Two-Echelon Prize-Collecting Vehicle Routing with Time Windows and Vehicle Synchronization: A Branch-and-Price Approach

Transportation Research Part C 171 (2025) 104987

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Two-echelon prize-collecting vehicle routing with time windows and vehicle synchronization: A branch-and-price approach

I. Edhem Sakarya [a], Milad Elyasi [b,*], S.U.K. Rohmer [c], O. Örsan Özener [b], Tom Van Woensel [a], Ali Ekici [b]

[a] Department of Industrial Engineering & Innovation Sciences, Eindhoven University of Technology, Eindhoven, Netherlands
[b] Industrial Engineering Department, Özyeğin University, İstanbul, Turkey
[c] Department of Logistics and Operations Management, HEC Montréal, Montréal, Canada

ARTICLE INFO

Keywords:
Routing
Vehicle synchronization
Two-echelon
Prize-collecting
Urban logistics

ABSTRACT

The steady growth in e-commerce and grocery deliveries within cities strains the available infrastructure in urban areas by increasing freight movements, aggravating traffic congestion, and air and noise pollution. This research introduces the *Two-Echelon Prize-Collecting Vehicle Routing Problem with Time Windows and Vehicle Synchronization*, where deliveries are carried out by smaller low- or zero-emission vehicles and larger trucks. Given their capacity restrictions, the smaller vehicles can only deliver small-sized orders and must be replenished via depot locations or larger-sized trucks. Besides replenishing smaller vehicles at satellite locations, larger trucks can deliver small orders and larger items. Managing these two types of fleets in an urban setting under consideration of capacity limitations, tight delivery time windows, vehicle synchronization, and selective order fulfillment is challenging. We model this problem on a time-expanded network and apply network reduction by considering the time window constraints. In addition, we propose a branch-and-price algorithm capable of solving instances with up to 200 customers, which continuously outperforms a state-of-the-art general-purpose optimization solver. Moreover, we present several managerial insights concerning synchronization, vehicles, and the placement of depot/satellite locations.

1. Introduction

The significant growth in e-commerce over recent years, as a result of changing consumer behaviors and lifestyles, has led to a steady increase in urban freight movements, contributing to traffic congestion as well as air and noise pollution (Savelsbergh and Van Woensel, 2016; Seidel and Wickerath, 2020; Aslan et al., 2021; Han et al., 2022). Adverse health effects linked to this pollution demand the implementation of appropriate countermeasures that reduce the number of vehicles and mitigate the environmental impact of logistics within cities (Wen et al., 2016; Mohri et al., 2024). In this context, one possible solution is the use of *small vehicles* (SVs), such as electric vehicles and cargo bikes. However, while SVs are more environmentally friendly and maneuverable in urban settings, their limited range and capacity pose challenges, particularly for larger orders (Ferrero et al., 2016). This necessitates a hybrid approach, utilizing both small and large vehicles. In these hybrid settings, an integrated planning approach, where the big vehicles (BVs) may be used as moving depots for the SVs, can present opportunities (Grangier et al., 2016). At the same time, this requires a high level of synchronization between BVs and SVs at the replenishment points. Synchronized planning and integrated decision-making approaches are thus crucial for the logistics transition in urban areas. An illustrative example of this problem

* Corresponding author.
E-mail address: milad.elyasi@ozyegin.edu.tr (M. Elyasi).

https://doi.org/10.1016/j.trc.2024.104987
Received 16 May 2024; Received in revised form 21 November 2024; Accepted 23 December 2024
Available online 3 January 2025

context is provided by Getir, a leading on-demand delivery service, facing challenges related to vehicle synchronization and route planning in their two-echelon delivery system. Inspired by this real-life context, this paper aims to address these practical challenges by introducing the *Two-Echelon Prize-Collecting Vehicle Routing Problem with Time Windows and Vehicle Synchronization*, solving it mathematically using a Branch & Price approach. In this problem setting, arising in last-mile delivery contexts of e-commerce and e-grocery companies such as Getir, two vehicle types (BVs and SVs) are responsible for the delivery of customer orders, which vary in size. Given the capacity limits of SVs, large customer orders are exclusively handled by BVs, while small orders can be handled by both vehicle types. In addition to fulfilling customer orders, BVs may, furthermore, be used to replenish the SVs at specific locations (e.g., parking and loading zones), which requires synchronization between vehicles due to the lack of storage capacity at these locations. Reserving the right to reject (unprofitable) customer orders, the decision-maker at the delivery company adopts a prize-collecting approach with the goal of maximizing total profit, by optimizing the vehicle routes while strategically choosing the customers to visit. By modeling the Two-Echelon Vehicle Routing Problem (2E-VRP) as a prize-collecting problem, we are able to better capture the strategic decision-making process of order prioritization arising in contemporary urban logistic systems, where cost considerations often make it impractical for service providers to accept all orders. In these situations, certain orders may be strategically deferred or canceled to optimize overall efficiency.

To the best of our knowledge, we are the first to study this problem, with the study of Anderluh et al. (2021) being the closest in the literature while showcasing several key differences with respect to (i) the customer service, (ii) the replenishment of SVs, (iii) the considered objective function, and (iv) the proposed solution method. First, they pre-assign customers to delivery zones, determining which vehicle type may be used for delivery, whereas in our model the assignment of customers to vehicles is only restricted by the capacity limits of the vehicles. Second, the model of Anderluh et al. (2021) restricts the replenishment of SVs to the use of big vehicles at satellites, whereas our approach offers more flexibility by allowing for replenishment at both satellites (through BVs) as well as the depot (without the aid of BVs). Third, contrasting to Anderluh et al. (2021), we adopt a prize-collecting approach so that not all customers need to be served. Finally, unlike Anderluh et al. (2021), which models the problem as a multi-objective problem and proposes a metaheuristic algorithm as the solution approach, in the present study, we model the problem to maximize the total profit gained by satisfying the customer demands and develop a branch-and-price algorithm as an exact solution approach. As such, the contributions of this paper are threefold:

1. Focusing on important practical developments in last-mile logistics, we introduce a complex new variant of the 2E-VRP, addressing a real-life operational challenge faced by e-commerce and logistics companies. Presenting an innovative solution for contemporary urban logistics systems, this variant features a unique distribution setting with overlapping echelons, advanced vehicle synchronization, and selective order fulfillment, which captures the strategic decision-making process of order prioritization in these systems.
2. From a methodological perspective, we model this problem on a time-expanded network and develop an exact solution approach based on a branch-and-price algorithm. Capable of solving instances with up to 200 customers, this solution approach consistently outperforms state-of-the-art general-purpose optimization solvers while providing a flexible yet accessible way of solving a highly complex practical problem.
3. Focusing on vehicle synchronization for the integration of low-emission vehicles within last-mile delivery settings, we provide valuable managerial insights for e-commerce and e-grocery companies by examining how various factors, including the degree of vehicle synchronization, the number and placement of depots and satellites, and fleet composition, impact the system performance.

The remainder of the paper is organized as follows. Section 2 presents an overview of the related literature. Section 3 provides a formal description of the problem. Section 4 introduces the problem's notation and mathematical formulation of the problem, while Section 5 describes the proposed branch-and-price algorithm for solving the problem. Numerical experiments are presented in Section 6, and a general discussion and conclusion follow in Section 7.

2. Literature review

The topic of city logistics and, in this context, the study of 2E-VRPs has received growing attention within the scientific literature of recent years. An overview of this literature is provided by the reviews of Cuda et al. (2015) and Sluijk et al. (2023), presenting different research streams and problem variants. An important variant, which has been studied extensively, is the 2E-VRP with time windows, where vehicles must adhere to certain time constraints at the customer locations. Most of the research conducted in this area focuses on hard time constraints, setting strict delivery time windows at customer locations (e.g., Lehmann and Winkenbach (2024), Li et al. (2020), Dellaert et al. (2019), Mhamedi et al. (2022)). Lehmann and Winkenbach (2024) consider a 2E-VRP with time windows and mixed demand, and propose an efficient matheuristic which effectively solves medium and large instances. The studies of Li et al. (2020) and Dellaert et al. (2019) both focus on such hard time constraints, with the former developing a heuristic method based on a large neighborhood search and the latter proposing a branch-and-price algorithm. Mhamedi et al. (2022) further contribute to the advancement of exact methods for this type of problem by developing a branch-price-and-cut algorithm, optimizing the 2E-VRP with time windows, and introducing a novel route-based formulation. Alternatively to the use of hard time constraints, some studies (e.g., Wang and Wen (2020)) employ soft constraints where time window violations incur a penalty cost. The use of time-expanded networks, as applied in the studies of Lagos et al. (2020) and Boland et al. (2019), presents another approach for incorporating time constraints. However, while effective in integrating time window requirements, this method faces challenges in computational tractability due to the significant increase in network size, as noted by Belieres et al. (2021).

Building on the 2E-VRP with time windows, the integration of synchronization decisions forms another interesting development in the context of the 2E-VRP and city logistics. The notion of synchronization in the area of VRPs has been discussed in more detail in the survey of Drexl (2012), presenting a classification of synchronization aspects by distinguishing between five main types of synchronization, namely task, operation, movement, load and resource synchronization. To remove ambiguity and gain focus regarding the concept of synchronization, the recent literature review of Soares et al. (2024) reduces this classification to two types of synchronization: operation and movement synchronization. The latter focuses on the synchronization of task sequences, which often relates to the simultaneous traversal of arcs (see, e.g., the VRP with trailers and transshipments (VRPTT) as described in Drexl (2013)). Given the focus of this research, this section focuses predominantly on operation synchronization, concerned with the temporary coordination of tasks, and, in particular, on synchronization as a requirement for transfer or cross-docking operations. In the context of two-echelon routing problems, this coordination generally envisions the synchronization of activities and exchanges between the first- and second-echelon vehicles in order to optimize the distribution process.

Minimizing fleet size as well as total travel costs,Grangier et al. (2016) propose, in this context, a variant of the 2E-VRP with time windows and synchronization constraints that allows for multiple trips at the second echelon. Li et al. (2021) propose a variant of 2E-VRP with bi-synchronization, satellites, multiple vehicles, and time window constraints. In their setting, the first echelon represents inter-satellite deliveries, whereas the second echelon covers pickups and deliveries between customer locations and satellite nodes. Their model considers synchronization in cargo collection satellites and cargo delivery satellites to minimize the operating cost of both echelons. Enthoven et al. (2020) introduce a two-echelon distribution system that allows for both direct delivery at the customer's home as well as indirect delivery via collection points, while considering synchronization between trucks and zero-emission vehicles at satellite locations. The goal in this system is then to minimize the combined travel and connection costs under consideration of the customer preferences for specific delivery methods. Anderluh et al. (2017) consider another variant of a 2E-VRP with satellite synchronization, minimizing total travel cost. In their problem, all customers are preassigned to an echelon and served using the echelon's vehicles, thus distinguishing between "bike customers" and "van customers." Synchronization between the vehicles takes place as the first-echelon vehicles (vans) may supply goods to the second-echelon vehicles (cargo bikes) at transshipment points (satellites). Anderluh et al. (2021) build on this research by introducing a new set of customers in the grey zone. The demand of these customers can be satisfied by vehicles of both echelons, while the demands of other customers may only be satisfied by a specific vehicle type. The model aims to minimize the total transportation cost, consisting of time- and distance-related as well as vehicle-related fixed costs.

The study of Anderluh et al. (2021) is the closest to ours in the literature. However, our approach offers greater flexibility in customer-vehicle assignments. While larger demands are generally served by bigger vehicles, our model allows any vehicle to serve any customer, provided capacity permits, offering a more adaptable solution to the 2E-VRP problem. In addition, we do allow the replenishment of small vehicles to be carried out both at the depots as well as the satellite locations, which results in more flexibility in order fulfillment. Further contributing to this flexibility, we consider a prize-collecting 2E-VRP, so that not all customers need to be visited. For an overview of the field of prize-collecting routing problems, we refer the interested reader to Balas (1989), Feillet et al. (2005), Vansteenwegen et al. (2011), Stenger et al. (2013), Long et al. (2019), and Trachanatzi et al. (2020). Moreover, for a similar application we refer the reader to the study of Senna et al. (2024) addressing a variant of the VRP with Time Windows and Multiple Deliverymen (VRPTWMD), which shows close similarities in the considered problem context, yet does not account for any synchronization or explicit load transfer between vehicles. Focusing on the solution methods that have been proposed in the existing literature, we observe a pervasiveness of heuristic algorithms, focusing either on single solution-based (Belgin et al., 2018; Kancharla and Ramadurai, 2019; Anderluh et al., 2021; Enthoven et al., 2020; Li et al., 2021) or population-based (Anderluh et al., 2017; Sahraeian and Esmaeili, 2018; Bevilaqua et al., 2019; He and Li, 2019) heuristics as solution methods. In contrast to the research on heuristic methods, the literature on exact algorithms for 2E-VRP is still limited (Perboli et al., 2011). In this context, according to Sluijk et al. (2023), the solution approaches can be divided into two major categories of branch-and-cut-based (Perboli and Tadei, 2010; Dang et al., 2013; Jepsen et al., 2013; Liu et al., 2018; Bianchessi et al., 2018) and decomposition-based approaches (Santos et al., 2013; Baldacci et al., 2013; Santos et al., 2015; Marques et al., 2020). In the context of decomposition-based algorithms, Baldacci et al. (2013) accelerate the column generation process by using a Tabu Search heuristic to quickly find routes with negative reduced costs, thus reducing the reliance on more time-consuming exact labeling algorithms. Using route-based formulations and branch-price-and-cut algorithms, Santos et al. (2015) identify rounded capacity cuts, multi-star, and strengthened comb inequalities. Compared to Baldacci et al. (2013)'s route-based model, Marques et al. (2020) present exponentially more constraints that can be separated in polynomial time. The new formulation allows the dynamic generation of variables associated with first-echelon vehicles despite enumerating routes for first-echelon vehicles (as in most decomposition methods). In general, the branch-and-price algorithm has been widely used in the literature on routing problems (see, for example, Dabia et al. (2013), Santos et al. (2015), Munari et al. (2019), Dellaert et al. (2019), Costa et al. (2019), Marques et al. (2020), Wang et al. (2022), and Moreno et al. (2024), Diao et al. (2024)). However, the use of exact methods and, as such, branch-and-price algorithms is still scarce in the context of highly complex 2E-VRPs with synchronization requirements, presenting opportunities for new adaptations and developments.

3. Problem description

In this paper, we address a novel routing problem arising in the last-mile context, where a company, operating, e.g., in e-commerce or e-groceries, aims to integrate two fleets of vehicles to promote the use of low-emission lightweight vehicles in its delivery operations. Within the context of this problem, the delivery of customer orders is, as such, carried out by two different sets

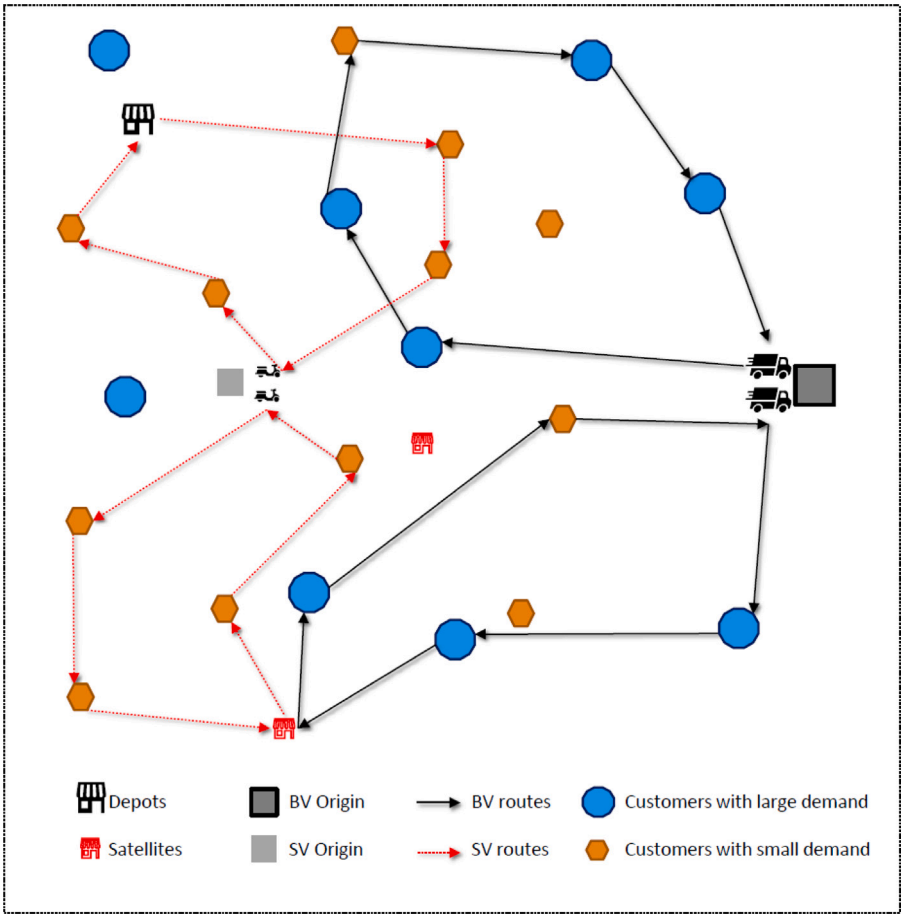

Fig. 1. An illustration of the problem.

of vehicles: (i) BVs and (ii) SVs. Based on observations from practical settings in food and pharmaceutical deliveries, where average order volumes are comparatively small, the capacity of BVs is assumed to be large enough to satisfy all the visited customers during the considered planning horizon. The capacity of SVs is, however, considered to be significantly lower, given the nature of these vehicles. As customer orders may vary considerably in size, it is important to classify them according to the order size. Large-sized orders exceed the SVs total capacity and can, as such, only be carried out by BVs, while small-sized orders can be delivered by both vehicle types. All vehicles of a certain type start their delivery operations with full inventory from a specified location (referred to as their origin) from which they can subsequently make multiple trips before returning to their origin at the end of the day. To replenish, SVs can then receive inventory either directly, by returning to a depot location, or indirectly from a BV at a satellite location. The transfer of goods between vehicles at satellite locations requires vehicle synchronization, i.e., both vehicles must be at the location simultaneously, as these locations, in the form of, for example, parking and loading zones, do not have any storage capacity. Waiting at satellite locations is allowed, and the service time associated with the transfer of goods at these locations is considered fixed.

Promising overall fast delivery as well as convenient customer time windows, companies may choose in practice to reject certain customer orders on a given day for economic reasons. As such, we assume a prize-collecting structure within our system, where not all customers have to be satisfied. However, if a customer order is accepted, the customer must be served by a single vehicle, prohibiting split or partial deliveries. Moreover, order delivery at customer locations needs to adhere to predefined customer time windows. While vehicles may arrive at the customer locations before the start of a time window, they must wait until its official start before commencing service. Service times at customer locations are considered known, yet may vary between customer locations and the considered time stamp. The company aims to maximize the total profit by satisfying customer demands within their specified time windows while minimizing total transportation costs, ensuring efficient yet reliable delivery operations.

The resulting decision problem can be classified as the *Two-Echelon Prize-Collecting Vehicle Routing Problem with Time Windows and Vehicle Synchronization*. The two-echelon structure of the problem is illustrated by Fig. 1, which provides a simple representation of the problem by presenting possible routes for both BVs (solid lines) and SVs (dashed lines) under consideration of vehicle synchronization. In this context, we visualize several distinct features of our problem. Firstly, the BVs can serve both echelons'

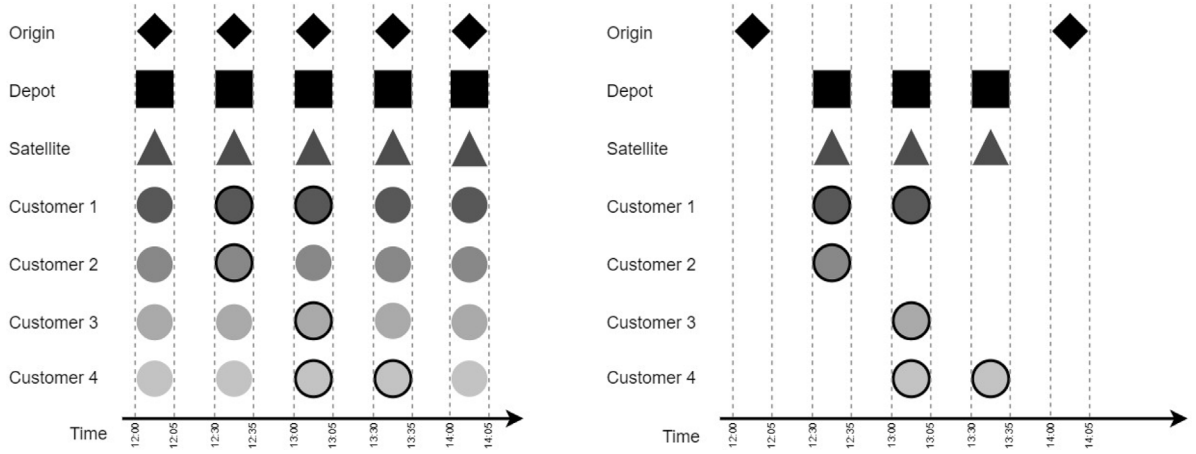

Fig. 2. Time-expanded network example - Before (a) and after (b) the network reduction.

customers, while the SVs only serve customers with small-sized demands. Secondly, SVs can replenish their inventory either at the depots (SV route in the top left) or by meeting with a BV at a replenishment/satellite location (SV route in the bottom left), requiring exact synchronization (i.e., arriving at a satellite location at the same time) between the vehicles. Lastly, Fig. 1 also shows that not all customers have to be visited, and vehicles, especially SVs, continue their trips after replenishing their inventory at the depot or satellite locations.

4. Mathematical formulation

In this section, we present the mathematical modeling approach and elaborate on the details of our mixed-integer programming model. For this purpose, we first present the underlying network structure and the pre-processing steps we followed to reduce the network in Section 4.1 before presenting the model in Section 4.2.

4.1. Time-expanded network and pre-processing

To omit time indices in our variables, implicitly control vehicle synchronization, and meet time window constraints, we define our problem on a time-expanded, directed graph, generating copies of all vertices for each time step in the planning horizon. An illustrative example of such a time-expanded network graph (i.e., before pre-processing) is shown in Fig. 2(a). The network in this example consists of seven locations, which are depicted by multiple nodes representing different discrete time stamps in the planning horizon (e.g., 12:00, 12:30, 13:00, 13:30, 14:00). The considered locations include an origin, a depot, a satellite, and four customer locations. In the graphical representation of Fig. 2, a black line around the circles, representing customer nodes, indicates whether the considered time stamp falls into the customer's delivery time window, which may span multiple time stamps. In the example, the time windows of customers 1 and 4 span across two time stamps, while the time windows of customers 2 and 3 only align with a single time stamp. The notion of service times can be easily incorporated in this context by extending the time stamp with the duration of the service, thus only allowing the vehicle to continue its journey after the service's completion, reducing the available travel time. For reasons of simplicity, we do not depict arcs in Fig. 2(a). However, it should be noted that arcs only connect nodes of a time stamp t with nodes of later time stamps t' (i.e., $t < t'$) while taking into account whether the distance between the nodes can be traversed within the given amount of time (which in this example corresponds to 25 min between time stamps).

As a result, three different types of arcs may be considered: (i) *direct arcs*, representing the most direct path between two locations; (ii) *extended arcs*, which take longer than the required travel time, symbolizing a situation where a vehicle arrives at a location but skips the node with the earliest available time stamp, visiting the location at one of the subsequent time stamps, and (iii) *waiting arcs*, which connect two nodes of the same location but at different time stamps for the purpose of waiting. The latter are exclusive to depot, satellite and origin nodes as customer nodes may only be visited once (and waiting at customer locations is covered by extended arcs).

Given that the size of the graph increases quickly with larger numbers of customers and/or time stamps, solving this type of problem can quickly become computationally challenging. In order to address this, we propose a pre-processing procedure, removing non-essential duplicate nodes, their corresponding arcs as well as other inefficient arcs (such as redundant waiting arcs) within the network. Fig. 2(b) illustrates this procedure for our example graph from earlier. Since all vehicles must start and finish their journeys at their origin and may not visit the origin nodes at any intermediate time in the planning horizon, we only keep the first and the last time stamp nodes of the origin in the pre-processing step. Copies of the origin nodes at other time stamps are removed. Following the same reasoning, the customer, depot, and satellite nodes of the first and last time stamps are consequently also removed from the network. In addition, the pre-processing eliminates all copies of customer nodes outside the specified delivery time windows. When

Table 1
Nomenclature.

Sets	
L	set of vehicle types indexed by l. For BVs, $l = 1$, and for SVs, $l = 2$
K_l	set of vehicles of type l.
K	set of all vehicles $K = K_1 \cup K_2$
$v_{o,l}$	origin node of vehicle type l at first time stamp
$v_{f,l}$	origin node of vehicle type l at last time stamp
V_d	set of depot nodes in the time-expanded network
V_s	set of satellite nodes in the time-expanded network
C	set of unique customer IDs indexed by u
V_c	set of all customer nodes in the time-expanded network
V_u	set of all nodes corresponding to customer $u \in C$ in the time-expanded network
$\bar{V}$	set of all nodes except the first and the last time stamps
V	set of all nodes
Parameters	
M	significantly large number
d_j	total volume of demand at node j
p_j	revenue gained by visiting customer j
c_{ij}^l	cost of traveling from node i to node j by a vehicle of type l
α_{ij}^l	1, if there is an arc from node i to j for vehicle type l; 0, otherwise
Q	capacity of SVs
Decision variables	
x_{ijk}	1, if vehicle k arrives to node j from node i; 0, otherwise
I_{jk}	total vehicle load of vehicle k just before visiting node j

a customer's time window spans multiple time stamps, we keep a copy of the customer node for each of the relevant time stamps, allowing vehicles to visit and serve the customer at any time stamp within the specified window while removing the remaining nodes. This ensures that only feasible delivery options are considered in the model. This targeted reduction of nodes and arcs allows us to effectively reduce the computational complexity of our problem. Other network reduction and arc elimination techniques have been proposed by, for example, Irnich et al. (2010), Kramer et al. (2019), de Lima et al. (2022), and could be explored in future research.

4.2. Model

The notation, parameters, and decision variables are provided in Table 1, followed by our mixed-integer programming formulation.

$$\max \quad \sum_{i\in V}\sum_{j\in V_c}\sum_{k\in K} p_j x_{ijk} - \sum_{l\in L}\sum_{k\in K_l}\sum_{i,j\in V} c_{ij}^l x_{ijk} \tag{1}$$

$$\text{s.t.} \quad x_{ijk} \le \alpha_{ij}^l \qquad \forall i \in V, j \in V,\ k \in K_l, l \in L \tag{2}$$

$$\sum_{i\in v_{o,l}}\sum_{j\in V} x_{ijk} = 1 \qquad \forall k \in K_l, l \in L \tag{3}$$

$$\sum_{i\in V} x_{ijk} = \sum_{i'\in V} x_{ji'k} \qquad \forall k \in K, j \in \bar{V} \tag{4}$$

$$\sum_{i\in V}\sum_{j\in v_{f,l}} x_{ijk} = 1 \qquad \forall k \in K_l, l \in L \tag{5}$$

$$\sum_{i\in V}\sum_{j\in V_u}\sum_{k\in K} x_{ijk} \le 1 \qquad \forall u \in C \tag{6}$$

$$I_{jk} \le Q \qquad \forall j \in V, \forall k \in K_2 \tag{7}$$

$$I_{jk} \le I_{ik} - d_i + M(1 - x_{ijk}) \qquad \forall i \in V_c, \forall j \in V, k \in K_2 \tag{8}$$

$$I_{ik} - d_i - M(1 - x_{ijk}) \le I_{jk} \qquad \forall i \in V_c, \forall j \in V, k \in K_2 \tag{9}$$

$$\sum_{i\in V}\sum_{k\in K_2} x_{ijk} \le |K_2| \sum_{i\in V}\sum_{k\in K_1} x_{ijk} \qquad \forall j \in V_s \tag{10}$$

$$I_{jk} \in \mathbb{N} \qquad \forall j \in V, k \in K \tag{11}$$

$$x_{ijk} \in \{0, 1\} \qquad \forall i, j \in V, k \in K \tag{12}$$

The objective function (1) aims to maximize the total profit, which is calculated based on the total revenue obtained from the customers minus the total traveling cost of the vehicles. The revenue from visiting a customer is the summation of the profits of all products included in the customer's demand. Constraints (2) guarantee that the vehicles can traverse only the available arcs in

the corresponding vehicle's network. Constraints (3), (4) and (5) are network flow constraints, and they impose that all vehicles start and end their trip in the first and last stamp at their origins. Constraints (6) ensure that split delivery is not allowed, and each customer's demand can be satisfied by at most one vehicle. Constraints (7) restrict the vehicle capacity of SVs. Constraints (8) and (9) are inventory balance constraints, ensuring that only vehicles with a load larger or equal to the demand of the customer can carry out the delivery; in other words, it prevents partial delivery. Constraints (10) are synchronization constraints, stating that SVs need at least one BV in the satellite node to replenish their inventory. This condition suffices under the assumption that BVs have sufficiently large capacities, which eliminates the need to track vehicle loads explicitly. Finally, constraints (11) and (12) are integrality and binary constraints, respectively.

5. Branch-and-price algorithm

In this section, we present a *branch-and-price* (B&P) method (Barnhart et al., 1998) to solve the problem for larger instances. A B&P method consists of a branch-and-bound algorithm where the lower bounds in the tree are computed by column generation. Therefore, column generation is utilized to solve the LP-relaxation of the route-based reformulation of the proposed mathematical model, called the master problem.

5.1. Master problem

We derive the *master problem* (MP) by decomposing the original formulation presented in Section 4.2 using a column generation approach. While the original problem maximizes profit, we reformulate the master problem as a minimization problem (following standard column generation conventions) by negating the objective function coefficients. This transformation allows us to directly search for columns with negative reduced costs in the pricing problems. For this purpose, we define a vehicle's route as a sequence of customer, depot, and/or satellite nodes within our time-expanded network structure, which starts from the vehicle's origin. In this context, each customer location (represented by multiple nodes associated with a specific time stamp) can be visited by at most one vehicle. The resulting route-based reformulation is then as follows:

$$\min \quad \sum_{l\in L}\sum_{r\in\Psi_l} \boldsymbol{\sigma}_r \boldsymbol{\Phi}_r \tag{13}$$

$$\text{s.t.} \quad \sum_{l\in L}\sum_{r\in\Psi_l}\sum_{j\in V_u} A_{rj}\boldsymbol{\Phi}_r \le 1 \qquad \forall u \in C \tag{14}$$

$$\sum_{r\in\Psi_2} A_{rj}\boldsymbol{\Phi}_r - |K_2| \sum_{r\in\Psi_1} A_{rj}\boldsymbol{\Phi}_r \le 0 \qquad \forall j \in V_s \tag{15}$$

$$\sum_{r\in\Psi_l} \boldsymbol{\Phi}_r \le |K_l| \qquad \forall l \in L \tag{16}$$

$$\boldsymbol{\Phi}_r \in \{0,1\} \qquad \forall l \in L, r \in \Psi_l \tag{17}$$

where $\boldsymbol{\sigma}_r$ is the total cost of the route $r \in \Psi_l$, which is defined as the total traveling cost minus the revenue obtained by satisfying the demands of the customers, and Ψ_l is the set of routes for vehicle type $l \in L$. We introduce the decision variables $\boldsymbol{\Phi}_r$, which will take the value 1 if route r is selected and 0 otherwise. In addition, the parameter A_{rj} is set to 1 if node j is visited in route r, and 0 otherwise. In this proposed model, the objective function (13) aims to minimize the total cost associated with the routes/delivery of products. Constraints (14) guarantee that each customer's demand can be satisfied by at most one vehicle, taking into account the different nodes of a specific customer in the time-expanded network. The replenishment of SVs at satellite nodes is defined in constraints (15), which ensure synchronization. Constraints (16) ensure that the total number of routes for BVs and SVs is less than/or equal to the number of BVs and SVs, respectively. Finally, constraints (17) define the decision variables as binary.

Enumeration of all routes is generally impractical and quickly becomes computationally challenging as the size of the problem increases. To address this, we employ a column generation algorithm, solving a restricted version of the master problem (RMP) and two distinct pricing problems at each node of the search tree. By definition, the RMP is an LP relaxation of the MP that considers only a subset of the variables. This problem is solved to optimality in every column generation iteration. The pricing subproblems then seek to find columns with negative reduced costs or to prove that no negative reduced cost columns exist. If the algorithm finds negative reduced cost columns, those columns (or a subset of them) are added to the RMP before the next iteration; otherwise, the column generation algorithm is terminated. Exploiting the structure of our problem, we consider two types of pricing subproblems, one for BV routes and one for SV routes, which will be discussed in the following.

5.2. Pricing problems

Exploiting the structure of our problem, we consider two distinct pricing problems tailored to generate cost-effective routes per vehicle type, i.e., one for BVs and one for SVs. The reduced cost of the columns for these pricing problems is given by Eq. (18) for BVs and by Eq. (19) for SVs, where $\pi_u^{(14)}$, $\pi_j^{(15)}$, and $\pi_l^{(16)}$ ($l \in \{1,2\}$) denote the dual variables associated with constraints (14), (15), and (16) in the RMP, respectively.

$$\min_{r\in\Psi_1}\left\{ \boldsymbol{\sigma}_r - \sum_{u\in C}\sum_{j\in V_u} A_{rj}\pi_u^{(14)} + |K_2| \sum_{j\in V_s} A_{rj}\pi_j^{(15)} - \pi_1^{(16)} \right\} \tag{18}$$

$$\min_{r\in\Psi_2}\left\{\boldsymbol{\sigma}_r - \sum_{u\in C}\sum_{j\in V_u} A_{rj}\pi_u^{(14)} - \sum_{j\in V_s} A_{rj}\pi_j^{(15)} - \pi_2^{(16)}\right\} \tag{19}$$

The complete mixed-integer-programming formulations for the BV and the SV pricing problem are presented in Appendix A and Appendix B, respectively. Solving these mixed-integer programming formulations of the pricing problems requires a high computational effort. We reduce this effort by implementing a dynamic programming-based labeling algorithm, which considers the pricing problem as a variant of the *elementary shortest path problem with resource constraints* (ESPPRC), for which we refer the interested reader to the review of Irnich and Desaulniers (2005). The following subsection provides a detailed description of this labeling algorithm for the case of SVs. The same algorithmic framework can be applied to the case of BVs by removing the capacity constraints and using the corresponding dual values.

5.3. Labeling algorithm

In our study, the pricing problem is solved exactly using a forward labeling algorithm. This algorithm incrementally constructs feasible routes across our time-expanded network without enumerating all possible paths. In this context, a label L, representing a partial or complete path $p(L)$ in the network, is characterized by a tuple $L = (\bar{c}, v, V_L, q)$, where $\bar{c}$ denotes the reduced cost of the path, v the end vertex, V_L the set of visited customers, and q the total delivered quantity since the last replenishment, which is reset after visiting a depot or satellite location. For clarity and easier representation of the components of a specific label L, we refer to these components as $\bar{c}(L)$, $v(L)$, $V_L(L)$, and $q(L)$ respectively. Enabling the reconstruction of complete paths at the end of the algorithm, each label maintains a link to its predecessor. The algorithm initiates at the origin node (node 0 at time stamp 0) with an initial label $L_0 = (-\pi_l^{(16)}, 0, \{0\}, 0)$, representing the starting point. Label extension is then performed iteratively at each node $i \in V$ along all arcs $(i, j) : \forall j \in V$, resulting in a new label with the following properties:

$$\begin{aligned}
\bar{c}(L') &= \bar{c}(L) + \bar{c}_{ij} \\
v(L') &= j \\
V_L(L') &= \begin{cases} V_L(L) \cup \{V_j\}, & \text{if } j \in V_c \\ V_L(L), & \text{o.w.} \end{cases} \\
q(L') &= \begin{cases} q(L) + d_j, & \text{if } j \in V_c \\ 0, & \text{o.w.} \end{cases}
\end{aligned}$$

This determines the reduced cost and end vertex of the new label. Moreover, if j is a customer node, the visited customer set is updated to include the customer ID of node j (denoted by V_j, which maps each node j in the time-expanded network to its corresponding customer) and the demand of node j is added to the total quantity delivered since the last replenishment. Otherwise, the visited customer set remains the same as the predecessor label, and the delivered quantity is reset to 0, as non-customer nodes only consist of depot and satellite nodes, at which the small vehicles get replenished.

Taking this into account, each resulting new label is then assessed with regard to its feasibility in terms of the vehicle's total load and the customers visited. Feasible labels are retained for further extension until the sink node is reached, while infeasible ones are discarded. To manage the potentially large number of feasible labels and enhance computational efficiency, we implement a set of dominance rules. In this context, we consider a label L_1 to be dominant over a label L_2 if the following dominance conditions are met,

$$v(L_1) = v(L_2) \tag{20}$$

$$\bar{c}(L_1) \leq \bar{c}(L_2) \tag{21}$$

$$q(L_1) \leq q(L_2) \tag{22}$$

$$V_L(L_1) \subseteq V_L(L_2) \tag{23}$$

stating that (i) both labels must end at the same node, (ii) L_1 must have a lower or equal reduced cost (including the revenue made from visiting customers), (iii) a lower or equal load, and (iv) the set of customers visited by L_1 must be a subset of those visited by L_2. These conditions collectively ensure that, any feasible extension of L_2 is known to be feasible for L_1 and therefore L_2 can be discarded. However, as rule (23) is highly restrictive and computationally expensive to include in the algorithm, we modify this rule by implementing the idea of unreachable nodes sets $\mathcal{U}$, based on already visited customers and resource limitations, as introduced by Feillet et al. (2004). More specifically, $\mathcal{U}$ is defined for each label, considering customer nodes that have already been visited and therefore cannot be revisited, as well as customer nodes whose demand cannot be served due to insufficient inventory. As such, this method effectively identifies nodes that cannot be feasibly visited in future extensions of the current path, replacing rule (23) by:

$$\mathcal{U}_{(L_1)} \subseteq \mathcal{U}_{(L_2)} \tag{24}$$

This new rule ensures that any infeasible extension of L_1 is also infeasible for L_2, thereby respecting rule (23).

Despite the improvements resulting from the introduction of formulation (24), solving the ESPPRC remains computationally demanding. To address this, we introduce several acceleration strategies, transforming our labeling algorithm into a three-phased

approach, in which we first relax the dominance rule (24). For this purpose, we remove for each label all nodes in their corresponding unreachable vertex set from the neighbor set. While this significantly speeds up the algorithm and ensures the feasibility of generated labels, this may lead to discarding potentially useful labels, rendering the solution suboptimal. If the columns generated in this phase have negative reduced costs, these columns are incorporated into the RMP. Once the algorithm can no longer find solutions with negative reduced costs, we transition to the second phase, in which we no longer remove the unreachable vertex set from the neighbor set of labels, but instead focus on addressing only 2-cycle violations in the routes, solving a modified problem referred to as the 2-cyc-SPPRC. In this phase, we add only the columns which are found to be elementary and have a negative reduced cost to the RMP. In case this 2-cyc-SPPRC leads to a solution with non-negative reduced cost, the algorithm terminates as no elementary routes with negative reduced cost remain. Conversely, if an elementary solution with negative reduced cost is found, it is incorporated into the RMP for the subsequent iteration. For routes with k cycles, where $k \geq 3$, the algorithm moves to the third phase.

In this final phase, we reinstate the full dominance rule (24) and solve the ESPPRC in its entirety. If the solution has a negative reduced cost, the corresponding columns are added to the RMP. If no such solution is found, it indicates that there are no more columns with negative reduced cost to be generated. This ensures that the solution obtained is optimal, thus concluding the labeling algorithm. This approach is particularly effective since solving the ESPPRC to optimality in every iteration of the column generation is unnecessary, except for the final iteration at each node of the branch and bound tree. Similar acceleration procedures have been proposed by, for example, Costa et al. (2019) and Ghoniem et al. (2015). Preliminary experiments have shown that the absence of heuristic pricing leads to a significantly worse performance, underscoring its importance in the algorithm's overall efficacy.

5.4. Branching and computational considerations

In order to obtain integer solutions in case the LP relaxation yields fractional results, we employ the branching strategy proposed by Desrosiers et al. (1984). This well-established strategy presents a simple approach, branching on the arc flow variables, denoted as x_{ij}, taking the value of 1 if there is an arc between nodes i and j in the associated column with a fractional value. After enforcing the branching decision, both the RMP and the pricing problems must ensure that the returned columns adhere to this decision. To speed up the process of finding integer feasible solutions, we furthermore apply a mixed integer linear programming (MILP) heuristic at the end of each CG iteration before we start branching (Munari et al., 2019). For this purpose, we impose the variables in the RMP as integers and solve the resulting MILP. We use a general-purpose solver with a time limit (30 s in our study) to update the bounds on the column generation tree. This approach, though simple, has been proven to positively impact the convergence of the branch-and-price tree (Alvarez and Munari, 2017).

To obtain an initial solution to our problem, we propose a clustering-based algorithm that consists of three main steps. In this algorithm, each cluster corresponds to a set of nodes of a route for BVs or SVs. In the first step, to generate a pre-specified number of randomly selected but potentially good clusters, we first utilize a modified version of the *Approximate Stability Assessment* procedure developed by Özener et al. (2013). In the second step, considering the vehicle's capacity, a random constructive algorithm generates routes for BVs and SVs. After the clusters are generated and associated routes and costs are calculated, the final step implements a tractable set packing problem to decide the delivery schedules of the BVs and SVs. A detailed description of the individual steps of this algorithm is presented in Appendix C.

6. Computational experiments

We conducted extensive numerical experiments to validate our problem formulation and test the performance of our solution approach in comparison with a state-of-the-art general-purpose optimization solver. These experiments have, moreover, been designed to investigate the effect of different instance structures on the algorithmic performance and the obtained solution structures. The MILP formulation presented in Section 4.2 and the proposed B&P algorithm are coded in Python and Gurobi 9.5. All experiments were executed on a machine equipped with an AMD EPYC 2.0 GHz Processor (with IBPB, 4 Cores) and 32 GB RAM.

6.1. Instances

We tested our method on two sets of instances. The first set consists of 100 randomly generated instances with up to five vehicles (containing both SVs and BVs), five time stamps, two depots, and two satellite locations, considering up to 50 customers. The second set of instances consists of larger instances adapted from the instances proposed by Solomon (1987), with 100 customers, and Gehring and Homberger (1999), with 200 customers. Using these sources, we obtain 12 distinct geographical datasets (6 with 100 customers and 6 with 200 customers), based on which we create 720 instances. These instances can then be classified in two ways; (i) based on the tightness of the considered time windows (where type 1 problems have tight time windows and type 2 problems have wide time windows) or (ii) according to the distribution of the demand locations resulting in random, clustered, and mixed (i.e., random and clustered) instances. As these instances do not have predefined time slots for deliveries and do not consider multiple depots and satellite locations, we adapt these instances slightly to fit our problem. For this purpose, we first transform the continuous time window data provided by the Solomon instances into a discrete time stamp framework, by dividing the total planning horizon, defined by the depot's ready time and due date in the original instances, into equal intervals (creating either 10 or 20 timestamps depending on the instance setting). Each customer's time window and service time are then mapped to these timestamps by identifying all intervals that overlap with their ready time and due date, creating corresponding customer nodes for each feasible timestamp. In this context, it should be noted that a customer's time window may span multiple time stamps. To

Table 2
Performances of algorithms.

Case	Cust.	T.S.	Ins.	Time (s)		Gap		#Opt		#Feasible	
				MILP	B&P	MILP	B&P	MILP	B&P	MILP	B&P
I	100	10	120	3682	1790	0.19	0.01	1	88	120	120
		20	120	3477	2148	0.22	0.06	11	60	119	120
	200	10	60	3694	3219	0.19	0.02	1	6	18	60
		20	60	3788	3598	0.37	0.20	0	1	42	60
	Total		360	–	–	–	–	13	155	299	360
	Average			3660	2689	0.24	0.07	–	–	–	–
II	100	10	120	3008	108	0.05	0.00	32	120	120	120
		20	120	2116	851	0.08	0.01	52	113	120	120
	200	10	60	3662	918	0.06	0.03	0	53	56	60
		20	60	3824	2823	0.16	0.11	1	20	58	60
	Total		360	–	–	–	–	85	306	354	360
	Average			3153	1175	0.09	0.04	–	–	–	–
Total	100	10	240	3345	949	0.12	0.00	33	208	240	240
		20	240	2797	1500	0.15	0.04	63	173	239	240
	200	10	120	3678	2069	0.13	0.03	1	59	74	120
		20	120	3806	3211	0.27	0.16	1	21	100	120
	Total		720	–	–	–	–	98	461	653	720
	Average			3406	1932	0.17	0.05	–	–	–	–

address the issue of the missing depot and satellite locations, we then consider the given depot location as the origin location for both SVs and BVs, while generating additional depot and satellite locations for our problem. When generating these locations, we distinguish between two cases: *Case I*, where the depots are far from the center of the grid, and *Case II*, where the depots are in the center of the grid. In both cases, the satellites are in/around the center, and the total number of depots and satellites is based on the average number of customers per time stamp ($|V_d| + |V_s| \leq \alpha \frac{|V_c|}{|T|}, 0.2 \leq \alpha \leq 0.3$). For instance, if there are, on average, ten customers per time stamp, possible combinations for depot-satellite number pairs are (1,1) or (1,2). The total number of vehicles is also limited by the average number of customers per time stamp, such that $|K_1| + |K_2| \leq 0.75 \frac{|V_c|}{|T|}$. The ratio between the number of BVs and SVs is $1/1, 1/2$, and $1/3$. The number of time stamps in the planning horizon is set to either 10 or 20, which allows us to assign, on average, 5, 10, and 20 customers per time stamp, depending on the size of the instance.

This setup results in a total of 720 instances, comprising 480 instances which were generated from the six base-instances with 100-customers (considering 20 different parameter settings for each base instance-case-time horizon combination, resulting in $6 \times 2 \times 2 \times 20$ instances), and 240 instances generated from our six base instances with 200 customers, (considering 10 different parameter settings for each combination, which leads to $6 \times 2 \times 2 \times 10$ instances). The 100-customer instances can be further divided into 240 instances with tight time windows and 240 with wide time windows, which can be further categorized into 120 instances for each depot location case. Similarly, the 200-customer instances are divided into 120 instances with tight time windows and 120 with wide time windows, which can be further categorized into 60 instances for each depot location case. The generated instances are available upon request.

6.2. B&P algorithm results

This section focuses on evaluating the performance of the proposed B&P algorithm. For this purpose, we solve the instances presented in Section 6.1 with the B&P algorithm and the MILP model from Section 4.2, setting a time limit of 3600 s for both methods. When applying both methods to the instances we generated randomly, we see that both methods can quickly solve these relatively small instances, proving optimality. Therefore, we mostly used these instances to inspect, analyze, and understand the behavior of our methods against the changes in the instance settings. The findings from this preliminary analysis show that the most significant factors affecting the performance of the solution methods are the locations of the depots and satellites, their distances to customer locations, the number of time stamps in the planning horizon, as well as the length of the considered time windows. The results of both methods for the 720 larger-sized instances are, thus, structured according to these factors.

A comparison of the performance of the MILP and the B&P algorithm on the 720 larger-sized instances, taking into account the different case settings, as well as the number of customers and time stamps, is shown in Table 2, while Table 3 presents a similar comparison under consideration of the different time window lengths. Both tables present for this purpose the number of customers (Cust.) in column 2, the number of time stamps (T.S.) in column 3, and the total number of instances (Ins.) within the specified instance setting in column 4. The subsequent columns in each table focus on evaluating the performance of the MILP and the B&P algorithm by comparing their average run times (Time), optimality gaps (Gap), and the number of optimal (#Opt) and feasible (#Feasible) solutions found.

From these tables, we can observe that the B&P algorithm is considerably faster than the MILP in terms of run times, with an overall average time saving of approximately 43% (equivalent to about 1500 s). The comparison between Case I and Case II, presented in Table 2, shows, furthermore, that the B&P performs even better in Case II, with an average difference of around 2000

Table 3
Performances of algorithms w.r.t. the length of demand time windows.

Window Length	Cust.	T.S.	Ins.	Time (s)		Gap		#Opt		#Feasible	
				MILP	B&P	MILP	B&P	MILP	B&P	MILP	B&P
Tight	100	10	120	3121	642	0.13	0.02	23	115	120	120
		20	120	2271	1093	0.15	0.04	54	97	120	120
	200	10	60	3752	2165	0.08	0.02	0	31	38	60
		20	60	3737	3081	0.15	0.06	1	13	49	60
Wide	100	10	120	3586	1177	0.13	0.01	3	90	120	120
		20	120	3373	2049	0.16	0.04	10	68	117	120
	200	10	60	3776	2215	0.09	0.02	0	29	37	60
		20	60	3860	3371	0.27	0.19	0	8	49	60

Table 4
Settings of the scenarios.

Scenario	Case	Synchronization
S1	I	No
S2	II	No
S3	I	Yes
S4	II	Yes

seconds. More generally, it can be seen that placing the depots far from the center of the grid (i.e., as in Case I) increases the overall run times for both methods. A similar effect can be observed when the number of customers increases. In addition to being generally faster, the B&P also outperforms the MILP in terms of solution quality, providing consistently lower optimality gaps. Overall, the B&P provides an optimality gap of 5%, while the average optimality gap of the MILP is at 17%. At the more detailed level, the results show that even for the more challenging larger instances with 200 customers the B&P still maintains a good performance. Comparing the number of optimal and feasible solutions found by each method further highlights the superiority of the B&P. While the MILP struggles to find feasible solutions for almost 10% of the instances, the B&P always finds a feasible solution. In terms of optimal solutions, the B&P is able to prove optimality for 461 of the 720 instances, while the MILP provides optimal solutions for only 98 instances. Aside from the results presented in Table 2, we also investigate the impact of the number of depots and satellites on the performance of the B&P. Our findings indicate a general increase in both the optimality gaps and run times as the number of depots and satellites increases, highlighting their essential role in the performance of both methods. Notably, the B&P significantly outperforms the MILP in terms of both optimality gaps and run times, further demonstrating its efficiency and effectiveness.

Investigating the impact of the time window lengths on the performance of the B&P, Table 3 indicates, furthermore, that the problem becomes more challenging as the length of the time window increases, showcasing an increase in the average run times for long time windows. Despite the longer run times, the solution quality only seems to be affected in the case of the largest instances (with 200 customers and 20 time stamps), for which wide time windows may trigger a substantial increase in the number of customer node copies.

6.3. Managerial insights

This section presents valuable managerial insights into different aspects of the problem, particularly on the effect of synchronization and the impact of the number of depots, satellites, and vehicles used.

Insights into the effect of synchronization

We compare four scenarios to investigate synchronization's impact and potential benefits within different planning settings. In this context, scenarios S1 and S2 assume a situation where no synchronization is allowed/possible, representing the planning setting of *Case I* and *Case II*, respectively. For our experiments, we implement these scenarios by removing the synchronization-related constraints ((15) in Master Problem), as well as the satellite nodes and associated arcs from the considered network structure. Scenarios S3 and S4, in contrast, consider the situation with synchronization while again representing the setting of *Case I* and *Case II*, respectively. These scenarios are overviewed in Table 4.

Highlighting differences in profit, the number of depot visits, and the revenue and cost associated with each vehicle type, Fig. 3 compares these four scenarios, presenting the results as a percentage of scenario S1. It must be noted that all results are based on average values, considering only the instances that could be solved to optimality for all four scenarios. Comparing these results shows that synchronization has an overall positive effect on the total profits in the system. This is in line with expectations, as the consideration of synchronization offers more flexibility for planning the routes of the SVs. Comparing the results for the two cases, it can be seen that synchronization's positive effect is more pronounced when the depots are located far from the center (i.e., *Case I*, as in scenario S3). In this case, synchronization leads to significant cost savings in the routing cost of the SVs, which can now be replenished by BVs in the center instead of returning to the depots located far away. This quick replenishment in the center further allows SVs to satisfy more customer orders, which is clearly visible from their increased revenues. This effect can also be seen in the

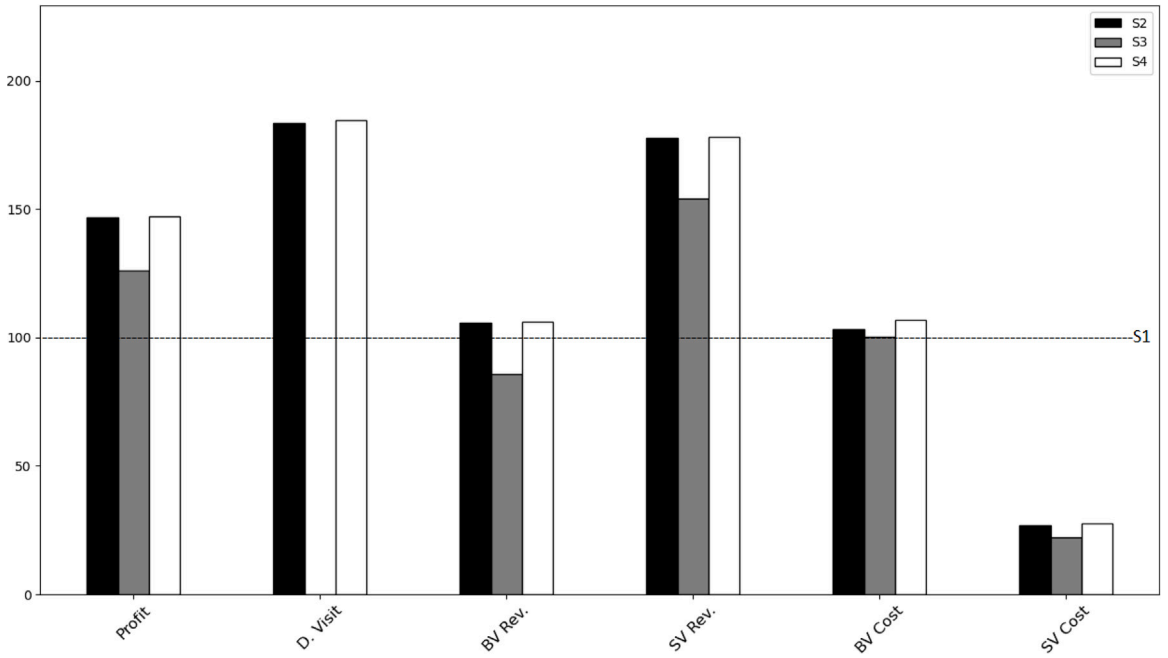

Fig. 3. Comparison of different scenarios as a percentage of scenario S1.

drastic decrease in depot visits. Furthermore, this also affects the revenue of the BVs, as they now spend time visiting satellites to replenish SVs instead of serving customers. In contrast, comparing scenarios S2 and S4, which both consider the *Case II* setting, we observe that the effect of synchronization is significantly lower than for the *Case I* setting. This is a result of the central placement of the depots, which reduces the benefits of using BVs to replenish SVs at satellites due to the short travel distance to the depots. As such, SVs mainly get replenished at the depots so that BVs can focus on serving customers. However, even though the effect is small, we can still see a positive effect of synchronization on the profit due to reduced depot visits and lower costs for both BVs and SVs.

Insights into the effect of the number of depots and satellites

Comparing the results of the scenarios shown in Fig. 3, we observe that the placement of the depots plays an essential role in synchronization and its potential impact on profit and cost savings. Based on our results, we identify the number of satellite locations as another critical factor impacting the success of synchronization. More specifically, as the number of centrally located satellites increases (i.e., more synchronization locations become available), the SVs' cost decreases due to the shorter distances to these replenishment locations. A similar effect can be observed for an increase in the number of centrally located depot locations, resulting in a decrease in the SV cost due to closer replenishment facilities. However, unlike when increasing the number of satellite locations in the center, increasing the number of centrally located depots reduces synchronization between vehicles. As a result of this general lower need for synchronization, we can conclude that a simultaneous increase in the number of satellite locations does not have a notable effect on the profit, revenue, and cost of the solutions.

Insights into the effect of the number of vehicles

Another critical factor for customer demand satisfaction and cost management in city logistics is the fleet size, i.e., the number of available BVs and SVs (Anand et al., 2015). Hence, we try to understand the effect of the number of available BVs and SVs on the solution structure under consideration of the placement of the depots. Comparing the optimal solutions for different numbers of vehicles with respect to different performance indicators, Fig. 4 shows, in this context, that an increase in the number of BVs in *Case I* leads to an increase in the total profit as the additional BVs can be used to satisfy more customer demands. This is accompanied by an increase in the revenue of the BVs and a general increase in vehicle synchronization.

The use of synchronization in *Case I* is, however, mostly driven by the number of SVs. As the number of SVs increases, there is a notable rise in synchronization, indicating a direct correlation between these two factors. Despite the significant effect of synchronization and the revenue gained by the SVs, the effect of increasing the number of SVs on total profit is less clear. This is because as synchronization increases, BVs focus on the replenishment of SVs instead of serving customer demands, affecting the BVs' revenue and, thus, the potential total profit. As such, increasing the number of BVs takes priority when trying to satisfy customer demands and increase total profit, considering the fixed and other operating costs to be the same.

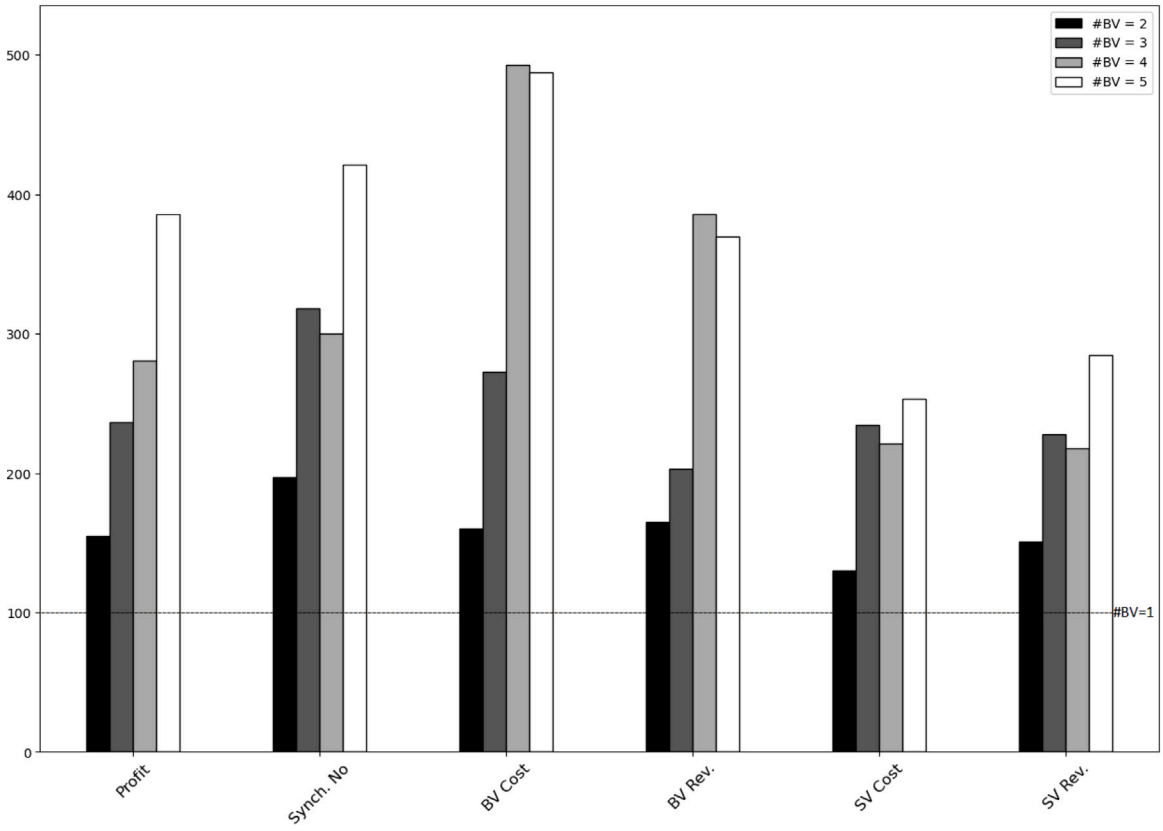

Fig. 4. (*Case I*) Comparison of the solutions for different numbers of big vehicles (BV) with regards to different performance indicators represented as a percentage of the case with only one BV (indicated by the dashed line).

In the case where the depot is centrally located, i.e., *Case II*, SVs replenish at the depots instead of satellite locations and focus on the demand satisfaction as much as possible, which eliminates the need for synchronization (see Fig. 5). Consequently, increasing the number of vehicles of any type directly correlates with an increase in the revenue of the considered vehicle type as well as the overall profit.

7. Conclusions

This study introduces a new variant of the *Two-Echelon Vehicle Routing Problem* (2E-VRP) in which the echelons partially overlap, both echelons' vehicles make delivery to customers, and one echelon's vehicles are replenished by using the other. The problem is highly complex due to the synchronization between the vehicles and multi-trip between the depots, satellites, and customer nodes. We formulate the problem mathematically on a time-expanded network and propose a branch-and-price algorithm to solve it. The results from our computational experiments on a set of 720 instances, adapted from the literature, show that the branch-and-price outperforms a state-of-the-art general-purpose optimization solver and provides optimal solutions and reasonable run times for instances with up to 200 customers. However, the number of instances solved to optimality decreases as customers, time stamps, depots, satellites, and time window lengths increase. In our computational experiments, we investigated the effect of the placement of the depots on the performance of the algorithms as well as the solution structure.

In addition, we conduct deeper analysis to obtain valuable managerial insights, particularly emphasizing the effect of synchronization and the impact of the number of satellites, depots, and vehicles used. As expected, the consideration of synchronization provides more flexibility for planning vehicle routes and positively affects the total profit. This effect is, however, more pronounced when depots are far from the center since synchronization, i.e., the replenishment of *small vehicles* (SVs) by *big vehicles* (BVs), can significantly reduce the travel distances of the SVs, which no longer need to return to the depots.

The number of satellite and depot locations is another critical factor affecting the success of synchronization. More specifically, an increase in the number of satellite locations will lead to an increase in synchronization and a decrease in the SV cost. In contrast, increasing the number of centrally located depots reduces vehicle costs and the need for vehicle synchronization. In addition, we observe a close link between the number of vehicles and the obtained total profit. In this context, we can see that increasing the number of BVs takes priority in case depots are located far from the center, whereas increasing the number of SVs leads to more synchronization without necessarily increasing profit, considering the fixed and operational costs to be the same.

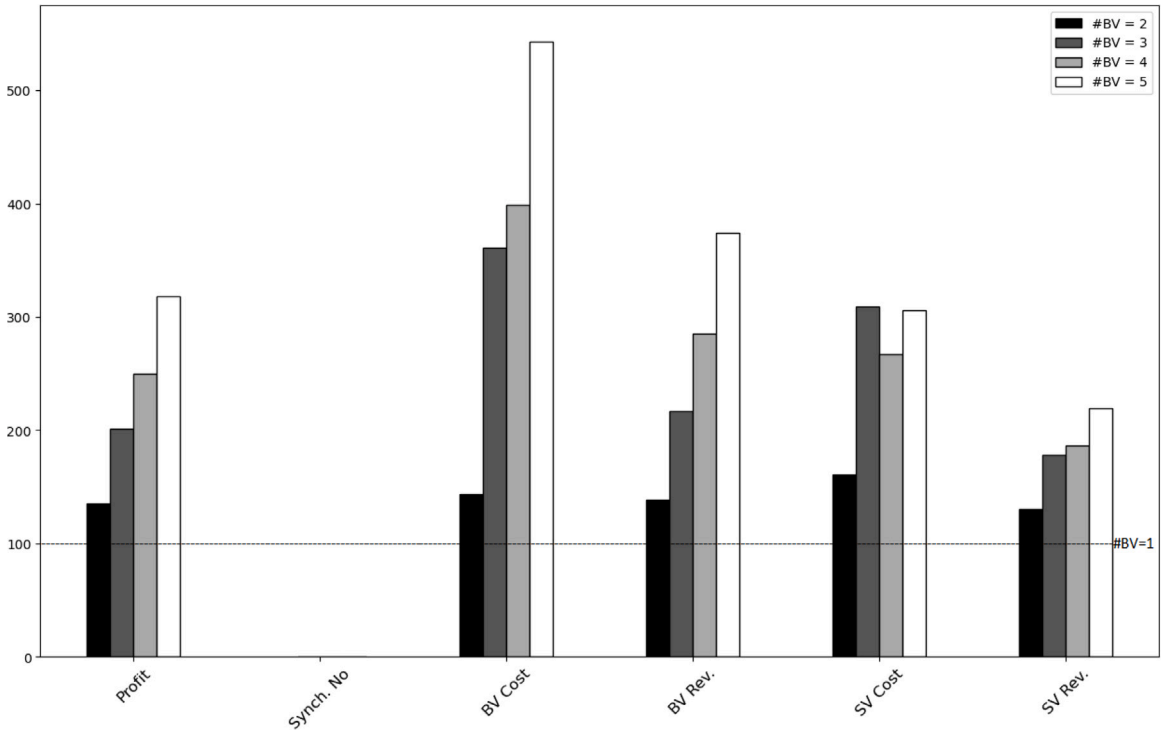

Fig. 5. (*Case II*) Comparison of the solutions for different numbers of big vehicles (BV) with regards to different performance indicators represented as a percentage of the case with only one BV (indicated by the dashed line).

Future research could build on the findings in this research by incorporating stochastic elements, such as variable customer demand and travel times, enhancing the practical applicability of the proposed solutions. In addition, from a methodological perspective, the potential of other labeling algorithms, e.g., bidirectional labeling and branching strategies, could be explored.

CRediT authorship contribution statement

I. Edhem Sakarya: Writing – review & editing, Writing – original draft, Visualization, Methodology, Formal analysis, Conceptualization. **Milad Elyasi:** Writing – review & editing, Writing – original draft, Methodology, Formal analysis, Conceptualization. **S.U.K. Rohmer:** Writing – review & editing, Writing – original draft, Supervision. **O. Örsan Özener:** Supervision. **Tom Van Woensel:** Supervision. **Ali Ekici:** Supervision.

Appendix A. Single BV pricing problem

Pricing problem PP-y, in which the decision variable z_{ij} denotes if there is an arc between node i and j, is provided below:

PP-y

$$\min \quad \sum_{i\in V}\sum_{j\in V} c^1_{ij} z_{ij} - \sum_{i\in V}\sum_{j\in V} p_j d_j z_{ij} - \pi^{(16)}_1 - \sum_{i\in V}\sum_{j\in V_c} \pi^{(14)}_j z_{ij} + \sum_{i\in V}\sum_{j\in V_s} |K_2| \pi^{(15)}_j z_{ij} \tag{25}$$

$$\text{s.t.} \quad z_{ij} \le \alpha^1_{ij} \qquad \forall i,j \in V \tag{26}$$

$$\sum_{i\in v_{o,1}}\sum_{j\in V} z_{ij} = 1 \tag{27}$$

$$\sum_{i\in V}\sum_{j\in v_{f,1}} z_{ij} = 1 \tag{28}$$

$$\sum_{i\in V} z_{ij} = \sum_{i'\in V} z_{ji'} \qquad \forall j \in \bar{V} \tag{29}$$

$$\sum_{i\in V}\sum_{j\in V_u} z_{ij} \le 1 \qquad \forall u \in C \tag{30}$$

$$z_{ij} \in \{0,1\} \qquad \forall i,j \in V \tag{31}$$

In PP-y, the objective function (25) aims to minimize the total delivery cost of a BV, which is calculated as the total traveling cost minus the revenue gained by satisfying the demand of the customers. Constraints (26) ensure, in this context, that the BV can traverse only available arcs, while constraints (27) and (28) impose that the BV leaves from and returns to its origin when starting and ending its trip. Constraints (29) are general network flow constraints, constraints (30) make sure at most one node of a customer location is visited, and constraints (31) define the decision variables as binary.

Appendix B. Single SV pricing problem

Pricing problem PP-x, in which w_{ij} decision variable denotes if there is an arc between node i and node j, is given as follows:

PP-x

$$\min \quad \sum_{i\in V}\sum_{j\in V} c^2_{ij} w_{ij} - \sum_{i\in V}\sum_{j\in V} p_j d_j w_{ij} - \pi_2^{(16)} - \sum_{i\in V}\sum_{j\in V_c} \pi_j^{(14)} w_{ij} - \sum_{j\in V_s}\sum_{i\in V} \pi_j^{(15)} w_{ij} \tag{32}$$

$$\text{s.t.} \quad w_{ij} \le \alpha^2_{ij} \qquad \forall i,j \in V \tag{33}$$

$$\sum_{i\in v_{o,2}}\sum_{j\in V} w_{ij} = 1 \tag{34}$$

$$\sum_{i\in V} w_{ij} = \sum_{i'\in V} w_{ji'} \qquad j \in \bar{V} \tag{35}$$

$$\sum_{i\in V}\sum_{j\in v_{f,2}} w_{ij} = 1 \tag{36}$$

$$\sum_{i\in V}\sum_{j\in V_u} w_{ij} \le 1 \qquad \forall u \in C \tag{37}$$

$$I_j \le I_i - d_i + M(1 - w_{ij}) \qquad \forall i \in V_c, j \in V \tag{38}$$

$$I_i - d_i - M(1 - w_{ij}) \le I_j \qquad \forall i \in V_c, j \in V \tag{39}$$

$$0 \le I_j \le Q \qquad \forall j \in V \tag{40}$$

$$w_{ij} \in \{0,1\} \qquad \forall i,j \in V \tag{41}$$

In PP-x, the objective function (32) aims to minimize the total delivery cost of an SV, which is calculated as the total traveling cost minus the revenue gained by satisfying the demand of the customers. Constraints (33) ensure that the SV can only traverse arcs that are available to it, and constraints (34) require the SV to leave from the origin. Constraints (35) and (36) are network flow balance constraints with the latter one ensuring that the SV returns to the origin at the end of its trip. Constraints (38) and (39) are inventory balance constraints for the customer nodes with small-sized demands. Constraints (37) state that no more than one customer node can be visited. Constraints (40) restrict the inventory of the SV and finally constraints (41) define the decision variables as binary.

Appendix C. Initial solution

The steps of the proposed algorithm for the initial solution are presented as follows:

[***Step 1***] ***Iterative cluster generation***

We generate different routing patterns for BV and SV by using an iterative clustering algorithm. In each iteration, a random point is chosen on the map as the base point, and a probability is assigned for each customer concerning its distance from the base point. The customers in each cluster are then chosen using a roulette wheel selection approach. Since we seek to generate unique clusters when a repetitive cluster is generated, such a generation is denoted as a failure. The associated clustering approach and the probability calculation used in the roulette wheel selection are inspired by Ekici et al. (2015). We should note that we limit the number of unsuccessful trials terminating each vehicle type's cluster generation to $25000\times|K_l|$.

[***Step 2***] ***Determining the routes for BVs and SVs and the associated costs***

After the clusters are generated, considering the capacities of the vehicles, we utilize a random constructive algorithm to generate routes for BVs and SVs. Assuming I_{cl} denotes the set of nodes on the cluster c associated with vehicle type l, cs denotes the 'current node' at the related iteration of the algorithm, *CurrentCapacity* stands for the available capacity of SV, which at the start of the algorithm is equal to Q, *TotalCost* stands for the delivery cost of the vehicles with the initial value of 0. Finally, *VisitedSites* stands for the set of the nodes the vehicles visit. The proposed constructive algorithm sorts the nodes on the clusters according to their distance from the current node. Following that, each node is added to the set of visited nodes while controlling the feasibility of the solution. The steps of the constructive algorithm for BVs and SVs are presented as follows:

[*Step 2.1*] Using the following equation, calculate the score of choosing each node for being on each vehicle (s_{il}) :

$$s_{il} = \begin{cases} \eta_1 \frac{p_i}{\sum_{j \in I_{cl}} p_j} - \eta_2 \frac{c^l_{cs,i}}{\sum_{j \in I_{cl}} c^l_{cs,j}} & , \quad \text{if } \alpha^l_{cs,i} = 1 \\ -M, & \quad \text{otherwise.} \end{cases} \qquad \forall i \in I_{cl}, l \in L$$

[*Step 2.2*] Sort the nodes in I_{cl} in a descending order of scores and call this array I'_{cl}.

[*Step 2.3*] **If** $l = 1$:

For each the node j in I'_{c1} **do**:

If a generated random number is less than ξ, do the following updates:

$VisitedSites \leftarrow \{j\}$

$TotalCost = TotalCost$ - $p_j + c^1_{cs,j}$

$cs = j$.

Else:

Calculate the score of different satellites and depots (s'_{i1}) using the following equation:

$$s'_{i1} = \begin{cases} \frac{c^1_{cs,i}}{\sum_{j \in V_s \cup V_d} c^1_{cs,j}} & , \quad \text{if } \alpha^1_{cs,i} = 1 \\ -M, & \quad \text{otherwise.} \end{cases} \qquad \forall i \in V_s \cup V_d$$

$Candidate = \min_i \{s'_{i1}\}$

$VisitedSites \leftarrow \{Candidate\}$

$TotalCost = TotalCost + c^1_{cs,candidate}$

$cs = Candidate$.

Else:

For each the node j in I'_{c2} **do**:

If $CurrentCapacity - d_j \geq 0$, do the following updates:

$CurrentCapacity = CurrentCapacity - d_j$

$VisitedSites \leftarrow \{j\}$

$TotalCost = TotalCost$ - $p_j + c^2_{cs,j}$

$cs = j$

Else:

Calculate the score of different satellites (s'_{i2}) using the following equation:

$$s'_{i2} = \begin{cases} \frac{c^2_{cs,i}}{\sum_{j \in V_s \cup V_d} c^2_{cs,j}} & , \quad \text{if } \alpha^2_{cs,i} = 1 \\ -M, & \quad \text{otherwise.} \end{cases} \qquad \forall i \in V_s \cup V_d$$

$Candidate = \min_i \{s'_{i1}\}$

$VisitedSites \leftarrow \{Candidate\}$

$TotalCost = TotalCost + c^2_{cs,j}$

$cs = Candidate$.

[*Step 3*] *Determining the clusters*

In the third step, after the clusters are generated and their delivery costs are calculated, we decide on the delivery schedules of the vehicles by solving a tractable set packing problem. We use Δ_l to denote the set of clusters for vehicle type l and ρ_{δ_l} to denote the total cost of the cluster δ_l which is calculated in *Step 2*. Also, we define the binary parameter $\psi_{i\delta_l}$, which takes 1 if node i is visited on cluster δ_l and 0 otherwise. We choose the best cluster combination for vehicles while guaranteeing that each node is on a vehicle or not selected. We solve the following set packing problem **[SPP]** . The decision variable is as follows:

$$H_{\delta_l} = \begin{cases} 1, & \text{if cluster } \delta_l \text{ is chosen for vehicle types } l \\ 0, & \text{otherwise.} \end{cases} \qquad \delta_l \in \Delta_l, l \in L$$

We solve the following mathematical model to determine the best combination of clusters:

SPP :

$$\min \sum_{\delta_l \in \Delta_l} \sum_{l \in L} \rho_{\delta_l} H_{\delta_l} \tag{42}$$

$$\text{s.t.} \quad \sum_{\delta \in \Delta_l} H_\delta \leq |K_l| \qquad \forall l \in L \tag{43}$$

$$\sum_{\delta \in \Delta_l} \sum_{l \in L} \psi_{i\delta_l} H_{\delta_l} \leq 1 \qquad \forall i \in V_c \tag{44}$$

$$H_{\delta_l} \in \{0, 1\} \qquad \forall \delta_l \in \Delta_l, l \in L \tag{45}$$

In [**SPP**], the objective is to minimize the total delivery cost. Constraints (43) make sure that the number of the clusters for each vehicle type are limited by the number of vehicles for that type. Constraints (44) guarantee that each customer node is included in at most one of the chosen clusters. Constraints (45) are the binary restrictions.

References

Alvarez, Aldair, Munari, Pedro, 2017. An exact hybrid method for the vehicle routing problem with time windows and multiple deliverymen. Comput. Oper. Res. 83, 1–12.

Anand, Nilesh, Van Duin, Ron, Quak, Hans, Tavasszy, Lori, 2015. Relevance of city logistics modelling efforts: A review. Transp. Rev. 35 (6), 701–719.

Anderluh, Alexandra, Hemmelmayr, Vera C., Nolz, Pamela C., 2017. Synchronizing vans and cargo bikes in a city distribution network. CEJOR Cent. Eur. J. Oper. Res. 25 (2), 345–376.

Anderluh, Alexandra, Nolz, Pamela C, Hemmelmayr, Vera C, Crainic, Teodor Gabriel, 2021. Multi-objective optimization of a two-echelon vehicle routing problem with vehicle synchronization and 'grey zone' customers arising in urban logistics. European J. Oper. Res. 289 (3), 940–958.

Aslan, Alper, Altinoz, Buket, Ozsolak, Baki, 2021. The link between urbanization and air pollution in Turkey: Evidence from dynamic autoregressive distributed lag simulations. Environ. Sci. Pollut. Res. 28 (37), 52370–52380.

Balas, Egon, 1989. The prize collecting traveling salesman problem. Networks 19 (6), 621–636.

Baldacci, Roberto, Mingozzi, Aristide, Roberti, Roberto, Calvo, Roberto Wolfler, 2013. An exact algorithm for the two-echelon capacitated vehicle routing problem. Oper. Res. 61 (2), 298–314.

Barnhart, Cynthia, Johnson, Ellis L, Nemhauser, George L, Savelsbergh, Martin WP, Vance, Pamela H, 1998. Branch-and-price: Column generation for solving huge integer programs. Oper. Res. 46 (3), 316–329.

Belgin, Onder, Karaoglan, Ismail, Altiparmak, Fulya, 2018. Two-echelon vehicle routing problem with simultaneous pickup and delivery: Mathematical model and heuristic approach. Comput. Ind. Eng. 115, 1–16.

Belieres, Simon, Hewitt, Mike, Jozefowiez, Nicolas, Semet, Frédéric, 2021. A time-expanded network reduction matheuristic for the logistics service network design problem. Transp. Res. Part E: Logist. Transp. Rev. 147, 102203.

Bevilaqua, Andre, Bevilaqua, Diego, Yamanaka, Keiji, 2019. Parallel island based memetic algorithm with Lin–Kernighan local search for a real-life two-echelon heterogeneous vehicle routing problem based on Brazilian wholesale companies. Appl. Soft Comput. 76, 697–711.

Bianchessi, Nicola, Mansini, Renata, Speranza, M. Grazia, 2018. A branch-and-cut algorithm for the team orienteering problem. Int. Trans. Oper. Res. 25 (2), 627–635.

Boland, Natashia, Hewitt, Mike, Marshall, Luke, Savelsbergh, Martin, 2019. The price of discretizing time: A study in service network design. EURO J. Transp. Logist. 8 (2), 195–216.

Costa, Luciano, Contardo, Claudio, Desaulniers, Guy, 2019. Exact branch-price-and-cut algorithms for vehicle routing. Transp. Sci. 53 (4), 946–985.

Cuda, Rosario, Guastaroba, Gianfranco, Speranza, Maria Grazia, 2015. A survey on two-echelon routing problems. Comput. Oper. Res. 55, 185–199.

Dabia, Said, Ropke, Stefan, Van Woensel, Tom, De Kok, Ton, 2013. Branch and price for the time-dependent vehicle routing problem with time windows. Transp. Sci. 47 (3), 380–396.

Dang, Duc-Cuong, El-Hajj, Racha, Moukrim, Aziz, 2013. A branch-and-cut algorithm for solving the team orienteering problem. In: International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research. Springer, pp. 332–339.

de Lima, Vinícius L, Alves, Cláudio, Clautiaux, François, Iori, Manuel, de Carvalho, José M Valério, 2022. Arc flow formulations based on dynamic programming: Theoretical foundations and applications. European J. Oper. Res. 296 (1), 3–21.

Dellaert, Nico, Dashty Saridarq, Fardin, Van Woensel, Tom, Crainic, Teodor Gabriel, 2019. Branch-and-price–based algorithms for the two-echelon vehicle routing problem with time windows. Transp. Sci. 53 (2), 463–479.

Desrosiers, Jacques, Soumis, François, Desrochers, Martin, 1984. Routing with time windows by column generation. Networks 14 (4), 545–565.

Diao, Xudong, Qiu, Meng, Xu, Gangyan, 2024. Electric vehicle-based express service network design with recharging management: A branch-and-price approach. Comput. Oper. Res. 162, 106469.

Drexl, Michael, 2012. Synchronization in vehicle routing—a survey of VRPs with multiple synchronization constraints. Transp. Sci. 46 (3), 297–316.

Drexl, Michael, 2013. Applications of the vehicle routing problem with trailers and transshipments. European J. Oper. Res. 227 (2), 275–283.

Ekici, Ali, Özener, Okan Örsan, Kuyzu, Gültekin, 2015. Cyclic delivery schedules for an inventory routing problem. Transp. Sci. 49 (4), 817–829.

Enthoven, David LJU, Jargalsaikhan, Bolor, Roodbergen, Kees Jan, uit het Broek, Michiel AJ, Schrotenboer, Albert H, 2020. The two-echelon vehicle routing problem with covering options: City logistics with cargo bikes and parcel lockers. Comput. Oper. Res. 118, 104919.

Feillet, Dominique, Dejax, Pierre, Gendreau, Michel, 2005. Traveling salesman problems with profits. Transp. Sci. 39 (2), 188–205.

Feillet, Dominique, Dejax, Pierre, Gendreau, Michel, Gueguen, Cyrille, 2004. An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. Networks: Int. J. 44 (3), 216–229.

Ferrero, Enrico, Alessandrini, Stefano, Balanzino, Alessia, 2016. Impact of the electric vehicles on the air pollution from a highway. Appl. Energy 169, 450–459.

Gehring, Hermann, Homberger, Jörg, 1999. A parallel hybrid evolutionary metaheuristic for the vehicle routing problem with time windows. In: Proceedings of EUROGEN99, vol. 2, Springer Berlin, pp. 57–64.

Ghoniem, Ahmed, Farhadi, Farbod, Reihaneh, Mohammad, 2015. An accelerated branch-and-price algorithm for multiple-runway aircraft sequencing problems. European J. Oper. Res. 246 (1), 34–43.

Grangier, Philippe, Gendreau, Michel, Lehuédé, Fabien, Rousseau, Louis-Martin, 2016. An adaptive large neighborhood search for the two-echelon multiple-trip vehicle routing problem with satellite synchronization. European J. Oper. Res. 254 (1), 80–91.

Han, Brian Rongqing, Sun, Tianshu, Chu, Leon Yang, Wu, Lixia, 2022. COVID-19 and E-commerce operations: Evidence from Alibaba. Manuf. Serv. Oper. Manag. 24 (3), 1388–1405.

He, Pengfei, Li, Jing, 2019. The two-echelon multi-trip vehicle routing problem with dynamic satellites for crop harvesting and transportation. Appl. Soft Comput. 77, 387–398.

Irnich, Stefan, Desaulniers, Guy, 2005. Shortest path problems with resource constraints. In: Column Generation. Springer, pp. 33–65.

Irnich, Stefan, Desaulniers, Guy, Desrosiers, Jacques, Hadjar, Ahmed, 2010. Path-reduced costs for eliminating arcs in routing and scheduling. INFORMS J. Comput. 22 (2), 297–313.

Jepsen, Mads, Spoorendonk, Simon, Ropke, Stefan, 2013. A branch-and-cut algorithm for the symmetric two-echelon capacitated vehicle routing problem. Transp. Sci. 47 (1), 23–37.

Kancharla, Surendra Reddy, Ramadurai, Gitakrishnan, 2019. Multi-depot two-echelon fuel minimizing routing problem with heterogeneous fleets: Model and heuristic. Netw. Spat. Econ. 19 (3), 969–1005.

Kramer, Arthur, Lalla-Ruiz, Eduardo, Iori, Manuel, Voß, Stefan, 2019. Novel formulations and modeling enhancements for the dynamic berth allocation problem. European J. Oper. Res. 278 (1), 170–185.

Lagos, Felipe, Boland, Natashia, Savelsbergh, Martin, 2020. The continuous-time inventory-routing problem. Transp. Sci. 54 (2), 375–399.

Lehmann, Jonas, Winkenbach, Matthias, 2024. A matheuristic for the two-echelon multi-trip vehicle routing problem with mixed pickup and delivery demand and time windows. Transp. Res. C 160, 104522.
Li, Hongqi, Wang, Haotian, Chen, Jun, Bai, Ming, 2020. Two-echelon vehicle routing problem with time windows and mobile satellites. Transp. Res. B 138, 179–201.
Li, Hongqi, Wang, Haotian, Chen, Jun, Bai, Ming, 2021. Two-echelon vehicle routing problem with satellite bi-synchronization. European J. Oper. Res. 288 (3), 775–793.
Liu, Tian, Luo, Zhixing, Qin, Hu, Lim, Andrew, 2018. A branch-and-cut algorithm for the two-echelon capacitated vehicle routing problem with grouping constraints. European J. Oper. Res. 266 (2), 487–497.
Long, Jianyu, Sun, Zhenzhong, Pardalos, Panos M, Hong, Ying, Zhang, Shaohui, Li, Chuan, 2019. A hybrid multi-objective genetic local search algorithm for the prize-collecting vehicle routing problem. Inform. Sci. 478, 40–61.
Marques, Guillaume, Sadykov, Ruslan, Deschamps, Jean-Christophe, Dupas, Rémy, 2020. An improved branch-cut-and-price algorithm for the two-echelon capacitated vehicle routing problem. Comput. Oper. Res. 114, 104833.
Mhamedi, Tayeb, Andersson, Henrik, Cherkesly, Marilène, Desaulniers, Guy, 2022. A branch-price-and-cut algorithm for the two-echelon vehicle routing problem with time windows. Transp. Sci. 56 (1), 245–264.
Mohri, Seyed Sina, Mohammadi, Mehrdad, Van Woensel, Tom, 2024. Designing zero-emissions containerized last-mile delivery systems: A case study for melbourne. Transp. Res. C 159, 104492.
Moreno, Alfredo, Munari, Pedro, Alem, Douglas, 2024. Crew scheduling and routing problem in road restoration via branch-and-price algorithms. Transp. Sci..
Munari, Pedro, Moreno, Alfredo, De La Vega, Jonathan, Alem, Douglas, Gondzio, Jacek, Morabito, Reinaldo, 2019. The robust vehicle routing problem with time windows: Compact formulation and branch-price-and-cut method. Transp. Sci. 53 (4), 1043–1066.
Özener, Okan Örsan, Ergun, Özlem, Savelsbergh, Martin, 2013. Allocating cost of service to customers in inventory routing. Oper. Res. 61 (1), 112–125.
Perboli, Guido, Tadei, Roberto, 2010. New families of valid inequalities for the two-echelon vehicle routing problem. Electron. Notes Discrete Math. 36, 639–646.
Perboli, Guido, Tadei, Roberto, Vigo, Daniele, 2011. The two-echelon capacitated vehicle routing problem: Models and math-based heuristics. Transp. Sci. 45 (3), 364–380.
Sahraeian, Rashed, Esmaeili, Mehraneh, 2018. A multi-objective two-echelon capacitated vehicle routing problem for perishable products. J. Ind. Syst. Eng. 11 (2), 62–84.
Santos, Fernando Afonso, da Cunha, Alexandre Salles, Mateus, Geraldo Robson, 2013. Branch-and-price algorithms for the two-echelon capacitated vehicle routing problem. Optim. Lett. 7 (7), 1537–1547.
Santos, Fernando Afonso, Mateus, Geraldo Robson, da Cunha, Alexandre Salles, 2015. A branch-and-cut-and-price algorithm for the two-echelon capacitated vehicle routing problem. Transp. Sci. 49 (2), 355–368.
Savelsbergh, Martin, Van Woensel, Tom, 2016. 50Th anniversary invited article—city logistics: Challenges and opportunities. Transp. Sci. 50 (2), 579–590.
Seidel, Tobias, Wickerath, Jan, 2020. Rush hours and urbanization. Reg. Sci. Urban Econ. 85, 103580.
Senna, Fernando, Coelho, Leandro C, Morabito, Reinaldo, Munari, Pedro, 2024. An exact method for a last-mile delivery routing problem with multiple deliverymen. European J. Oper. Res. 317 (2), 550–562.
Sluijk, Natasja, Florio, Alexandre M., Kinable, Joris, Dellaert, Nico, Van Woensel, Tom, 2023. Two-echelon vehicle routing problems: A literature review. European J. Oper. Res. 304 (3), 865–886.
Soares, Ricardo, Marques, Alexandra, Amorim, Pedro, Parragh, Sophie N, 2024. Synchronisation in vehicle routing: classification schema, modelling framework and literature review. European J. Oper. Res. 313 (3), 817–840.
Solomon, Marius M., 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints. Oper. Res. 35 (2), 254–265.
Stenger, Andreas, Schneider, Michael, Goeke, Dominik, 2013. The prize-collecting vehicle routing problem with single and multiple depots and non-linear cost. EURO J. Transp. Logist. 2 (1–2), 57–87.
Trachanatzi, Dimitra, Rigakis, Manousos, Marinaki, Magdalene, Marinakis, Yannis, 2020. A firefly algorithm for the environmental prize-collecting vehicle routing problem. Swarm Evol. Comput. 57, 100712.
Vansteenwegen, Pieter, Souffriau, Wouter, Van Oudheusden, Dirk, 2011. The orienteering problem: A survey. European J. Oper. Res. 209 (1), 1–10.
Wang, Ziqi, Wen, Peihan, 2020. Optimization of a low-carbon two-echelon heterogeneous-fleet vehicle routing for cold chain logistics under mixed time window. Sustainability 12 (5), 1967.
Wang, Mengtong, Zhang, Canrong, Bell, Michael GH, Miao, Lixin, 2022. A branch-and-price algorithm for location-routing problems with pick-up stations in the last-mile distribution system. European J. Oper. Res. 303 (3), 1258–1276.
Wen, Min, Linde, Esben, Ropke, Stefan, Mirchandani, Pitu B., Larsen, Allan, 2016. An adaptive large neighborhood search heuristic for the electric vehicle scheduling problem. Comput. Oper. Res. 76, 73–83.