CIRRELT

Centre interuniversitaire de recherche
sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre
on Enterprise Networks, Logistics and Transportation

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

Nico Dellaert
Fardin Dashty Saridarq
Tom Van Woensel
Teodor Gabriel Crainic

August 2016

CIRRELT-2016-45

Bureaux de Montréal :
Université de Montréal
Pavillon André-Aisenstadt
C.P. 6128, succursale Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
Pavillon Palasis-Prince
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

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

Nico Dellaert[1], Fardin Dashty Saridarq[1], Tom Van Woensel[1], Teodor Gabriel Crainic[2,*]

[1] School of Industrial Engineering, Eindhoven University of Technology, 5600MB Eindhoven, The Netherlands

[2] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and Department of Management and Technology, Université du Québec à Montréal, P.O. Box 8888, Station Centre-Ville, Montréal, Canada H3C 3P8

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), while 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) a path-based formulation, where paths are defined over both first and second echelon tours, and (2) a path-based formulation, in which the first and second echelon paths are decomposed. Branch-and-price based algorithms are developed for both formulations. We compare all formulations and solution methods on a comprehensive set of instances and are able to solve instances up to 5 satellites and 100 customers to optimality. This paper is the first paper in the literature which solves such large instance sizes.

Keywords: Two-echelon vehicle routing problem with time windows, branch-and-price, column generation.

Acknowledgements. This research has been funded by the Dutch Institute for Advanced Logistics (DINALOG), under the grant titled ConCoord. Partial funding for this project has been provided by the Natural Sciences and Engineering Council of Canada (NSERC), through its Discovery Grant program. We also gratefully acknowledge the support of Fonds de recherche du Québec through their infrastructure grants. 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.

---

* Corresponding author: teodorgabriel.crainic@cirrelt.net

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2016

1 Introduction

Consider a multi-echelon 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 due 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, while the second echelon consists of transferring freight from these facilities to the final customers. This problem arises in different applications such as multi-modal transportation systems, spare parts distribution, e-commerce, grocery and hyper market products distribution and home delivery services (Perboli et al. 2011). Another important application of the 2E-VRP arises in city logistics where a set of external depots which 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 et al. 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 et al. (2015) and in Savelsbergh and Woensel (2016).

To our knowledge, only a limited number of studies exist in the literature which 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 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. We describe the issues and define the problem in detail. We propose an arc-based mathematical formulation and two path-based mathematical formulations. The first path-based formulation is based on an integrated approach which defines interconnected first and second echelon routes as the components of the problem. In contrast, the second path-based formulation exploits

the structure of the problem to define separate first echelon routes and second echelon routes as the components of the problem where the interconnectivity of the first and second echelon routing problems are assured employing extra connectivity constraints. We propose branch-and-price-based algorithms. Each branch-and-price algorithm uses a column generation approach based on specifically designed labeling algorithms. An extensive computational study is performed using new instances demonstrating that the proposed algorithms are successful both in terms of the quality of the achieved (near) optimal solutions and the size of the solved instances.

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 outline a number of alternative efficient 2E-VRPTW path formulations for the problem on-hand. For each model 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*). For this formulation, an exact branch-and-price algorithm is proposed where a special pricing problem is obtained, handled by a specifically designed labeling algorithm.
   (b) A second path-based formulation is designed where the two echelon routes are separated (denoted as the two-echelon two-path formulation, or *2E-2P*). By using this formulation, the paths to be generated are shorter, but part of the complexity moves to the first and second echelon paths that need to be re-connected again in this formulation. A branch-and-price-based algorithm is proposed where 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 5 satellites. These instance sizes correspond to the two-echelon city logistics systems in medium-sized cities, where 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. In Section 4, we present the

2E-1P formulation and the branch-and-price algorithm. In Section 5, 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 6, followed by conclusions and suggestions for future research in Section 7.

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. The basic variant of the 2E-VRP is addressed in few papers, where multiple models and solution approaches are proposed. The temporal aspects of the problem and particularly specification of time windows for customers are only mentioned and discussed in a few papers (e.g., Crainic et al. (2009) and Mancini (2013)). However, the 2E-VRPTW in our paper on-hand is not treated explicitly from an exact optimization point of view and no formal descriptions exist yet in the literature. We introduce the 2E-VRPTW, propose multiple mathematical formulations and exact solution methods.

The literature on the 2E-VRP started few years ago 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, multi-depot, 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 et al. (2011), Crainic et al. (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 (TSP) and the capacitated vehicle routing problem (CVRP), the network flow formulation, and the connectivity of the transportation system graph. Perboli et al. (2011) presented a new mathematical model for 2E-VRP, a number of valid inequalities, and two math-heuristics based on the proposed model.

Limited number of exact algorithms for the problem 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 5 for city logistic applications, but because of number of customers (meta)heuristic approaches should be investigated. The reported gap for the medium-sized instances is around 12 percent. Jepsen et al. (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 5 satellites and 50 customers. Another exact algorithm is proposed by Baldacci et al. (2013) where, the problem is decomposed into a limited set of multi-depot 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 et al. (2014) and Santos et al. (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 5 satellites and 50 customers.

Due to complexity of the problem heuristics are also investigated for the problem. For example, Crainic et al. (2011a) present a family of multi-start 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 clustering and multi-depot approach. A meta-heuristic based on greedy randomized adaptive search procedure (GRASP) combined with path relinking has been proposed in Crainic et al. (2013). A hybrid GRASP with a variable neighborhood descent (VND) heuristic is proposed by Zeng et al. (2014). Hemmelmayr et al. (2012) propose an adaptive large neighborhood search heuristic.

Another problem seen in the literature, similar to the 2E-VRP, is the truck and trailer routing problem (TTRP). The TTRP consists of determining routes of truck and trailers to supply customers where each truck pulls a trailer which is detachable. Some customers are allowed to be visited by truck alone and the remaining customers can be served either by a full vehicle (i.e., a truck pulling a trailer) or by a truck alone. Thus, a truck may park its trailer in a location to continue its route toward a customer which can be visited only by a truck alone. Although the nature of the TTRP is different than the 2E-VRP, it can be considered as a two-echelon routing problem where some customers are served by a direct shipment by a first echelon vehicle (truck and trailer). We refer the interested reader to a recent survey on the two-echelon routing problems (Cuda et al. 2015).

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), i.e., the transportation cost (distance) for moving a vehicle form 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 neither before nor 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 both satellites and 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 $\upsilon \in \mathcal{V}$ 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 3 depots, 4 satellites and 8 customers.

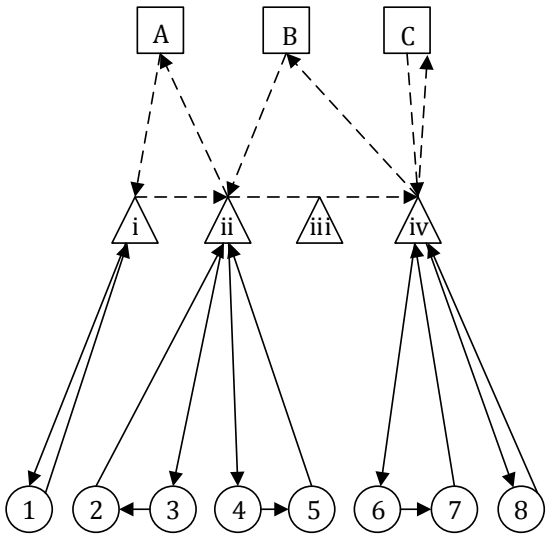


Figure 1: A 2E-VRPTW instance

In Appendix 8.1, we present an explicit arc-based three-index mixed integer programming problem for the 2E-VRPTW, inspired by ideas from Crainic et al. (2011b). As usual, the arc-based formulations require the definition of a large number of decision variables and constraints. This also makes our formulations hard to solve using an industrial solver, like CPLEX.

It is known from the literature that although set covering formulations have a huge number of variables for the paths, it can account for a large variety of quite complex routing requirements (Crainic et al. 2011b). In the remainder of this paper, we present a set partitioning formulation for the problem, called two-echelon one-path set formulation (2E-1P). This formulation is based on specifying only one set of paths, where each path includes routes for both echelons of the problem (see Section 4). Additionally, we propose an alternative set covering formulation exploiting the structure of the problem, called the two-echelon two-path sets formulation (2E-2P). This formulation is based on specifying the urban vehicle tours and the city freighter tours (see Section 5). This path-based formulation is inspired based on the ideas from the path-based formulation as proposed by Baldacci et al. (2013) for the 2E-VRP.

4 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. The associated urban vehicle tour starts from (and ends at) a depot and visits a subset of satellites. From each of these satellites, at least one city freighter tour starts (and ends) visiting a subset of customers. Note that a city freighter can depart from a satellite only after the associated urban vehicle has arrived to the satellite and the unloading and loading of goods are performed within a service time. A solution to the 2E-VRPTW consists of feasible tour-trees where each customer is served by exactly one tour-tree. Figure 1 consists of 3 urban vehicle tours and 5 city freighter tours, where 4 of these city freighter tours may have 2 different options for their corresponding urban vehicle tour, if this is time-wise and capacity-wise feasible. There are multiple options to combine these city freighter tours with urban vehicle tours and build tour-trees. For example, one can represent this solution by 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 constant that, $\alpha_{rz} = 1$ 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).

Based on these definitions, the 2E-1P formulation can be written as the following set partitioning problem:

$$\text{minimize} \quad \sum_{r\in R} c_r y_r \tag{1}$$

$$\text{subject to} \quad \sum_{r\in R} \alpha_{rz} y_r = 1, \qquad \forall z \in Z \tag{2}$$

$$y_r \in \{0, 1\}, \qquad \forall r \in R \tag{3}$$

Objective function (1) minimizes the total cost of the used tour-trees. Constraints (2) ensure each customer is served by one tour-tree. Constraints (3) are the domain constraint for the decision variables. We do not have to include capacity and time window constraints here, as we only consider feasible tour-trees.

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 LP-relaxation of problem (1 - 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 (2). Then, the pricing problem, aiming to generate columns with the most negative reduced cost, is formulated as follows:

$$\min_{r \in R} \{\bar{c_r} = c_r - \sum_{z \in Z} \alpha_{rz} \lambda_z\} \tag{4}$$

In the following sections, we present the used labeling algorithm for the pricing problem, the used branching strategy and our tree exploring strategy. We also presents the limits of the proposed algorithm and the need of new alternative solution methods to solve the problem. 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 Appendix (see Appendix 8.3).

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. In order to solve this problem, we propose a labeling algorithm which 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 is performed, where labels are extended both in 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. In a similar way, 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 employed when extending labels. Using the breadth-first search method leads to generating lots of labels which are not very long (with respect to the number of visited nodes). Additionally, in order to speed up the labeling algorithm a depth-first search method is designed. This method uses a greedy approach to find a tour-tree in a shorter time. Whenever processing a label, it is

extended toward the cheapest next label. Thus, an immediate reduction in the reduced cost is aimed which results in extending labels more quickly towards a tour-tree. At the end of the procedure, the label representing a tour-tree with the best reduced cost is reported as the nominated column for the algorithm.

Branching strategy

The branching strategy aims to select a pair of customer-customer or satellite-customer or a tour-tree such that branching on it would improve the lower bounds of the new children nodes. Let R be the set of all existing tour-trees for the current node. Moreover, let Z^* and $y_r^*, r \in R$, be the objective function and the computed LP-relaxation solution of the restricted master problem (RMP) of the problem (1 - 3) at the current node respectively. Furthermore, for each pair $\iota = (i, j)$, let $R_\iota \subset R$ show the set of columns where either i and j are served together in a column or none of them is in a column. In a similar way, $R_{\neg\iota} \subset R$ is the set of columns where in any column either i or j, but not both, is served. Let Λ be the set of all pairs. We initially, search for best columns or pairs which are potentially good options to be used in branching. Such columns are found using the most infeasible branching strategy. We choose the column (variable) with the fractional part closest to 0.5. Thus, the candidate columns are calculated by $\arg\max_{r\in R}(y_r^*)(1-y_r^*)$. A similar idea is used to find the candidate pairs. Therefore, the solution of $\arg\max_{\iota\in\Lambda}(\sum_{r\in R_\iota} y_r).(\sum_{r\in R_{\neg\iota}} y_r)$ is considered as the best candidate pair.

The final candidate (a pair or a column) is selected using the strong branching strategy. To reduce the computational cost, we find the candidate which gives the best improvement to the objective function by solving each of the corresponding LP-relaxations only with existing columns.

Tree exploring strategy

Typically, the depth-first strategy is used for branch and bound solution methods for the vehicle routing problems. Although this strategy is useful to reach some leaves and generate upper bounds early, it may lead to exploring a large number of nodes before finding the optimal solution. On the other hand the best first strategy, which selects the branch with the best lower bound value, may end up exploring lots of nodes without reaching leaves and generating good upper bounds. It is reasonable to mix these two strategies to use the benefits of both strategies. Three criteria to select a candidate node b as the next branch to explore are considered:

1. The lower bound value of the node (denoted by $lb(b)$).

2. The number of columns with non-fractional value in the optimal solution of the associated LP-relaxation problem (denoted by $N_f(b)$).

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.

Limits of the algorithm

The proposed branch-and-price algorithm is applicable for the problem without any assumption on size of the problem in terms of number of depots, satellites and customers. However, this algorithm has some limits for large problem settings. Large instances with large number of customers yield in long tour-trees which makes the labeling algorithm take more time. Another challenge is that large number of depots and satellites increase the number of subproblems which should be considered to solve the pricing problem to find the tour-tree with the most negative reduced cost value. As discussed before, such subproblems are considered for any particular pair of depot and satellite where the depot is the start (and end) point and the satellite is the first visited satellite in the associated tour-tree. Apart from these challenges, the proposed algorithm forms the basis for research in exact algorithms for the problem. We numerically examine how large instances could be solved by this algorithm in Section 6.

5 The two-echelon two-path formulation

Instead of formulating the 2E-VRPTW on a complete tour tree as we did in Section 4, we look into an alternative formulation where first and second echelon tours are decomposed.

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, where 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 $\mathscr{M}$ be the set of all urban vehicle tours and let $A(m)$ be the subset of arcs traversed by the urban vehicle tour $m \in \mathscr{M}$. An urban vehicle tour $m \in \mathscr{M}$ 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 $m \in \mathscr{M}$ is $c_m = \sum_{(i,j)\in A(m)} c_{ij} + h^1$.

Let $\mathscr{L}_m$ be the set of all feasible city freighter tours which 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 $\mathscr{L} = \cup_{m\in\mathscr{M}} \mathscr{L}_m$ 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 constant that denotes if customer z is visited by the city freighter tour $l \in \mathscr{L}$. Moreover, let $A(l)$ be the subset of arcs traversed by the city freighter tour $l \in \mathscr{L}$. A load $\mathscr{D}_l = \sum_{z\in Z} \xi_{lz} d_z$ 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 $l \in \mathscr{L}_m$ starts from a satellite $\S(l) \in S(m)$ at time $t^m_{\S(l)} + s_{\S(l)}$, delivers demands of subset of customers, and ends at the same satellite.

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

$$\text{minimize} \quad \sum_{m\in\mathscr{M}} c_m \mathscr{X}_m + \sum_{l\in\mathscr{L}} c_l \mathscr{Y}_l \tag{5}$$

$$\text{subject to} \quad \sum_{l\in\mathscr{L}} \xi_{lz} \mathscr{Y}_l = 1, \qquad \forall z \in Z \tag{6}$$

$$\sum_{l\in\mathscr{L}_m} \mathscr{D}_l \mathscr{Y}_l \leq K^1 \mathscr{X}_m, \qquad \forall m \in \mathscr{M} \tag{7}$$

$$\mathscr{X}_m \in \{0,1\}, \qquad \forall m \in \mathscr{M} \tag{8}$$

$$\mathscr{Y}_l \in \{0,1\}, \qquad \forall l \in \mathscr{L} \tag{9}$$

The objective function (5) minimizes the total first and second echelon transportation costs and the vehicle fixed costs. Constraints (6) ensure each customer is served by one city freighter tour. Constraints (7) ensure that the city freighter tours are selected such that the capacity of their associated urban vehicle tour is respected. Constraints (8)

and (9) are the domains of the decision variables. Again, we do not include the (city freighter) capacity and time window constraints here, as we only consider feasible city freighter tours.

Note that this formulation presents a number of challenges. The network flow formulation for the pricing problem to generate city freighter tours is a mixed integer *nonlinear* programming problem (due to the dependency between first and second echelon) with a nonlinear objective function and linear constraints. This non-linear programming problem proved to be hard to solve. Additionally, this formulation does not give any information to solve a pricing problem for finding urban vehicle tours with negative reduced cost. Please refer to Appendix 8.4 for more information on the algorithmic challenges of the above formulation.

Reformulation

In this section, we propose an alternative reformulation, based on the 2E-2P. Considering that in real-life applications the number of satellites is not too large (CIVITAS Initiative 2015), we can enumerate the set of all urban vehicle tours. In this respect, this assumption is similar to Baldacci et al. (2013) and Santos et al. (2013).

Let $\mathscr{P} = \{M \subset \mathscr{M} : |M|K^1 \geq \sum_{z \in Z} d_z\}$. Each element $M \in \mathscr{P}$ is called a configuration. Each configuration $M \in \mathscr{P}$ is defined as a solution for the first echelon routing problem. It thus combines a number of first echelon routes into a complete first echelon solution. We calculate the total (first echelon) transportation cost of the configuration M as $c(M) = \sum_{m \in M} c_m$. Moreover, $S(M) = \cup_{m \in M} S(m)$ is the set of satellites visited by urban vehicle tours of the configuration M. The above formulation for the 2E-VRPTW is then equivalent to $\min_{M \in \mathscr{P}}\{c(M) + z(SP(M))\}$, where $z(SP(M))$ is the objective function value of the optimal solution for the following problem denoted as $SP(M)$:

$$\text{minimize} \quad \sum_{l \in \mathscr{L}} c_l \mathscr{y}_l \tag{10}$$

$$\text{subject to} \quad \sum_{l \in \mathscr{L}} \xi_{lz} \mathscr{y}_l = 1, \qquad \forall z \in Z \tag{11}$$

$$\sum_{l \in \mathscr{L}_m} \mathscr{D}_l \mathscr{y}_l \leq K^1, \qquad \forall m \in M \tag{12}$$

$$\mathscr{y}_l \in \{0, 1\}, \qquad \forall l \in \mathscr{L} \tag{13}$$

For a given configuration $M \in \mathscr{P}$, a feasible solution for the $SP(M)$ consists of feasible city freighter tours starting (and ending) at satellites $S(M)$. A column generation

procedure can be used to solve the LP-relaxation of problem (10 - 13): 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 (11) and δ_m, $m \in M$ be a dual variable associated with constraints (12). Then, the pricing subproblem can be formulated as follows:

$$\min_{l \in L} \{\bar{c}_l = c_l - \sum_{z \in Z} \xi_{lz} \zeta_z + \sum_{m \in M} d_l \delta_m\} \tag{14}$$

Solution algorithm

In this section, we propose a branch-and-price-based algorithm based on the described reformulation of the 2E-2P formulation. The main idea is to first fix the first echelon routing solution to a particular configuration $M \in \mathscr{P}$ and then solve $SP(M)$ problem in a branch-and-price framework. To this end, we enumerate the set of all urban vehicle tours, $\mathscr{M}$, to combine multiple urban vehicles tours and generate different configurations. If the number of depots and satellites is not too large, the enumeration of urban vehicle tours is not too time-consuming. For larger number of depots and satellites an efficient heuristic approach can be investigated to generate a limited number of urban vehicle tours and prevent an exhaustive search of all possible tours. However, this yields to a branch-and-price-based matheuristic which is beyond the scope of the present paper.

In the following sections, we describe the algorithm and discuss the cases. The proposed branch-and-price-based algorithm uses the same branching strategy and tree exploring strategy as described in Section 4.

Outline of the algorithm

The proposed algorithm consists of the following steps:

1. Generate the set of all urban vehicle tours $\mathscr{M}$ (Exhaustive search with dominance)
2. Generate the set of sorted θ-best configurations (based upon their lower bound) by combining different urban vehicle tours from the set $\mathscr{M}$ (Labeling algorithm)
3. For each generated configuration, initiate a search subtree and fix the first echelon routing solution to the associated configuration

4. Select the next cheapest configuration. If there are no more (existing) configurations to consider go to Step 2
5. While the best (minimum) lower bound found so far is less than the lower bound of the next configuration:
   (a) Select a node from the tree, do the branching and solve the associated LP-relaxation of RMP of problem (10 - 13)
   (b) Fix the best lower bound equal to the lowest lower bound found so far
6. Go to Step 4

Let $\theta \leq |\mathscr{P}|$ 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 total cost, if the configuration M is used as the first echelon solution ($LB_{AP}(M) \geq LB_{AG}(M)$). The outline of the proposed algorithm is given in Algorithm 1. 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 (Line 19). Then W is sorted again in a non-decreasing order of LB_{AP} values. For each configuration $M \in W$ a search subtree is created to solve the associated $SP(M)$, where for all nodes of the subtree, the solution of the first echelon routing problem is fixed to M. 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 22). For each selected node of the associated search tree, the RMP of problem (10 - 13) is achieved by replacing $\mathscr{L}$ by $\mathscr{L}'$ (Line 6). The pricing problem consists of finding city freighter tours with a negative reduced cost (problem 14).

Urban vehicle tours

An urban vehicle tour can visit any subset of satellites in any order (sequence). Moreover, for any sequence of satellites, we only consider non-dominated depots (as the start and end point of the urban vehicle tour). A depot (when considering an specific sequence of satellites, $(s_1, ..., s_n)$) is called non-dominated if and only if no other depot would lead to a lower transportation cost and 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 non-dominated 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 straight forward to add the starting (and ending) depot to generate urban vehicle tours.

For settings with a large number of satellites, the full enumeration can be replaced by a step wise enumeration, similar to what we do for the configuration set. In such a situation you start with the subset of urban vehicle tours with the lowest costs. As we

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

Input: the sorted list of all urban vehicle tours $\mathscr{M}$
Output: the lower bound LB and the upper bound UB of the problem

1: Initialization:
Initiate a search tree with a root node $T = \{root\}$, set of active configurations $W = \{\}$
Generate θ-best configurations (with respect to aggregate-based lower bounds) and add to W
Sort elements of W in a non-decreasing order of their LB_{AP} values ($W = \{M_1, ..., M_{|W|}\}$)
$LB = LB_{AP}(M_1)$, $UB = \infty$, $MaxLB = LB_{AG}(M_\theta)$, $i = 1$
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 $\mathscr{L}'$)
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: $\mathscr{L}' = L' + \{l\}$
10: Solve problem (5.20)-(5.24) by replacing $\mathscr{L}$ with $\mathscr{L}'$
11: **end while**
12: Set the LB equal to the lowest lower bound value of the leaf nodes of the tree T
13: if improved integer solution is found: update UB
14: **if** $LB = UB$ **then**
15: Terminate and report the optimal solution
16: **end if**
17: **end while**
18: **if** $LB > MaxLB$ **then**
19: Generate next θ-best configurations and add to W
20: Fix $MaxLB$ equal to the aggregate-based lower bound of the worst configuration within the next θ-best configurations
21: Re-sort elements of W in a non-decreasing order of their LB_{AP} values
22: **else**
23: $i = i + 1$; goto step 2
24: **end if**

have a lower bound for the costs on the first and second echelon, we can easily determine when it is necessary to extend our subset. The extension could be done for instance with a cost criterion or with a fixed number of additional tours.

Configurations

Any combination of at least $\left\lceil \sum_{z \in Z} d_z / K^1 \right\rceil$ 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 where each node represents an urban vehicle tour. Let $G_{\mathscr{M}} = (V_{\mathscr{M}}, A_{\mathscr{M}})$ be a directed graph with the vertex set $V_{\mathscr{M}} = \mathscr{M} \cup \{s, t\}$ and the arc set $A_{\mathscr{M}} = \{(i,j) | C(i) \leq C(j), i \in \mathscr{M}, j \in V_{\mathscr{M}}\} \cup \{(s,j) | j \in \mathscr{M}\} \cup \{(i,t) | i \in \mathscr{M}\}$. Nodes s and t are source and destination nodes respectively. Figure 2 shows an example of graph $G_{\mathscr{M}} = (V_{\mathscr{M}}, A_{\mathscr{M}})$ with $\mathscr{M} = \{m_1, m_2, m_3, m_4\}$. A path from s to t with at least $\left\lceil \sum_{z \in Z} d_z / K^1 \right\rceil + 1$ 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 towards t. Please refer to Appendix 8.5 for more detail.

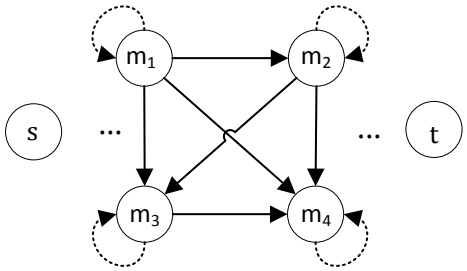


Figure 2: An example of graph $G_{\mathscr{M}} = (V_{\mathscr{M}}, A_{\mathscr{M}})$ with $\mathscr{M} = \{m_1, m_2, m_3, m_4\}$

θ-best configurations

In order to generate all possible configurations, any combination of the urban vehicle tours should be considered, which may be highly time-consuming. To prevent this, in each iteration of the proposed algorithm a subset of $\mathscr{P}$ is considered. Thus, we generate the set of the θ-best configurations where θ is a parameter to be tuned. In this regard, we prevent an exhaustive search of all paths and discard all partial paths which would not lead to a path (configuration) of the set of the θ-best configurations.

In order to find the θ-best configurations, we start with an empty set $\mathscr{P}' = \{\}$. In each

iteration a configuration is generated using the described labeling algorithm (Figure 2), and it is inserted in $\mathscr{P}'$ such that it is sorted in a non-decreasing order of the aggregate-based lower bound values. It is repeated until the size of $\mathscr{P}'$ becomes θ.

Lower bounds

For each used configuration, $M \in \mathscr{P}$, 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 $LB_{AG}(M) = c(M) + \sum_{z \in Z} \min_{m \in M, s \in S(m)}\{\gamma_{zs}\}$. γ_{zs} is the minimum cost of serving the customer z from the satellite s (see also the Appendix 8.2 for more details).

Additionally, we define an assignment-based lower bound for the configuration, $LB_{AP}(M) = c(M)+Z^*(SL(M))$, 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:

$$\text{minimize} \sum_{i \in Z} \sum_{m \in M} \sum_{k \in S} g_{imk}\gamma_{ik} \tag{15}$$

$$\text{subject to} \sum_{m \in M} \sum_{k \in S} g_{imk} = 1, \qquad \forall i \in Z \tag{16}$$

$$\sum_{i \in Z} \sum_{k \in S} g_{imk} d_i \leq K^1, \qquad \forall m \in M \tag{17}$$

$$g_{imk} \geq 0, \qquad \forall i \in Z, m \in M, k \in S \tag{18}$$

Pricing problem

We consider each $m \in M$ 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, i.e., on a directed graph

$\tilde{G} = (\tilde{V}, \tilde{A})$ with the vertex set $\tilde{V} = \{s\} \cup Z$ and the arc set $\tilde{A} = \{(i,j)|i \in \tilde{V}, j \in \tilde{V}\}$. An ESPPRC is used where 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). Employing 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 $d^l_{i,j}$ be the consumption of the resource l, $l \in L$, along an arc $(i,j) \in \tilde{V}$. Then it is straightforward to define $d^Q_{i,j} = d_j, j \in Z$ and $d^T_{i,j} = t_{ij}$. We use the Desrochers' label correcting algorithm (Desrochers et al. (1992)), adapted to ESPPRC by Feillet et al. (2004).

6 Computational study

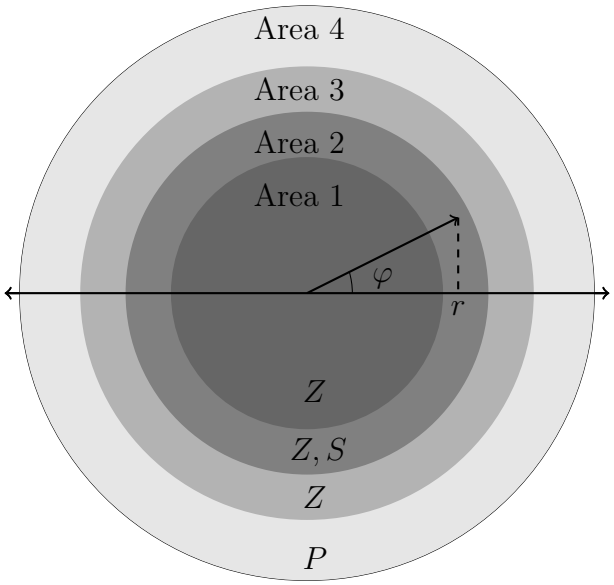
In this section, we present a computational study to test the proposed algorithms on newly generated 2E-VRPTW instances. All tests are performed using Delphi XE6 on a PC with Intel Core i7-4770 Processor (8M Cache, up to 3.90 GHz) and 8.00 GB of RAM. First, we explain the instance generation procedure, secondly, the analysis of the results is presented.

6.1 Instance generation

We developed an instance generator in order to generate specific instances for the 2E-VRPTW. The instance generator simulates an urban area where the area is divided into four sections. Figure 3 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 the area 2. Area 3 represents the housing area of the urban area where there exists some customers. Depots are located inside area 4 which represents the external area of the city which is a less populated section. In order to represent a point in one of the defined sections, a polar coordinate system is used. The center of the presented urban area is considered as the pole. Any point is defined by a radius $0 \leq r \leq 40$ and a polar angle $0 \leq \varphi \leq 2\pi$. Thus, a point with a radius $0 \leq r \leq 20$, $20 \leq r \leq 25$, $25 \leq r \leq 30$, $30 \leq r \leq 40$ belongs respectively to area 1, area 2, area 3 and area 4.

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. Let $|P|$, $|S|$ and $|Z|$ denote the size of an instance regarding the number of depots, satellites and customers. The procedure generates randomly polar coordinates (r, φ) for depots, satellites and

Figure 3: Schematic representation of the urban area



customers locations. In order to locate depots and satellites in a reasonable way, we divide area 2 and area 4 regarding to the number of facilities. Area 2 is divided into $|S|$ subsections. In a similar way, area 4 is divided into $|P|$ subsections. The procedure generates locations of customer, satellites and customers in the following manner:

- For each customer z: randomly generate $0 \leq r \leq 30$ and $0 \leq \varphi \leq 2\pi$.
- For each satellite s : randomly generate $20 \leq r \leq 25$ and $\frac{2\pi(s-1)}{|S|} \leq \varphi \leq \frac{2\pi(s)}{|S|}$.
- For each depot d : randomly generate $30 \leq r \leq 40$ and $\frac{2\pi(d-1)}{|P|} \leq \varphi \leq \frac{2\pi(d)}{|P|}$.

Four categories of instances generated differing in the time windows and demands of customers. We introduce categories ce, cf, cg and ci. The time window and the demand value for each customer of an instance from these sets is generated in the following order:

- For each customer z of an instance of category ce: randomly generate $20 \leq a_z \leq 260$ and $a_z \leq b_z \leq a_z + 20$ and $d_Z = 10$ *or* 20.
- For each customer z of an instance of category cf: randomly generate $20 \leq a_z \leq 260$ and $a_z \leq b_z \leq a_Z + 20$ and $5 \leq d_z \leq 25$, where d_z is an integer.
- For each customer z of an instance of category cg: randomly generate $60 \leq a_z \leq 360$ and $a_z \leq b_z \leq a_Z + 90$ and $d_Z = 10$ *or* 20.

- For each customer z of an instance of category ci: randomly generate $60 \leq a_z \leq 360$ 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 which consists of the associated category name, an index, number of depots, number of satellites and number of customers.For example, "ce3-2-3-15" denotes the third instance of the category ce, with two depots, three satellites and 15 customers (2-P, 3-S, 15-Z).

6.2 Computational results

We designed experiments to numerically examine how large instances the branch-and-price algorithm of the 2E-1P formulation can solve and also to check the effect of labeling approaches used in this algorithm. 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 15-Z instances due to the long CPU time for all other instances. The branch-and-price algorithm for the 2E-2P formulation is tested on all instances. Throughout this section, the CPU times are reported in seconds.

Results for the 2E1P formulation

Table 1 gives the results for the 2E-1P formulation for the 2-P, 3-P, 15-Z instances. We impose a time limit of one hour for CPU time. The upper bound, lower bound and the achieved gap are reported in columns "UB", "LB" and "Gap" respectively.

Effect of labeling approaches We first compare the effect of different labeling approaches, namely, the bidirectional breath first labeling approach and the forward depth-first labeling approach are denoted by "Bidirectional" and "Depth First. Table 1 shows that the depth-first approach solves 13 instances to optimality in a shorter time compared to the bidirectional approach which solves 9 instances to optimality. Moreover, the bidirectional approach fails to find any upper bound for two instances. This is mainly because of the fact that the tour-trees are very long and the bidirectional approach generates lots of labels, for which it takes a long time to check which labels can be merged. Therefore, for the remaining 15-Z instances, we use the depth-first approach (Table 2). The algorithm solves 25 instances out of 60 instances with 15 customer to optimality. The average gap for the remaining instances is 0.08 percent with a maximum

CPU time of one hour. All in all, the 2E-1P formulation is limited in terms of its performance.

Table 1: 2E-1P formulation: 2-P, 3-S, 15-Z instances

	Bidirectional				*DepthFirst*			
Instance	*LB*	*UB*	*GAP*	*CPU*	*LB*	*UB*	*GAP*	*CPU*
ce1-2-3-15	608.701	612.385	0.006	3600	612.385	612.385	0	305.2
ce2-2-3-15	548.953	548.953	0	119.6	548.953	548.953	0	53.7
ce3-2-3-15	551.985	551.985	0	127.6	551.985	551.985	0	429.8
ce4-2-3-15	569.579	569.579	0	1665.7	569.579	569.579	0	1569.7
ce5-2-3-15	555.796	555.796	0	1576.5	555.796	555.796	0	2219.4
cf1-2-3-15	573.724	635.667	0.097	3600	624.178	624.178	0	2042
cf2-2-3-15	516.739	516.739	0	36.9	516.739	516.739	0	7.6
cf3-2-3-15	601.897	601.897	0	154.8	601.897	601.897	0	543.7
cf4-2-3-15	512.664	590.866	0.132	3600	543.458	555.339	0.021	3600
cf5-2-3-15	486.21	578.556	0.160	3600	492.969	516.708	0.046	3600
cg1-2-3-15	526.172	611.145	0.139	3600	501.246	619.908	0.191	3600
cg2-2-3-15	482.985	482.985	0	982.2	482.985	482.985	0	65.0
cg3-2-3-15	518.163	539.685	0.040	3600	536.25	539.685	0.006	3600
cg4-2-3-15	537.023	620.58	0.135	3600	537.023	594.829	0.097	3600
cg5-2-3-15	-	-	-	3600	436.467	436.467	0	119.4
ci1-2-3-15	518.864	-	-	3600	526.957	597.698	0.118	3600
ci2-2-3-15	483.133	483.133	0	78.7	483.133	483.133	0	24.2
ci3-2-3-15	512.412	512.412	0	2734.4	512.412	512.412	0	12.1
ci4-2-3-15	568.937	585.301	0.028	3600	584.282	585.301	0.002	3600
ci5-2-3-15	522.011	536.764	0.027	3600	536.764	536.764	0	1471.5

Table 2: 2E-1P formulation: 3-P, 5-S, 15-Z and 6-P, 4-S, 15-Z instances

Instance	*LB*	*UB*	*Gap*	*CPU*	*Instance*	*LB*	*UB*	*Gap*	*CPU*
ce1-3-5-15	590.276	603.456	0.022	3600	ce1-6-4-15	551.457	551.457	0	2485.8
ce2-3-5-15	604.153	665.265	0.092	3600	ce2-6-4-15	554.189	573.343	0.033	3600
ce3-3-5-15	481.435	533.263	0.097	3600	ce3-6-4-15	556.642	556.642	0	1460.7
ce4-3-5-15	514.399	514.399	0	316.5	ce4-6-4-15	465.226	465.226	0	8.2
ce5-3-5-15	503.775	537.805	0.063	3600	ce5-6-4-15	416.632	416.632	0	7.2
cf1-3-5-15	532.340	532.340	0	6.7	cf1-6-4-15	558.886	567.151	0.015	3600
cf2-3-5-15	574.780	612.606	0.062	3600	cf2-6-4-15	631.512	631.512	0	28.5
cf3-3-5-15	488.654	540.482	0.096	3600	cf3-6-4-15	545.104	561.536	0.029	3600
cf4-3-5-15	438.674	438.674	0	17.4	cf4-6-4-15	510.954	510.954	0	6.3
cf5-3-5-15	499.227	538.904	0.074	3600	cf5-6-4-15	460.156	548.742	0.161	3600
cg1-3-5-15	473.553	473.553	0	122.1	cg1-6-4-15	556.676	601.575	0.075	3600
cg2-3-5-15	494.727	494.727	0	78.2	cg2-6-4-15	540.631	575.550	0.061	3600
cg3-3-5-15	472.158	540.432	0.126	3600	cg3-6-4-15	536.259	540.875	0.009	3600
cg4-3-5-15	466.209	518.545	0.101	3600	cg4-6-4-15	513.800	547.350	0.061	3600
cg5-3-5-15	520.063	650.872	0.201	3600	cg5-6-4-15	425.830	425.830	0	45.7
ci1-3-5-15	535.604	968.446	0.447	3600	ci1-6-4-15	544.169	551.492	0.013	3600
ci2-3-5-15	506.612	506.612	0	43.2	ci2-6-4-15	540.100	554.830	0.027	3600
ci3-3-5-15	474.118	534.214	0.112	3600	ci3-6-4-15	559.021	559.021	0	794.3
ci4-3-5-15	471.060	512.287	0.080	3600	ci4-6-4-15	491.521	535.492	0.082	3600
ci5-3-5-15	515.207	544.296	0.053	3600	ci5-6-4-15	442.518	506.695	0.127	3600

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 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, 4 and 5, we report the results of the 2E-2P formulation. The columns denoted as "Obj."and "CPU" 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 upper bounds reached respectively. In table 3, the first and second columns show the category and index of instances. We do not show the instance name explicitly as it is straightforward to implicitly extract the instance name from the category, index, number of depot, number of satellites and number of customers. For example, the first entry of the Table 3 is associated with the instance named as "ce1-2-3-15". We label the instances by reporting number of depots ($|P|$) and number of satellites ($|S|$) of each instance.

Performance of 2E-1P and 2E-2P formulations Comparing the results of 15-Z customers in tables 1, 2 and 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 ("ce1-6-4-15"). 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, where the pricing problem of the 2E-2P formulation is concerned with the second echelon and it is easier to solve.

Effect of instance characteristics: time window, number of depots and number of satellites It is observed that the instances of the "cg" 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 3-P, 5-S instances 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 2-P, 3-S, 50-Z instances, 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 0.604 percent. For 3-P, 5-S, 50-Z instances, 19 instances out of 20 instanced are solved to optimality in an average CPU time of 1269 seconds. Similarly, for 6-P, 4-S, 50-Z instances, we are able to solve 18 instances out of 20 customers to optimality in an average CPU time of 1526 seconds. Instances with 100 customers are more complex to solve and we are able to solve 11 instances out of 60 instances. 6 of these optimally solved instances are 2-P, 3-S, 100-Z instances. The average gap for the remaining 2-P, 3-S, 100-Z instances is 0.721 percent. For 3-P, 5-S, 100-Z instances the average gap is 0.586 percent. Similarly, for 6-P, 4-S, 100-Z instances the average gap is 0.957 percent. As the number of depots and satellites increase the problem is more complex to solve.

Table 3: 2E-2P formulation: 15-Z and 30-Z instances

$\|Z\| = 15$		$\|P\| = 2, \|S\| = 3$				$\|P\| = 3, \|S\| = 5$				$\|P\| = 6, \|S\| = 4$			
Cat.	*Ins.*	*Obj.*	*CPU*	*B&B N.*	*#UB*	*Obj.*	*CPU*	*B&B N.*	*#UB*	*Obj.*	*CPU*	*B&B N.*	*#UB*
ce	1	612.385	0.086	11	4	603.456	2.737	10	6	551.457	0.163	11	4
	2	548.953	0.065	10	3	628.405	5.073	212	3	560.919	0.378	31	6
	3	551.985	0.292	60	3	527.679	0.766	12	6	556.642	0.476	29	3
	4	569.579	0.061	7	3	514.399	0.304	45	4	465.226	0.326	22	5
	5	555.796	0.057	6	7	521.399	3.137	185	7	416.632	0.139	5	3
cf	1	624.178	0.071	6	4	532.340	0.070	6	2	567.151	0.116	8	4
	2	516.739	0.059	6	4	611.574	3.848	34	7	631.512	0.125	28	6
	3	601.897	0.023	5	6	537.256	0.926	6	3	561.536	0.226	25	3
	4	546.314	0.171	9	3	438.674	0.178	7	4	510.954	0.118	12	4
	5	494.395	0.260	35	5	538.570	2.513	20	3	460.569	0.267	24	4
cg	1	586.856	1.033	90	7	473.553	0.825	12	5	566.013	0.200	7	6
	2	482.985	0.103	4	2	494.727	0.900	30	6	549.229	0.418	16	2
	3	539.685	0.257	16	6	522.837	2.795	11	4	540.875	0.320	7	6
	4	562.798	0.344	32	8	506.350	2.545	14	4	521.621	0.868	66	5
	5	436.467	1.772	59	4	545.077	3.264	58	6	425.830	0.365	8	4
ci	1	597.698	0.083	7	2	568.629	3.240	42	4	551.492	0.135	6	2
	2	483.133	0.103	5	5	506.612	0.403	22	5	554.830	0.197	18	1
	3	512.412	0.047	6	2	522.800	2.763	15	4	558.714	0.193	8	1
	4	585.301	0.068	11	5	506.825	2.883	44	7	507.891	0.607	58	4
	5	536.764	0.085	13	2	544.084	2.451	32	6	488.801	0.386	30	6
$\|Z\| = 30$		$\|P\| = 2, \|S\| = 3$				$\|P\| = 3, \|S\| = 5$				$\|P\| = 6, \|S\| = 4$			
Cat.	*Ins.*	*Obj.*	*CPU*	*B&B N.*	*#UB*	*Obj.*	*CPU*	*B&B N.*	*#UB*	*Obj.*	*CPU*	*B&B N.*	*#UB*
ce	1	981.374	353.2	6855	6	892.746	18.6	127	4	922.215	919.4	9139	8
	2	968.324	2.6	30	4	960.533	24.6	90	5	956.349	2.4	20	1
	3	1069.810	0.9	8	5	924.509	17.4	15	3	869.334	4.9	27	4
	4	948.047	4.2	31	6	909.869	18.3	70	1	867.416	11.4	40	5
	5	968.231	9.5	60	6	892.660	15.4	15	5	851.516	1.9	9	2
cf	1	985.763	5.7	79	7	928.046	48.1	591	1	968.054	4.6	57	4
	2	937.119	9.2	45	2	958.302	73.2	449	4	920.495	3.1	12	7
	3	1076.254	1.1	9	1	900.905	8.4	25	1	929.483	3.6	36	2
	4	981.344	2.0	17	1	903.874	23.9	121	3	877.540	12.8	70	3
	5	834.854	7.7	19	2	977.906	45.5	190	2	895.742	9.3	86	2
cg	1	799.562	16.2	7	3	943.740	62.2	732	2	861.127	39.9	142	5
	2	937.712	10.8	50	6	897.767	65.2	163	4	912.379	10.4	41	3
	3	1013.546	562.3	2747	7	943.295	29.5	110	5	899.833	20.1	89	1
	4	943.279	130.9	492	6	890.652	325.5	514	6	833.637	190.0	230	6
	5	949.387	25.6	68	3	772.331	306.6	149	4	813.910	476.1	1406	7
ci	1	807.247	3.1	6	4	955.083	36.8	621	2	918.387	21.6	347	3
	2	945.304	4.1	29	4	944.603	38.3	168	3	914.267	3.7	16	3
	3	1030.606	9.6	79	6	907.529	29.2	203	2	880.593	8.4	37	2
	4	958.674	19.8	176	1	917.202	95.0	459	2	885.413	12.6	73	4
	5	965.894	55.5	286	3	917.008	83.7	224	2	943.678	16.2	320	6

7 Conclusions and future research

In this paper, we introduced and modeled the 2E-VRPTW in different ways. Two path-based formulations are proposed where 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 where the interconnectivity of them is assured using explicit connecting constraints. We designed an exact branch-and-price algorithm which uses the 2E-1P formulation to define the tour-trees as columns where a labeling approach is used to generate columns. Both mono-directional and bidirectional 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

Table 4: 2E-2P formulation: 50-Z instances

$\|P\| = 2, \|S\| = 3, \|Z\| = 50$						
Ins.	*LB*	*UB*	*Gap*	*CPU*	*B&B N.*	*#UB*
ce1-2-3-50	1501.80	1501.80	0	2.7	8	3
ce2-2-3-50	1579.53	1586.21	0.421	10800	20296	5
ce3-2-3-50	1479.56	1479.56	0	40.2	149	1
ce4-2-3-50	1485.70	1485.70	0	3427.1	6449	4
ce5-2-3-50	1397.05	1397.05	0	195.8	477	1
cf1-2-3-50	1455.11	1455.11	0	27.4	45	2
cf2-2-3-50	1468.47	1468.47	0	840.8	1170	4
cf3-2-3-50	1492.25	1492.25	0	3365.1	3419	5
cf4-2-3-50	1469.01	1469.01	0	566.2	618	2
cf5-2-3-50	1415.23	1415.23	0	1147.9	1891	3
cg1-2-3-50	1403.49	1411.24	0.549	10800	7824	15
cg2-2-3-50	1466.71	1468.88	0.148	10800	3462	12
cg3-2-3-50	1366.31	1381.33	1.087	10800	3507	8
cg4-2-3-50	1411.20	1427.86	1.167	10800	3397	10
cg5-2-3-50	1442.65	1442.65	0	326.8	187	5
ci1-2-3-50	1458.04	1458.04	0	30.7	44	7
ci2-2-3-50	1566.42	1570.43	0.255	10800	21138	3
ci3-2-3-50	1449.70	1449.70	0	133.4	384	2
ci4-2-3-50	1453.24	1453.24	0	127.9	115	1
ci5-2-3-50	1462.30	1462.30	0	31	83	1
$\|P\| = 3, \|S\| = 5, \|Z\| = 50$						
Ins.	*LB*	*UB*	*Gap*	*CPU*	*B&B N.*	*#UB*
ce1-3-5-50	1351.83	1351.83	0.0	662.9	1430	12
ce2-3-5-50	1354.12	1354.12	0.0	172.1	291	2
ce3-3-5-50	1271.67	1271.67	0.0	488.6	582	2
ce4-3-5-50	1299.86	1299.86	0.0	371.1	756	4
ce5-3-5-50	1267.49	1267.49	0.0	123.2	144	1
cf1-3-5-50	1355.50	1355.50	0.0	631.5	872	1
cf2-3-5-50	1391.31	1391.31	0.0	794.7	1467	2
cf3-3-5-50	1368.97	1368.97	0.0	847.4	1150	1
cf4-3-5-50	1298.34	1298.34	0.0	258	346	2
cf5-3-5-50	1270.39	1270.39	0.0	462.3	324	1
cg1-3-5-50	1293.88	1293.88	0.0	1507.4	687	3
cg2-3-5-50	1317.82	1317.82	0.0	150	47	1
cg3-3-5-50	1288.90	1288.90	0.0	248	97	3
cg4-3-5-50	1342.39	1348.84	0.478	10800	9362	1
cg5-3-5-50	1276.47	1276.47	0.0	901.9	607	10
ci1-3-5-50	1314.59	1314.59	0.0	4997	5984	6
ci2-3-5-50	1326.05	1326.05	0.0	472.3	622	4
ci3-3-5-50	1351.50	1351.50	0.0	8526.9	12464	2
ci4-3-5-50	1315.54	1315.54	0.0	445.1	534	5
ci5-3-5-50	1289.71	1289.71	0.0	2068.8	4189	5
$\|P\| = 6, \|S\| = 4, \|Z\| = 50$						
Ins.	*LB*	*UB*	*Gap*	*CPU*	*B&B N.*	*#UB*
ce1-6-4-50	1334.43	1335.55	0.084	10800	14424	10
ce2-6-4-50	1376.13	1376.13	0	5749.5	8488	5
ce3-6-4-50	1337.15	1337.15	0	1313.9	2686	4
ce4-6-4-50	1242.56	1242.56	0	29.6	92	3
ce5-6-4-50	1268.84	1268.84	0	81	67	3
cf1-6-4-50	1355.92	1355.92	0	63.8	182	1
cf2-6-4-50	1456.85	1456.85	0	3179.1	2600	2
cf3-6-4-50	1345.02	1345.02	0	75	146	1
cf4-6-4-50	1281.67	1281.67	0	36.3	78	1
cf5-6-4-50	1341.25	1341.25	0	244.4	370	3
cg1-6-4-50	1262.60	1262.60	0	1214	429	2
cg2-6-4-50	1304.68	1304.68	0	7221	2414	8
cg3-6-4-50	1315.85	1315.85	0	6100	2108	6
cg4-6-4-50	1204.31	1206.64	0.193	10800	8171	10
cg5-6-4-50	1453.39	1453.39	0	423.5	645	5
ci1-6-4-50	1300.53	1300.53	0	59.1	71	4
ci2-6-4-50	1338.28	1338.28	0	73.2	120	2
ci3-6-4-50	1375.06	1375.06	0	50.1	144	2
ci4-6-4-50	1236.79	1236.79	0	341.1	728	3
ci5-6-4-50	1332.86	1332.86	0	1229.7	1963	7

Table 5: 2E-2P formulation: 100-Z instances

$\|P\| = 2, \|S\| = 3, \|Z\| = 100$						
Ins.	*LB*	*UB*	*Gap*	*CPU*	*B&B N.*	*#UB*
ce1-2-3-100	2769.89	2780.05	0.366	10800	2554	5
ce2-2-3-100	2861.80	2863.57	0.062	10800	4200	10
ce3-2-3-100	2676.69	2678.51	0.068	10800	3125	6
ce4-2-3-100	3037.91	3037.91	0	8647.7	3236	5
ce5-2-3-100	2757.45	2757.45	0	1487.9	281	2
cf1-2-3-100	2478.67	2500.31	0.866	10800	148	3
cf2-2-3-100	2720.85	2756.17	1.282	10800	380	9
cf3-2-3-100	2706.98	2734.22	0.996	10800	328	4
cf4-2-3-100	2833.38	2863.69	1.058	10800	616	2
cf5-2-3-100	2785.22	2808.35	0.824	10800	314	8
cg1-2-3-100	2758.58	2778.64	0.722	10800	827	7
cg2-2-3-100	2639.09	2664.51	0.954	10800	737	10
cg3-2-3-100	2665.60	2665.85	0.009	10800	1111	4
cg4-2-3-100	2878.72	2898.31	0.676	10800	1927	8
cg5-2-3-100	2757.45	2757.45	0	1758.5	374	14
ci1-2-3-100	2666.77	2666.77	0	669.6	110	9
ci2-2-3-100	2852.93	2852.93	0	4021.9	1407	17
ci3-2-3-100	2616.30	2638.30	0.834	10800	2275	20
ci4-2-3-100	2882.78	2923.18	1.382	10800	1994	11
ci5-2-3-100	2794.66	2794.66	0	1269.4	330	8
$\|P\| = 3, \|S\| = 5, \|Z\| = 100$						
Ins.	*LB*	*UB*	*Gap*	*CPU*	*B&B N.*	*#UB*
ce1-3-5-100	2608.82	2608.82	0	3325.2	548	10
ce2-3-5-100	2471.23	2487.10	0.638	10800	2105	13
ce3-3-5-100	2455.60	2459.23	0.148	10800	1328	7
ce4-3-5-100	2700.77	2716.11	0.565	10800	2365	10
ce5-3-5-100	2621.14	2624.99	0.147	10800	1164	12
cf1-3-5-100	2635.64	2646.86	0.424	10800	931	6
cf2-3-5-100	2478.15	2478.15	0	10525.2	892	8
cf3-3-5-100	2538.03	2553.16	0.592	10800	778	10
cf4-3-5-100	2761.56	2814.27	1.873	10800	448	8
cf5-3-5-100	2778.78	2813.71	1.241	10800	526	10
cg1-3-5-100	2555.39	2563.99	0.335	10800	564	14
cg2-3-5-100	2441.19	2453.13	0.487	10800	919	10
cg3-3-5-100	2538.31	2547.48	0.360	10800	442	7
cg4-3-5-100	2618.64	2628.48	0.374	10800	514	4
cg5-3-5-100	2706.53	2714.37	0.289	10800	250	7
ci1-3-5-100	2530.20	2530.20	0	7912.3	1669	11
ci2-3-5-100	2442.22	2459.92	0.720	10800	2165	9
ci3-3-5-100	2528.87	2530.25	0.054	10800	1260	16
ci4-3-5-100	2549.35	2594.54	1.742	10800	1127	12
ci5-3-5-100	2697.60	2707.59	0.369	10800	2119	13
$\|P\| = 6, \|S\| = 4, \|Z\| = 100$						
Ins.	*LB*	*UB*	*Gap*	*CPU*	*B&B N.*	*#UB*
ce1-6-4-100	2522.81	2550.84	1.099	10800	1794	13
ce2-6-4-100	2380.65	2380.65	0	1881	59	4
ce3-6-4-100	2691.56	2706.99	0.570	10800	2741	7
ce4-6-4-100	2373.43	2377.60	0.176	10800	1743	16
ce5-6-4-100	2572.80	2579.80	0.271	10800	2791	17
cf1-6-4-100	2360.73	2386.73	1.089	10800	542	7
cf2-6-4-100	2554.32	2614.02	2.284	10800	459	7
cf3-6-4-100	2625.69	2694.58	2.557	10800	669	15
cf4-6-4-100	2736.90	2780.80	1.579	10800	449	2
cf5-6-4-100	2564.62	2592.88	1.090	10800	701	10
cg1-6-4-100	2511.23	2544.37	1.303	10800	1193	8
cg2-6-4-100	2548.80	2557.28	0.331	10800	261	9
cg3-6-4-100	2375.30	2402.34	1.125	10800	567	5
cg4-6-4-100	2550.38	2578.48	1.090	10800	183	12
cg5-6-4-100	2456.19	2463.28	0.288	10800	365	7
ci1-6-4-100	2581.31	2604.61	0.895	10800	2374	18
ci2-6-4-100	2535.57	2535.57	0	7767.3	1177	4
ci3-6-4-100	2406.84	2410.13	0.136	10800	2343	9
ci4-6-4-100	2667.30	2689.50	0.825	10800	2312	10
ci5-6-4-100	2570.76	2584.49	0.531	10800	2684	14

works on the set of all urban vehicle tours. The urban vehicle tour enumeration may form the basis of a heuristic which is a topic for further research. Designing this algorithm is the most important contribution of this paper. It is observed that the 2E-2P formulation outperforms the 2E-1P formulation due to the wise decomposition scheme which decomposes the problem into easier sub-problems. It succeeded to solve instance with up to 5 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 which solves such large instance sizes.

Acknowledgments

This research has been funded by the Dutch Institute for Advanced Logistics (DINALOG), under the grant titled ConCoord. Partial funding for this project has been provided by the Natural Sciences and Engineering Council of Canada (NSERC), through its Discovery Grant program. We also gratefully acknowledge the support of Fonds de recherche du Québec through their infrastructure grants. 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

R. Baldacci, A. Mingozzi, R. Roberti, and R. W. Calvo. An exact algorithm for the two-echelon capacitated vehicle routing problem. *Oper. Res.*, 61(2):298–314, 2013.

T. Bektaş, T. G. Crainic, and T. V. Woensel. *From managing urban freight to smart city logistics networks*. Technical report CIRRELT-2015-17, CIRRELT, Montréal, 2015.

CIVITAS Initiative. CIVITAS Policy Note - Making urban freight logistics more sustainable, 2015. `http://www.civitas.eu`.

T. G. Crainic, S. Mancini, G. Perboli, and R. Tadei. *Clustering-based heuristics for the two-echelon vehicle routing problem*. Technical report CIRRELT-2008-46, CIRRELT, Montréal, 2008.

T. G. Crainic, , N. Ricciardi, and G. Storchi. Models for evaluating and planning city logistics systems. *Transportation Sci.*, 43(4):432–454, 2009.

T. G. Crainic, G. Perboli, S. Mancini, and R. Tadei. Two-echelon vehicle routing problem: a satellite location analysis. *Procedia-Social and Behavioral Sciences*, 2(3):5944–5955, 2010.

T. G. Crainic, S. Mancini, G. Perboli, and R. Tadei. Multi-start heuristics for the two-echelon vehicle routing problem. In *European Conference on Evolutionary Computation in Combinatorial Optimization*, pages 179–190. Springer, 2011a.

T. G. Crainic, A. Sforza, and C. Sterle. *Location-routing models for two-echelon freight distribution system design*. Technical report CIRRELT-2011-40, CIRRELT, Montréal, 2011b.

T. G. Crainic, S. Mancini, G. Perboli, and R. Tadei. Grasp with path relinking for the two-echelon vehicle routing problem. In *Advances in Metaheuristics*, pages 113–125. Springer, 2013.

R. Cuda, G. Guastaroba, and M. G. Speranza. A survey on two-echelon routing problems. *Computers & Operations Research*, 55:185–199, 2015.

M. Desrochers, J. Desrosiers, and M. Solomon. A new optimization algorithm for the vehicle routing problem with time windows. *Oper. Res.*, 40(2):342–354, 1992.

D. Feillet, P. Dejax, M. Gendreau, and C. Gueguen. An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks*, 44(3):216–229, 2004.

J. Gonzalez-Feliu. *Models and methods for the city logistics: The two-echelon capacitated vehicle routing problem*. PhD thesis, Politecnico di Torino, 2008.

V. C. Hemmelmayr, J. F. Cordeau, and T. G. Crainic. An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics. *Computers & operations research*, 39(12):3215–3228, 2012.

M. Jepsen, S. Spoorendonk, and S. Ropke. A branch-and-cut algorithm for the symmetric two-echelon capacitated vehicle routing problem. *Transportation Sci.*, 47(1):23–37, 2013.

M. E. Lübbecke and J. Desrosiers. Selected topics in column generation. *Oper. Res.*, 53(6): 1007–1023, 2005.

S. Mancini. Multi-echelon distribution systems in city logistics. *European Transport\ Trasporti Europei*, (54), 2013.

G. Perboli and R. Tadei. New families of valid inequalities for the two-echelon vehicle routing problem. *Electronic notes in discrete mathematics*, 36:639–646, 2010.

G. Perboli, R. Tadei, and D. Vigo. The two-echelon capacitated vehicle routing problem: models and math-based heuristics. *Transportation Sci.*, 45(3):364–380, 2011.

F. A. Santos, A. S. da Cunha, and G. R. Mateus. Branch-and-price algorithms for the two-echelon capacitated vehicle routing problem. *Optimization Letters*, 7(7):1537–1547, 2013.

F. A. Santos, G. R. Mateus, and A. S. da Cunha. A branch-and-cut-and-price algorithm for the two-echelon capacitated vehicle routing problem. *Transportation Sci.*, 49(2):355–368, 2014.

M. W. P. Savelsbergh and T. V. Woensel. 50th anniversary invited articlecity logistics: Challenges and opportunities. *Transportation Sci.*, 50(2):579–590, 2016.

Z. Y. Zeng, W. S. Xu, Z. Y. Xu, and W. H. Shao. A hybrid grasp+vnd heuristic for the two-echelon vehicle routing problem arising in city logistics. *Mathematical Problems in Engineering*, 2014, 2014.

8 Appendices

8.1 Arc-based formulation

We formulate the 2E-VRPTW as a three-index mixed integer programming problem. In this formulation, $r_{ij}^{\tau} = 1$ if arc $(i \in P \cup S, j \in P \cup S)$ is traveled by vehicle $\tau \in \mathcal{T}$; 0 otherwise. If arc $(i \in S \cup Z, j \in S \cup Z)$ is traveled by vehicle $\upsilon \in \mathcal{V}$ then $q_{ij}^{\upsilon} = 1$; otherwise $q_{ij}^{\upsilon} = 0$. Customer i is assigned to satellite j if w_{ij} is equal to 1. If an urban vehicle $\tau \in \mathcal{T}$ is used to perform a first echelon tour then $u_{\tau} = 1$. Similarly, $u_{\upsilon} = 1$ if a city freighter $\upsilon \in \mathcal{V}$ 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^{υ} is used to represent the arrival time of the city freighter υ to customer z. Then, the network flow formulation for the 2E-VRPTW becomes:

$$\text{minimize} \sum_{\tau\in\mathcal{T}}\sum_{i\in P\cup S}\sum_{j\in P\cup S} c_{ij}r_{ij}^{\tau} + \sum_{\upsilon\in\mathcal{V}}\sum_{i\in S\cup Z}\sum_{j\in S\cup Z} c_{ij}q_{ij}^{\upsilon} + \sum_{\tau\in\mathcal{T}} h^1 u_\tau + \sum_{\upsilon\in\mathcal{V}} h^2 u_\upsilon \tag{19}$$

$$\text{subject to} \sum_{l\in P\cup S} r_{lj}^{\tau} - \sum_{l\in P\cup S} r_{jl}^{\tau} = 0, \qquad \forall j \in P\cup S, \tau\in\mathcal{T} \tag{20}$$

$$\sum_{i\in P\cup S}\sum_{j\in P} r_{ij}^{\tau} \le 1, \qquad \forall \tau\in\mathcal{T} \tag{21}$$

$$K^1 \sum_{k\in P\cup S} r_{ik}^{\tau} - x_{ji}^{\tau} \ge 0, \qquad \forall \tau\in\mathcal{T}, i\in S, j\in P \tag{22}$$

$$K^1 \sum_{k\in P\cup S} r_{jk}^{\tau} - x_{ji}^{\tau} \ge 0, \qquad \forall \tau\in\mathcal{T}, i\in S, j\in P \tag{23}$$

$$\sum_{i\in P}\sum_{j\in S} x_{ij}^{\tau} \le K^1 u_\tau, \qquad \forall \tau\in\mathcal{T} \tag{24}$$

$$\sum_{\upsilon\in\mathcal{V}}\sum_{j\in S\cup Z} q_{ij}^{\upsilon} = 1, \qquad \forall i\in Z \tag{25}$$

$$\sum_{j\in S\cup Z} q_{lj}^{\upsilon} - \sum_{l\in S\cup Z} q_{jl}^{\upsilon} = 0, \qquad \forall j\in S, \upsilon\in\mathcal{V} \tag{26}$$

$$\sum_{i\in S\cup Z}\sum_{j\in S} q_{ij}^{\upsilon} \le 1, \qquad \forall \upsilon\in\mathcal{V} \tag{27}$$

$$\sum_{k\in S\cup Z} q_{ik}^{\upsilon} + \sum_{k\in S\cup Z} q_{jk}^{\upsilon} - w_{ij} \le 1, \qquad \forall i\in Z, j\in S, \upsilon\in\mathcal{V} \tag{28}$$

$$\sum_{j\in S} w_{ij} = 1, \qquad \forall i\in Z \tag{29}$$

$$\sum_{i\in Z} d_i \sum_{j\in S\cup Z} q_{ij}^{\upsilon} \le K^2 u_\upsilon, \qquad \forall \upsilon\in\mathcal{V} \tag{30}$$

$$\sum_{i\in P}\sum_{\tau\in\mathcal{T}} x_{ik}^{\tau} - \sum_{j\in Z} d_j w_{jk} = 0, \qquad \forall k\in S \tag{31}$$

$$\omega_j^{\tau} \ge \omega_i^{\tau} + t_{ij} + s_i - M(1 - r_{ij}^{\tau}), \qquad \forall i\in S\cup P, j\in S, \tau\in\mathcal{T} \tag{32}$$

$$\omega_j^{\tau} \ge \omega_i^{\tau} + t_{ij} + s_i - M(2 - q_{kj}^{\upsilon} - \sum_{h\in P\cup S} r_{hi}^{\tau}), \qquad \forall i\in S, j\in Z, \tau\in\mathcal{T}, \upsilon\in\mathcal{V} \tag{33}$$

$$\omega_j^{\upsilon} \ge \omega_i^{\upsilon} + t_{ij} + s_i - M(1 - q_{ij}^{\upsilon}), \qquad \forall i\in Z, j\in Z, \upsilon\in\mathcal{V} \tag{34}$$

$$\omega_i^{\upsilon} \ge a_i, \qquad \forall i\in Z, \upsilon\in\mathcal{V} \tag{35}$$

$$\omega_i^{\upsilon} \le b_i, \qquad \forall i\in Z, \upsilon\in\mathcal{V} \tag{36}$$

$$r_{ij}^{\tau} \in \{0,1\}, \qquad \forall i,j\in S\cup P, \tau\in\mathcal{T} \tag{37}$$

$$q_{ij}^{\upsilon} \in \{0,1\}, \qquad \forall i,j\in S\cup Z, \upsilon\in\mathcal{V} \tag{38}$$

$$x_{ij}^{\tau} \ge 0, \qquad \forall i\in P, j\in S, \tau\in\mathcal{T} \tag{39}$$

$$w_{ij} \in \{0,1\}, \qquad \forall i\in Z, j\in S \tag{40}$$

$$u_i \in \{0,1\}, \qquad \forall i\in\mathcal{T}\cup\mathcal{V} \tag{41}$$

$$\omega_i^{j} \ge 0, \qquad \forall i\in S\cup Z, j\in\mathcal{T}\cup\mathcal{V} \tag{42}$$

Objective function (19) minimizes the first level transportation cost, second level transportation cost and vehicle fixed costs. Constraints (20 - 24) are associated with the first echelon of the problem. Constraint (20) is the flow conservation constraint for the first level. Constraint (21) ensures that each urban vehicle is used at most once. Constraints (22) and (23) relate the flow and routing variables. Constraints (24) is the capacity constraint for urban vehicles. Constraints (25 - 30) are associated with the second echelon of the problem. Constraint (25) ensures that each customer is visited only by one city freighter. Constraint (26) is the flow conservation constraint for the second level. Constraint (27) ensures that each city freighter is used at most once. Constraints (28) and (29) assure that each customer is assigned to a satellite and it is visited by one city freighter tour, starting (and ending) at the assigned satellite. Constraint (30) is the capacity constraint for city freighters. Constraint (31) links the first and second level decision variables by calculating the incoming freight flow to each satellite. Constraints (32) and (34) calculate the arrival time of vehicles to satellite and customers. Constraint (33) 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. Constraints (35) and (36) are hard time window constraints. Constraints (37 - 42) are the domain constraints.

The problem consists of a first echelon routing problem and a second echelon routing problem connected to each other. The set of constraints (20 - 36) can be categorized mainly in three groups. Constraints (20 - 24) and (32) are associated with the first echelon, Constraints (25 - 30) and (34 - 36) are associated with the second echelon, and constraints (31) and (34) are associated with both first and second echelons. Constraint (31) is a flow conservation constraint for satellites which connects the two echelon problems. It denotes that the total freight which is delivered to a satellite should be equal to the total freight which is distributed to the customers from the satellite. It is clear that the freight delivery to a satellite occurs in the first echelon and then the delivery to the customers happen in the second echelon. The amount of freight which each satellite receives (from urban vehicles) and delivers (to city freighters) should be jointly decided by both first and second echelon routing problems. Moreover, a city freighter can depart from a satellite only after the fright is delivered to the satellite by urban vehicles and the freight is consolidated and it is ready to be distributed. Thus, the ready time of freight at each satellite is a result of the the arrival time of the urban vehicles.

The interconnectivity of and the coordination between the first and second echelon routing problems makes the problem highly complicated. Even without considering constraints (31) and (34), both first and second echelon routing problems are not straightforward to address. The first echelon routing problem is a multi-depot capacitated vehicle routing problem (MDCVRP) and the second echelon routing problem is a multi-depot capacitated vehicle routing problem with time windows (MDCVRPTW) which are both extensions of the classical vehicle routing problem. We know from the literature that these problems are very complex and the arc-based formulations are hard to solve.

Table 6 shows the result of the CPLEX solver for customers belong to the *ce* category, with 2-P, 3-S, 15-Z. We limit the CPU time with 5 minutes (These is a memory overload error for longer running times) and report the upper bound, lower bound and the achieved gap in columns "UB", "LB" and "Gap". CPLEX fails to solve the instances to optimality and except one instance the reported gap is over 20 percent. This is due to the complexity of the problem and large number of nodes generated by CPLEX to solve the problem within a branch and bound tree.

Table 6: CPLEX results for customers with 2 depots, 3 satellites and 15 customers

Instance	*UB*	*LB*	*Gap*
ce1-2-3-15	637.376	507.849	20.32
ce2-2-3-15	548.950	523.996	4.55
ce3-2-3-15	612.451	440.080	28.14
ce4-2-3-15	576.740	455.293	21.06
ce5-2-3-15	611.498	476.299	22.11

8.2 Minimal cost of serving a customer from a satellite γ_{ij}

In order to find the minimal cost of serving a customer from a satellite, we solve a single depot vehicle routing problem with time windows (VRPTW) for each satellite. For each associated VRPTW, the satellite is considered as the depot with unlimited source of supply. The earliest time to start distribution from a satellite is the summation of the traveling time from the nearest depot and the service time of the satellite. For each satellite, we call a customer a reachable customer if it is feasible to meet the time window constraint of the customer by performing a city freighter tour starting from the satellite and visiting the customer as the first customer. For each satellite s and its reachable customers Z_s, The problem is defined on the second echelon of the graph G on a directed graph $\tilde{G} = (\tilde{V}, \tilde{A})$ with the vertex set $\tilde{V} = \{s\} \cup Z_s$ and the arc set $\tilde{A} = \{(i,j)|i \in \tilde{V}, j \in \tilde{V}\}$. Let L_s denote the set of city freighter tours which start (and end) at satellite s. Let κ_l denote a binary variable that takes the value 1 if and only if the city freighter our $l \in L(s)$ is included in the solution. Furthermore, let ρ_{lz} be a binary constant that, $\rho_{lz} = 1$ if customer z is visited by the city freighter tour l; 0 otherwise.Then, the VRPTW for each satellite s, is presented as the following set partitioning problem:

$$\text{minimize} \sum_{l \in L_s} c_l \kappa_l \tag{43}$$

$$\text{subject to} \sum_{l \in L_s} \rho_{lz} \kappa_l = 1, \qquad \forall z \in Z_s \tag{44}$$

$$\kappa_l \in \{0, 1\}, \qquad \forall l \in L_s \tag{45}$$

A column generation method is used to solve the LP-relaxation of a set partitioning formulation of the VRPTW to find a lower bound. New columns are generated using the labeling algorithm proposed by Feillet et al. (2004) for the ESPPRC. Let γ_{ij}, $\forall i \in Z, j \in S$ be the dual variable values for each pair of customer and satellite (the dual variable associated with the constraints 44). For customers which are not reachable from a satellite, this value is set to infinity. Note that γ_{ij} can be considered as the minimal cost of serving customer i from satellite j. For any subset $\bar{Z} \subset Z$ of customers, the total cost of serving them from satellite j will be at least $C_{\bar{Z}} \geq \sum_{i \in \bar{Z}} \gamma_{ij}$.

8.3 Labeling algorithm for 2E1P

Let's denote $\rho(L)$ as the partial tour-tree corresponding to the label L.

Forward labeling algorithm We extend labels from a depot $p \in P$ (as the starting node on the corresponding partial tour-tree) to its successors toward satellites and customers. Let L_f denote a forward label with the following components:

- $v(L_f)$ Last node visited on $\rho(L_f)$
- $\sigma(L_f)$ Last visited satellite on $\rho(L_f)$
- $\Omega(L_f)$ Set of usable satellites
- $\pi(L_f)$ Ready time at the node $v(L_f)$ when reached through $\rho(L_f)$
- $\eta(L_f)$ Ready time at satellite $\sigma(L_f)$
- $\kappa_\tau(L_f)$ Remaining urban-vehicle capacity at node $v(L_f)$ when reached by $\rho(L_f)$
- $\kappa_v(L_f)$ Remaining city-freighter capacity at node $v(L_f)$ when reached by $\rho(L_f)$
- $Z(L_f)$ Set of customers visited along $\rho(L_f)$
- $C(L_f)$ Reduced cost of the partial tour-tree $\rho(L_f)$

The algorithm extends a label L'_f along an arc $(v(L'_f), w)$ to a node w to create a new label L_f. The components are calculated as follows:

$$\sigma(L_f) = \begin{cases} w & \text{if } v(L'_f) \in P \cup S \cup Z, w \in S \\ v(L'_f) & \text{if } v(L'_f) \in S, w \in Z \\ \sigma(L'_f) & \text{if } v(L'_f) \in Z, w \in Z \\ \oslash & \text{if } v(L'_f) \in S, w \in P \end{cases} \tag{46}$$

$$\Omega(L_f) = \begin{cases} \Omega(L'_f) - \{v(L'_f)\} & \text{if } v('_f) \in S, w \in S \\ \emptyset & \text{if } v(L'_f) \in S, w \in P \\ \Omega(L'_f) & \text{otherwise} \end{cases} \tag{47}$$

$$\pi(L_f) = \begin{cases} \pi(L'_f) + t_{v(L'_f)w} + s_w & \text{if } v(L'_f) \in P, w \in S \\ \eta(L'_f) + t_{v(L'_f)w} + s_w & \text{if } v(L'_f) \in S, w \in S \\ \eta(L'_f) & \text{if } v(L'_f) \in Z, w \in S \\ \max(\pi(L'_f) + t_{v(L'_f)w}, a_w) + s_w & \text{if } v(L'_f) \in S \cup Z, w \in Z \\ \infty & \text{otherwise} \end{cases} \tag{48}$$

$$\eta(L_f) = \begin{cases} \eta((L'_f) & \text{if } v(L'_f) \in S \cup Z, w \in Z, \text{ or if } v(L'_f) \in Z, w \in S \\ \pi(L'_f) + t_{v(L'_f)w} + s_w & \text{if } v(L'_f) \in P, w \in S \\ \eta(L'_f) + t_{v(L'_f)w} + s_w & \text{if } v(L'_f) \in S, w \in S \\ \infty & \text{if } v(L'_f) \in S, w \in P \end{cases} \tag{49}$$

$$\kappa_\tau(L_f) = \begin{cases} \kappa_\tau(L'_f) & \text{if } v(L'_f) \in P \cup S \cup Z, w \in S \\ \kappa_\tau(L'_f) - d_w & \text{if } v(L'_f) \in S \cup Z, w \in Z \\ 0 & \text{if } v(L'_f) \in S, w \in P \end{cases} \tag{50}$$

$$\kappa_v(L_f) = \begin{cases} K_v & \text{if } v(L'_f) \in P \cup S \cup Z, w \in S \\ \kappa_v(L'_f) - d_w & \text{if } v(L'_f) \in S \cup Z, w \in Z \\ 0 & \text{if } v(L'_f) \in S, w \in P \end{cases} \tag{51}$$

$$Z(L_f) = \begin{cases} Z(L'_f) + \{w\} & \text{if } v(L'_f) \in S \cup Z, w \in Z \\ Z(L'_f) & \text{otherwise} \end{cases} \tag{52}$$

$$C(L_f) = \begin{cases} C(L'_f) + c_{v(L'_f)w} - \lambda_w & \text{if } v(L'_f) \in S \cup Z, w \in Z \\ C(L'_f) + c_{v(L'_f)w} & \text{otherwise} \end{cases} \tag{53}$$

$$\tag{54}$$

The extension of label L'_f to L_f is feasible if

$$w \notin S - \{\sigma(L'_f)\} \wedge w \notin S \setminus \Omega(L'_f) \wedge \pi(L_f) - s_w \leq b_w \wedge \kappa_\tau(L_f) \geq 0 \wedge \kappa_v(L_f) \geq 0$$

The labeling algorithm extends all labels toward all feasible directions. It ends when all labels are processed. Typically the total number of the labels to process is very huge which makes the algorithm more time consuming. We limit the number of labels by considering each pair of depot and satellite (p, s) one by one to find the best tour-tree which starts from the depot p and visits satellite s immediately as the first satellite.

A dominance test is performed to discard labels which are dominated by other labels. The dominance rule is defined in the following.

Definition 1 *The label L_f^2 is dominated by the label L_f^1 if and only if*

1. $v(L_f^1) = v(L_f^2)$
2. $v(L_f^1) \in S$
3. $Z(L_f^2) \subseteq Z(L_f^1)$
4. $C(L_f^1) \leq C(L_f^2)$
5. $\pi(L_f^1) \leq \pi(L_f^2)$

Definition 1 states that if L_f^1 and L_f^2 represent two partial tour-trees ($\rho(L_f^1)$ and $\rho(L_f^2)$) where the last node visited on $\rho(L_f^1)$ and $\rho(L_f^2)$ is the same, extending L_f^1 should always result in a tour-tree with a lower reduced cost. That is because L_f^1 represents a label where the same set of customers (and even maybe extra customers) are visited so far with a lower cost and earliest ready time at $v(L_f^1)$ when compared it to L_f^2. Conditions 1 and 2 ensure that $v(L_f^1)$ and $v(L_f^2)$ are the same node which is associated with a satellite. Condition 3 assures that the set customers visited so far for L_f^2 is visited in L_f^1. Conditions 4 and 5 ensure that L_f^1 has a less reduced cost and the ready time is earlier when compared to L_f^2.

An extra improvement is, that each label L_f (where $v(L_f) \in S$) is extended if it brings a good reduction in the reduced cost. We calculate a lower bound for the future cost of extending the label toward a complete tour-tree. The biggest reduction in the reduced cost is achieved by visiting customers with a small transportation cost and a large dual value. Due to the urban vehicle capacity constraint only a subset of the remaining customers should be considered. Among all customers which are not visited by $\rho(L_f)$, customers which bring a large reduction in the reduced cost with a small demand are

preferred. Each customer $z \in Z \backslash Z(L_f)$ can be reached from $v(L_f)$ with a transportation cost not less than $D_z^* = \min_{s \in \Omega(L_f)} \gamma_{zs}$. Thus, the reduction in the reduced cost by vising a customer z is $D_z^* - \lambda_z$. We assign a score to each customer z which is defined as $\frac{D_z^* - \lambda_z}{d_z}$. Let's assume that $z_1, ..., z_N$ is the list of remaining customers which is sorted in a non-decreasing order of their scores. Moreover, let $\bar{Z}$ be a set of customers which is created by adding customers $z_i, 1 \leq i \leq N$ one by one as long as the urban vehicle capacity allows. Let's assume L_d represents a label associated with a complete tour-tree, then we can define:

Definition 2 *L_v is promising if*

1. $v(L_f) \in S$
2. $C(L_f) + \sum_{z \in \bar{Z}} D_z^* - \lambda_z \leq C(L_d)$

Using definition 2 only promising states for satellites are extended. Condition 1 ensures that the last node visited on ($\rho(L_f)$ is associated with a satellite. Condition 2 assures that the summation of the current reduced cost (so far) and the minimum future reduced cost is not greater than the reduced cost of an already existing tour-tree.

Backward labeling algorithm Here, we extend labels from a depot $p \in P$ (as the last visited node on the corresponding partial tour-tree) to its predecessors toward satellites and customers. Let L_b denote a backward label which has the same components of a forward label. We keep the same definition for $\Omega(L_b)$, $Z(L_b)$ and $C(L_b)$ and define the remaining components as the following:

- $v(L_b)$ First node visited on $\rho(L_b)$
- $\sigma(L_b)$ First visited satellite on $\rho(L_b)$
- $\pi(L_b)$ Latest time which the service should start at $v(L_b)$ when reached through $\rho(L_b)$
- $\eta(L_b)$ Latest time which the service should start at satellite $\sigma(L_b)$
- $\kappa_\tau(L_b)$ Total demand of the visited customers by $\rho(L_b)$
- $\kappa_\upsilon(L_b)$ Total demand of visited customers by $\rho(L_b)$ since visiting $\sigma(L_b)$

The algorithm extends a label L_b' along an arc $(w, v(L_b'))$ to a node w to create a new label L_b. Appendix D describes how the backward labels are extended and also how the dominance rule is adapted to the backward labeling. In the backward labeling, the

components $\sigma(L_b)$, $\Omega(L_b)$, $Z(L_b)$ and $C(L_b)$ are calculated in the same way as in the forward label algorithm. The other components are calculated as follows:

$$\pi(L_b) = \begin{cases} \pi(L'_b) - t_{v(L'_b)w} - s_w & \text{if } v(L'_b) \in P \cup Z, w \in S \\ \eta(L'_b) - t_{v(L'_b)w} - s_w & \text{if } v(L'_b) \in S, w \in S \\ \min(\pi(L'_b) - t_{v(L'_b)w} - s_w, b_w) & \text{if } v(L'_b) \in S \cup Z, w \in Z \end{cases} \tag{55}$$

$$\eta(L_b) = \begin{cases} \eta((L'_b) & \text{if } v(L'_b) \in S \cup Z, w \in Z, \text{ or if } v(L'_b) \in P, w \in S \\ \max(\pi(L'_b) - t_{v(L'_b)w} - s_w, \eta(L'_b)) & \text{if } v(L'_b) \in Z, w \in S \\ \eta(L'_b) - t_{v(L'_b)w} - s_w & \text{if } v(L'_f) \in S, w \in S \\ 0 & \text{if } v(L'_b) \in S, w \in P \end{cases} \tag{56}$$

$$\kappa_\tau(L_b) = \begin{cases} \kappa_\tau(L'_b) & \text{if } v(L'_b) \in P \cup S \cup Z, w \in S \\ \kappa_\tau(L'_b) + d_w & \text{if } v(L'_b) \in S \cup Z, w \in Z \end{cases} \tag{57}$$

$$\kappa_\upsilon(L_b) = \begin{cases} 0 & \text{if } v(L'_b) \in P \cup S \cup Z, w \in S \\ \kappa_\upsilon(L'_b) + d_w & \text{if } v(L'_b) \in S \cup Z, w \in Z \end{cases} \tag{58}$$

$$\tag{59}$$

The extension of label L'_b to L_b is feasible if

$$w \notin S - \{\sigma(L'_b)\} \wedge w \notin S \setminus \Omega(L'_b) \wedge \pi(L_b) \geq a_w \wedge \kappa_\tau(L_b) \leq K_\tau \wedge \kappa_\upsilon(L_b) \leq K_\upsilon$$

Again, only promising backward labels are extended. Definition 2 remains true for the backward labels as well. A similar dominance test is performed for backward labels. Definition 3 states the required condition for the dominance rule in the backward labeling algorithm.

Definition 3 *The label L_b^2 is dominated by the label L_b^1 if and only if*

1. $v(L_b^1) = v(L_b^2)$
2. $v(L_b^1) \in S$
3. $Z(L_b^2) \subseteq Z(L_b^1)$
4. $C(L_b^1) \leq C(L_b^2)$
5. $\pi(L_b^1) \geq \pi(L_b^2)$

According to the definition 3, the first node visited on $\rho(L_b^1)$ and $\rho(L_b^2)$ is the same and $\rho(L_b^1)$ visits the same set of customers (and even maybe extra customers) with a lower cost. Moreover, the latest time which the service should start at $v(L_b^1)$ when reached through $\rho(L_b^1)$ is later than the latest time which the service should start at $v(L_b^2)$ when reached through $\rho(L_b^2)$. Thus, extending L_b^1 should always result in a tour-tree with a lower reduced cost.

Merging forward and backward labels This is performed to construct a complete tour-tree where a forward and a backward label are joined. A forward label L_f and a backward label L_b can be merged only if they meet at a customer with the following conditions:

- $v(L_f) = v(L_b)$
- $\sigma(L_f) = \sigma(L_b)$
- $[S \setminus \Omega(L_f)] \cup [S \setminus \Omega(L_b)] = \{\sigma(L_f)\}$
- $\pi(L_f) \leq \pi(L_b)$
- $\kappa_\tau(L_f) \geq \kappa_\tau(L_b) - d_{v(L_f)}$
- $\kappa_\upsilon(L_f) \geq \kappa_\upsilon(L_b) - d_{v(L_f)}$
- $Z(L_f) \cup Z(L_b) = \{v(L_f)\}$

The resulting label (L) represents a complete tour-tree where its reduced cost is calculated as $C(L) = C(L_f) + C(L_b) - \lambda_{v(L_f)}$.

8.4 Algorithmic challenges fo the 2E-2P formulation

One should solve the LP-relaxation of the 2E-2P formulation: starting with a small set of initial city-freighter tours (columns) $\mathscr{L}' \subset \mathscr{L}$ and a small set of initial urban-vehicle tours (columns) $\mathscr{M}' \subset \mathscr{M}$, new city-freighter tours and new urban-vehicle tours (columns) are generated in an iterative manner by solving two pricing problems. Each pricing problem finds new columns with negative reduced cost. Let ζ_z, $z \in Z$, be the dual variable associated with constraints 6 and δ_m, $m \in \mathscr{M}$ be a dual variable associated with constraints 7. Then, the pricing subproblem aiming to find city-freighter tours with negative reduced cost is formulated as follows:

$$\min_{l \in \mathscr{L}} \{\bar{c}_l = c_l - \sum_{z \in Z} \xi_{lz} \zeta_z + \sum_{m \in \mathscr{M}_l'} d_l \delta_m\} \tag{60}$$

The optimal solution for the pricing problem is a city-freighter l^* with the most negative reduced cost which departs from a satellite $s(l^*)$ at a specific time $t(l^*)$ with the possibility of connecting to urban-vehicle tours set $\mathscr{M}_{l^*}$ where $t(l^*) \leq \min_{m \in \mathscr{M}_{l^*}} \{t^m_{s(l^*)}\} + s_{s(l^*)}$. This problem is defined on the second echelon of the graph G which is represented as a directed graph $G' = (V', A')$ with the vertex set $V' = S \cup Z$ and the arc set $A' = A^2$. We formulate this problem as a two-index mixed integer programming problem. In this formulation, $q_{ij} = 1$ if arc $(i, j) \in A'$ is traveled by a city freighter; 0 otherwise. If the urban-vehicle tour $m \in \mathscr{M}'$ connected to the city-freighter tour then $y_m = 1$; 0 otherwise. ω_z is used to represent the arrival time of the city-freighter to customer z. Then, the network flow formulation for the pricing problem becomes:

$$\text{minimize} \quad \sum_{i \in S \cup Z} \sum_{j \in S \cup Z} c_{ij} q_{ij} - \sum_{z \in Z} \sum_{j \in S \cup Z} q_{ij} \zeta_z + \sum_{m \in \mathscr{M}'} \sum_{i \in Z} d_i \sum_{j \in S \cup Z} q_{ij} y_m \delta_m \tag{61}$$

$$\text{subject to} \quad \sum_{j \in S \cup Z} q_{lj} - \sum_{l \in S \cup Z} q_{jl} = 0, \qquad \forall j \in S \tag{62}$$

$$\sum_{i \in S \cup Z} \sum_{j \in S} q_{ij} = 1 \tag{63}$$

$$\sum_{i \in Z} d_i \sum_{j \in S \cup Z} q_{ij} \leq K^2 \tag{64}$$

$$\sum_{m \in \mathscr{M}'} y_m \geq 1 \tag{65}$$

$$\omega_j \geq t^m_i + t_{ij} + s_i - M(2 - y_m - q_{ij}), \qquad \forall i \in S, j \in Z, m \in \mathscr{M}_i \tag{66}$$

$$\omega_j \geq \omega_i + t_{ij} + s_i - M(1 - q_{ij}), \qquad \forall i \in Z, j \in Z \tag{67}$$

$$\omega_i \geq a_i \xi_i, \qquad \forall i \in Z \tag{68}$$

$$\omega_i \leq b_i \xi_i, \qquad \forall i \in Z \tag{69}$$

$$q_{ij} \in \{0, 1\}, \qquad \forall i, j \in S \cup Z \tag{70}$$

$$y_m \in \{0, 1\}, \qquad \forall m \in \mathscr{M}' \tag{71}$$

$$\omega_i \geq 0, \qquad \forall i \in S \cup Z \tag{72}$$

The problem (61 - 72) is a mixed integer *nonlinear* programming problem with a nonlinear objective function and linear constrains. The set of constraints (62 - 69) can be categorized mainly in three groups. Constraints (62 - 64) and (67 - 69) are associated with the routing decision, Constraint (65) is associated with the decision of selecting the set of urban-vehicle tours which could connect to the optimal city-freighter tour, and constraint (66) are associated with both decisions.

Another challenge is solving the pricing subproblem to generate urban-vehicle tours with negative reduced costs. Due to presence of constraints 7 the reduced cost of any urban vehicle tours $m \notin \mathscr{M}'$ only depends on the actual cost (c_m) and the dual variable

δ_m which is not known before finding the urban vehicle m.

8.5 Labeling problem for 2E-2P (to generate configurations)

Let $\rho(L_{\mathscr{P}})$ be the partial path corresponding to the label $L_{\mathscr{P}}$ with the following components:

- $m(L_{\mathscr{P}})$ Last node (urban vehicle tour) visited on $\rho(L_{\mathscr{P}})$
- $\mathscr{M}(L_{\mathscr{P}})$ Set of urban vehicle tours visited along $\rho(L_{\mathscr{P}})$
- $\Omega(L_{\mathscr{P}})$ Set of visited satellites by urban vehicle tours $\mathscr{M}(L_{\mathscr{P}})$
- $C(L_{\mathscr{P}})$ Aggregate lower bound of the partial path $\rho(L_{\mathscr{P}})$

The algorithm extends a label $L'_{\mathscr{P}}$ along an arc $(m(L'_{\mathscr{P}}), w)$ to a node $w \in G_{\mathscr{P}}$ to create a new label $L_{\mathscr{P}}$. The components are calculated as follows:

$$\mathscr{M}(L_{\mathscr{P}}) = \mathscr{M}(L'_{\mathscr{P}}) + w \tag{73}$$

$$\Omega(L_{\mathscr{P}}) = \Omega(L'_{\mathscr{P}}) + S(w) \tag{74}$$

$$C(L_{\mathscr{P}}) = C(L'_{\mathscr{P}}) + C(w) + \sum_{z \in Z} \min\{0, \sum_{j \in S(w)} \gamma_{zj} - \sum_{j \in \Omega(L_{\mathscr{P}})} \gamma_{zj}\} \tag{75}$$