



Transportation Science

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

Branch-and-Price-Based Algorithms for the Two-Echelon Vehicle Routing Problem with Time Windows

Nico Dellaert, Fardin Dashty Saridarq, Tom Van Woensel, Teodor Gabriel Crainic

To cite this article:

Nico Dellaert, Fardin Dashty Saridarq, Tom Van Woensel, Teodor Gabriel Crainic (2018) Branch-and-Price-Based Algorithms for the Two-Echelon Vehicle Routing Problem with Time Windows. Transportation Science

Published online in Articles in Advance 23 Oct 2018

. <https://doi.org/10.1287/trsc.2018.0844>

Full terms and conditions of use: <https://pubsonline.informs.org/page/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 © 2018, INFORMS

Please scroll down for article—it is on subsequent pages





INFORMS is the largest professional society in the world for professionals in the fields of operations research, management science, and analytics.

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

Branch-and-Price–Based Algorithms for the Two-Echelon Vehicle Routing Problem with Time Windows

Nico Dellaert,^a Fardin Dashty Saridarq,^a Tom Van Woensel,^a Teodor Gabriel Crainic^b

^a School of Industrial Engineering, Eindhoven University of Technology, 5600MB Eindhoven, Netherlands; ^b Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and Department of Management and Technology, Université du Québec à Montréal, Station Centre-Ville, Montréal, Québec H3C 3P8, Canada

Contact: n.p.dellaert@tue.nl,  <http://orcid.org/0000-0003-2343-7574> (ND); f.dashty.saridarq@tue.nl,  <http://orcid.org/0000-0001-6446-251X> (FDS); t.v.woensel@tue.nl,  <http://orcid.org/0000-0003-4766-2346> (TVW); teodorgabriel.crainic@cirrelt.net,  <http://orcid.org/0000-0002-4424-0984> (TGC)

Received: August 17, 2016

Revised: June 19, 2017

Accepted: February 25, 2018

Published Online in Articles in Advance:
October 23, 2018

<https://doi.org/10.1287/trsc.2018.0844>

Copyright: © 2018 INFORMS

Abstract. This paper studies the two-echelon capacitated vehicle routing problem with time windows. The first echelon consists of transferring freight from depots to intermediate facilities (i.e., satellites), whereas the second echelon consists of transferring freight from these facilities to the final customers, within their time windows. We propose two path-based mathematical formulations for our problem: (1) in one formulation, paths are defined over both first- and second-echelon tours, and (2) in the other one, the first- and second-echelon paths are decomposed. Branch-and-price–based algorithms are developed for both formulations. We compare both formulations and solution methods on a comprehensive set of instances and are able to solve instances up to five satellites and 100 customers to optimality. This paper is the first paper in the literature that solves such large instance sizes.

Funding: This work was funded by the Natural Sciences and Engineering Research Council of Canada through its Discovery Grant program; the Dutch Institute for Advanced Logistics [Grant ConCoord]; and Fonds de Recherche du Québec through their infrastructure grants.

Supplemental Material: The online appendix is available at <https://doi.org/10.1287/trsc.2018.0844>.

Keywords: branch-and-price • column generation • two-echelon vehicle routing problem with time windows

1. Introduction

Consider a multiechelon distribution system in which the transportation chain is split into multiple echelons. In each echelon, goods are consolidated at facilities and transferred to customers or another set of facilities. Vehicles with different characteristics (e.g., capacity) are used in each echelon to transfer goods. Such systems bring significant economic advantage owing to the efficient use of vehicle capacities at each echelon. An important problem is how to efficiently route vehicles in each echelon to distribute goods to their final customers. In this paper, we consider a two-echelon vehicle routing problem (2E-VRP) with time windows for customers.

The first echelon of the 2E-VRP consists of transferring freight from depots to intermediate facilities, whereas the second echelon consists of transferring freight from these facilities to the final customers. This problem arises in different applications, such as multimodal transportation systems, spare parts distribution, e-commerce, grocery and hyper market products distribution, and home delivery services (Perboli, Tade, and Vigo 2011). Another important application of the 2E-VRP arises in city logistics, whereby a set of external depots that are located at the city limits act as sources

of freight that need to be delivered to final customers using intermediate facilities called satellites. Consolidation and coordination activities are performed at these satellites located close to or within the city center (Crainic, Ricciardi, and Storchi 2009). A two-echelon distribution system intends to keep large vehicles out of the city center, by using smaller vehicles with lower pollution levels for the distribution of the consolidated freight from the satellites to the final customers. A comprehensive study on city logistics is presented in Bektaş, Crainic, and Woensel (2015) and in Savelsbergh and Woensel (2016).

To our knowledge, only a limited number of studies exist in the literature that consider the 2E-VRP and propose exact and heuristic solution approaches for the problem. These studies mostly focus on the routing aspect without considering time windows, and the ones that consider time windows are mostly solved using (meta)heuristics. This paper considers models with hard time windows at the final customers. In the remainder of this paper, we will refer to the described problem as the *two-echelon vehicle routing problem with time windows* (2E-VRPTW). We will also use the terminology proposed by Crainic et al. (2010) for the intermediate consolidation facilities and vehicles. Therefore, the

intermediate facilities are called satellites, the first-echelon vehicles are called urban vehicles (e.g., trucks or tramways), and the second-echelon vehicles are called city freighters.

The goal of this paper is to study the 2E-VRPTW and develop a number of alternative formulations and exact solution methods. The contributions of this paper are as follows:

1. In the literature, the 2E-VRPTW is not treated from an exact optimization point of view, and few formal model descriptions exist in the literature. In this paper, we introduce the generic problem and propose multiple mathematical formulations and a number of solution methods. Specifically, we propose an arc-based mathematical formulation and two path-based mathematical formulations. For each path-based formulation, we develop specific solution methods based on the well-known branch-and-price framework.

- (a) A path-based formulation integrating both first- and second-echelon routes is developed (denoted as the two-echelon one-path formulation, or *2E-1P*). This formulation defines interconnected first- and second-echelon routes as the components of the problem. For this formulation, an exact branch-and-price algorithm is proposed in which a special pricing problem is obtained, handled by a specifically designed labeling algorithm.

- (b) A second path-based formulation is designed in which the two echelon routes are separated (denoted as the two-echelon two-path formulation, or *2E-2P*). This formulation exploits the structure of the problem to define separate first-echelon routes and second-echelon routes as the components of the problem whereby the interconnectivity of the first and second echelon routing problems are assured, using extra connectivity constraints. A branch-and-price-based algorithm is proposed in which a special structure for the search tree is introduced.

2. The numerical evaluation through a large set of instances demonstrates the power of the newly developed models and their respective solution methods. We examine how large instances can be solved with the proposed branch-and-price algorithms. We are able to exactly solve instances up to 100 customers and up to five satellites. These instance sizes correspond to the two-echelon city logistics systems in medium-sized cities, in which urban consolidation centers (i.e., satellites) are used to consolidate loads (see e.g., CIVITAS Initiative 2015, for examples of urban consolidation centers in different European cities). As far as we know, this is the first paper in the literature able to solve such large instance sizes.

The remainder of the paper is organized as follows. A literature review is given in Section 2. We describe the problem in detail in Section 3. The arc-based formulation is presented in Section 4. In Section 5, we present the 2E-1P formulation and the branch-and-price algorithm.

In Section 6, we give the 2E-2P formulation of the problem and present the branch-and-price-based algorithm for this formulation. The computational experiments are presented and discussed in Section 7, followed by conclusions and suggestions for future research in Section 8.

2. Literature Review

The 2E-VRPTW is an extension of the 2E-VRP, but additionally, hard time windows are considered for the final customers. Only recently, the 2E-VRP received more and more attention in the literature, mainly discussed in a city logistics context (for more details see Cuda, Guastaroba, Speranza 2015). The basic variant of the 2E-VRP is addressed in few papers, in which multiple models and solution approaches are proposed. The temporal aspects of the problem and particularly specification of time windows for customers are only discussed and treated from a computational point of view in a few papers (e.g., Crainic, Ricciardi, and Storchi 2009; Mancini 2013; Grangier et al. 2016; and Li et al. 2016). However, the 2E-VRPTW in the present paper is not treated explicitly from an exact optimization point of view. We propose multiple mathematical formulations and exact solution methods for the 2E-VRPTW.

The literature on the 2E-VRP started recently with the introduction of an integrated short-term scheduling problem of operations and management of resources involving a two-tiered distribution structure in city logistics (Crainic et al. 2009). These authors introduced a two-echelon, synchronized, scheduled, multidepot, multiple-tour, heterogeneous vehicle routing problem with time windows and proposed a general model and formulations for the main system components. The 2E-VRP can be considered as a special case of this general planning problem. The 2E-VRP is officially introduced in Perboli, Tadei, and Vigo (2011); Crainic, Sforza, and Sterle (2011b); and Gonzalez-Feliu (2008). Different integer programming formulations for 2E-VRP and valid inequalities were proposed by different authors. Gonzalez-Feliu (2008) introduced a flow-based mathematical model and proposed some valid inequalities. Perboli and Tadei (2010) present valid inequalities based on the traveling salesman problem and the capacitated vehicle routing problem, the network flow formulation, and the connectivity of the transportation system graph. Perboli, Tadei, and Vigo (2011) presented a new mathematical model for 2E-VRP, a number of valid inequalities, and two math heuristics based on the proposed model.

A limited number of exact algorithms for the 2E-VRP are proposed in the literature. Gonzalez-Feliu (2008) proposed column generation approaches to find lower bounds for the problem. The author argues that although the number of satellites is not greater than five for city logistic applications, because of the number of

customers, (meta)heuristic approaches should be investigated. The reported gap for the medium-sized instances is approximately 12%. Jepsen, Spoorendonk, and Ropke (2013) present a general branch-and-cut algorithm for the symmetric version of 2E-VRP based on an edge flow model that is a relaxation and provides a valid lower bound. The authors tested the proposed algorithm on instances with maximum five satellites and 50 customers. Another exact algorithm is proposed by Baldacci et al. (2013), in which the problem is decomposed into a limited set of multidepot capacitated vehicle routing problems with side constraints. The proposed algorithm is based on the assumption that the number of satellite is less than 10 and thus the set of first-echelon routes is enumerable. Santos, Mateus, and da Cunha (2014) and Santos, da Cunha, and Mateus (2013) propose two branch-and-price and one branch-and-price-and-cut implementations. The authors assume that the set of all first-echelon routes is known. They test these algorithms on instances with maximum five satellites and 50 customers.

Because of the complexity of the problem, heuristics are also investigated for the 2E-VRP. For example, Crainic et al. (2011a) present a family of multistart heuristics based on separating the depot-to-satellite transfer and the satellite-to-customer delivery. Crainic et al. (2008) propose several meta-heuristics based on a clustering and multidepot approach. A meta-heuristic based on the greedy randomized adaptive search procedure (GRASP) combined with path relinking has been proposed by Crainic et al. (2013). A hybrid GRASP with a variable neighborhood descent heuristic is proposed by Zeng et al. (2014). Hemmelmayr, Cordeau, and Crainic (2012) propose an adaptive large neighborhood search heuristic. Breunig et al. (2016) provide a hybrid meta-heuristic combining enumerative local searches with destroy-and-repair principles. A few contributions proposing heuristic approaches for variants of the 2E-VRP with time windows were also proposed. For example, Grangier et al. (2016) introduces a two-echelon multiple-trip vehicle routing problem with satellite synchronization and provides an adaptive large neighborhood search. Li et al. (2016) propose a Clarke and Wright savings heuristic algorithm for the two-echelon time-constrained vehicle routing problem in linehaul-delivery systems.

3. Problem Description

The 2E-VRPTW is a two-echelon distribution network problem with a set of depots, a set of satellites, and a set of final customers. A homogeneous vehicle fleet is used for each echelon. First-echelon vehicle (i.e., urban vehicle) tours are used to transfer freight from depots to the satellites. Each urban vehicle tour starts from a depot and, after visiting a subset of satellites,

ends at the same depot. Second-echelon vehicles (i.e., city freighters) deliver freight to final customers by performing a tour starting from a satellite, visiting a subset of final customers, and ending at the same satellite. Each vehicle type has a specific capacity and operational cost paid per performed tour. The capacity of a city freighter is usually less than the capacity of an urban vehicle.

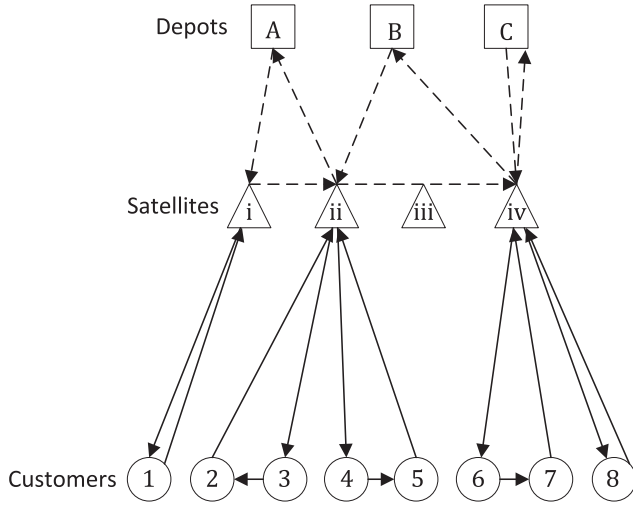
We define the problem on a directed graph $G = (V, A)$ with vertex set $V = P \cup S \cup Z$, where P is the set of depot locations, S is the set of satellite locations, and Z is the set of customer locations. The arc set A consists of two different sets: $A^{(1)}$ is the set of first-echelon arcs from node $i \in P \cup S$ to node $j \in S$, and $A^{(2)}$ is the set of second-echelon arcs from node $i \in S \cup Z$ to node $j \in S \cup Z$. There is a cost c_{ij} associated with each arc (i, j) , that is, the transportation cost (distance) for moving a vehicle from node i to node j on the arc. Moreover, t_{ij} is the travel time of a vehicle on arc (i, j) . Each customer $z \in Z$ has a given demand of d_z , which should be satisfied during a time window $[a_z, b_z]$. This time window is a hard time window, such that the start of the delivery is not allowed either before or after the time window. Waiting is allowed at all locations at no cost. Furthermore, no time windows (other than the total time horizon) are considered for either satellites or depots. A fleet of urban vehicles is represented by set \mathcal{T} . There is a capacity of $K^{(1)}$ and an associated fixed cost $h^{(1)}$ paid for each urban vehicle $\tau \in \mathcal{T}$. The fleet of city freighters is depicted by set \mathcal{V} . Each city freighter $v \in \mathcal{V}$ has the same capacity of $K^{(2)}$ and an associated fixed cost $h^{(2)}$.

At the first echelon, freight can be delivered from any depot. Each urban vehicle tour starts from a depot, visits some satellites, and ends at the same depot. Upon arrival to satellite i , delivery of freight requires a service time s_i . The second-echelon routing is performed using city freighter tours. These tours start from a satellite, visit a set of customers, and end at the same satellite. There is a service time s_i needed to deliver to each customer i . Each customer is visited only once. No direct shipments from a depot to the customers are allowed.

The objective of the 2E-VRPTW is to minimize the total fixed vehicle costs and transportation costs (including the service costs). A solution consists of a set of urban vehicle and city freighter tours, the amount of delivered freight to each satellite, the assignment of customers to satellites, and the arrival times of vehicles to each node. Figure 1 illustrates a solution for an instance with three depots, four satellites, and eight customers.

4. Arc-Based Formulation

We formulate the 2E-VRPTW as a three-index mixed integer programming formulation, inspired by ideas from Crainic, Sforza, and Sterle (2011b). In this formulation,

Figure 1. A 2E-VRPTW Instance

S includes multiple copies of each satellite. Let $r_{ij}^\tau = 1$ if a first echelon arc ($i \in P \cup S, j \in P \cup S$) is traveled by an urban vehicle $\tau \in T$; 0 otherwise. Similarly, if a second-echelon arc ($i \in S \cup Z, j \in S \cup Z$) is traveled by a city freighter $v \in \mathcal{V}$, then $q_{ij}^v = 1$; otherwise $q_{ij}^v = 0$. If a customer i is assigned to satellite j , then $w_{ij} = 1$. If an urban vehicle $\tau \in T$ is used to perform a first-echelon tour, then $u_\tau = 1$. Similarly, $u_v = 1$ if a city freighter $v \in \mathcal{V}$ is used in the second-echelon routing. Flow from depot i to satellite j on urban vehicle τ is represented by x_{ij}^τ . The arrival time of the urban vehicle τ to satellite s is represented by ω_s^τ . In a similar way, ω_z^v is used to represent the arrival time of the city freighter v to customer z . Then the network flow formulation for the 2E-VRPTW becomes:

$$\begin{aligned} \text{minimize } & \sum_{\tau \in T} \sum_{i \in P \cup S} \sum_{j \in P \cup S} c_{ij} r_{ij}^\tau + \sum_{v \in \mathcal{V}} \sum_{i \in S \cup Z} \sum_{j \in S \cup Z} c_{ij} q_{ij}^v \\ & + \sum_{\tau \in T} h^{(1)} u_\tau + \sum_{v \in \mathcal{V}} h^{(2)} u_v \end{aligned} \quad (4.1)$$

$$\text{subject to } \sum_{i \in P \cup S} r_{ij}^\tau - \sum_{i \in P \cup S} r_{ji}^\tau = 0, \quad \forall j \in P \cup S, \tau \in T \quad (4.2)$$

$$\sum_{i \in P \cup S} \sum_{j \in P} r_{ij}^\tau \leq 1, \quad \forall \tau \in T \quad (4.3)$$

$$K^{(1)} \sum_{k \in P \cup S} r_{ik}^\tau - x_{ji}^\tau \geq 0, \quad \forall \tau \in T, i \in S, j \in P \quad (4.4)$$

$$K^{(1)} \sum_{k \in P \cup S} r_{jk}^\tau - x_{ji}^\tau \geq 0, \quad \forall \tau \in T, i \in S, j \in P \quad (4.5)$$

$$\sum_{i \in P} \sum_{j \in S} x_{ij}^\tau \leq K^{(1)} u_\tau, \quad \forall \tau \in T \quad (4.6)$$

$$\sum_{v \in \mathcal{V}} \sum_{j \in S \cup Z} q_{ij}^v = 1, \quad \forall i \in Z \quad (4.7)$$

$$\sum_{j \in S \cup Z} q_{ij}^v - \sum_{i \in S \cup Z} q_{ji}^v = 0, \quad \forall j \in S, v \in \mathcal{V} \quad (4.8)$$

$$\sum_{i \in S \cup Z} \sum_{j \in S} q_{ij}^v \leq 1, \quad \forall v \in \mathcal{V} \quad (4.9)$$

$$\sum_{k \in S \cup Z} q_{ik}^v + \sum_{k \in S \cup Z} q_{jk}^v - w_{ij} \leq 1, \quad \forall i \in Z, j \in S, v \in \mathcal{V} \quad (4.10)$$

$$\sum_{j \in S} w_{ij} = 1, \quad \forall i \in Z \quad (4.11)$$

$$\sum_{i \in Z} d_i \sum_{j \in S \cup Z} q_{ij}^v \leq K^{(2)} u_v, \quad \forall v \in \mathcal{V} \quad (4.12)$$

$$\sum_{i \in P} \sum_{\tau \in T} x_{ik}^\tau - \sum_{j \in Z} d_j w_{jk} = 0, \quad \forall k \in S \quad (4.13)$$

$$\omega_j^\tau \geq \omega_i^\tau + t_{ij} + s_i - M(1 - r_{ij}^\tau), \quad \forall i \in S \cup P, j \in S, \tau \in T \quad (4.14)$$

$$\begin{aligned} \omega_j^v & \geq \omega_i^\tau + t_{ij} + s_i - M \left(2 - q_{kj}^v - \sum_{h \in P \cup S} r_{hi}^\tau \right), \\ & \forall i \in S, j \in Z, \tau \in T, v \in \mathcal{V} \end{aligned} \quad (4.15)$$

$$\omega_j^v \geq \omega_i^v + t_{ij} + s_i - M(1 - q_{ij}^v), \quad \forall i \in Z, j \in Z, v \in \mathcal{V} \quad (4.16)$$

$$\omega_i^v \geq a_i, \quad \forall i \in Z, v \in \mathcal{V} \quad (4.17)$$

$$\omega_i^v \leq b_i, \quad \forall i \in Z, v \in \mathcal{V} \quad (4.18)$$

$$r_{ij}^\tau \in \{0, 1\}, \quad \forall i, j \in S \cup P, \tau \in T \quad (4.19)$$

$$q_{ij}^v \in \{0, 1\}, \quad \forall i, j \in S \cup Z, v \in \mathcal{V} \quad (4.20)$$

$$x_{ij}^\tau \geq 0, \quad \forall i \in P, j \in S, \tau \in T \quad (4.21)$$

$$w_{ij} \in \{0, 1\}, \quad \forall i \in Z, j \in S \quad (4.22)$$

$$u_i \in \{0, 1\}, \quad \forall i \in T \cup \mathcal{V} \quad (4.23)$$

$$\omega_i^j \geq 0, \quad \forall i \in S \cup Z, j \in T \cup \mathcal{V}. \quad (4.24)$$

Objective function (4.1) minimizes the first-level transportation cost, second-level transportation cost, and vehicle fixed costs. Constraint (4.2) is the flow conservation constraint for the first level. Constraint (4.3) ensures that each urban vehicle is used at most once. Constraints (4.4) and (4.5) relate the flow and routing variables. Constraint (4.6) is the capacity constraint

for urban vehicles. Constraint (4.7) ensures that each customer is visited only by one city freighter. Constraint (4.8) is the flow conservation constraint for the second level. Constraint (4.9) ensures that each city freighter is used at most once. Constraints (4.10) and (4.11) ensure that each customer is assigned to a satellite and that the customer is visited by one city freighter tour, starting (and ending) at the assigned satellite. Constraint (4.12) is the capacity constraint for city freighters.

Constraint (4.13) links the first- and second-level decision variables by calculating the incoming freight flow to each satellite. It denotes that the total freight that is delivered to a satellite should be equal to the total freight that is distributed to the customers from the satellite. Constraints (4.14) and (4.16) calculate the arrival time of vehicles to satellite and customers. Constraint (4.15) relates the arrival time of an urban vehicle and departure time of a city freighter if they meet at a satellite to carry a demand. It denotes that a city freighter can depart from a satellite only after the freight is delivered to the satellite by urban vehicles and that the freight is consolidated and is ready to be distributed. Constraints (4.17) and (4.18) are hard time window constraints for the customers. Constraints (4.19–4.24) are the domain constraints.

The interconnectivity of and the coordination between the first- and second-echelon routing problems makes the problem highly complicated. Even without considering constraints (4.13) and (4.16), both first- and second-echelon routing problems are not straightforward to address. The first-echelon routing problem is a multidepot capacitated vehicle routing problem, and the second echelon routing problem is a multidepot capacitated vehicle routing problem with time windows, which are both extensions of the classic vehicle routing problem. We know from the literature that these problems are very complex and that the arc-based formulations are hard to solve.

This formulation is a compact formulation with respect to the number of constraints. However, it is not computationally attractive owing to the poor lower bound and the long run time. These issues are known problems for these types of formulations (see Crainic, Sforza, and Sterle 2011b). We present the results of this formulation in Section 7.

5. The Two-Echelon One-Path Formulation

Define a tour-tree as exactly one urban vehicle tour and at least one city freighter tour starting from each of the visited satellites by the urban vehicle tour. Note that a city freighter can depart from a satellite only after the associated urban vehicle has arrived to the satellite and that the unloading and loading of goods are performed within a service time. A solution to the 2E-VRPTW

consists of feasible tour-trees in which each customer is served by exactly one tour-tree. Figure 1 consists of three urban vehicle tours and five city freighter tours. One can represent this solution by the following tour-trees: $[(A - i - ii - A), \{(i - 1 - i), (ii - 3 - 2 - ii)\}]$, $[(B - ii - iv - B), \{(ii - 4 - 5 - ii), (iv - 6 - 7 - iv)\}]$ and $[(C - iv - C), \{(iv - 8 - iv)\}]$.

Let R be the set of all feasible tour-trees. A tour-tree is feasible if the capacity of the associated urban vehicles and city freighter(s) and also the time windows for all served customers are respected. For each tour-tree $r \in R$, let $\tau(r)$ be the used urban vehicle, and let $\mathcal{V}(r)$ be the set of the used city freighters. Let $A(r)$ be the subset of arcs covered by the tour-tree $r \in R$. An arc (i, j) belongs to $A(r)$ if it is traversed by either $\tau(r)$ or a city freighter $v \in \mathcal{V}(r)$. For each tour-tree r , we define $c_r = \sum_{(i,j) \in A(r)} c_{ij} + |\mathcal{V}(r)|h^{(2)} + h^{(1)}$ as the total transportation and vehicle fixed cost. Furthermore, let α_{rz} be a binary coefficient equal to 1 if customer z is visited by the tour-tree r (0 otherwise). Let y_r denote a binary variable that takes the value 1 if and only if the tour-tree $r \in R$ is included in the solution (0 otherwise).

On the basis of these definitions, the 2E-1P formulation can be written as the following set partitioning problem:

$$\text{minimize } \sum_{r \in R} c_r y_r \quad (5.1)$$

$$\text{subject to } \sum_{r \in R} \alpha_{rz} y_r = 1, \quad \forall z \in Z \quad (5.2)$$

$$y_r \in \{0, 1\}, \quad \forall r \in R. \quad (5.3)$$

Objective function (5.1) minimizes the total cost of the used tour-trees. Constraints (5.2) ensure each customer is served by one tour-tree. Constraints (5.3) are the domain constraint for the decision variables. We do not have to include capacity and time window constraints here, because we only consider feasible tour-trees. Note that for implementation issues, it is common to relax the equality constraints (5.2) by \geq inequalities. Moreover, the binary domain constraints (5.3) can be replaced by integer domain constraints.

5.1. Solution Algorithm

We propose a branch-and-price algorithm for the above 2E-1P formulation. A column generation method (see Lübbecke and Desrosiers 2005 for an overview of column generation approaches) is used to solve the linear programming (LP) relaxation of problem (5.1–5.3): starting with a small set of initial tour-trees (columns) $R' \subset R$, new columns are generated in an iterative manner by solving the pricing problem. Let $\lambda_z, z \in Z$ be the dual variable associated with constraints (5.2). Then the pricing problem, aiming to generate columns

with the most negative reduced cost, is formulated as follows:

$$\min_{r \in R} \left\{ \bar{c}_r = c_r - \sum_{z \in Z} \alpha_{rz} \lambda_z \right\}. \quad (5.4)$$

In the following sections, we present the used labeling algorithm for the pricing problem, the used branching strategy, and our tree exploring strategy. To keep the discussions clear and in an attempt to keep the overview structured, we decided to keep the high-level algorithmic choices in the main text and describe most of the algorithmic details in the online appendix.

5.2. Labeling Algorithm

The pricing problem aims to find a tour-tree with the most negative reduced cost. Considering that such a tour-tree could start from (and end at) any depot, it is reasonable to search for it by considering each depot independently. To solve this problem, we propose a labeling algorithm that is a modification of the labeling algorithm proposed by Feillet et al. (2004) for the elementary shortest path problem with resource constraints (ESPPRC). A bidirectional search (for example see Righini and Salani 2006, 2008) is performed, in which labels are extended in both forward and backward directions. In the forward direction, labels are extended from a depot toward a satellite as the first visited satellite in the associated tour-tree. Similarly, in the backward direction, labels are extended from a depot toward a satellite as the last visited satellite. Forward and backward labels are merged to construct a complete tour-tree. Breadth-first and depth-first search methods are used when extending labels.

5.3. Branching Strategy

The branching is done on (1) whether a column (tour-tree) is used, or not; and (2) whether a pair of two customers (or a pair of a satellite and a customer) are used in a tour-tree, or not. Initially, we use a most infeasible branching strategy for selecting the candidate pairs and columns. The final candidate (a pair or a column) is selected using the strong branching strategy.

The candidate column r is the one that maximizes $(y_r)(1 - y_r)$. Moreover, the candidate pair (i, j) is the one that maximizes $(\sum_{r \in R_{(i,j)}} y_r) \cdot (\sum_{r \in R_{-(i,j)}} y_r)$, where $R_{(i,j)}$ represents the set of columns in which either i and j are served together in a column, or none of them is in a column. Similarly, $R_{-(i,j)}$ is the set of columns in which in any column either i or j , but not both, is served.

5.4. Tree Exploring Strategy

Branch-and-price methods use the best-first and depth-first strategies. We use a mixture of these two strategies to explore a minimum number of nodes and also quickly

find good integer solutions. To select a candidate node b as the next branch to explore, three criteria are considered: (1) the lower bound value of the node (denoted by $lb(b)$), (2) the number of columns with nonfractional value in the optimal solution of the associated LP relaxation problem (denoted by $N_f(b)$), and (3) the level of the node (denoted by $L(b)$). Let w_1 , w_2 , and w_3 define the random weights used to diversify the search. The mixed strategy picks the node b with the minimum value of $w_1 * lb(b) + w_2 * L(b) + w_3 * N_f(b)$ for the branching.

6. The Two-Echelon Two-Path Formulation

Instead of formulating the 2E-VRPTW on a complete tour tree as we did in Section 5, we look into an alternative formulation in which first- and second-echelon tours are decomposed. This path-based formulation is inspired from the formulation proposed by Baldacci et al. (2013) for the 2E-VRP. This model and its associated branch-and-price-based algorithm is based on a decomposition introduced by Baldacci, Mingozzi, and Calvo (2011) for the capacitated location-routing problem.

A tour-tree can be decomposed into its building blocks (urban vehicle tour and city freighter tours). Therefore, a solution for the problem can be considered as feasible urban vehicle tours and feasible city freighter tours, which together build feasible tour-trees. We define the main components of the problem as separate urban vehicle and city freighter tours, whereby each demand request is jointly satisfied by exactly one urban vehicle tour and one city freighter tour, which meet at a satellite. Decomposing the tour-trees of the solution of the instance in Figure 1 results in the set of urban vehicle tours $\{(A - i - ii - A), (B - ii - iv - B), (C - iv - C)\}$ and the set of city freighter tours $\{(i - 1 - i), (ii - 3 - 2 - ii), (ii - 4 - 5 - ii), (iv - 6 - 7 - iv), (iv - 8 - iv)\}$.

Let \mathcal{M} be the set of all urban vehicle tours, and let $A(m)$ be the subset of arcs traversed by the urban vehicle tour $m \in \mathcal{M}$. An urban vehicle tour $m \in \mathcal{M}$ starts from a depot at time zero, visits a subset of satellites $S(m)$, and ends at the same depot. The urban vehicle tour m arrives at each visited satellite $i \in S(m)$ at time t_i^m . If satellite $j \in S(m)$ is visited immediately after satellite $i \in S(m)$ by the urban vehicle tour m , $(i, j) \in A(m)$, then the arrival time at satellite j can be calculated as $t_j^m = t_i^m + t_{ij} + s_i$. The total transportation and vehicle fixed cost c_m of an urban vehicle tour $m \in \mathcal{M}$ is $c_m = \sum_{(i,j) \in A(m)} c_{ij} + h^{(1)}$.

Let \mathcal{L}_m be the set of all feasible city freighter tours that could be fed from the urban vehicle tour m . A city freighter tour could be fed from an urban vehicle tour if they meet at a satellite to unload freight (from the urban vehicle) and load it (to the city freighter) to serve the

demand of a customer and the departure time of the city freighter tour is after the unloading time of the urban vehicle at the satellite. We indicate with $\mathcal{L} = \cup_{m \in \mathcal{M}} \mathcal{L}_m$ the set of all feasible city freighter tours. A city freighter tour is feasible if the associated city freighter vehicle capacity and the time windows of all of its customers are respected. Let ξ_{lz} be a binary coefficient equal to 1 if customer z is visited by the city freighter tour $l \in \mathcal{L}$ (0 otherwise). Moreover, let $A(l)$ be the subset of arcs traversed by the city freighter tour $l \in \mathcal{L}$. A load $\mathcal{D}_l = \sum_{z \in Z} \xi_{lz} d_z$ and (transportation and vehicle fixed) cost $c_l = \sum_{(i,j) \in A(l)} c_{ij} + h^{(2)}$ are associated with the city freighter tour l . A city freighter tour $l \in \mathcal{L}_m$ starts from a satellite $\S(l) \in S(m)$ at time $t_{\S(l)}^m + s_{\S(l)}$, delivers demands to a subset of customers, and ends at the same satellite.

Let \mathcal{X}_m denote a binary variable that takes the value 1 if and only if the urban vehicle tour $m \in \mathcal{M}$ is included in the solution, and let \mathcal{Y}_l denote a binary variable that takes the value 1 if and only if the city freighter tour $l \in \mathcal{L}$ is included in the solution. Then the 2E-2P formulation for the 2E-VRPTW is:

$$\text{minimize } \sum_{m \in \mathcal{M}} c_m \mathcal{X}_m + \sum_{l \in \mathcal{L}} c_l \mathcal{Y}_l \quad (6.1)$$

$$\text{subject to } \sum_{l \in \mathcal{L}} \xi_{lz} \mathcal{Y}_l = 1, \quad \forall z \in Z \quad (6.2)$$

$$\sum_{l \in \mathcal{L}_m} \mathcal{D}_l \mathcal{Y}_l \leq K^{(1)} \mathcal{X}_m, \quad \forall m \in \mathcal{M} \quad (6.3)$$

$$\mathcal{X}_m \in \{0, 1\}, \quad \forall m \in \mathcal{M} \quad (6.4)$$

$$\mathcal{Y}_l \in \{0, 1\}, \quad \forall l \in \mathcal{L}. \quad (6.5)$$

The objective function (6.1) minimizes the total first- and second-echelon transportation costs and the vehicle fixed costs. Constraints (6.2) ensure each customer is served by one city freighter tour. Constraints (6.3) ensure that the city freighter tours are selected such that the capacity of their associated urban vehicle tour is respected. Constraints (6.4) and (6.5) are the domains of the decision variables. We relax the equality constraints (6.2) by \geq inequalities. Moreover, the binary domain constraints (6.4) and (6.5) are replaced by integer domain constraints.

6.1. Solution Algorithm

A direct branch-and-price approach is not suitable for the 2E-2P formulation because of some difficulties (i.e., nonlinear pricing problem and unknown dual variables, which are explained in detail in the online appendix). Therefore, we propose a branch-and-price-based algorithm, in which the main idea is to enumerate all possible urban vehicle tours for the first echelon, combine multiple urban vehicle tours and generate

different first-echelon solutions (called a configuration), fix one configuration, and solve the resulting 2E-2P formulation. However, we do not consider all possible configurations and work on a limited number of configurations. The proposed branch-and-price-based algorithm uses the same branching strategy and tree exploring strategy as described in Section 5.

Let M denote a configuration (a first-echelon solution that is a combination of possibly multiple urban vehicle tours), and let θ be a parameter for the algorithm. $LB_{AG}(M)$ denotes an aggregate-based lower bound, and $LB_{AP}(M)$ denotes an assignment-based lower bound for the problem, if the configuration M is used as the first-echelon solution (note that $LB_{AP}(M) \geq LB_{AG}(M)$). The outline of the proposed algorithm is given in algorithm 1. The algorithm starts with generating the set of all possible urban vehicle tours (line 1.a). In each iteration of the algorithm, the set of next θ -best configurations is generated and is added to the set of current configurations, W (lines 1.c and 21). Then W is sorted again in a non-decreasing order of LB_{AP} values (lines 1.d and 23). For each configuration $M \in W$ a search subtree is created, in which for all nodes of the subtree, the solution of the first-echelon routing problem is fixed to M (line 2). Column generation is used to solve the LP relaxation of the problem (6.1)–(6.5) by replacing the set of all urban vehicle tours, \mathcal{M} , with a configuration M , and fixing $\mathcal{X}_m = 1$ for all urban vehicle tours $m \in M$ (lines 7–13). Whenever the lower bound of the next configuration is better than the best lower bound found so far, the creation of the next subtree is initiated (line 24). Each step is described in detail in the following sections.

6.2. Urban Vehicle Tours (Line 1.a)

An urban vehicle tour can visit any subset of satellites in any order (sequence). Moreover, for any sequence of satellites, we only consider nondominated depots (as the start and end point of the urban vehicle tour). A depot (when considering a specific sequence of satellites, (s_1, \dots, s_n)) is called nondominated if and only if no other depot would lead to a lower transportation cost or an earlier arrival time visiting the first satellite in its tour. We generate all urban vehicle tours by exhaustively considering any sequence of satellites and the related nondominated depots. The number of urban vehicle tours depends on the number of depots and satellites, capacity of urban vehicle, and the size of demands. Note that once all possible sequences of satellites are generated, it is straightforward to add the starting (and ending) depot to generate urban vehicle tours.

6.3. Configurations (Lines 1.c and 21)

Any combination of at least $\lceil \sum_{z \in Z} d_z / K^{(1)} \rceil$ urban vehicle tours could lead to a configuration. We define the problem of finding the best configuration with minimum

LB_{AG} value on a graph in which each node represents an urban vehicle tour. Let $G_M = (V_M, A_M)$ be a directed graph with the vertex set $V_M = \mathcal{M} \cup \{s, t\}$ and the arc set $A_M = \{(i, j) \mid C(i) \leq C(j), i \in \mathcal{M}, j \in V_M\} \cup \{(s, j) \mid j \in \mathcal{M}\} \cup \{(i, t) \mid i \in \mathcal{M}\}$. Nodes s and t are source and destination nodes, respectively. A path from s to t with at least $\lceil \sum_{z \in Z} d_z / K^{(1)} \rceil + 1$ arcs could lead to a configuration. A modified labeling algorithm based on Feillet et al. (2004) is used to solve this problem. Labels are extended in a forward direction starting from s toward t . Please refer to the online appendix for more detail.

6.4. θ -Best Configurations (Lines 1.c and 21)

Generating all possible configurations may be highly time-consuming and unnecessary. To prevent this, in each iteration of the proposed algorithm a limited number of configurations is considered. Thus, we generate the set of θ -best configurations in which θ is a parameter to be tuned. In this regard, we prevent an exhaustive search of all paths and discard all partial paths that would not lead to a path (configuration) of the set of θ -best configurations.

To find the θ -best configurations, we start with an empty set $\mathcal{P}' = \{\}$. In each iteration, a configuration is generated using the described labeling algorithm, and it is inserted in \mathcal{P}' such that it is sorted in a non-decreasing order of the aggregate-based lower bound values. It is repeated until the size of \mathcal{P}' becomes θ .

Algorithm 1. Pseudo-code of the branch-and-price-based algorithm for 2E-2P

- 1: Initialization:
 - a: Generate the set of all urban vehicle tours \mathcal{M} (Exhaustive search with dominance)
 - b: Initiate a search tree with a root node $T = \{root\}$, set of active configurations $W = \{\}$
 - c: Add θ -best configurations (with respect to aggregate-based lower bound) to W
 - d: Sort elements of W in a nondecreasing order of their LB_{AP} values ($W = \{M_1, \dots, M_{|W|}\}$)
 - e: $LB = LB_{AP}(M_1)$, $UB = \infty$, $MaxLB = LB_{AG}(M_\theta)$, $i = 1$
- 2: Add a child node to the root node and fix its first echelon solution equal to M_i
- 3: **while** $LB < LB_{AP}(M_{i+1})$ and $LB < MaxLB$ **do**
- 4: Select a node from the tree T
- 5: Branch
- 6: Initialization (generate \mathcal{L}')
- 7: **while** there exists a negative reduced cost city freighter tour **do**
- 8: Solve the pricing problem (find a negative reduced cost city freighter tour l)
- 9: $\mathcal{L}' = \mathcal{L}' + \{l\}$
- 10: Replace the set of all urban vehicle tours, \mathcal{M} , with a configuration M
- 11: Fix $\mathcal{X}_m = 1$ for all urban vehicle tours $m \in M$

- 12: Solve the resulted LP relaxation of the problem (6.1)–(6.5) by replacing \mathcal{L} with \mathcal{L}'
- 13: **end while**
- 14: Set the LB equal to the lowest lower bound value of the leaf nodes of the tree T
- 15: if improved integer solution is found: update UB
- 16: **if** $LB = UB$ **then**
- 17: Terminate and report the optimal solution
- 18: **end if**
- 19: **end while**
- 20: **if** $LB > MaxLB$ **then**
- 21: Generate next θ -best configurations and add to W
- 22: Fix $MaxLB$ equal to the aggregate-based lower bound of the worst configuration within the next θ -best configurations
- 23: Resort elements of W in a nondecreasing order of their LB_{AP} values
- 24: **else**
- 25: $i = i + 1$; goto step 2
- 26: **end if**

6.5. Lower Bounds (Lines 1.c, 1.d, 1.e, 3, 22, and 23)

For each used configuration, M , we determine two lower bounds for total costs: an aggregate-based bound (LB_{AG}) and an assignment-based bound (LB_{AP}).

We define the aggregate-based lower bound of a configuration M as $LB_{AG}(M) = \sum_{m \in M} C_m + \sum_{z \in Z} \min_{m \in M, s \in S(m)} \{\gamma_{zs}\}$. γ_{zs} is the minimum cost of serving the customer z from the satellite s (see also the online appendix for more details).

Additionally, we define an assignment-based lower bound for the configuration, $LB_{AP}(M) = \sum_{m \in M} C_m + Z^*(SL(M))$, where $Z^*(SL(M))$ is the optimal objective function value of the problem $SL(M)$, which calculates the lower bound for the second echelon if the configuration M is used as the first echelon routing solution. Let the decision variable g_{ijk} be the fraction of the demand of customer i that is supplied by urban vehicle tour m , $m \in M$, and satellite k . Of course, $g_{imk} = 0$ when satellite k is not part of the urban vehicle tour m . The problem $SL(M)$ is defined as follows:

$$\text{minimize } \sum_{i \in Z} \sum_{m \in M} \sum_{k \in S} g_{imk} \gamma_{ik} \quad (6.6)$$

$$\text{subject to } \sum_{m \in M} \sum_{k \in S} g_{imk} = 1, \quad \forall i \in Z \quad (6.7)$$

$$\sum_{i \in Z} \sum_{k \in S} g_{imk} d_i \leq K^{(1)}, \quad \forall m \in M \quad (6.8)$$

$$g_{imk} \geq 0, \quad \forall i \in Z, m \in M, k \in S \quad (6.9)$$

6.6. Column Generation (Lines 7–13)

The LP relaxation of problem (6.1–6.5) is solved with column generation. Starting with a small set of initial city freighter tours (columns) $L' \subset L$, new columns are generated in an iterative manner by solving a pricing

problem. The pricing problem finds new columns with negative reduced cost. Let ζ_z , $z \in Z$ be the dual variable associated with constraints (6.2) and δ_m , $m \in M$ be a dual variable associated with constraints (6.3). The pricing problem is formulated as follows:

$$\min_{l \in L} \left\{ \bar{c}_l = c_l - \sum_{z \in Z} \xi_{lz} \zeta_z + \sum_{m \in M} \mathcal{D}_l \delta_m \right\}. \quad (6.10)$$

We consider each urban vehicle tour $m \in M$ separately and search for city freighter tours (columns) starting from $S(m)$. Each satellite $s \in S(m)$ can be considered independently as a source of freight (i.e., depot) where freight is ready to be distributed to customers only after the associated urban vehicle route m has delivered it to the satellite. Thus, it is reasonable to search for city freighter tours with negative reduced cost for each pair of m and s , $s \in S(m)$.

This problem is defined on the second echelon of the graph G , that is, on a directed graph $\tilde{G} = (\tilde{V}, \tilde{A})$ with the vertex set $\tilde{V} = \{s\} \cup Z$ and the arc set $\tilde{A} = \{(i, j) \mid i \in \tilde{V}, j \in \tilde{V}\}$. An ESPPRC is used in which s is the origin and the destination. Let $L = \{Q, T\}$ be the resources where Q is the city freighter capacity and T is time (windows). Using the notation from Feillet et al. (2004), time windows constraints and city freighter capacity constraints are defined using intervals $[a_i, b_j]$ and $[0, K^{(2)}]$ for each node $i \in V'$, respectively. Let d_{ij}^l be the consumption of the resource l , $l \in L$ along an arc $(i, j) \in \tilde{V}$. Then it is straightforward to define $d_{ij}^Q = d_j$, $j \in Z$ and $d_{ij}^T = t_{ij}$. We use Desrochers' label correcting algorithm (Desrochers, Desrosiers, and Solomon 1992), adapted to ESPPRC by Feillet et al. (2004).

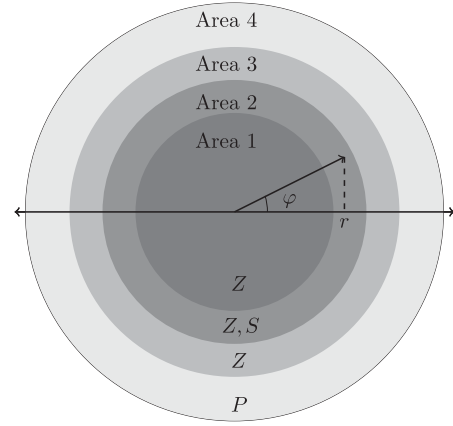
7. Computational Study

In this section, we present a computational study to test the proposed algorithms on newly generated 2E-VRPTW instances. All tests are performed using Delphi XE6 on a personal computer with Intel Core i7-4770 processor (8M cache, up to 3.90 GHz) and 8.00 GB of random access memory. CPLEX v12.6.0 is used to solve the arc-based formulation and the LP relaxations. First we explain how the new instances are generated, and second the analysis of the results is presented.

7.1. Instances

To generate instances, we simulate a circular urban area where the area is divided into four sections. Figure 2 shows the schematic representation used for an urban area and the four sections. Area 1 represents the city center where customers are located. Area 2 represents the surrounding area of the city center where satellites are located to serve the customers. In addition, there could exist customers inside area 2.

Figure 2. Schematic Representation of the Urban Area



Area 3 represents the housing area of the urban area where some customers are located. Depots are located inside area 4, which represents the external area of the city, which is a less populated section.

For all generated instances, urban vehicle capacity, city freighter capacity, urban vehicle cost, and city freighter cost are set to $K^{(1)} = 200$, $K^{(2)} = 50$, $h^{(1)} = 50$, $h^{(2)} = 25$. Moreover, the service time for all customers and satellites is set to 10. To locate depots and satellites in a reasonable way, we divide area 2 and area 4 according to the number of facilities. Area 2 is divided into $|S|$ subsections, and each satellite is located in one of such subsections. In a similar way, area 4 is divided into $|P|$ subsections, and each depot is located in one of such subsections.

Four categories of instances are generated, differing in the time windows and demands of customers: (1) Ca: $20 \leq a_z \leq 260$ and $a_z \leq b_z \leq a_z + 20$ and $d_z = 10$ or 20 ; (2) Cb: $20 \leq a_z \leq 260$ and $a_z \leq b_z \leq a_z + 20$ and $5 \leq d_z \leq 25$, where d_z is an integer; (3) Cc: $60 \leq a_z \leq 360$ and $a_z \leq b_z \leq a_z + 90$ and $d_z = 10$ or 20 ; (4) Cd: $60 \leq a_z \leq 360$ and $a_z \leq b_z \leq a_z + 20$ and $d_z = 10$ or 20 .

Using different combinations of $|P|$, $|S|$ and $|Z|$, different instances are generated for each category. We generate instances with four different values of $|Z|$, namely, $|Z| = \{15, 30, 50, 100\}$. Moreover, three combinations are used for values of $|P|$ and $|S|$, namely, $(|P|, |S|) = \{(2, 3), (3, 5), (6, 4)\}$. Each instance is represented by a notation that consists of the associated category name, an index, number of depots, number of satellites, and number of customers. For example, "Ca3-2-3-15" denotes the third instance of the category "Ca," with two depots, three satellites, and 15 customers. All instances can be found at the website <http://smartlogisticslab.nl>.

7.2. Computational Results

We designed experiments to numerically examine how large instances the branch-and-price algorithms can

Table 1. 2E-1P Formulation and Arc-Based Formulation: Instances with Two Depots, Three Satellites, and 15 Customers

Instance	2E-1P formulation (breadth-first)				2E-1P formulation (depth-first)				Arc-based formulation (CPLEX)			
	LB	UB	GAP%	CPU(s)	LB	UB	GAP%	CPU(s)	LB	UB	GAP%	CPU(s)
Ca1-2-3-15	608.701	612.385	0.602	3,600	612.385	612.385	0	305.2	523.725	612.782	14.533	3,600
Ca2-2-3-15	548.953	548.953	0	119.6	548.953	548.953	0	53.7	536.519	548.953	2.265	3,600
Ca3-2-3-15	551.985	551.985	0	127.6	551.985	551.985	0	429.8	463.629	563.628	17.742	3,600
Ca4-2-3-15	569.579	569.579	0	1,665.7	569.579	569.579	0	1,569.7	484.749	569.579	14.894	3,600
Ca5-2-3-15	555.796	555.796	0	1,576.5	555.796	555.796	0	2,219.4	478.194	583.886	18.101	3,600
Cb1-2-3-15	573.724	635.667	9.745	3,600	624.178	624.178	0	2,042	531.704	628.591	15.413	3,600
Cb2-2-3-15	516.739	516.739	0	36.9	516.739	516.739	0	7.6	480.661	516.739	6.982	3,600
Cb3-2-3-15	601.897	601.897	0	154.8	601.897	601.897	0	543.7	496.362	641.917	22.675	3,600
Cb4-2-3-15	512.664	590.866	13.235	3,600	543.458	555.339	2.139	3,600	522.549	546.314	4.350	3,600
Cb5-2-3-15	486.21	578.556	15.961	3,600	492.969	516.708	4.594	3,600	389.423	564.988	31.074	3,600
Cc1-2-3-15	526.172	611.145	13.904	3,600	501.246	619.908	19.142	3,600	478.812	609.192	21.402	3,600
Cc2-2-3-15	482.985	482.985	0	982.2	482.985	482.985	0	65.0	413.513	490.068	15.621	3,600
Cc3-2-3-15	518.163	539.685	3.988	3,600	536.25	539.685	0.636	3,600	435.667	563.000	22.617	3,600
Cc4-2-3-15	537.023	620.58	13.464	3,600	537.023	594.829	9.718	3,600	384.005	568.472	32.450	3,600
Cc5-2-3-15	—	—	—	3,600	436.467	436.467	0	119.4	311.510	442.467	29.597	3,600
Cd1-2-3-15	518.864	—	—	3,600	526.957	597.698	11.836	3,600	545.555	597.698	8.724	3,600
Cd2-2-3-15	483.133	483.133	0	78.7	483.133	483.133	0	24.2	403.247	483.133	16.535	3,600
Cd3-2-3-15	512.412	512.412	0	2,734.4	512.412	512.412	0	12.1	445.684	512.412	13.022	3,600
Cd4-2-3-15	568.937	585.301	2.796	3,600	584.282	585.301	0.174	3,600	500.442	594.105	15.765	3,600
Cd5-2-3-15	522.011	536.764	2.749	3,600	536.764	536.764	0	1,471.5	481.406	536.764	10.313	3,600
Average	—	—	—	2,353.8	—	—	2.412	1,703.2	—	—	16.704	3,600

Table 2. 2E-1P Formulation: Instances with 15 Customers (Either with Three Depots and Five Satellites or with Six Depots and Four Satellites)

Instance	LB	UB	Gap%	CPU(s)	Instance	LB	UB	Gap%	CPU(s)
Ca1-3-5-15	590.276	603.456	2.184	3,600	Ca1-6-4-15	551.457	551.457	0	2,485.8
Ca2-3-5-15	604.153	665.265	9.186	3,600	Ca2-6-4-15	554.189	573.343	3.341	3,600
Ca3-3-5-15	481.435	533.263	9.719	3,600	Ca3-6-4-15	556.642	556.642	0	1,460.7
Ca4-3-5-15	514.399	514.399	0	316.5	Ca4-6-4-15	465.226	465.226	0	8.2
Ca5-3-5-15	503.775	537.805	6.328	3,600	Ca5-6-4-15	416.632	416.632	0	7.2
Cb1-3-5-15	532.340	532.340	0	6.7	Cb1-6-4-15	558.886	567.151	1.457	3,600
Cb2-3-5-15	574.780	612.606	6.175	3,600	Cb2-6-4-15	631.512	631.512	0	28.5
Cb3-3-5-15	488.654	540.482	9.589	3,600	Cb3-6-4-15	545.104	561.536	2.926	3,600
Cb4-3-5-15	438.674	438.674	0	17.4	Cb4-6-4-15	510.954	510.954	0	6.3
Cb5-3-5-15	499.227	538.904	7.363	3,600	Cb5-6-4-15	460.156	548.742	16.143	3,600
Cc1-3-5-15	473.553	473.553	0	122.1	Cc1-6-4-15	556.676	601.575	7.464	3,600
Cc2-3-5-15	494.727	494.727	0	78.2	Cc2-6-4-15	540.631	575.550	6.067	3,600
Cc3-3-5-15	472.158	540.432	12.633	3,600	Cc3-6-4-15	536.259	540.875	0.853	3,600
Cc4-3-5-15	466.209	518.545	10.093	3,600	Cc4-6-4-15	513.800	547.350	6.130	3,600
Cc5-3-5-15	520.063	650.872	20.097	3,600	Cc5-6-4-15	425.830	425.830	0	45.7
Cd1-3-5-15	535.604	968.446	44.694	3,600	Cd1-6-4-15	544.169	551.492	1.328	3,600
Cd2-3-5-15	506.612	506.612	0	43.2	Cd2-6-4-15	540.100	554.830	2.655	3,600
Cd3-3-5-15	474.118	534.214	11.249	3,600	Cd3-6-4-15	559.021	559.021	0	794.3
Cd4-3-5-15	471.060	512.287	8.048	3,600	Cd4-6-4-15	491.521	535.492	8.211	3,600
Cd5-3-5-15	515.207	544.296	5.344	3,600	Cd5-6-4-15	442.518	506.695	12.666	3,600
Average			8.135	2,549.2	Average			3.462	2,401.8

solve and also to check the effect of labeling approaches. We compare the performance of proposed algorithms and study the effect of instance characteristics on the algorithms' performance, with several experiments. We observe that the branch-and-price-based algorithm for the 2E-2P formulation outperforms the branch-and-price algorithm for the 2E-1P formulation. Therefore, we test the latter one only on instances with 15 customers, owing to the long central processing unit (CPU) time for all other instances. The branch-and-price-based algorithm for the 2E-2P formulation is tested on all instances. Throughout this section, the CPU times are reported in seconds.

7.2.1. Results for the 2E-1P Formulation. Table 1 gives the results for the 2E-1P formulation and the results of the CPLEX solver for the arc-based formulation for instances with two depots, three satellites, and 15 customers. We impose a time limit of 1 hour for CPU time. The upper bound, lower bound, and the achieved gap (in percentage) are reported in columns "UB," "LB," and "Gap%," respectively. The branch-and-price algorithm for the 2E-1P formulation outperforms the CPLEX for the arc-based formulation. The latter fails to solve any of the instances to optimality and gives an average gap of 16.704%.

7.2.2. Effect of Labeling Approaches. We compare the effect of different labeling approaches, namely the bidirectional breadth-first labeling approach and the monodirectional depth-first labeling approach, denoted by "Breadth-First" and "Depth First." Table 1 shows

that the monodirectional depth-first approach solves more instances to optimality in a shorter time compared with the bidirectional breadth-first approach. Moreover, the bidirectional breadth-first approach fails to find any upper bound for two instances. This is mainly because the tour-trees are very long and the bidirectional breadth-first approach generates lots of labels, for which it takes a long time to check which labels can be merged.

Table 2 shows the results of the monodirectional depth-first approach for the remaining instances with 15 customers. The algorithm solves 25 instances out of 60 instances to optimality. The average gap for the remaining instances is 0.08% with a maximum CPU time of 1 hour. All in all, the 2E-1P formulation is limited in terms of its performance. Observe that the instances with wide time windows (i.e., the "Cc" category) are more time-consuming to solve owing to long tour-trees, which makes the labeling algorithm take more time. Instances with a large number of customers lead to a large number of tour-trees, which makes the column generation more time-consuming. Another challenge is that the large number of depots and satellites increases the number of subproblems that should be considered to solve the pricing problem.

7.2.3. Results for the 2E-2P Formulation. After extensive experiments with different values for θ , it was concluded that the value of θ has almost no influence on the final performance of the branch-and-price-based algorithm for the 2E-2P formulation. This is mainly because of the possibility of moving to set of

Table 3. 2E-2P Formulation: Instances with 15 Customers

Instance	Obj.	CPU(s)	B&B N.	#UB
$ P = 2, S = 3, Z = 15$				
Ca1-2-3-15	612.385	0.086	11	4
Ca2-2-3-15	548.953	0.065	10	3
Ca3-2-3-15	551.985	0.292	60	3
Ca4-2-3-15	569.579	0.061	7	3
Ca5-2-3-15	555.796	0.057	6	7
Cb1-2-3-15	624.178	0.071	6	4
Cb2-2-3-15	516.739	0.059	6	4
Cb3-2-3-15	601.897	0.023	5	6
Cb4-2-3-15	546.314	0.171	9	3
Cb5-2-3-15	494.395	0.260	35	5
Cc1-2-3-15	586.856	1.033	90	7
Cc2-2-3-15	482.985	0.103	4	2
Cc3-2-3-15	539.685	0.257	16	6
Cc4-2-3-15	562.798	0.344	32	8
Cc5-2-3-15	436.467	1.772	59	4
Cd1-2-3-15	597.698	0.083	7	2
Cd2-2-3-15	483.133	0.103	5	5
Cd3-2-3-15	512.412	0.047	6	2
Cd4-2-3-15	585.301	0.068	11	5
Cd5-2-3-15	536.764	0.085	13	2
Average		0.252		
$ P = 3, S = 5, Z = 15$				
Ca1-3-5-15	603.456	2.737	10	6
Ca3-3-5-15	628.405	5.073	212	3
Ca5-3-5-15	527.679	0.766	12	6
Ca4-3-5-15	514.399	0.304	45	4
Ca5-3-5-15	521.399	3.137	185	7
Cb1-3-5-15	532.34	0.070	6	2
Cb3-3-5-15	611.574	3.848	34	7
Cb5-3-5-15	537.256	0.926	6	3
Cb4-3-5-15	438.674	0.178	7	4
Cb5-3-5-15	538.57	2.513	20	3
Cc1-3-5-15	473.553	0.825	12	5
Cc3-3-5-15	494.727	0.900	30	6
Cc5-3-5-15	522.837	2.795	11	4
Cc4-3-5-15	506.35	2.545	14	4
Cc5-3-5-15	545.077	3.264	58	6
Cd1-3-5-15	568.629	3.240	42	4
Cd3-3-5-15	506.612	0.403	22	5
Cd5-3-5-15	522.8	2.763	15	4
Cd4-3-5-15	506.825	2.883	44	7
Cd5-3-5-15	544.084	2.451	32	6
Average		2.081		
$ P = 6, S = 4, Z = 15$				
Ca1-6-4-15	551.457	0.163	11	4
Ca2-6-4-15	560.919	0.378	31	6
Ca3-6-4-15	556.642	0.476	29	3
Ca4-6-4-15	465.226	0.326	22	5
Ca5-6-4-15	416.632	0.139	5	3
Cb1-6-4-15	567.151	0.116	8	4
Cb2-6-4-15	631.512	0.125	28	6
Cb3-6-4-15	561.536	0.226	25	3
Cb4-6-4-15	510.954	0.118	12	4
Cb5-6-4-15	460.569	0.267	24	4
Cc1-6-4-15	566.013	0.200	7	6
Cc2-6-4-15	549.229	0.418	16	2
Cc3-6-4-15	540.875	0.320	7	6
Cc4-6-4-15	521.621	0.868	66	5
Cc5-6-4-15	425.83	0.365	8	4
Cd1-6-4-15	551.492	0.135	6	2
Cd2-6-4-15	554.83	0.197	18	1

Table 3. (Continued)

$ P = 6, S = 4, Z = 15$				
Cd3-6-4-15	558.714	0.193	8	1
Cd4-6-4-15	507.891	0.607	58	4
Cd5-6-4-15	488.801	0.386	30	6
Average		0.301		

next θ -best configurations when required. In this regard, we put $\theta = 5000$ for all instances to start with relatively enough initial configurations.

In Tables 3–6, we report the results of the 2E-2P formulation. The columns denoted as “Obj.” and “CPU(s)” indicate the optimal objective function value and the time (in seconds) spent to solve an instance. Note that this time includes the time spent to generate urban vehicle tours. “B&B N.” and “#UB” show the size of the branch and bound tree (number of branches) and the number of feasible solutions found (i.e., how many times the upper bound is updated).

7.2.4. Performance of 2E-1P and 2E-2P Formulations.

Comparing the results of instances with 15 customers in Tables 1–3 shows that the branch-and-price-based algorithm for the 2E-2P formulation outperforms the branch-and-price algorithm for the 2E-1P formulation. The 2E-1P formulation is not able to solve almost half of the 15-Z instances to optimality, and for the other half the CPU time could go up to 40 minutes (e.g., “Ca1-6-4-15”). However, using the 2E-2P formulation, we are able to solve all instances with 15 customers to optimality in a reasonable CPU time. The average CPU time is 0.878 seconds. This difference is mainly because of the structure of the pricing problems. The pricing problem of the 2E-1P formulation (which combines both echelon routings) is hard to solve, whereas the pricing problem of the 2E-2P formulation is concerned only with the second echelon and is easier to solve.

7.2.5. Effect of Instance Characteristics: Time Window, Number of Depots, and Number of Satellites.

It is observed that the instances of the “Cc” category are more time-consuming to solve because of the wide time windows, which allows for longer city freighter tours. The number of depots is not playing a critical role in the complexity of the problem. However, as expected, instances with more satellites are more complex. Therefore the instances with three depots and five satellites are more time-consuming to solve. It is observed that the CPU time has a direct linear relationship with the size of the branch and bound tree.

For all instances with two depots, three satellites, and 50 customers, we are able to solve 14 instances out of 20 instances up to optimality in an average CPU time of 733 seconds. For the remaining instances, the average gap is 0.603%. Observe that for these instances, the

Table 4. 2E-2P Formulation: Instances with 30 Customers

Instance	Obj.	CPU(s)	B&B N.	#UB
$ P = 2, S = 3, Z = 30$				
Ca1-2-3-30	981.374	353.2	6,855	6
Ca2-2-3-30	968.324	2.6	30	4
Ca3-2-3-30	1,069.81	0.9	8	5
Ca4-2-3-30	948.047	4.2	31	6
Ca5-2-3-30	968.231	9.5	60	6
Cb1-2-3-30	985.763	5.7	79	7
Cb2-2-3-30	937.119	9.2	45	2
Cb3-2-3-30	1,076.254	1.1	9	1
Cb4-2-3-30	981.344	2	17	1
Cb5-2-3-30	834.854	7.7	19	2
Cc1-2-3-30	799.562	16.2	7	3
Cc2-2-3-30	937.712	10.8	50	6
Cc3-2-3-30	1,013.546	562.3	2,747	7
Cc4-2-3-30	943.279	130.9	492	6
Cc5-2-3-30	949.387	25.6	68	3
Cd1-2-3-30	807.247	3.1	6	4
Cd2-2-3-30	945.304	4.1	29	4
Cd3-2-3-30	1,030.606	9.6	79	6
Cd4-2-3-30	958.674	19.8	176	1
Cd5-2-3-30	965.894	55.5	286	3
Average		61.7		
$ P = 3, S = 5, Z = 30$				
Ca1-3-5-30	892.746	18.6	127	4
Ca3-3-5-30	960.533	24.6	90	5
Ca5-3-5-30	924.509	17.4	15	3
Ca4-3-5-30	909.869	18.3	70	1
Ca5-3-5-30	892.66	15.4	15	5
Cb1-3-5-30	928.046	48.1	591	1
Cb3-3-5-30	958.302	73.2	449	4
Cb5-3-5-30	900.905	8.4	25	1
Cb4-3-5-30	903.874	23.9	121	3
Cb5-3-5-30	977.906	45.5	190	2
Cc1-3-5-30	943.74	62.2	732	2
Cc2-3-5-30	897.767	65.2	163	4
Cc3-3-5-30	943.295	29.5	110	5
Cc4-3-5-30	890.652	325.5	514	6
Cc5-3-5-30	772.331	306.6	149	4
Cd1-3-5-30	955.083	36.8	621	2
Cd3-3-5-30	944.603	38.3	168	3
Cd5-3-5-30	907.529	29.2	203	2
Cd4-3-5-30	917.202	95	459	2
Cd5-3-5-30	917.008	83.7	224	2
Average		68.3		
$ P = 6, S = 4, Z = 30$				
Ca1-6-4-30	922.215	919.4	9,139	8
Ca2-6-4-30	966.855	2.4	7	1
Ca3-6-4-30	869.334	4.9	27	4
Ca4-6-4-30	867.416	11.4	40	5
Ca5-6-4-30	851.516	1.9	9	2
Cb1-6-4-30	968.054	4.6	57	4
Cb2-6-4-30	920.495	3.1	12	7
Cb3-6-4-30	929.483	3.6	36	2
Cb4-6-4-30	877.54	12.8	70	3
Cb5-6-4-30	895.742	9.3	86	2
Cc1-6-4-30	861.127	39.9	142	5
Cc2-6-4-30	912.379	10.4	41	3
Cc3-6-4-30	899.833	20.1	89	1
Cc4-6-4-30	833.637	190	230	6
Cc5-6-4-30	813.91	476.1	1,406	7
Cd1-6-4-30	918.387	21.6	347	3
Cd2-6-4-30	914.267	3.7	16	3

Table 4. (Continued)

$ P = 6, S = 4, Z = 30$				
Cd3-6-4-30	880.593	8.4	37	2
Cd4-6-4-30	885.413	12.6	73	4
Cd5-6-4-30	943.678	16.2	320	6
Average		88.6		

value of “#UB” is larger, which shows that the algorithm finds more feasible solutions and spends more time to update the upper bound. For instances with three depots, five satellites, and 50 customers, 19 instances out of 20 are solved to optimality in an average CPU time of 1,269 seconds. Similarly, for instances with six depots, four satellites, and 50 customers, we are able to solve 18 instances out of 20 to optimality in an average CPU time of 1,526 seconds. Instances with 100 customers are more complex to solve, and we are able to solve 11 instances out of 60 instances. Six of these optimally solved instances are instances with two depots and three satellites. The average gap for the remaining instances with two depots and three satellites is 0.721%. For instances with three depots, five satellites, and 100 customers the average gap is 0.654%. Similarly, for instances with six depots, four satellites, and 100 customers the average gap is 0.415%. As the number of depots and satellites increases, the problem is more complex to solve.

8. Conclusions and Future Research

In this paper, we introduced and modeled the 2E-VRPTW in different ways. Two path-based formulations are proposed, in which the 2E-1P formulation uses an integrated approach to define interconnected first- and second-echelon tours as the components of the problem. The 2E-2P formulation defines separate urban vehicle tours and city freighter tours as the components of the problem, whereby the interconnectivity of them is assured using explicit connecting constraints. We designed an exact branch-and-price algorithm that uses the 2E-1P formulation to define the tour-trees as columns, whereby a labeling approach is used to generate columns. Both monodirectional and bi-directional labeling approaches are tested. Moreover, we designed a branch-and-price-based algorithm, based on the 2E-2P formulation, which solves the problem by investigating different first-echelon routing solutions and searching for the optimal city freighter tours. This algorithm works on the set of all urban vehicle tours. The urban vehicle tour enumeration is not too time-consuming if the number of depots and satellites is not too large. For larger numbers of depots and satellites an efficient heuristic approach can be investigated to generate a limited number of urban vehicle tours, which is a topic for further research. Designing this algorithm is the most important contribution

Table 5. 2E-2P Formulation: Instances with 50 Customers

Instance	LB	UB	Gap%	CPU(s)	B&B N.	#UB
$ P = 2, S = 3, Z = 50$						
Ca1-2-3-50	1,501.80	1,501.80	0	2.7	8	3
Ca2-2-3-50	1,579.53	1,586.21	0.421	10,800	20,296	5
Ca3-2-3-50	1,479.56	1,479.56	0	40.2	149	1
Ca4-2-3-50	1,485.70	1,485.70	0	3,427.1	6,449	4
Ca5-2-3-50	1,397.05	1,397.05	0	195.8	477	1
Cb1-2-3-50	1,455.11	1,455.11	0	27.4	45	2
Cb2-2-3-50	1,468.47	1,468.47	0	840.8	1,170	4
Cb3-2-3-50	1,492.25	1,492.25	0	3,365.1	3,419	5
Cb4-2-3-50	1,469.01	1,469.01	0	566.2	618	2
Cb5-2-3-50	1,415.23	1,415.23	0	1,147.9	1,891	3
Cc1-2-3-50	1,403.49	1,411.24	0.549	10,800	7,824	15
Cc2-2-3-50	1,466.71	1,468.88	0.148	10,800	3,462	12
Cc3-2-3-50	1,366.31	1,381.33	1.087	10,800	3,507	8
Cc4-2-3-50	1,411.20	1,427.86	1.167	10,800	3,397	10
Cc5-2-3-50	1,442.65	1,442.65	0	326.8	187	5
Cd1-2-3-50	1,458.04	1,458.04	0	30.7	44	7
Cd2-2-3-50	1,566.42	1,570.43	0.255	10,800	21,138	3
Cd3-2-3-50	1,449.70	1,449.70	0	133.4	384	2
Cd4-2-3-50	1,453.24	1,453.24	0	127.9	115	1
Cd5-2-3-50	1,462.30	1,462.30	0	31	83	1
Average			0.181	3,753.1		
$ P = 3, S = 5, Z = 50$						
Ca1-3-5-50	1,351.83	1,351.83	0.0	662.9	1,430	12
Ca2-3-5-50	1,354.12	1,354.12	0.0	172.1	291	2
Ca3-3-5-50	1,271.67	1,271.67	0.0	488.6	582	2
Ca4-3-5-50	1,299.86	1,299.86	0.0	371.1	756	4
Ca5-3-5-50	1,267.49	1,267.49	0.0	123.2	144	1
Cb1-3-5-50	1,355.50	1,355.50	0.0	631.5	872	1
Cb2-3-5-50	1,391.31	1,391.31	0.0	794.7	1,467	2
Cb3-3-5-50	1,368.97	1,368.97	0.0	847.4	1,150	1
Cb4-3-5-50	1,298.34	1,298.34	0.0	258	346	2
Cb5-3-5-50	1,270.39	1,270.39	0.0	462.3	324	1
Cc1-3-5-50	1,293.88	1,293.88	0.0	1,507.4	687	3
Cc2-3-5-50	1,317.82	1,317.82	0.0	150	47	1
Cc3-3-5-50	1,288.90	1,288.90	0.0	248	97	3
Cc4-3-5-50	1,342.39	1,348.84	0.478	10,800	9,362	7
Cc5-3-5-50	1,276.47	1,276.47	0.0	901.9	607	10
Cd1-3-5-50	1,314.59	1,314.59	0.0	4,997	5,984	6
Cd2-3-5-50	1,326.05	1,326.05	0.0	472.3	622	4
Cd3-3-5-50	1,351.50	1,351.50	0.0	8,526.9	12,464	6
Cd4-3-5-50	1,315.54	1,315.54	0.0	445.1	534	5
Cd5-3-5-50	1,289.71	1,289.71	0.0	2,068.8	4,189	5
Average			0.023	1,746.5		
$ P = 6, S = 4, Z = 50$						
Ca1-6-4-50	1,334.43	1,335.55	0.084	10,800	14,424	10
Ca2-6-4-50	1,376.13	1,376.13	0	5,749.5	8,488	5
Ca3-6-4-50	1,337.15	1,337.15	0	1,313.9	2,686	4
Ca4-6-4-50	1,242.56	1,242.56	0	29.6	92	3
Ca5-6-4-50	1,268.84	1,268.84	0	81	67	3
Cb1-6-4-50	1,355.92	1,355.92	0	63.8	182	1
Cb2-6-4-50	1,456.85	1,456.85	0	3,179.1	2,600	2
Cb3-6-4-50	1,345.02	1,345.02	0	75	146	1
Cb4-6-4-50	1,281.67	1,281.67	0	36.3	78	1
Cb5-6-4-50	1,341.25	1,341.25	0	244.4	370	3
Cc1-6-4-50	1,262.60	1,262.60	0	1,214	429	2
Cc2-6-4-50	1,304.68	1,304.68	0	7,221	2,414	8
Cc3-6-4-50	1,315.85	1,315.85	0	6,100	2,108	6
Cc4-6-4-50	1,204.31	1,206.64	0.193	10,800	8,171	10
Cc5-6-4-50	1,453.39	1,453.39	0	423.5	645	5
Cd1-6-4-50	1,300.53	1,300.53	0	59.1	71	4
Cd2-6-4-50	1,338.28	1,338.28	0	73.2	120	2

Table 5. (Continued)

$ P = 6, S = 4, Z = 50$						
Cd3-6-4-50	1,375.06	1,375.06	0	50.1	144	2
Cd4-6-4-50	1,236.79	1,236.79	0	341.1	728	3
Cd5-6-4-50	1,332.86	1,332.86	0	1,229.7	1,963	7
Average			0.014	2,454.2		

Table 6. 2E-2P Formulation: Instances with 100 Customers

Instance	LB	UB	Gap%	CPU(s)	B&B N.	#UB
$ P = 2, S = 3, Z = 100$						
Ca1-2-3-100	2,769.89	2,780.05	0.366	10,800	2,554	5
Ca2-2-3-100	2,861.80	2,863.57	0.062	10,800	4,200	10
Ca3-2-3-100	2,676.69	2,678.51	0.068	10,800	3,125	6
Ca4-2-3-100	3,037.91	3,037.91	0	8,647.7	3,236	5
Ca5-2-3-100	2,757.45	2,757.45	0	1,487.9	281	2
Cb1-2-3-100	2,478.67	2,500.31	0.866	10,800	148	3
Cb2-2-3-100	2,720.85	2,756.17	1.282	10,800	380	9
Cb3-2-3-100	2,706.98	2,734.22	0.996	10,800	328	4
Cb4-2-3-100	2,833.38	2,863.69	1.058	10,800	616	2
Cb5-2-3-100	2,785.22	2,808.35	0.824	10,800	314	8
Cc1-2-3-100	2,758.58	2,778.64	0.722	10,800	827	7
Cc2-2-3-100	2,639.09	2,664.51	0.954	10,800	737	10
Cc3-2-3-100	2,665.60	2,665.85	0.009	10,800	1,111	4
Cc4-2-3-100	2,878.72	2,898.31	0.676	10,800	1,927	8
Cc5-2-3-100	2,757.45	2,757.45	0	1,758.5	374	14
Cd1-2-3-100	2,666.54	2,666.54	0	497	118	3
Cd2-2-3-100	2,852.93	2,852.93	0	4,021.9	1,407	17
Cd3-2-3-100	2,616.30	2,638.30	0.834	10,800	2,275	20
Cd4-2-3-100	2,882.78	2,923.18	1.382	10,800	1,994	11
Cd5-2-3-100	2,794.66	2,794.66	0	1,269.4	330	8
Average			0.505	8,444.1		
$ P = 3, S = 5, Z = 100$						
Ca1-3-5-100	2,608.81	2,608.81	0	796.4	432	1
Ca2-3-5-100	2,468.67	2,485.70	0.685	10,800	1,378	4
Ca3-3-5-100	2,449.93	2,454.65	0.192	10,800	762	6
Ca4-3-5-100	2,644.52	2,649.56	0.190	10,800	2,476	6
Ca5-3-5-100	2,618.54	2,622.47	0.150	10,800	2,214	8
Cb1-3-5-100	2,634.84	2,652.34	0.660	10,800	1,065	6
Cb2-3-5-100	2,477.35	2,478.14	0.032	10,800	1,305	8
Cb3-3-5-100	2,528.77	2,559.93	1.217	10,800	622	6
Cb4-3-5-100	2,681.83	2,746.73	2.363	10,800	837	4
Cb5-3-5-100	2,735.03	2,777.09	1.514	10,800	1,013	4
Cc1-3-5-100	2,557.46	2,563.99	0.254	10,800	2,327	9
Cc2-3-5-100	2,425.35	2,438.28	0.531	10,800	852	11
Cc3-3-5-100	2,513.80	2,550.53	1.440	10,800	100	4
Cc4-3-5-100	2,566.61	2,572.92	0.245	10,800	700	7
Cc5-3-5-100	2,632.24	2,652.84	0.777	10,800	727	3
Cd1-3-5-100	2,530.20	2,530.20	0	4,985.9	2,766	3
Cd2-3-5-100	2,429.08	2,458.48	1.196	10,800	2,863	6
Cd3-3-5-100	2,523.66	2,523.66	0	2,352.4	716	7
Cd4-3-5-100	2,519.34	2,555.03	1.397	10,800	2,368	3
Cd5-3-5-100	2,663.88	2,670.25	0.238	10,800	2,156	5
Average			0.654	9,586.7		
$ P = 6, S = 4, Z = 100$						
Ca1-6-4-100	2,524.98	2,545.68	0.813	10,800	4,057	5
Ca2-6-4-100	2,380.64	2,380.64	0	429.1	71	1
Ca3-6-4-100	2,693.05	2,698.97	0.219	10,800	4,602	8
Ca4-6-4-100	2,373.88	2,375.87	0.083	10,800	1,794	6
Ca5-6-4-100	2,573.94	2,575.28	0.052	10,800	4,994	2
Cb1-6-4-100	2,361.55	2,386.04	1.027	10,800	518	8
Cb2-6-4-100	2,555.59	2,567.85	0.477	10,800	345	4
Cb3-6-4-100	2,629.39	2,641.34	0.452	10,800	1,374	7
Cb4-6-4-100	2,738.30	2,768.51	1.091	10,800	1,305	3

Table 6. (Continued)

$ P = 6, S = 4, Z = 100$						
Cb5-6-4-100	2,565.61	2,574.51	0.346	10,800	1,082	7
Cc1-6-4-100	2,510.37	2,539.31	1.140	10,800	1,338	11
Cc2-6-4-100	2,548.60	2,556.62	0.314	10,800	171	2
Cc3-6-4-100	2,376.64	2,391.93	0.639	10,800	1,088	9
Cc4-6-4-100	2,555.55	2,559.58	0.157	10,800	871	6
Cc5-6-4-100	2,457.58	2,461.53	0.160	10,800	1,686	8
Cd1-6-4-100	2,583.12	2,598.30	0.584	10,800	4,295	8
Cd2-6-4-100	2,535.57	2,535.57	0	5,934.1	2,078	4
Cd3-6-4-100	2,407.67	2,409.22	0.064	10,800	4,985	8
Cd4-6-4-100	2,668.64	2,681.10	0.464	10,800	4,364	10
Cd5-6-4-100	2,573.41	2,579.05	0.218	10,800	5,059	7
Average			0.415	10,038.2		

of this paper. It is observed that the 2E-2P formulation outperforms the 2E-1P formulation owing to the wise decomposition scheme, which decomposes the problem into easier subproblems. It succeeded in solving instances with up to five satellites and 100 customers to optimality. This number of satellites corresponds to what is typically seen in European medium-sized cities. This paper is the first paper in the literature that solves such large instance sizes.

Acknowledgments

While working on this project, the fourth author was adjunct professor with the Department of Computer Science and Operations Research of the Université de Montréal.

References

- Baldacci R, Mingozzi A, Calvo RW (2011) An exact method for the capacitated location-routing problem. *Oper. Res.* 59(5):1284–1296.
- Baldacci R, Mingozzi A, Roberti R, Calvo RW (2013) An exact algorithm for the two-echelon capacitated vehicle routing problem. *Oper. Res.* 61(2):298–314.
- Bektaş T, Crainic TG, Woensel TV (2015) From managing urban freight to smart city logistics networks. Technical report CIRRELT-2015-17, CIRRELT, Montréal.
- Breunig U, Schmid V, Hartl RF, Vidal T (2016) A large neighbourhood based heuristic for two-echelon routing problems. *Comput. Oper. Res.* 76(C):208–225.
- CIVITAS Initiative (2015) CIVITAS policy note—Making urban freight logistics more sustainable. Accessed August 1, 2016, <http://www.civitas.eu>.
- Crainic TG, Ricciardi N, Storchi G (2009) Models for evaluating and planning city logistics systems. *Transportation Sci.* 43(4):432–454.
- Crainic TG, Mancini S, Perboli G, Tadei R (2008) Clustering-based heuristics for the two-echelon vehicle routing problem. Technical report CIRRELT-2008-46, CIRRELT, Montréal.
- Crainic TG, Mancini S, Perboli G, Tadei R (2011a) Multi-start heuristics for the two-echelon vehicle routing problem. *European Conf. on Evolutionary Computation in Combinatorial Optimization* (Springer, New York), 179–190.
- Crainic TG, Mancini S, Perboli G, Tadei R (2013) Grasp with path re-linking for the two-echelon vehicle routing problem. Di Gaspero L, Schaerf A, Stützle T, eds. *Advances in Metaheuristics* (Springer, New York), 113–125.
- Crainic TG, Perboli G, Mancini S, Tadei R (2010) Two-echelon vehicle routing problem: A satellite location analysis. *Procedia Soc. Behav. Sci.* 2(3):5944–5955.
- Crainic TG, Ricciardi N, Storchi G (2004) Advanced freight transportation systems for congested urban areas. *Transportation Res. C Emerging Technol.* 12(2):119–137.
- Crainic TG, Sforza A, Sterle C (2011b) Location-routing models for two-echelon freight distribution system design. Technical report CIRRELT-2011-40, CIRRELT, Montréal.
- Cuda R, Guastaroba G, Speranza MG (2015) A survey on two-echelon routing problems. *Comput. Oper. Res.* 55(C):185–199.
- Desrochers M, Desrosiers J, Solomon M (1992) A new optimization algorithm for the vehicle routing problem with time windows. *Oper. Res.* 40(2):342–354.
- 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.
- Gonzalez-Feliu J (2008) Models and methods for the city logistics: The two-echelon capacitated vehicle routing problem. PhD thesis, Politecnico di Torino, Turin, Italy.
- Grangier P, Gendreau M, Lehuédé F, Rousseau LM (2016) An adaptive large neighborhood search for the two-echelon multiple-trip vehicle routing problem with satellite synchronization. *Eur. J. Oper. Res.* 254(1):80–91.
- Hemmelmayr VC, Cordeau JF, Crainic TG (2012) An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics. *Comput. Oper. Res.* 39(12):3215–3228.
- Jepsen M, Spoorendonk S, Ropke S (2013) A branch-and-cut algorithm for the symmetric two-echelon capacitated vehicle routing problem. *Transportation Sci.* 47(1):23–37.
- Li H, Zhang L, Lv T, Chang X (2016) The two-echelon time-constrained vehicle routing problem in linehaul-delivery systems. *Transportation Res. B Methodol.* 94:169–188.
- Lübbecke ME, Desrosiers J (2005) Selected topics in column generation. *Oper. Res.* 53(6):1007–1023.
- Mancini S (2013) Multi-echelon distribution systems in city logistics. *Eur. Transport* 54(2):1–24.
- Perboli G, Tadei R (2010) New families of valid inequalities for the two-echelon vehicle routing problem. *Electron. Notes Discrete Math.* 36:639–646.
- Perboli G, Tadei R, Vigo D (2011) The two-echelon capacitated vehicle routing problem: Models and math-based heuristics. *Transportation Sci.* 45(3):364–380.
- 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.
- Righini G, Salani M (2008) New dynamic programming algorithms for the resource constrained elementary shortest path problem. *Networks* 51(3):155–170.

- Santos FA, da Cunha AS, Mateus GR (2013) Branch-and-price algorithms for the two-echelon capacitated vehicle routing problem. *Optim. Lett.* 7(7):1537–1547.
- Santos FA, Mateus GR, da Cunha AS (2014) A branch-and-cut-and-price algorithm for the two-echelon capacitated vehicle routing problem. *Transportation Sci.* 49(2):355–368.
- Savelsbergh MWP, Woensel TV (2016) 50th anniversary invited article—city logistics: Challenges and opportunities. *Transportation Sci.* 50(2):579–590.
- Zeng ZY, Xu WS, Xu ZY, Shao WH (2014) A hybrid grasp+vnd heuristic for the two-echelon vehicle routing problem arising in city logistics. *Math. Probl. Engrg.* 2014:1–11.