see Online Appendix 6). Integrality gaps for the instances in set \mathcal{B} are even lower than for those in set \mathcal{A} , with all averages being below 1.5% and with a maximum gap of only 3.23%.

5.3. VRPPO vs. VRPPC

In Sections 5.3.1 and 5.3.2, the solutions of the VRPPO are compared with the solutions of the VRPPC on the total costs for instance sets \mathcal{A} and \mathcal{B} , respectively. In Section 5.4, we explore the impact of allowing splitting on the structure of the routes. The solutions of the VRPPC are obtained by running the algorithm by Dabia, Lai, and Vigo (2019) with the fixed-fee

Table 2. Aggregated Results on Instance Set \mathcal{A}

| Topology | N | High outsourcing cost | | | | | Low outsourcing cost | | | | |
|----------|-----|-----------------------|-----------|--------|------|--------|----------------------|-----------|--------|------|--------|
| | | #Opt. | #All out. | $T(s)$ | Tree | Gap(%) | #Opt. | #All out. | $T(s)$ | Tree | Gap(%) |
| R (69) | 25 | 43 | 0 | 29 | 1.7 | 0.13 | 42 | 0 | 14 | 0.7 | 0.05 |
| | 50 | 26 | 0 | 485 | 12.2 | 0.44 | 30 | 0 | 391 | 5.7 | 0.26 |
| | 100 | 3 | 0 | 78 | 10.7 | 0.13 | 4 | 0 | 299 | 15.0 | 0.11 |
| C (51) | 25 | 33 | 17 | 273 | 2.9 | 1.12 | 51 | 51 | 0 | 0 | 0.00 |
| | 50 | 26 | 16 | 457 | 16.6 | 0.62 | 51 | 51 | 27 | 0 | 0.00 |
| | 100 | 25 | 15 | 354 | 0.9 | 0.01 | 48 | 48 | 110 | 0 | 0.00 |
| RC (48) | 25 | 35 | 0 | 127 | 1.1 | 0.14 | 40 | 15 | 106 | 0.1 | 0.06 |
| | 50 | 18 | 0 | 963 | 99.2 | 1.43 | 26 | 12 | 338 | 7.8 | 0.68 |
| | 100 | 9 | 0 | 870 | 6.7 | 0.08 | 17 | 9 | 671 | 4.8 | 0.08 |

Table 3. Aggregated Results on Instance Set \mathcal{B}

| Topology | N | High outsourcing cost | | | | | Low outsourcing cost | | | | |
|----------|-----|-----------------------|-----------|------|-------|--------|----------------------|-----------|------|------|--------|
| | | #Opt. | #All out. | T(s) | Tree | Gap(%) | #Opt. | #All out. | T(s) | Tree | Gap(%) |
| R (46) | 25 | 46 | 0 | 0 | 0 | 0.00 | 46 | 0 | 0 | 0.3 | 0.03 |
| | 50 | 46 | 0 | 17 | 4.3 | 0.11 | 46 | 0 | 19 | 5.0 | 0.10 |
| | 100 | 29 | 0 | 250 | 11.9 | 0.04 | 30 | 0 | 212 | 13.1 | 0.05 |
| C (34) | 25 | 34 | 0 | 3 | 12.1 | 0.52 | 34 | 34 | 0 | 0 | 0.00 |
| | 50 | 34 | 0 | 255 | 492.0 | 0.49 | 34 | 34 | 0 | 0 | 0.00 |
| | 100 | 25 | 0 | 161 | 34.8 | 0.08 | 34 | 26 | 0 | 0 | 0.00 |
| RC (32) | 25 | 32 | 0 | 7 | 40.5 | 1.48 | 32 | 16 | 2 | 7.6 | 0.26 |
| | 50 | 32 | 0 | 2 | 3.9 | 0.53 | 32 | 16 | 29 | 47.8 | 0.23 |
| | 100 | 31 | 0 | 357 | 16.5 | 0.07 | 32 | 0 | 75 | 5.9 | 0.05 |

outsourcing cost. The results are aggregated over all instance sizes. The columns in the tables report the topology of the instances (T), the type of time windows (TW), the fixed vehicle costs (C) for set \mathcal{A} being high, medium and low (a,b and c, respectively), and the vehicle capacity (Q) for set \mathcal{B} . For both outsourcing costs (high and low), the tables give the average percentage cost improvement (Avg. %), the highest improvement (Max. %), and the number of solved instances in the category (#Opt.). Finally, the “All out.” column indicates the fraction of the solved instances in which all demand is outsourced in the optimal solution: all (1), none (0), or two-thirds ($\frac{2}{3}$) of the instances. If for an instance all demand is outsourced in the optimal solution of the VRPPO, then this is also the case for the VRPPC. Hence, for these instances, allowing part of a customer’s demand to be outsourced does not result in a cost improvement of the VRPPO compared with the VRPPC. For each instance, the percentage improvement in total cost between VRPPO and VRPPC is computed as $(\text{cost VRPPC} - \text{cost}$

VRPPO)/cost VRPPC. Only instances for which both the VRPPO and the VRPPC result in an optimal solution are included in the averages. Note that there are some instances for which an optimal VRPPO solution is found, but no optimal VRPPC solution.

5.3.1. Improvements on Set \mathcal{A} Instances. In Table 4, the solutions of the VRPPO are compared with the solutions of the VRPPC for instance set \mathcal{A} .

First, it is observed that the improvements that can be achieved by allowing a part of the demand to be outsourced are relatively small. For high outsourcing costs, a better improvement can be achieved than for low outsourcing costs for most instance categories. This can be explained by the fact that for a high outsourcing cost, covering more distance with a private vehicle is more likely to be cost efficient compared with a situation in which outsourcing is cheap. For the R and RC instances, higher improvements are reached than for the C instances, which can be explained by the fact that if one customer in a cluster is outsourced, the

Table 4. Comparison of VRPPO and VRPPC for Instance Set \mathcal{A}

| T | TW | C | High outsourcing cost | | | | Low outsourcing cost | | | |
|----|----|---|-----------------------|--------|-------|----------|----------------------|--------|-------|----------|
| | | | Avg. % | Max. % | #Opt. | All out. | Avg. % | Max. % | #Opt. | All out. |
| R | 1 | a | 0.32 | 1.30 | 20 | 0 | 0.68 | 1.29 | 25 | 0 |
| | | b | 0.12 | 0.93 | 22 | 0 | 0.08 | 0.66 | 22 | 0 |
| | | c | 0.08 | 1.05 | 22 | 0 | 0.03 | 0.76 | 22 | 0 |
| | 2 | a | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| | | b | 0 | 0 | 2 | 0 | 0 | 0 | 2 | 0 |
| | | c | 0 | 0 | 4 | 0 | 0 | 0 | 4 | 0 |
| C | 1 | a | 0 | 0 | 24 | 1 | 0 | 0 | 27 | 1 |
| | | b | 0 | 0.02 | 17 | 0 | 0 | 0 | 27 | 1 |
| | | c | 0 | 0.02 | 17 | 0 | 0 | 0 | 27 | 1 |
| | 2 | a | 0 | 0 | 16 | 1 | 0 | 0 | 21 | 1 |
| | | b | 0 | 0 | 1 | 0 | 0 | 0 | 21 | 1 |
| | | c | 0 | 0 | 1 | 0 | 0 | 0 | 21 | 1 |
| RC | 1 | a | 0.12 | 0.73 | 15 | 0 | 0 | 0 | 24 | 1 |
| | | b | 0.19 | 1.31 | 17 | 0 | 0.12 | 0.60 | 18 | 0 |
| | | c | 0.21 | 1.46 | 17 | 0 | 0.14 | 0.69 | 18 | 0 |
| | 2 | a | 0 | 0 | 1 | 0 | 0 | 0 | 9 | 1 |
| | | b | 0 | 0 | 5 | 0 | 0 | 0 | 4 | 0 |
| | | c | 0 | 0 | 7 | 0 | 0.01 | 0.06 | 7 | 0 |

whole cluster of customers tends to be outsourced as for the VRPPC (Dabia, Lai, and Vigo 2019). For the limited number of solved instances with wide time windows (type 2), hardly any improvement is achieved by allowing outsourcing. There are no consistent results for the different vehicle costs because, for the R instances, better improvements are achieved for high vehicle costs (a), whereas for the RC instances, better improvements are found for low vehicle costs (c).

5.3.2. Improvements on Set \mathcal{B} Instances. Table 5 shows that for set \mathcal{B} , substantially larger improvements are obtained by allowing outsourcing of part of the demand compared with set \mathcal{A} . Improvements of approximately 10% are achieved. Also observe that higher improvements are obtained when vehicle capacity is tight ($Q = 30$), as the percentage improvements are much higher for $Q = 30$ than for $Q = 50$. This implies that splitting demand over a private vehicle and a common carrier is more beneficial if the customer demands are closer to or higher than vehicle capacity. For vehicle capacity $Q = 30$, larger improvements can be achieved for high outsourcing costs than for low outsourcing costs for the same reasons as indicated for set \mathcal{A} . For vehicle capacity $Q = 50$, the improvements are smaller and comparable for both outsourcing cost levels. For high outsourcing costs, the best improvements can be achieved for instances with both clustered and randomly located customers (RC), whereas for low outsourcing costs, the best improvements are found for instances with only randomly located customers (R). The type of time windows does not have a big impact on the improvements for the instances in set \mathcal{B} .

To see the impact of time windows, we also conducted experiments for a subset of instances in data set \mathcal{B} in which we discarded the time windows and adjusted the algorithm accordingly by, for example, removing the time condition from the dominance

criteria. We observed that the results are quite comparable to those with time windows for these instances.

5.4. Insights

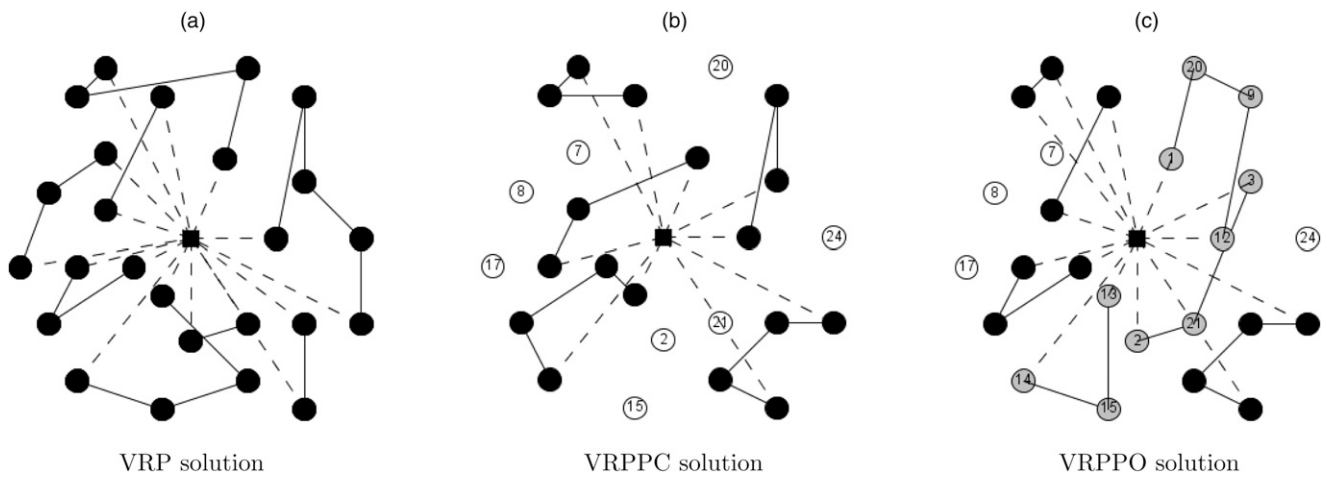
To get further insight into the obtained results, we examine the structure of some individual optimal solutions by visualizing them in Figures 3 and 4. In both figures, routes are indicated by lines connecting the visited customers. The customers that are not connected by the lines have their demand fully outsourced. The customers indicated in gray are in a route that requires a split delivery because total demand exceeds vehicle capacity. Note that any customer with sufficiently high demand can be the split customer without changing the total costs; therefore, all customers in these routes are colored gray.

In a solution of the VRPPC, some routes may not fully utilize the vehicle capacity. One might expect that the VRPPO solution contains the same routes as the VRPPC solution in which more customers are added to the routes and a split delivery is performed to fully utilize the vehicle capacity. However, the results show that this is not necessarily the case. Rather, the VRPPO solution of an instance can contain completely different routes than the corresponding VRPPC solution. As an example, consider Figure 3, which shows the optimal VRP, VRPPC, and VRPPO solutions of set \mathcal{A} , instance R101a, with 25 customers and high outsourcing costs. Consider customer 20 in the upper part of Figure 3(b). Customer 20 is not added to route 0-3-9-12-0 of the VRPPC solution to find the optimal VRPPO solution, but rather, it is combined with customers 9, 12, and 1 because of both efficiency and time windows. Note that customers 3 and 20 cannot be in the same route because of their time windows; temporarily widening the time windows of customer 20 to allow for customers 3 and 20 in the same route does not result in a different solution. Hence, in the VRPPO solution, customer 20 is not added to route 0-3-9-12-0

Table 5. Comparison of VRPPO and VRPPC for Instance Set \mathcal{B}

| T | TW | Q | High outsourcing cost | | | | Low outsourcing cost | | | |
|----|----|----|-----------------------|--------|-------|----------|----------------------|--------|-------|----------|
| | | | Avg. % | Max. % | #Opt. | All out. | Avg. % | Max. % | #Opt. | All out. |
| R | 1 | 30 | 3.87 | 10.05 | 36 | 0 | 2.55 | 5.49 | 36 | 0 |
| | | 50 | 0.61 | 2.69 | 28 | 0 | 0.88 | 2.22 | 29 | 0 |
| | 2 | 30 | 3.86 | 10.05 | 33 | 0 | 2.55 | 5.49 | 33 | 0 |
| | | 50 | 0.54 | 1.29 | 24 | 0 | 0.84 | 1.76 | 24 | 0 |
| C | 1 | 30 | 2.87 | 3.99 | 27 | 0 | 0 | 0 | 27 | 1 |
| | | 50 | 0 | 0 | 18 | 0 | 0 | 0 | 27 | 1 |
| | 2 | 30 | 1.82 | 3.21 | 24 | 0 | 0 | 0 | 24 | 1 |
| | | 50 | 0 | 0 | 24 | 0 | 0 | 0 | 24 | 2/3 |
| RC | 1 | 30 | 4.56 | 7.95 | 24 | 0 | 0.03 | 0.09 | 24 | 2/3 |
| | | 50 | 0.15 | 0.88 | 24 | 0 | 0.11 | 0.48 | 24 | 0 |
| | 2 | 30 | 4.56 | 7.95 | 24 | 0 | 0.03 | 0.09 | 24 | 2/3 |
| | | 50 | 0.13 | 0.76 | 23 | 0 | 0.10 | 0.39 | 24 | 0 |

Figure 3. Set \mathcal{A} Instance R101a, 25 Customers, High Outsourcing Cost

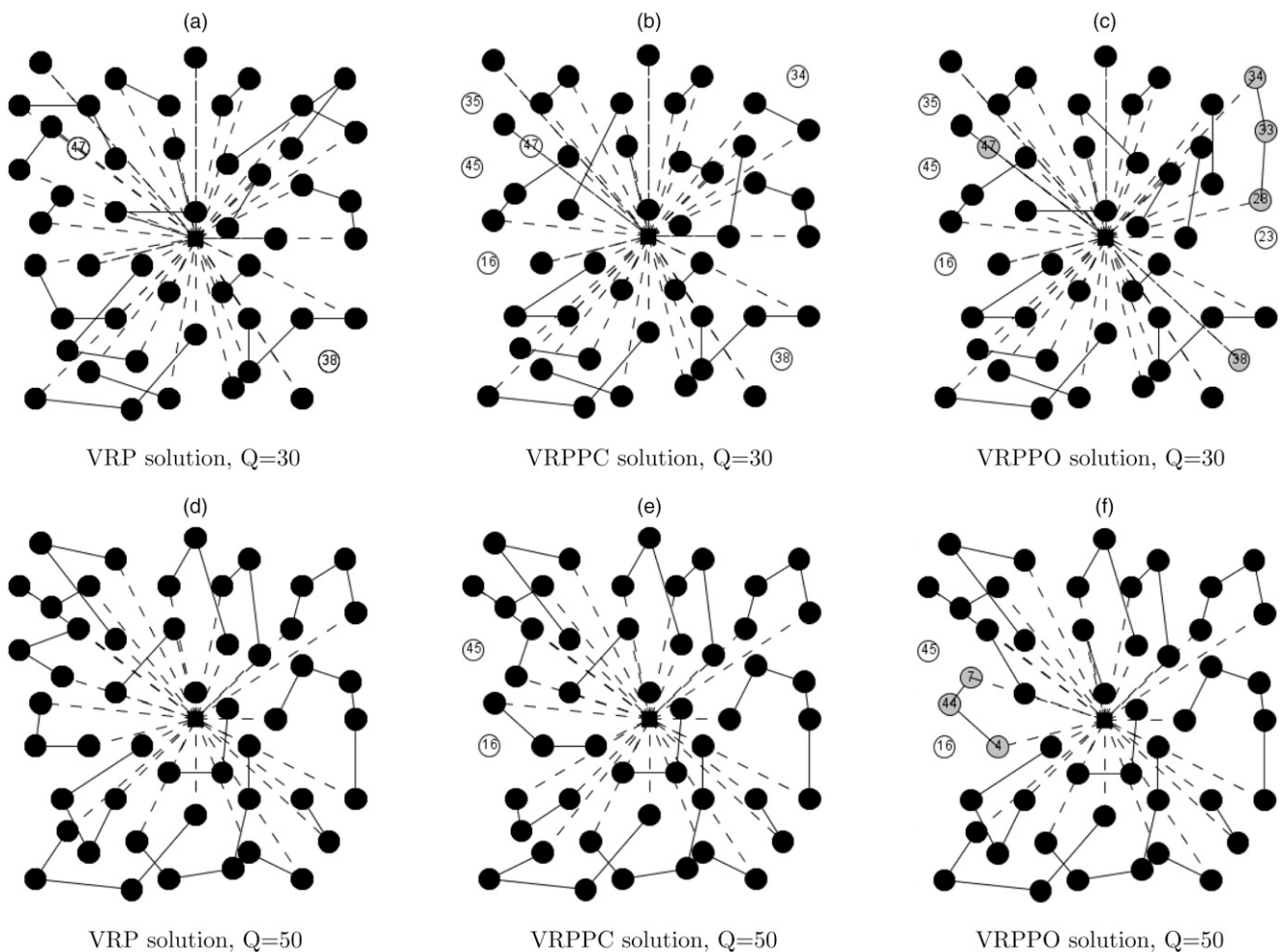


Note. Routes with gray customers require a split delivery which can be assigned to any customer with sufficiently high demand.

of the VRPPC solution because of route efficiency reasons. At the same time, customer 3 in the VRPPO solution is served together with customers whose

demand was fully outsourced in the VRPPC solution. One can observe that only one route is the same in both solutions (in the lower-right area). The VRPPO

Figure 4. Set \mathcal{B} Instance R102, 50 Customers, High Outsourcing Cost



Note. Routes with gray customers require a split delivery which can be assigned to any customer with sufficiently high demand.

solution contains seven routes, of which three contain a split delivery.

Next, it is also interesting to look at the number of units of demand outsourced. In both routes 0-2-21-3-0 and 0-14-15-13-0, just one unit of demand needs to be outsourced; hence, all customers are candidate to split the delivery because their demand is larger than one. In route 0-12-9-20-1-0, four units of demand need to be outsourced. Also for this route, all customers are candidate to be split, and four units of demand represent between 21% and 44% of each customer's demand.

Figure 4 presents the VRP, VRPPC, and VRPPO results for set \mathcal{B} , instance R102, with high outsourcing costs. The improvement in costs of the VRPPO compared with the VRPPC is 9.79% for $Q = 30$ and 1.55% for $Q = 50$, respectively. The VRP solution for $Q = 30$ is infeasible, because customers 38 and 47 have demands higher than the vehicle capacity. For $Q = 30$, two customers (38 and 47) with demand larger than the vehicle capacity must be fully outsourced in the VRPPC solution (Figure 4(b)), but full trucks can be sent to customers in the VRPPO solution, which reduces costs substantially (Figure 4(c)). For $Q = 50$, both customers 38 and 47 are combined with another customer in the route in both the VRPPC and the VRPPO solutions (Figure 4, (e) and (f), respectively). Note that for $Q = 30$, improvements are not only achieved because customers with a demand higher than vehicle capacity can be partially served by a private vehicle in the VRPPO, but also because other adjustments can be made to improve efficiency. For example, customer 34 that has demand of eight is outsourced in the VRPPC solution but is served by a private vehicle in the VRPPO solution. Moreover, note that customer 23 for $Q = 30$ is served in the VRPPC solution, whereas in the VRPPO solution it is more efficient to fully outsource this customer to service customer 34 by the private vehicle (with a split in the corresponding route). For $Q = 50$, the VRPPO solution contains one route less than the VRPPC solution. Hence, by allowing a split between private and outsourced delivery, the number of used private vehicles can be reduced in some cases.

For $Q = 30$, the quantities outsourced are one unit for routes 0-39-0 and 0-29-34-35-0, and six units for route 0-48-0, which is 17% of the demand. For $Q = 50$, only route 0-4-44-7-0 requires a split, and one unit of demand is outsourced, which is between 4% and 11% of the demand.

Concluding, a solution of the VRPPO can be rather different from the corresponding VRPPC solution. The routes are structurally different, customers fully served in a VRPPC solution can be fully outsourced in the VRPPO solution, and customers with demand higher than vehicle capacity can receive a full truck

load delivery in the VRPPO solution. Moreover, both small and large shares of the demand are being outsourced in the split delivery in the considered examples.

6. Conclusion and Future Research

This paper is the first to formally describe a vehicle routing problem in which splitting the delivery of demand to customers between the private and common fleet is allowed. For the so-called VRPPO, we developed a branch-and-price-and-cut solution framework. We proposed two master problem formulations for the VRPPO, and for both master problems we designed two pricing algorithms. In the first master problem, all outsourcing decisions are taken in the master problem, whereas in the second master problem, the decision on partially outsourcing a demand is referred to the pricing problem. The first pricing algorithm was inspired by a pricing algorithm for the SDVRPTW by Desaulniers (2010) in which multiple labels per extension are created to decide which customer is split. The second pricing algorithm exploits specific problem characteristics by creating at most one label per extension and by taking the splitting decision after creating a path. The first pricing algorithm leads to a higher number of labels, whereas in the second pricing algorithm the paths can be longer because vehicle capacity can be exceeded during the labeling algorithm. The performance of the algorithms is enhanced by applying (generalized) subset-row inequalities and dominance rules in the labeling algorithms. For the first pricing algorithm the dominance rules suggested by Desaulniers (2010) are applicable. For the second pricing algorithm, nontrivial problem-specific adjustments are made to be able to handle the postponed decision on splitting.

Extensive testing on two sets of instances derived from the literature provided insight into the different algorithms and the possible cost improvements of the VRPPO over the VRPPC. The results show that the second pricing algorithm performs much better than the first pricing algorithm, and that the difference between the master problems is small. Moreover, the results show that higher cost improvements can be achieved through outsourcing and split deliveries if customer demand is close to or higher than vehicle capacity. If outsourcing costs are low, it can be more beneficial to outsource all demand of a certain customer instead of having a split, thus resulting in larger cost improvements of the VRPPO over the VRPPC for high outsourcing costs. Finally, a topology with randomly located customers gives more room for improvement than settings with only clustered customers because outsourcing one customer in a cluster tends to lead to outsourcing all customers in the cluster, which was also observed for the VRPPC (Dabia, Lai, and Vigo 2019).

Because both routing problems with outsourcing or split deliveries are rich problems, multiple directions for future research can be identified. First, similar to Dabia, Lai, and Vigo (2019) and Gahm, Brabänder, and Tuma (2017), the outsourcing cost structure could be extended to include, for example, quantity discounts. Second, one could consider customer inconvenience constraints, such as a minimum delivery amount (Gulczynski, Golden, and Wasil 2010; Han and Chu 2016). Finally, also allowing for splits between private vehicles in the VRPPO could offer interesting research challenges.

References

- Archetti C, Bianchessi N, Speranza MG (2011) A column generation approach for the split delivery vehicle routing problem. *Networks* 58(4):241–254.
- Archetti C, Bianchessi N, Speranza MG (2015) A branch-price-and-cut algorithm for the commodity constrained split delivery vehicle routing problem. *Comput. Oper. Res.* 64:1–10.
- Archetti C, Savelsbergh MWP, Speranza MG (2008) To split or not to split: That is the question. *Transportation Res. Part E: Logist. Transportation Res.* 44(1):114–123.
- Archetti C, Speranza MG (2012) Vehicle routing problems with split deliveries. *Int. Transportation Oper. Res.* 19(1–2):3–22.
- Baldacci R, Mingozzi A, Roberti R (2011) New route relaxation and pricing strategies for the vehicle routing problem. *Oper. Res.* 59(5):1269–1283.
- Bianchessi N, Drexel M, Irnich S (2019) The split delivery vehicle routing problem with time windows and customer inconvenience constraints. *Transportation Sci.* 53(4):917–1212.
- Bolduc MC, Laporte G, Renaud J, Boctor FF (2010) A tabu search heuristic for the split delivery vehicle routing problem with production and demand calendars. *Eur. J. Oper. Res.* 202(1):122–130.
- Bolduc MC, Renaud J, Boctor F, Laporte G (2008) A perturbation metaheuristic for the vehicle routing problem with private fleet and common carriers. *J. Oper. Res. Soc.* 59(6):776–787.
- Chen S, Golden B, Wasil E (2007) The split delivery vehicle routing problem: Applications, algorithms, test problems, and computational results. *Networks* 49(4):318–329.
- Chu CW (2005) A heuristic algorithm for the truckload and less-than-truckload problem. *Eur. J. Oper. Res.* 165(3):657–667.
- Côté JF, Potvin JY (2009) A tabu search heuristic for the vehicle routing problem with private fleet and common carrier. *Eur. J. Oper. Res.* 198(2):464–469.
- Dabia S, Lai D, Vigo D (2019) An exact algorithm for a rich vehicle routing problem with private fleet and common carrier. *Transportation Sci.* 53(4):917–1212.
- Desaulniers G (2010) Branch-and-price-and-cut for the split-delivery vehicle routing problem with time windows. *Oper. Res.* 58(1):179–192.
- Feillet D, Dejax P, Gendreau M, Gueguen C (2004) An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks* 44(3):216–229.
- Gahm C, Brabänder C, Tuma A (2017) Vehicle routing with private fleet, multiple common carriers offering volume discounts, and rental options. *Transportation Res. Part E: Logist. Transportation Res.* 97:192–216.
- Goeke D, Gschwind T, Schneider M (2019) Upper and lower bounds for the vehicle-routing problem with private fleet and common carrier. *Discrete Appl. Math.* 264:43–61.
- Gulczynski D, Golden B, Wasil E (2010) The split delivery vehicle routing problem with minimum delivery amounts. *Transportation Res. Part E: Logist. Trans. Rev.* 46(5):612–626.
- Han AFW, Chu YC (2016) A multi-start heuristic approach for the split-delivery vehicle routing problem with minimum delivery amounts. *Transportation Res. Part E: Logist. Transportation Res.* 88:11–31.
- Jepsen M, Petersen B, Spoorendonk S, Pisinger D (2008) Subset-row inequalities applied to the vehicle-routing problem with time windows. *Oper. Res.* 56(2):497–511.
- Jin M, Liu K, Eksioğlu B (2008) A column generation approach for the split delivery vehicle routing problem. *Oper. Res. Lett.* 36(2):265–270.
- Keskin BB, Çapar İ, Sox CR, Freeman NK (2014) An integrated load-planning algorithm for outbound logistics at Webb Wheel. *Interfaces* 44(5):480–497.
- Lee J, Kim BI (2015) Industrial ship routing problem with split delivery and two types of vessels. *Expert Systems Appl.* 42(22):9012–9023.
- Liu FH, Shen SY (1999) The fleet size and mix vehicle routing problem with time windows. *J. Oper. Res. Soc.* 50(7):721–732.
- Luo Z, Qin H, Zhu W, Lim A (2016) Branch and price and cut for the split-delivery vehicle routing problem with time windows and linear weight-related cost. *Transportation Sci.* 51(2):668–687.
- Ozbaygin G, Karasan O, Yaman H (2018) New exact solution approaches for the split delivery vehicle routing problem. *Eur. J. Comput. Optim.* 6(1):85–115.
- Potvin JY, Naud MA (2011) Tabu search with ejection chains for the vehicle routing problem with private fleet and common carrier. *J. Oper. Res. Soc.* 62(2):326–336.
- Righini G, Salani M (2006) Symmetry helps: Bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints. *Discrete Optim.* 3(3):255–273.
- Salani M, Vacca I (2011) Branch and price for the vehicle routing problem with discrete split deliveries and time windows. *Eur. J. Oper. Res.* 213(3):470–477.
- Solomon MM (1987) Algorithms for the vehicle routing and scheduling problems with time window constraints. *Oper. Res.* 35(2):254–265.
- Spliet R, Desaulniers G (2015) The discrete time window assignment vehicle routing problem. *Eur. J. Oper. Res.* 244(2):379–391.
- Stenger A, Vigo D, Enz S, Schwind M (2013) An adaptive variable neighborhood search algorithm for a vehicle routing problem arising in small package shipping. *Transportation Sci.* 47(1):64–80.
- Tilk C, Rothenbächer AK, Gschwind T, Irnich S (2017) Asymmetry matters: Dynamic half-way points in bidirectional labeling for solving shortest path problems with resource constraints faster. *Eur. J. Oper. Res.* 261(2):530–539.
- Toth P, Vigo D (2014) *Vehicle Routing: Problems, Methods, and Applications*, MOS-SIAM Series on Optimization, vol. 18, 2nd ed. (Society for Industrial and Applied Mathematics, Philadelphia).
- Yan S, Chu JC, Hsiao FY, Huang HJ (2015) A planning model and solution algorithm for multi-trip split-delivery vehicle routing and scheduling problems with time windows. *Comput. Indust. Eng.* 87:383–393.