10-2021

Design of a two-echelon freight distribution system in last-mile logistics considering covering locations and occasional drivers

Vincent F. YU

Panca JODIAWAN

Ming-Lu HOU

Aldy GUNAWAN
Singapore Management University, aldygunawan@smu.edu.sg

Design of a two-echelon freight distribution system in last-mile logistics considering covering locations and occasional drivers

YU, Vincent F.[ab]; JODIAWAN, Panca [a*]; HOU, Ming-Lu[a]; GUNAWAN, Aldy[c]

a Department of Industrial Management, National Taiwan University of Science and Technology, Taipei 10607, Taiwan

b Center for Cyber-Physical System Innovation, National Taiwan University of Science and Technology, Taipei 10607, Taiwan

c School of Computing and Information Systems, Singapore Management University, Singapore

Published in Transportation Research Part E: Logistics and Transportation Review, 2021 October, 154, 102461. DOI: 10.1016/j.tre.2021.102461

Abstract:

This research addresses a new variant of the vehicle routing problem, called the two-echelon vehicle routing problem with time windows, covering options, and occasional drivers (2E-VRPTW-CO-OD). In this problem, two types of fleets are available to serve customers, city freighters and occasional drivers (ODs), while two delivery options are available to customers, home delivery and alternative delivery. For customers choosing the alternative delivery, their demands are delivered to one of the available covering locations for them to pick up. The objective of 2E-VRPTW-CO-OD is to minimize the total cost consisting of routing costs, connection costs, and compensations paid to ODs while satisfying all demands. We formulate a mixed integer linear programming model and propose an effective adaptive large neighborhood search (ALNS) for solving 2E-VRPTW-CO-OD. In addition, the proposed ALNS provides comparable results with those obtained by state-of-the-art algorithms for the two echelon vehicle routing problem, which is a special case of 2E-VRPTW-CO-OD. Lastly, the effects of occasional drivers and covering locations are presented.

Keywords: Two-echelon vehicle routing, City logistics, Covering location, Occasional driver

1. Introduction

The substantial growth of e-commerce has led to increasing freight distribution activities. Consequently, governments and logistics providers have put forth great efforts to reduce traffic congestions to improve the life quality of citizens. One possible solution is to limit the movement of large vehicles in the city and to utilize smaller and eco-friendly vehicles for serving customers. This strategy proposes a two-echelon distribution network in which freights originating from a depot are sent to intermediate facilities called satellites, and customers receive their orders from deliveries performed from those satellites (Hemmelmayr et al., 2012, Savelsbergh and Van Woensel, 2016).

Two echelon distribution networks have been widely implemented in various business areas, such as express delivery services, grocery and hypermarket-product distribution, spare-parts distribution, home delivery services, and newspaper and press distribution (Perboli et al., 2011). One of their recent applications is the utilization of cargo bikes in high-density city centers (Anderluh et al., 2017, Arnold et al., 2018, Schliwa et al., 2015).

Aside from the two-echelon distribution network, a decentralized facility and the concept of crowd-shipping are promising new strategies to address the negative impacts of growing freight volume. A decentralized facility requires customers' self-service; e.g.,

* Corresponding author. E-mail address: pancajodiawan@gmail.com (P. Jodiawan).

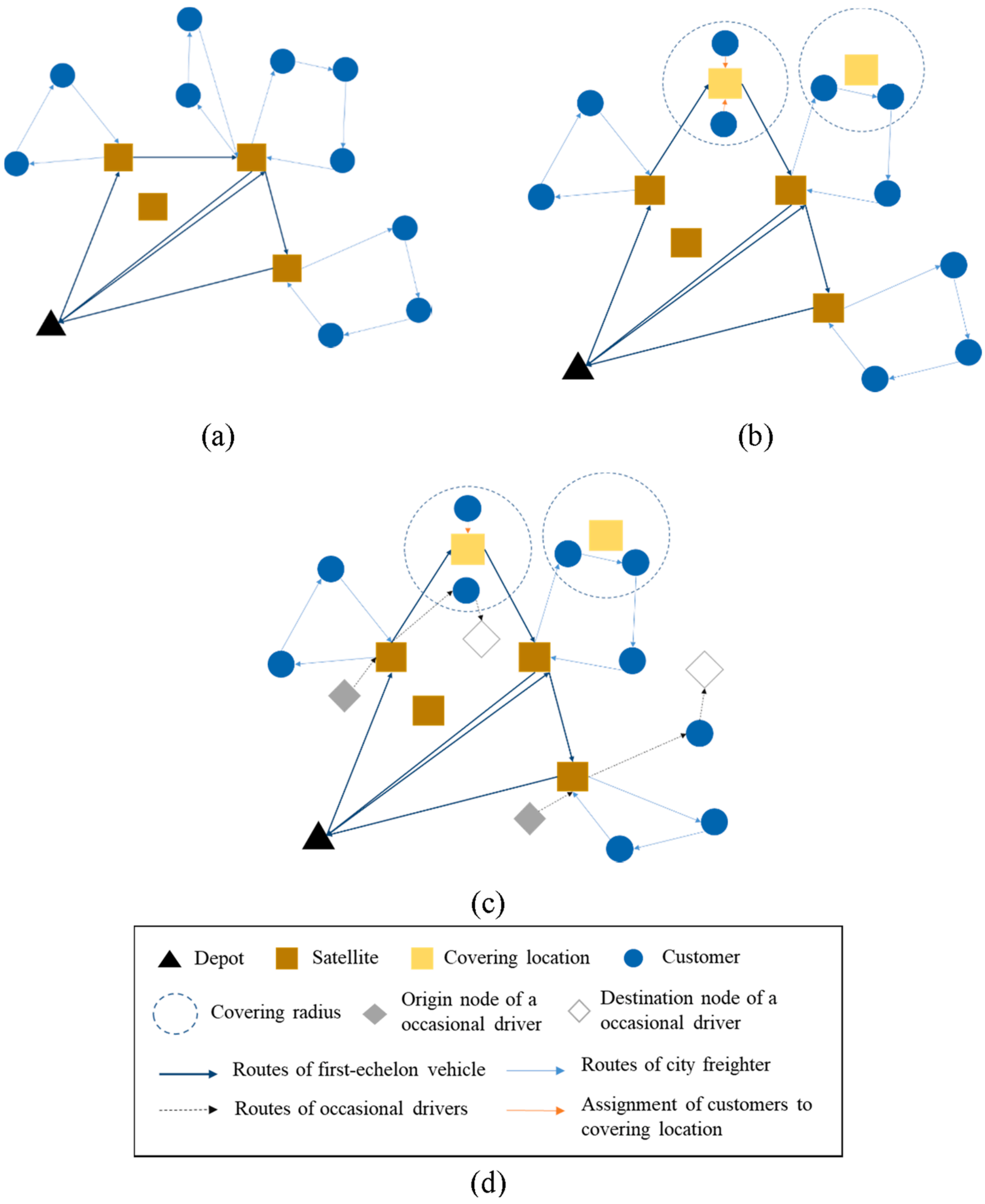

Fig. 1. Illustration of (a) a 2E-VRP solution, (b) a 2E-VRP-CO solution, (c) a 2E-VRPTW-CO-OD solution, and (d) the explanation of symbols.

customers pick up their packages at parcel lockers or stores (Boysen et al., 2020). Crowd-shipping allows logistics service providers to employ existing on-road crowds who are willing to detour to serve customers in exchange for monetary or other compensation (Rai et al., 2017; Sampaio et al., 2019). Recent research indicates that the utilization of decentralized facilities and crowd-shipping systems can be beneficial in terms of cost savings and the alleviation of negative environmental impacts (Archetti et al., 2016; Boysen et al., 2020; Deutsch and Golany, 2018; Macrina et al., 2017; Macrina et al., 2020; Simoni et al., 2019; Van Duin et al., 2020).

Despite the potential benefits of decentralized facilities and crowd-shipping systems, the literature on analyzing the impact of these strategies in a two-echelon distribution network is still scant. To the best of the authors' knowledge, Zhou et al. (2018) and Enthoven et al. (2020) are the only recent works investigating new variants of the Two-echelon Vehicle Routing Problem (2E-VRP) by allowing customers' demands to be delivered to parcel lockers. Thus, one of the aims of this research is to introduce a new variant of 2E-VRP, called the Two-Echelon Vehicle Routing Problem with Time Windows, Covering Options, and Occasional Drivers (2E-VRPTW-CO-OD), which simultaneously considers parcel lockers and crowd-shipping. We herein denote a parcel locker as a covering location in 2E-

VRPTW-CO-OD like in the Two-echelon Vehicle Routing Problem with Covering Option (2E-VRP-CO) in Enthoven et al. (2020). Fig. 1 provides an illustration of the differences between 2E-VRP, 2E-VRP-CO, and 2E-VRPTW-CO-OD solutions.

Meta-heuristics and heuristics have been developed and widely used to deal with various routing problems (Vidal et al., 2013). In particular, destroy-and-repair based heuristic variants are common to solve 2E-VRPs (Breunig et al., 2016; Enthoven et al., 2020; Grangier et al., 2016; Hemmelmayr et al., 2012). Among them, ALNS has been successfully employed to solve many hard combinatorial optimization problems (Ali et al., 2020; Gunawan et al., 2020; Li et al., 2020; Sun et al., 2020). Thus, this research develops an ALNS heuristic to solve 2E-VRPTW-CO-OD. In conclusion, the contributions of this research are listed as follows.

1. A new variant of 2E-VRP, the Two Echelon Vehicle Routing Problem with Time Windows, Covering Options and Occasional Drivers, is introduced.
2. A mixed integer linear programming (MILP) model of 2E-VRPTW-CO-OD is formulated.
3. An effective ALNS heuristic for solving 2E-VRPTW-CO-OD is developed.
4. Sensitivity analyses are conducted to provide insights into the implementation of 2E-VRPTW-CO-OD.

The rest of this paper runs as follows. Section 2 discusses relevant literature. Section 3 presents the formal description and the MILP model of 2E-VRPTW-CO-OD. Section 4 describes the proposed ALNS algorithm. Section 5 presents computational results. Finally, Section 6 concludes the research and offers several ideas for future works.

2. Literature review

In this section we review some publications on 2E-VRP and its closely related variant, location-routing problems (LRPs), since the problem addressed in this work is based on 2E-VRP. To show the current research progress of the integration of parcel lockers in delivery systems, we also present an overview of recent studies focusing on two different levels: strategic and operational. Lastly, we summarize the literature on crowd-shipping system developments.

2.1. Two-echelon vehicle routing problems

Our research considers a two-echelon routing problem first introduced in Perboli et al. (2011) as a new family of VRP. Their study addresses the basic version of 2E-VRP, which is the two-echelon capacitated VRP. A compact formulation with some valid inequalities and two math-heuristics are proposed as their problem solving approaches. Since then, several developments incorporating real-world characteristics and recent trends have been made. The property of time windows is one of the important real-world aspects for the extension of 2E-VRP (Dellaert et al., 2019; Grangier et al., 2016; Li et al., 2016). Grangier et al. (2016) and Anderluh et al. (2017) allowed second-echelon vehicles to perform multiple trips. A grouping constraint is another aspect that categorizes each customer into a particular group, and customers from the same group should be served from the same satellite (Liu et al., 2018). Soysal et al. (2015) and Wang et al. (2017) addressed an extension of 2E-VRP by considering different traffic densities over time, leading to the time-dependent 2E-VRP. Several latest extensions of 2E-VRP focus on the utilization of newly developed technologies, like the unmanned-aerial-vehicle (UAV) (Li et al., 2020) and small autonomous vehicle (SAV) (Yu et al., 2020a).

2.2. Location routing problems

LRPs have two main decisions: locating depot(s) and determining vehicle routes originating from the opened depot(s) to serve customers. Salhi and Rand (1989) and Salhi and Nagy (1999) showed that dealing with LRP brings forth benefits compared to independently dealing with the aforementioned two decisions. Since then, more research works have dealt with this problem, and due to its property of NP-hardness, different heuristics have been proposed. Schneider and Drexl (2017) and Mara et al. (2021) comprehensively reviewed recently developed heuristics to solve LRPs.

Prodhon and Prins (2014) and Drexl and Schneider (2015) presented timely reviews on LRP variants, of which 2E-LRP is one of them. They addressed 2E-LRP, because the opened depots are supplied by main plant(s), constituting a first-echelon network. Two ways are commonly employed to solve 2E-LRP: explicitly dealing with the routing decisions and approximating the routing decisions. Exact approaches and various heuristics are developed when the routing decisions are explicitly handled (Boccia et al., 2010; Gonzalez-Feliu et al., 2008; Hemmelmayr et al., 2012; Perboli and Tadei, 2010; Perboli et al., 2011). Continuum/continuous approximation methods commonly deal with the latter (Smilowitz and Daganzo, 2007; Winkenbach et al., 2016).

Recent innovative models dealing with extensions of 2E-LRP have also been proposed. Snoeck and Winkenbach (2020) addressed the design of a large-scale two-echelon distribution network by considering three types of flexibility under demand uncertainty. Janjevic et al. (2019) dealt with another extension of 2E-LRP by taking into account places serving as customers' secondary pick-up points called collection-and-delivery points. Zhao et al. (2018) proposed an extended model of 2E-LRP in which two or more depots exist and the second-echelon vehicles are heterogeneous. Yu et al. (2020b) presented another 2E-LRP variant in which no fixed cost for satellites is considered and deliveries to customers are outsourced. These recent 2E-LRP works have aimed to improve the standard 2E-LRP to capture real-world characteristics or to further enhance the benefits of its implementation.

2.3. Integration of parcel lockers in delivery systems

Research considering parcel lockers focuses on two different levels: strategic and operational (Rohmer and Gendron, 2020). The number of locker stations, the configuration of the locker stations, and the locations of the locker stations are decisions made at the strategic level. Deutsch and Golany (2018) pioneered the first comprehensive work at the strategic level by tackling an optimization problem for determining the optimal number, location, and size of locker facilities in a network of parcel lockers with an objective of maximizing profits. Guerrero-Lorente et al. (2020) addressed another strategic-level issue concerning the integration of parcel lockers as a new channel in an omnichannel distribution network for a company in Madrid. The main purpose of the proposed model is to find the location of the considered facilities in the network with respect to cost minimization.

At the operational level (e.g., routing decisions), Zhou et al. (2018) investigated a city logistics problem inspired from the last-mile distribution of e-commerce, called the multi-depot two-echelon vehicle routing problem with delivery options. The main feature of this problem is that customers are allowed to pick up their packages at intermediate pickup facilities which have a similar function as parcel lockers. Orenstein et al. (2019) developed a mathematical model for delivering small parcels to a set of available capacitated service points with an objective of minimizing total driving time, the penalty for not delivering a particular parcel, and the fixed cost of a vehicle. Jiang et al. (2019) dealt with the traveling salesman problem with time windows by considering available shared delivery facilities (SDFs) with an objective of finding the minimum cost of delivery, pick-up, and opening SDF(s). While the aforementioned works focused on forward logistics, Guerrero-Lorente et al. (2017) built a MILP model for an omnichannel distribution network that makes use of parcel lockers as collection points of returned packages.

2.4. Development of crowd-shipping systems

The concept of crowd-shipping has recently been considered by logistics providers due to potential benefits offered by its implementation (Macrina et al., 2020; Simoni et al., 2019). There are several aspects that need to be analyzed prior to the implementation, leading to a new opportunity in the field of operations research. Archetti et al. (2016) were the first to propose an integration of crowds - called occasional drivers (ODs) - into the classical capacitated vehicle routing problem. The problem is called the vehicle routing problem with occasional drivers (VRPOD). An integer programming model and a multi-start heuristic that combines variable neighborhood search and tabu search have been proposed to solve VRPOD. Aside from showing the effectiveness and efficiency of the proposed method at solving the newly generated instances of VRPOD, experiments have been conducted to show the impact of the number and flexibility of ODs and the compensation scheme employed toward the cost savings that result from utilizing ODs.

Macrina et al. (2017) extended the VRPOD by considering three new aspects. First, both customers and ODs have preferred time windows. Second, multiple deliveries are allowed for each OD. Lastly, a split-delivery policy is considered for ODs. All the aforementioned considerations are included into VRPOD to achieve a more realistic condition and to enhance the benefits of employing ODs.

Kafle et al. (2017) extended VRPOD by considering the existence of pick-up customers, a bidding mechanism to select ODs, the existence of relay points, and soft time windows. In Archetti et al. (2016), the available ODs are only able to pick up their assigned demands at the depot. Later on, an idea for adding alternative pickup points - also known as transshipment points - was proposed to reduce the distance travelled by ODs, providing greater benefits to companies that utilize crowd-shipping systems (Huang and Ardiansyah, 2019; Macrina et al., 2020; Sampaio et al., 2020). Table A15 lists the characteristics dealt with by previous works focusing on 2E-VRP, parcel locker deliveries, and crowd-shipping delivery problem variants.

3. Problem description and formulation

3.1. Description

The problem description of 2E-VRPTW-CO-OD is provided as follows. Each customer existing in the second-echelon network can pick up his/her own package at a covering location or is served at home within his/her preferred time window. The concept of covering location proposed by Enthoven et al. (2020) is that customers' demands can be assigned to a covering location as long as those customers are still located within a specified radius from the covering location. Two types of vehicles collecting customers' packages at satellites - a company's second-echelon vehicles (city freighters) and ODs - are available to serve home-delivery customers. First-echelon trucks originating from the depot need to deliver demands to satellites before city freighters, and ODs can serve home-delivery customers. The covering locations are also served by first-echelon trucks. In this problem we seek to minimize the total operational cost consisting of total traveling cost of first- and second-echelon vehicles, total compensation paid to the employed ODs, and the connection cost incurred by assigning customers to covering locations with respect to serving all customers. We are now ready to formally define 2E-VRPTW-CO-OD.

3.2. Formulation

We first define 2E-VRPTW-CO-OD on a directed graph $G = (V, A)$. The set of vertices V consists of the origin depot s, the destination depot t, the set of covering locations V^L, the set of satellite locations V^S, the set of customer locations V^C, and the set of origin and destination locations of available ODs, V^O and V^D. Two arc sets, $A^1 := \{(i,j)|$ and $A^2 := A \backslash A^1$, are defined as a collection of arcs in first

Table 1
Decision variables.

Notation	*Description*
x_{ijk}	Binary decision variables representing the selection of arc (i,j) to be traversed by first-echelon truck k, where $i,j \in s \cup t \cup V^L \cup V^S, k \in V^T$; 1 if first-echelon truck k traverses through arc (i,j) and 0 otherwise
y_{ijk}	Binary decision variables indicating whether a city freighter originating from satellite k traverses arc (i,j), where $i,j \in V^S \cup V^C, k \in V^S$; 1 if a city freighter originating from satellite k travels through arc (i,j) and 0 otherwise
v_k	Binary decision variables indicating that intermediate location k is utilized, where $k \in V^S \cup V^L$; 1 if intermediate location k is utilized and 0 otherwise
z_{ki}	Binary decision variables indicating that demand of customer i is assigned to intermediate location k, where $i \in V^C, k \in V^S \cup V^L$; 1 if customer i is served by intermediate location k and 0 otherwise
w^1_{ijk}	Amounts of demands being transported by first-echelon truck k on arc (i,j), where $i,j \in s \cup t \cup V^S \cup V^L, k \in V^T$
w^2_{ijk}	Amounts of demands being transported by a city freighter originating from satellite k along arc (i,j), where $i,j \in V^S \cup V^C, k \in V^S$
ω^1_{ik}	Arrival time of first-echelon truck k at node i, where $i \in s \cup t \cup V^S \cup V^L, k \in V^T$
ω^2_i	Arrival time of a city freighter at node i, where $i \in V^S \cup V^C$
γ^k_{pi}	Binary decision variables indicating whether OD k serves customer i by picking up the demand at satellite p, where $k \in V^{OD}, p \in V^S, i \in V^C$; 1 if OD k serves customer i by picking up the demand at satellite p and 0 otherwise.
g_{pk}	Arrival time of OD k at satellite p, where $k \in V^{OD}, p \in V^S$
ω^{OD}_i	Arrival time of the OD who serves customer i, where $i \in V^C$

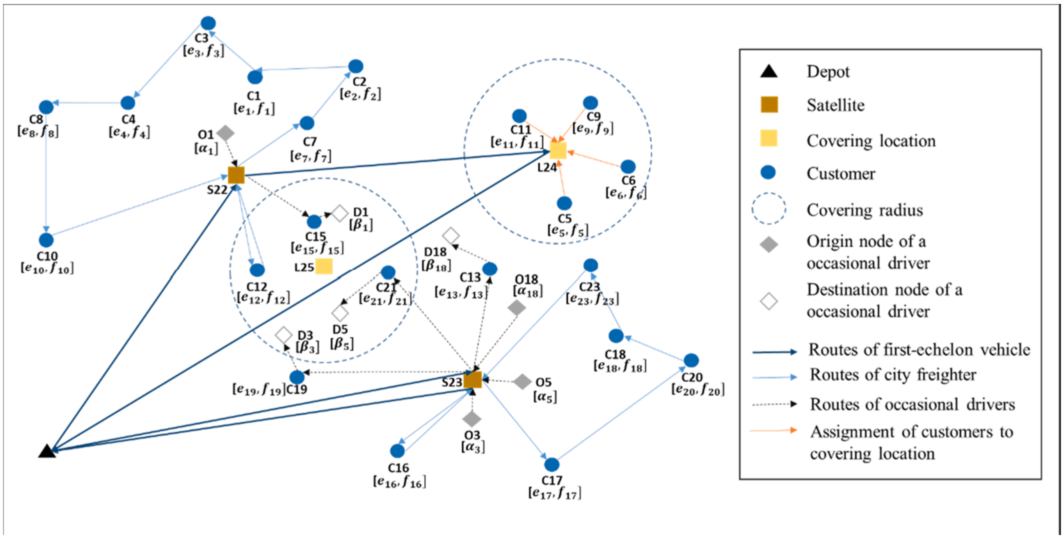

Fig. 2. An illustrative example of the 2E-VRPTW-CO-OD solution.

and second echelons, respectively. For all arcs $(i,j) \in A := \{(i,j)|i,j \in V, i \neq j\}$, the distance (unit of distance) is denoted by d_{ij}, for all arcs (i, j). In addition, we define c_{ij} as the travel cost (unit of cost) for arc (i, j) and t_{ij} as the traveling time (unit of time) for arc (i, j), where $d_{ij} = c_{ij} = t_{ij}$.

A set of homogeneous first-echelon trucks V^T with capacity Q^1 (unit of demand) is located at depot s, and m^2 homogeneous city freighters with capacity $Q^2 < Q^1$ (unit of demand) are available. The city freighters can be located at any satellite $S \in V^S$. A satellite can only have at most $\overline{m}^2$city freighters. In the second-echelon network, a set of ODs V^{OD} exists, for which each can serve at most one customer.

At the second echelon, customer $i \in V^C$ has a service time, ST^2, and a time window, $[e_i, f_i]$, with each representing the earliest and the latest time for starting the service. Moreover, customer i has demand $q_i$and can be served in two ways: either by a direct or by an alternative delivery option. If a customer is assigned to the direct delivery option, then demand needs to be sent to the customer's location by either a city freighter or an OD; otherwise, the demand is sent to a covering location $j \in V^L$ by a first-echelon truck provided that the distance between covering location j and customer i is still within the covering radius *rad* (unit of distance). A connection cost ς_{ij} (unit of cost) arises if customer i is assigned to covering location j. If a customer is served by either a city freighter or an OD, then the time window of the customer must be respected; otherwise, the delivery can be performed any time during the planning period.

A city freighter departs from a satellite, serves a subset of customers, and returns to the same satellite. An OD can visit any satellite $S \in V^S$ to pick up the demand of a customer assigned to the driver. OD i, $i \in V^{OD}$, has an origin location ν^O_i that is stored in V^O and a destination location ν^D_i that is stored in V^D. The origin and destination locations of an OD are obtained from the information provided by the OD when entering the system. In addition, OD i has the earliest time to set off from the origin location, α_i, and the latest time to arrive at the destination location, β_i. The compensation factor ρ (unit of cost/unit of distance) is used for calculating the cost of ODs. Here, ρ is multiplied with the extra distance travelled by an OD, which becomes the cost of employing the related OD.

A first-echelon truck departs from the origin depot and ends the journey at the destination depot, travelling to serve either satellite (s) and/or covering location(s). The satellites and covering locations are later mentioned as intermediate locations for the sake of convenience. Split delivery is allowed for intermediate locations, but we do not permit such delivery in the second echelon. In addition, there is a particular amount of service time, ST^1, for each intermediate location. The second-echelon vehicle, either a city freighter or an OD, can depart from a satellite only after first-echelon truck(s) have served that particular satellite. In addition, M represents a big number utilized during the formulation in Section 3. Table 1 lists the decision variables. Fig. 2 illustrates an example solution of 2E-VRPTW-CO-OD. A complete explanation of Fig. 2 is provided in the appendix.

The MILP model of 2E-VRPTW-CO-OD is as follows.

The objective of this problem is to minimize the sum of the total travel costs, the connection costs, and the total compensation paid to all ODs.

$$Min \qquad Z = Z_c + Z_d + \sum_{k \in V^{OD}} Z_k$$

$$Z_c = \sum_{k \in V^C} \sum_{j \in V^L} \varsigma_{kj} z_{kj}$$

$$Z_d = \sum_{i,j \in s \cup t \cup V^S \cup V^L, k \in V^T} c_{ij} x_{ijk} + \sum_{k \in V^S} \sum_{i,j \in V^S \cup V^C} c_{ij} y_{ijk}$$

$$Z_k = \rho \Big(\sum_{p \in V^S} \sum_{i \in V^C} \left(d_{v_k^O p} + d_{pi} + d_{i v_k^D} - d_{v_k^O v_k^D} \right) \gamma_{pi}^k \Big)$$

Subject to

$$\sum_{i \in s} \sum_{j \in V^L \cup V^S} x_{ijk} \leqslant 1 \quad \forall k \in V^T \tag{1}$$

$$\sum_{k \in V^T} \sum_{i \in t} \sum_{j \in V^L \cup V^S} x_{ijk} = 0 \tag{2}$$

$$\sum_{\substack{i \in s \ \cup \ V^S \cup V^L \\ i \neq j}} x_{ijk} = \sum_{\substack{i \in V^S \cup \ V^L \cup t \\ i \neq j}} x_{jik} \quad \forall j \in V^S \cup V^L, k \in V^T \tag{3}$$

$$\sum_{k \in V^T} \sum_{\substack{i \in V^S \cup \ V^L \\ i \neq j}} x_{jik} \leqslant M \times \left(\sum_{i \in V^C} z_{ji} + \sum_{k \in V^{OD}} \sum_{i \in V^C} \gamma_{ji}^k \right) \quad \forall j \in V^S \tag{4}$$

$$\sum_{k \in V^T} \sum_{\substack{i \in V^S \cup \ V^L \\ i \neq j}} x_{jik} \leqslant M \times \sum_{i \in V^C} z_{ji} \quad \forall j \in V^L \tag{5}$$

$$\sum_{k \in V^S} \sum_{i \in V^C} y_{kik} \leqslant m^2 \tag{6}$$

$$\sum_{\substack{i \in V^S \cup V^C \\ i \neq j}} y_{jik} = \sum_{\substack{i \in V^S \cup V^C \\ i \neq j}} y_{ijk} = z_{kj} \quad \forall j \in V^C, k \in V^S \tag{7}$$

$$\sum_{i \in V^C} y_{kik} = \sum_{i \in V^C} y_{ikk} \leqslant v_j \times \overline{m}^2 \quad \forall k \in V^S \tag{8}$$

$$v_j \leqslant \sum_{k \in V^T} \sum_{\substack{i \in t \cup V^S \cup V^L \\ i \neq j}} x_{jik} \quad \forall j \in V^S \cup V^L \tag{9}$$

$$\sum_{j \in V^C} y_{ijk} = 0 \quad \forall i, k \in V^S, i \neq k \tag{10}$$

$$\sum_{i \in V^S \cup V^L} z_{ji} + \sum_{k \in V^{OD}} \sum_{i \in V^S} \gamma_{ij}^k = 1 \quad \forall j \in V^C \tag{11}$$

$$d_{ki} z_{ki} \leqslant rad \times v_k \ \forall i \in V^C, k \in V^L \tag{12}$$

$$\sum_{i \in V^C} z_{li} q_i = \sum_{k \in V^T} \sum_{\substack{i \in s \cup V^S \cup V^L \\ i \neq l}} w_{ilk}^1 - \sum_{k \in V^T} \sum_{\substack{i \in V^S \cup V^L \cup t \\ i \neq l}} w_{lik}^1 \quad \forall l \in V^L \tag{13}$$

$$\sum_{i \in V^C} z_{li} q_i + \sum_{m \in V^{OD}, j \in V^C} \gamma_{lj}^m q_j = \sum_{k \in V^T} \sum_{i \in s \cup V^S \cup V^L i \neq l} w_{ilk}^1 - \sum_{k \in V^T} \sum_{i \in V^S \cup V^L \cup t i \neq l} w_{lik}^1 \quad \forall l \in V^S \tag{14}$$

$$\sum_{i \in V^C} q_i = \sum_{k \in V^T} \sum_{i \in s} \sum_{j \in V^S \cup V^L} w_{ijk}^1 \tag{15}$$

$$0 \leqslant w_{ijk}^1 \leqslant Q^1 x_{ijk} \quad \forall i, j \in s \cup t \cup V^S \cup V^L, k \in V^T \tag{16}$$

$$\sum_{\substack{i \in V^C \cup V^S \\ i \neq j}} w_{ijk}^2 - \sum_{\substack{i \in V^C \cup V^S \\ i \neq j}} w_{jik}^2 = z_{kj} q_j \quad \forall j \in V^C, k \in V^S \tag{17}$$

$$\sum_{i \in V^C} w_{ikk}^2 - \sum_{i \in V^C} w_{kik}^2 = -\sum_{i \in V^C} z_{ki} q_i \quad \forall k \in V^S \tag{18}$$

$$0 \leqslant w_{ijk}^2 \leqslant (Q^2 - q_i) y_{ijk} \quad \forall i, j \in V^C, k \in V^S \tag{19}$$

$$0 \leqslant w_{ijk}^2 \leqslant Q^2 y_{ijk} \forall i \in V^S, \quad \forall j \in V^C, k \in V^S \tag{20}$$

$$\sum_{k \in V^T} \sum_{i \in V^L \cup V^S} \sum_{j \in t} w_{ijk}^1 + \sum_{k \in V^S} \sum_{i \in V^C} w_{ikk}^2 = 0 \tag{21}$$

$$\omega_{jk}^1 \geqslant \omega_{ik}^1 + ST^1 + t_{ij} - M\left(1 - x_{ijk}\right) \quad \forall i \in s \cup V^S \cup V^L, j \in t \cup V^S \cup V^L, i \neq j, k \in V^T \tag{22}$$

$$\omega_j^2 \geqslant \omega_{kp}^1 + ST^1 + t_{kj} - M^* \left(2 - y_{kjk} - \sum_{\substack{h \in s \cup V^S \cup V^L \\ h \neq k}} x_{hkp} \right) \quad \forall j \in V^C, k \in V^S, p \in V^T \tag{23}$$

$$\omega_j^2 \geqslant \omega_i^2 + ST^2 + t_{ij} - M^* \left(1 - y_{ijk}\right) \quad \forall i, j \in V^C, k \in V^S, i \neq j \tag{24}$$

$$e_i - M(1 - z_{ki}) \leqslant \omega_i^2 \leqslant f_i + M(1 - z_{ki}) \quad \forall i \in V^C, k \in V^S \tag{25}$$

$$\sum_{p \in V^S} \sum_{i \in V^C} \gamma_{pi}^k \leqslant 1 \quad \forall k \in V^{OD} \tag{26}$$

$$g_p^k \geqslant \alpha_{u_k} + t_{u_k p} - M\left(1 - \gamma_{pi}^k\right) \quad \forall p \in V^S, k \in V^{OD}, i \in V^C \tag{27}$$

$$g_p^k \geqslant \omega_{pj}^1 + ST^1 - M\left(1 - \gamma_{pi}^k\right) \quad \forall p \in V^S, k \in V^{OD}, i \in V^C, j \in V^T \tag{28}$$

$$\omega_i^{OD} \geqslant g_p^k + t_{pi} - M\left(1 - \gamma_{pi}^k\right) \quad \forall i \in V^C, p \in V^S, k \in V^{OD} \tag{29}$$

$$e_i - M\left(1 - \gamma_{pi}^k\right) \leqslant \omega_i^{OD} \leqslant f_i + M\left(1 - \gamma_{pi}^k\right) \quad \forall i \in V^C, p \in V^S, k \in V^{OD} \tag{30}$$

$$\omega_i^{OD} + ST^2 + t_{iv_k} \leqslant \beta_{v_k} + M\left(1 - \gamma_{pi}^k\right) \quad \forall i \in V^C, p \in V^S, k \in V^{OD} \tag{31}$$

$$x_{ijk} \in \{0, 1\} \quad \forall i \in s \cup V^S \cup V^L, j \in t \cup V^S \cup V^L, k \in V^T \tag{32}$$

$$y_{ijk} \in \{0,1\} \quad \forall i,j \in V^S \cup V^C, k \in V^S \tag{33}$$

$$v_k \in \{0,1\} \quad \forall k \in V^S \cup V^L \tag{34}$$

$$z_{ki} \in \{0,1\} \quad \forall k \in V^S \cup V^L, \forall i \in V^C \tag{35}$$

$$w^1_{ijk} \geqslant 0 \quad \forall i \in s \cup V^S \cup V^L, j \in t \cup V^S \cup V^L, k \in V^T \tag{36}$$

$$w^2_{ijk} \geqslant 0 \quad \forall i,j \in V^S \cup V^C, k \in V^S \tag{37}$$

$$\omega^1_{ik} \geqslant 0 \quad \forall i \in s \cup t \cup V^S \cup V^L, \forall k \in V^T \tag{38}$$

$$\omega^2_i, \omega^{OD}_i \geqslant 0 \quad \forall i \in V^C \tag{39}$$

$$\gamma^k_{pi} \in \{0,1\} \quad \forall i \in V^C, \forall p \in V^S, \forall k \in V^{OD} \tag{40}$$

$$g_{pk} \geqslant 0 \quad \forall p \in V^S, \forall k \in V^{OD} \tag{41}$$

Constraints (1) guarantee that each first-echelon truck can only be employed at most once. Constraint (2) ensures that no first-echelon trucks can depart from the destination depot. Constraints (3) define the flow conservation for each first-echelon truck. Constraints (4) and (5) guarantee that an intermediate location is visited by first-echelon truck(s) if there are customer(s) assigned to the intermediate location. Constraint (6) limits the number of city freighters. Constraints (7) ensure that city freighters originating from a particular satellite serves customers assigned to the satellite. Constraints (8) to (9) limit the number of city freighters that can be assigned to a satellite. Constraints (10) ensure that no city freighter located at a satellite departs from another satellite. Constraints (11) guarantee that each customer is served by a city freighter, a covering location, or an OD. Constraints (12) state that only customers within the covering range can be assigned to a covering location.

Constraints (13) to (15) track the amount of demands carried by first-echelon trucks over all travelled arcs. Constraints (16) limit the total demands that can be carried by a first-echelon truck over arcs. Constraints (17) and (18) supervise the amount of demands carried by city freighters over all travelled arcs. Constraints (19) and (20) limit the maximum amount of demand that can be carried by a city freighter over arcs. Constraint (21) ensures that all vehicles, both first-echelon trucks and city freighters, should be empty upon their return.

Constraints (22) and (24) track the arrival time of first-echelon trucks at intermediate locations and city freighters at customers. Constraints (23) ensure that the arrival time of a city freighter at a customer location is greater than the time of a first-echelon truck departing from the satellite where the customer's demand is delivered. In other words, constraints (23) act as synchronization constraints between first-echelon trucks and city freighters. Constraints (25) guarantee that each customer is served within his/her time window.

Constraints (26) impose that each OD can serve at most one customer and pick-up demand of the customer at one satellite. Constraints (27) to (28) track the arrival time of an OD at a satellite and ensure the synchronization between first-echelon trucks and ODs. Constraints (29) define the arrival time of ODs at customer locations. Constraints (30) and (31) guarantees the time windows feasibility of both customers visited by ODs and the ODs themselves. Constraints (32) to (41) define the natural range of all decision variables.

4. Proposed ALNS algorithm for 2E-VRPTW-CO-OD

The proposed ALNS is inspired by several previous works (Breunig et al., 2016; Enthoven et al., 2020; Hemmelmayr et al., 2012) with several notable extensions. The previous algorithms proposed to solve the 2E-VRP (Breunig et al., 2016; Hemmelmayr et al., 2012) and the 2E-VRP-CO (Enthoven et al., 2020) deals with customers without time windows. Thus, the first feature is an efficient time windows feasibility checking procedure developed to handle the time windows and synchronization constraints in this problem. The heart of ALNS is the iterative implementation of destroy and repair operators. A huge solution space is considered when a solution is being repaired by a repair operator. Therefore, an efficient feasibility evaluation mainly aims to reduce the computational burden. Second, the aforementioned studies (Breunig et al., 2016; Enthoven et al., 2020; Hemmelmayr et al., 2012) did not consider the existence of ODs. Therefore, we propose a tailored local search procedure so that it can handle the presence of both covering locations and ODs.

The pseudocode of our ALNS is described in Algorithm 1. We first generate a feasible initial solution as an input for ALNS and set the initial solution as current solution, S^c, and best solution, S^*. We then initialize the selection probability of each destroy and repair operator. Here, π^S_R, π^L_R, and π_i represent the selection probability of small destroy operators, large destroy operators, and repair operators, respectively. Lines 5–21 explain the proposed ALNS. In particular, line 5 explains the termination criterion. ALNS terminates when it reaches one of the termination conditions - either a particular time limit t_{max} or a maximum iteration $iteration^{max}$.

The first step of the proposed ALNS is to remove several customers and/or reconfigure the first-echelon network from S^c and to store the resulting solution in S^w. For the removal phase, there are two types of destroy operators provided in ALNS: large and small destroy operators (Section 4.4). If the current iteration is smaller than w_{grace}, then an operator from a set of small destroy operators, O_R^S, is selected; otherwise, an operator from a set of large destroy operators, O_R^L, is chosen.

The ALNS algorithm then selects an operator from a set of repair operators (Section 4.5), O_I, to re-insert the removed customers to S^w. It then executes a local search procedure focusing on the second-echelon network structure; i.e., *LocalSearch*$2E()$. The algorithm next applies a large repair operator to re-create the first-echelon routes and implements a local search procedure focusing on the first-echelon network structure; i.e., *LocalSearch*$1E()$. We adopt the better solution acceptance, as described in Lines 13–16. Lines 17–18 state that the algorithm needs to update the selection probability whenever an η_s iteration has been performed. Here, σ_1 and σ_2 are respectively scores added to operators that can result in a new best solution and a better solution, while β is the reaction factor required for calculating operators' accumulated scores. One can refer to Hemmelmayr et al. (2012) and Enthoven et al. (2020) for a complete explanation of updating operators' selection probabilities. Lastly, we employ a restart mechanism; i.e., S^* replaces S^c, and every $\eta_{restart}$ iterations are performed by ALNS.

Algorithm 1: Overall structure of the ALNS heuristic

Input : $O_R^L, O_R^S, O_I, \omega_{grace}, \eta_s, t_{max}$
Output : S^*

1 $S^c \leftarrow InitialSolution$
2 $S^* \leftarrow S^w \leftarrow S^c$
3 $\pi_R^L, \pi_R^S, \pi_i \leftarrow InitializeProbabilities(\pi_R^L, \pi_R^S, \pi_i)$
4 $i \leftarrow 1, i_{restart} \leftarrow 1, i_{local} \leftarrow 1$
5 **while** $t < t_{max}$ ***and*** $i_{ALNS} < iteration^{max}$ **do**
6   *if* $i_{local} < \omega_{grace}$ **then**
7         $S^w \leftarrow Destroy(S^c, O_R^S, \pi_R^S)$
8   **else**
9         $S^w \leftarrow Destroy(S^c, O_R^L, \pi_R^L)$
10         $i_{local} \leftarrow 0$
11     $S^w \leftarrow LocalSearch2E(SmallRepair(S^w, \pi_i))$
12     $S^w \leftarrow LocalSearch1E(LargeRepair(S^w))$
13   **if** $f(S^w) < f(S^c)$ *or* $i_{local} \equiv 0$ **then**
14         $S^c \leftarrow S^w$
15   **if** $f(S^c) < f(S^*)$**then**
16             $S^* \leftarrow S^c$
17   **if** $(modulo(i, \eta_s) \equiv 0)$ **then**
18         $\pi_R^S, \pi_i \leftarrow UpdateOperatorProbabilities(\pi_R^S, \pi_i)$
19   if $(modulo(i_{restart}, \eta_{restart}) \equiv 0)$ **then**
20         $S^c \leftarrow S^*, i_{restart} = 0$
21     $i_{restart} \leftarrow i_{restart} + 1, i_{local} \leftarrow i_{local} + 1,\ i_{ALNS} \leftarrow i_{ALNS} + 1$
22 **return** S^*

4.1. *Solution representation*

The solution representation consists of four components. The first one is a two-dimensional array that keeps the routes of the first-echelon trucks. The second and the third ones are respectively a two-dimensional array that stores the routes of the city freighters and a two-dimensional array that keeps the routes of ODs. The last component is a one-dimensional array with length of $|V^C|$ consisting of the customer-covering location assignments.

Let $F = \{F(0), F(1), \ldots, F(|F|)\}$ be a route of a first-echelon truck. $F(0)$ and $F(|F|)$ both denote the depot node, while the remaining nodes belong to either V^S or V^L. $\Omega = \{\Omega(0), \Omega(1), \ldots, \Omega(|\Omega|)\}$ represents a route of a city freighter. $\Omega(0)$ and $\Omega(|\Omega|)$ both denote a satellite node, where $\Omega(0) = \Omega(|\Omega|)$, while the remaining nodes belong to V^C. Θ is a route of an OD consisting of the origin node of the OD, the satellite selected for the OD to pick up the assigned demand, a customer node, and the destination node of the OD. Let Λ be the customer-covering location assignments. $\Lambda(i) = 0$ if customer i is served by either a city freighter or an OD, while $\Lambda(i) = \xi$, where $\xi \in V^L$, states that customer i is served by covering location ξ.

A set of auxiliary information is also stored so that we can implement an efficient time windows feasibility checking procedure described in Section 4.2. For a given F, two types of information are saved: extended earliest time, $\widetilde{a}_{F(.)}$, and extended latest time, $\widetilde{z}_{F(.)}$. Both of these types of information are inspired from Nagata et al. (2010). Moreover, $l^a_{F(i)}$ denotes *the latest allowable arrival time for a* *first-echelon truck at node* $F(i)$, and $e^a_{F(i)}$ represents *the earliest allowable time for a first-echelon truck to start the service at node* $F(i)$. Equations (4.1) to (4.6) show the calculation for obtaining $\widetilde{a}_{F(.)}$ and $\widetilde{z}_{F(.)}$.

$$\widetilde{a}_{F(0)} = e_0 \tag{4.1}$$

$$\widetilde{a}'_{F(i)} = \widetilde{a}_{F(i-1)} + ST^1 + d_{F(i-1)F(i)}(i = 1, \dots, |F|) \tag{4.2}$$

$$\widetilde{a}_{F(i)} = max\left(\widetilde{a}'_{F(i)}, \ \ e^a_{F(i)}\right) if\ \widetilde{a}_{F(i)} \le l^a_{F(i)}(i = 1, \dots, |F|) \tag{4.3}$$

$$\widetilde{z}_{F(|F|)} = l_0 \tag{4.4}$$

$$\widetilde{z}'_{F(i)} = \widetilde{z}_{F(i+1)} - ST^1 - d_{F(i)F(i+1)}(i = 0, \dots, |F| - 1) \tag{4.5}$$

$$\widetilde{z}_{F(i)} = min\left(\widetilde{z}'_{F(i)}, l^a_{F(i)}\right) if\ \widetilde{z}_{F(i)} \ge e^a_{F(i)}(i = 0, \dots, |F| - 1) \tag{4.6}$$

We also store two similar types of information for a given Ω: $\widetilde{a}_{\Omega(.)}$ and $\widetilde{z}_{\Omega(.)}$. The calculations of $\widetilde{a}_{\Omega(.)}$ and $\widetilde{z}_{\Omega(.)}$ are similar to $\widetilde{a}_{F(.)}$ and $\widetilde{z}_{F(.)}$ with several differences. First, the value of $\widetilde{a}_{\Omega(0)}$ depends on which satellite becomes the origin of the city freighter. Here, $\widetilde{a}_{\Omega(0)} = e^D_{\Omega(0)}$, where $e^D_{\Omega(0)}$ denotes *the earliest allowable departure time for a city freighter originating from satellite* $\Omega(0)$. Second,$\widetilde{z}_{\Omega(|\Omega|)} = l_{\Omega(|\Omega|)}$, where $l_{\Omega(|\Omega|)}$ represents *the latest allowable arrival time for a city freighter to go back to satellite* $\Omega(|\Omega|)$. Third, the service time incurred in each customer is denoted by ST^2 instead of ST^1.

For the routes of ODs, there is one type of auxiliary information: *the latest allowable departure time from the visited satellite,* $l_{\Theta(1)}$. This information can be obtained through equation (4.7).

$$l_{\Theta(1)} = \min\left(l_{\Theta(3)} - d_{\Theta(2)\Theta(3)} - ST^2, l_{\Theta(2)}\right) - d_{\Theta(1),\Theta(2)} \tag{4.7}$$

The value of $e^a_{F(0)}$ is the same as the earliest departure time of first-echelon trucks from the depot, e_0, while $e^D_{\Omega(0)}$ is obtained from the time point when first-echelon truck(s) finish the service at satellite $\Omega(0)$. When total demands assigned to a satellite are higher than the maximum capacity of a first-echelon truck, the satellite is served by more than one truck. If such a condition occurs at satellite $\Omega(0)$, then the latest finishing time among the first-echelon trucks visiting the node is adopted for $e^D_{\Omega(0)}$. Another case worth mentioning is when a satellite s has no demand. If this case occurs, then the value of $e^D_s = d_{0,s} + ST^1$. In other words, we set the lowest possible value for e^D_s when satellite s is not used. Utilizing this technique, we prevent a time windows violation when satellite s is later assigned by city freighters and/or ODs.

If $F(i) \in V^L$, then $l^a_{F(i)} = l_0$, because no time windows are imposed upon the covering locations. On the other hand, if $F(i) \in V^s$, then the following steps are required to calculate $l^a_{F(i)}$. First, let φ denote the set of city freighters originating from satellite $F(i)$, and ψ represents the set of ODs who pick up the assigned demands at satellite $F(i)$. The value of $l^a_{F(i)}$ is then calculated using equation (4.8). Intuitively, the latest arrival time for a first-echelon truck at satellite $F(i)$ is the minimum value among the latest allowable departure time of city freighter $\kappa, \kappa \in \varphi$ and the latest allowable departure time of OD $\nu, \nu \in \psi$. Lastly, the value of $l_{\Omega(|\Omega|)} = l_0$ since there is no restriction on the arrival time of a city freighter to go back to satellite $\Omega(|\Omega|)$. Note that all aforementioned types of auxiliary information are updated whenever the associated routes (i.e., first-echelon trucks, city freighters, or ODs) change.

$$l^a_{F(i)} = \min\left(\min_{\kappa\in\varphi}\widetilde{z}_{\Omega(0)}, \min_{\nu\in\psi} l_{\Theta(1)}\right) - ST^1 \tag{4.8}$$

4.2. *An efficient time windows feasibility checking procedure for the 2E-VRPTW-CO-OD solution*

Our ALNS only works on a feasible solution space. Thus, ALNS needs to ensure the feasibility of the solution through iterations. This section explains the proposed time windows feasibility checking procedure for a 2E-VRPTW-CO-OD solution. The sequential time windows feasibility check is commonly utilized in literatures. However, such a procedure requires $O(|\Omega|)$ in the worst case for evaluating the time windows feasibility of a city freighter's route. Utilizing this method leads to a computational burden issue since ALNS requires a repetitive feasibility evaluation during the repair phase. Thus, an efficient time windows feasibility checking procedure is developed to alleviate this issue. This method is extended from Nagata et al. (2010) and Schneider et al. (2013) which only requires a time complexity of amortized $O(1)$ to obtain the amount of time windows violation. We apply this method in the repair phase (Section 4.5) and local search procedure (Section 4.6) for two reasons. First, we ensure that the starting time of serving a customer assigned to the home delivery option is within his predetermined time windows. Second, we guarantee the satellites are served accordingly so that all customers assigned to those satellites are served within their time windows.

Let $\Omega_k = \{\Omega_k(0), \Omega_k(1), \dots, \Omega_k(i), \Omega_k(j), \dots, \Omega_k(|\Omega_k|)\}$ be the route of city freighter k, consisting of a sequence that is generated from two partial paths: $\{\Omega_k(0), \Omega_k(1), \dots, \Omega_k(i)\}$and $\{\Omega_k(j), \dots, \Omega_k(|\Omega_k|)\}$. The time windows violation of Ω_k can be calculated using equation (4.9). Ω_l denotes the route of city freighter l, consisting of a sequence $\{\Omega_l(0), \Omega_l(1), \dots, \Omega_l(i), \nu, \Omega_l(j), \dots, \Omega_l(|\Omega_l|)\}$. Here, ν represents a visited customer node. The time windows violation of Ω_l is calculated using equation (4.10).

$$TW(\Omega_k) = \max\left\{\widetilde{a}_{\Omega_k(i)} + ST^2 + d_{\Omega_k(i)\Omega_k(j)} - \widetilde{z}_{\Omega_k(j)}, 0\right\} \tag{4.9}$$

$$\mathrm{TW}(\Omega_l) = \max\left\{\widetilde{a}_{\Omega_l(i)} + ST^2 + d_{\Omega_l(i)v} - f_v, 0\right\} + \max\left\{\max\left\{\min\left\{\widetilde{a}_{\Omega_l(i)} + ST^2 + d_{\Omega_l(i)v}, f_v\right\}, e_v\right\} + ST^2 + d_{v\Omega_l(j)} - \widetilde{z}_{\Omega_l(j)}, 0\right\} \tag{4.10}$$

Each route of the first-echelon truck must finish the service at each satellite so that the service for all customers assigned to the home delivery option starts within the pre-determined time windows. Thus, we adopt the aforementioned method to determine whether there is a time windows violation for routes of the first-echelon trucks. Note that the second-echelon related decisions have been determined completely at the time of dealing with the time windows feasibility check for the routes of the first-echelon trucks. Let F_m denote the route of first-echelon truck m, consisting of a sequence $\{F_m(0), F_m(1), \ldots, F_m(i), F_m(j), \ldots, F_m(|F_m|)\}$ that is generated from two partial paths: $\{F_m(0), F_m(1), \ldots, F_m(i)\}$ and $F_m(j), \ldots, F_m(|F_m|)\}$.

Let F_n represent the route of first-echelon truck n, consisting of a sequence $\{F_n(0), F_n(1), \ldots, F_n(i), v, F_n(j), \ldots, F_n(|F_n|)\}$, where v denotes a visited node, either a satellite or a covering location. The amounts of time windows violation for F_m and F_n are obtained using equations (4.11) and (4.12), respectively. In addition, we recall l_i^a and e_i^a as the latest allowable arrival time of first-echelon truck k at node i and the earliest allowable time of first-echelon truck k to start the service at node i, respectively, as explained in Section 4.1.

$$TW(F_m) = \max\left\{\widetilde{a}_{F_m(i)} + ST^1 + d_{F_m(i)F_m(j)} - \widetilde{z}_{F_m(j)}, 0\right\} \tag{4.11}$$

$$TW(F_n) = \max\left\{\widetilde{a}_{F_n(i)} + ST^1 + d_{F_n(i)v} - l_v^a, 0\right\} + \max\left\{\max\left\{\min\left\{\widetilde{a}_{F_n(i)} + ST^1 + d_{F_n(i)v}, l_v^a\right\}, e_v^a\right\} + ST^1 + d_{vF_n(j)} - \widetilde{z}_{F_n(j)}, 0\right\} \tag{4.12}$$

4.3. Initial solution

In order to construct an initial solution for 2E-VRPTW-CO-OD, a two-phase construction algorithm is proposed. First, the algorithm generates city freighter routes, OD routes, and assignment of customers into covering locations. Second, the algorithm deals with the construction of first-echelon trucks' routes.

The first phase focuses on the second-echelon network. All customer nodes in V^c are initially stored in the list of unprocessed customers (*UnprocessCustomer*). In each iteration, the algorithm utilizes the cheapest insertion procedure to insert a customer selected from *UnprocessCustomer*. Feasible insertion costs are evaluated from three options: city freighters' routes, ODs' routes, and assignments to covering locations. The selected customer to be inserted is the customer possessing the lowest insertion cost. The first phase repeats until no customer remains in *UnprocessCustomer*.

The second phase creates routes of the first-echelon trucks. After performing the first phase, we have the total demands accumulated in every existing intermediate location. Based on the provided information, we make use of the insertion procedure described in the large repair operator (Section 4.5). Finally, a feasible initial solution for an input of ALNS is generated by implementing the aforementioned two-phase algorithm.

4.4. Destroy operators

The destroy operators aim to remove a number of nodes based on a particular metric, depending on which operator is implemented. They are divided into two categories: small and large destroy operators. The small destroy operators aim to only remove a number of customers, while the large destroy operators deal with both intermediate locations and customers. There are in total seven small destroy operators and six large removal operators.

Before applying a small destroy operator, ALNS first determines η_R customer nodes that need to be removed. Here, η_R is an integer number generated randomly between an upper-bound $\overline{\eta}_R$ and a lower-bound $\underline{\eta}_R$. The small destroy operators involve (1) Random removal, (2) Route removal, (3) Worst Removal, (4) Perturbed Worst Removal, (5) Related Removal, (6) Perturbed Related Removal, and (7) Node Neighborhood Removal. Operators (1) to (6) were originally proposed by Pisinger and Ropke (2007) while operator (7) was first developed by Demir et al. (2012). While modifications are made to route and related removal operators, one can refer to the two aforementioned papers for further descriptions of the remaining operators.

For the route removal, instead of utilizing η_R, this operator first determines a number of routes, η_{route}, that will be removed from a solution. Here, η_{route} is randomly generated between 1 to $\left\lfloor \overline{\eta}_{route}/2 \right\rfloor$, where $\overline{\eta}_{route}$ denotes the total routes in a solution, obtained from both city freighters' routes and ODs' routes. In each iteration, this operator selects a route randomly, either a city freighter's route or an OD's route, and removes all customers from the selected route. The iteration stops when η_{route} routes have been removed.

For the related removal, the measurement metric used to remove customers is the relatedness index. There are four criteria to calculate the relatedness index between two customers: distance between two customers, demand of the customers, earliest time to start the service at the location of customers, and the current position of the customers in the evaluated solution. Equation (4.13) calculates the relatedness index.

$$S_{(i,j)} = \varphi_1 \frac{c_{ij}}{\max\limits_{i,j \in V^c} c_{ij}} + \varphi_2 \frac{|d_i - d_j|}{\max\limits_{i \in V^c} d_i - \min\limits_{i \in V^c} d_i} + \varphi_3 \frac{|e_i - e_j|}{\max\limits_{i \in V^c} e_i - \min\limits_{i \in V^c} e_i} + \varphi_4 Pos_{ij} \quad (4.13)$$

Here, φ_1-φ_4 are the weights for each criterion. The first, second, and third terms in equation (4.13) are normalized by dividing the value obtained from two customer nodes with the maximum value of each term. The last term, Pos_{ij}, is a binary value. $Pos_{ij} = 1$ when one of two conditions occurs: the customers are located in the same city freighter route or assigned to the same covering location; otherwise, $Pos_{ij} = 0$.

For the perturbed version of worst and related removal operators, a list of remaining unremoved customers L_{remain} is first created and sorted in a descending order based on the employed measurement metric. The operator then selects a customer with rank $\lceil U(0,1)^p * |L_{remain}| \rceil$ and removes the customer from the current solution. Here, p represents the randomness factor of this operator. This procedure repeats until η_R customers are removed.

There are six large destroy operators focusing on re-configuring the intermediate locations adopted from Hemmelmayr et al. (2012) and Enthoven et al. (2020). These operators are categorized as *location close*, *location open*, and *location swap* where the term location here represents satellite or covering location.

4.5. Repair operators

After the removal phase, ALNS has a number of customers that need to be re-inserted. There are two types of repair operators: small and large. The small repair operators aim to re-insert the removed customers to the solution, while the large repair operators aim to re-build routes of the first-echelon trucks.

There are six small repair operators that can be categorized into two types. The first one is greedy insertion, and the other is regret insertion. Both classes have deterministic and perturbed versions. Furthermore, 2-regret and 3-regret insertions are implemented. The mechanism of the perturbed versions is similar to the deterministic versions with one difference: the method of calculating the insertion cost. In the perturbed versions, the insertion cost of a customer is perturbed by multiplying the costs (regret values) in greedy (regret) insertions with a random number generated within the range $[1 - \tau_{perturb}, 1 + \tau_{perturb}]$. All small repair operators only aim to insert every customer from the list of removed customers into a feasible position from these available options: a city freighter route, an OD route, and a covering location. In other words, small repair operators only deal with the second-echelon network. Hemmelmayr et al. (2012) offered a detailed mechanism of the implementations of greedy and regret insertions.

The large repair procedure is implemented after small repair and local search for the second-echelon network have both been performed (see Algorithm 1). An insertion procedure described in Breunig et al. (2016) is executed. First, a simple pre-processing step is conducted by calculating how many full truckloads need to be performed at each intermediate location. Back-and-forth trips are created for each intermediate location based on the number obtained from the pre-processing step. The remaining demands at each intermediate location are then inserted using the cheapest feasible insertion procedure.

4.6. Local search procedure

The ALNS utilizes two types of local search, and each one is implemented on different level of networks: the first- and the second-echelon networks. Algorithm 2 shows the general structure of the implementation of a local search operator. The local search primarily involves four operators with their general descriptions provided as follows.

1. **Swap:** This operator selects two nodes and swaps their positions. There are two versions of this operator: swap-intra (one route) and swap-inter (two different routes).
2. **Reinsertion:** This operator selects one node and inserts it at a selected position. There are two versions of this operator: reinsertion-intra (one route) and reinsertion-inter (two different routes).
3. **2-opt:** This operator selects two arcs in a route, deletes them, and re-connects the broken components.
4. **2-opt*:** The mechanism is similar to 2-opt, but it involves two different routes instead of one route.

We employ the efficient time windows feasibility checking procedure in several local search operators: (1) swap-inter, (2) reinsertion-inter, and (3) 2-opt* operator. For swap-intra, reinsertion-intra, and the 2-opt operator, we employ the sequential feasibility check on time windows, requiring $O(|\Omega|)$ in the worst case.

4.6.1. Local search for the second-echelon network

We utilize all types of local search operators described in Section 4.6. For 2-opt and 2-opt*, we only apply these two operators for city freighter routes originating from the same satellite. Since there are three options in assigning a customer (a city freighter route, an OD route, and a covering location), we derive the remaining operators (i.e., swap and reinsertion) to deal with this issue. The intra versions are only applied to city freighter routes. For the inter versions, the derived operators consider (1) two different routes of city freighters, (2) a route of a city freighter and a route of OD, and (3) a route of a city freighter and customers assigned to a covering location.

The local search operators are implemented sequentially. To alleviate the expensive computational time issue, we do not consider all neighborhoods resulting from a particular local search operator. A pruning technique called *granular search* is widely used in developing algorithms for routing problems (Vidal et al., 2013). This technique requires a pre-processing step; i.e., for each customer node in V^c, v_i, we create a list, $L_{close}(v_i)$, representing a list of other customer nodes spatially related to v_i. Before evaluating a neighborhood further, a local search operator screens whether the change in the neighborhood resulting in customer nodes that are related to one another is placed adjacently. Consequently, the number of evaluated neighborhoods falls, and the computational time to perform the local search procedure is quicker.

4.6.2. Local search for the first-echelon network

The routes of the first-echelon truck can be categorized as full-truckload and less-than-truckload deliveries. The local search operators are only implemented to less-than-truckload routes. For the first-echelon network, we adopt all local search operators mentioned in Section 4.6, which all follow Algorithm 2 and are implemented sequentially. The pruning technique explained in Section 4.6.1 is not adopted in local search for the first-echelon network, because the number of intermediate locations is significantly less than the number of customers.

5. Computational study

This section describes the numerical experiments. They were carried out in a PC with i7-8700 CPU @ 3.20 GHz and 16 GB RAM, which has a single-thread rating of 2,679 (PassMark Software, 2021). A set of 2E-VRPTW-CO-OD instances was generated for the experiments and can be accessed at http://web.ntust.edu.tw/~vincent/vrpod/.

5.1. Generating the benchmark instances

Since 2E-VRPTW-CO-OD is a new variant of 2E-VRP, there is no existing benchmark instance. Thus, we develop a set of new benchmark instances from 2E to VRP benchmark instances. In particular, we utilize two small instance sets of 2E-VRP proposed by Perboli et al. (2011) and one large instance set of 2E-VRP developed by Hemmelmayr et al. (2012). The two small instances are called Sets 2 and 3, while the large instance is called Set 5. In Sets 2 and 3, the number of customers ranges from 21 to 50, and the number of satellites is either 2 or 4. In Set 5, the number of customers is 100 or 200, while the number of satellites is 5 or 10.

In order to construct a 2E-VRPTW-CO-OD instance, we need to add several types of information to a 2E-VRP instance. First, we add a time window for each customer i, $i \in V^c$; i.e., $[e_i, f_i]$. By utilizing the solution of 2E-VRP instances provided in Breunig et al. (2016), we add a time window for each customer by following these steps.

1. For each visited satellite, we calculate the time when first-echelon trucks finish serving the satellite. In addition, the service time of each satellite is set to 90.
2. For each customer, we calculate the arrival time of a city freighter that serves the customer. In addition, the service time of each customer is set to 30.
3. The arrival time of a city freighter at each customer is recorded as the center of the time window $\widehat{s}_i, \widetilde{i} \in V^c$.
4. For each customer i, a random number $\bar{r}_i$ ranging between 0 and $\widehat{s}_i/2$ is generated, and the customer's time window, $[e_i, f_i]$, are set to $[\widehat{s}_i - \bar{r}_i, \widehat{s}_i + \bar{r}_i]$.

Second, we increase the number of available fleets by multiplying the current number of fleets in each echelon by 2. Third, we need several types of information of covering locations: (1) number of covering locations $|V^L|$, (2) the locations of covering locations (3) covering radius *rad*, and (4) the connection cost of assigning customer i to covering location j, ς_{ij}. We adopt the rules proposed by Enthoven et al. (2020). In our cases, we set $\alpha = 0.075$ and $\beta = 0.075$. Moreover, the service time of each covering location is set to 90. Fourth, we need several types of information related to an OD, described as follows.

1. The number of available ODs is set to the number of customers.
2. The earliest time for OD i to depart from his/her origin, α_i, is randomly generated within the range of $[0.5\bar{e}, 1.0\bar{e}]$, and the latest time for OD i to arrive at his/her destination, β_i, is randomly generated within the range of $[0.5\bar{f}, 1.0\bar{f}]$. Herein, $\bar{e}$ and $\bar{f}$ denote the average of the earliest time to serve all customers and the average latest time to serve all customers, respectively.
3. The compensation factor of an OD is set to 1.
4. The origin node of each OD is generated by first selecting a satellite and selecting a location, resulting in a random distance within the range of $[0.3\bar{d}, 0.5\bar{d}]$ from the satellite. The destination node of each OD is constructed by selecting a customer node randomly and selecting a location, resulting in a random distance within the range of $[0.3\bar{d}, 0.5\bar{d}]$ from the customer node. Herein, $\bar{d}$ denotes the diagonal of a customer plane containing the smallest square that contains all customers.

5.2. Parameter calibration for the proposed ALNS

Before applying the ALNS heuristic, we randomly select 11 2E-VRPTW-CO-OD instances to conduct parameter tuning to select the best combination of parameters. The approach of one factor at a time (OFAT) is adopted for the parameter tuning (Yu et al., 2021; Yu et al., 2020c). The first step of OFAT is setting each parameter of the ALNS heuristic to its initial value. The list of final parameters can be found in Table A1, and the initial value of each parameter, if applicable, is adopted from Enthoven et al. (2020). A set of candidate

Table 2
The average gaps of state-of-the-art algorithms and ALNS for solving 2E-VRP benchmark instances.

Instances	$\|V^c\|$	$\|V^S\|$	Ave BKS	HCC – 500K	BSHV	EJRBS – ST	EJRBS – 4T	ALNS
Set 2								
2a	21–32	2	577.59	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
2b	50	2–4	549.50	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
2c	50	2–4	607.94		**0.00**	0.06	**0.00**	**0.00**
Set 3								
3a	21–32	2	591.30	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
3b	50	2	714.75	**0.00**	**0.00**	0.16	**0.00**	**0.00**
3c	50	2	668.40		**0.00**	0.16	**0.00**	0.12
Set 4								
4a	50	2–5	1419.95		**0.00**			0.08
4b	50	2–5	1397.06	0.27	**0.00**			0.04
Set 5	100–200	5–10	1118.81	1.58/0.85*	0.46	1.71	0.98	0.73/0.67+

* Average gap is obtained by using 5000 K iterations as the stopping criterion.
+ Average gap is obtained by merely using 1200 s as the stopping criterion.

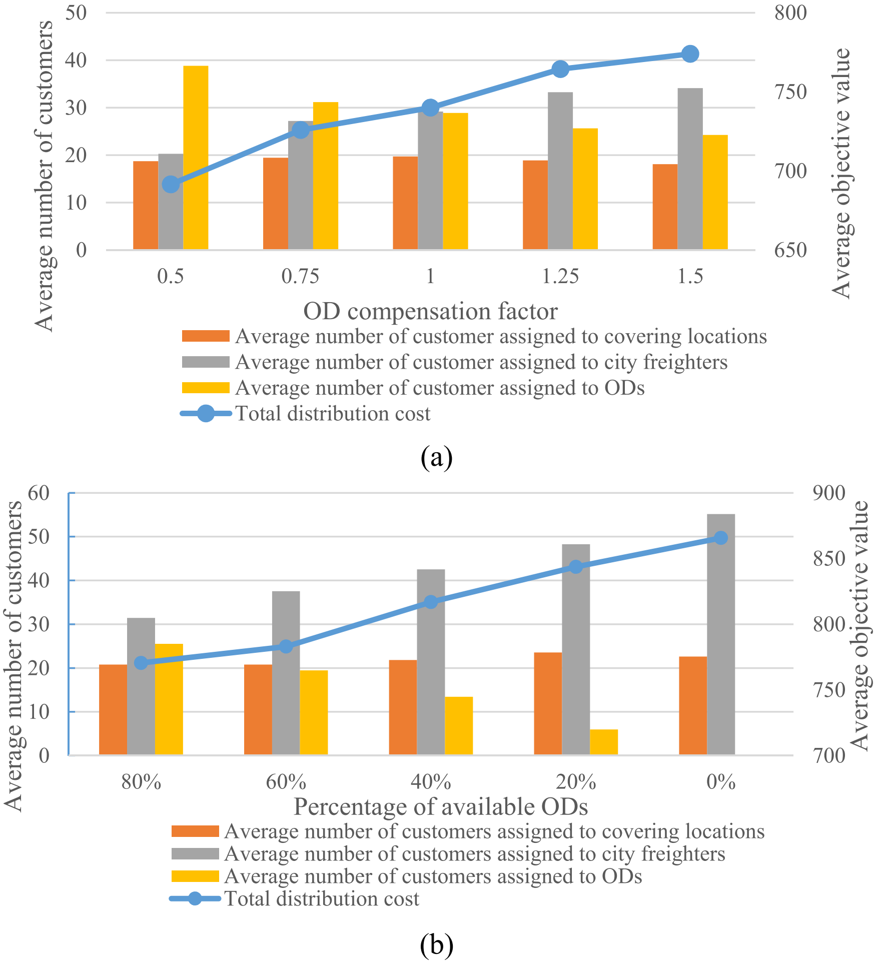

Fig. 3. Summary of the statistics under two varying conditions: (a) OD compensation factor, and (b) Percentage of available ODs.

values for each parameter is then tested and also listed in Table A1. We test each candidate value of a parameter with values fixed for the remaining parameters. While testing each candidate value, we run each selected instance five times and record the best obtained objective value. The final value for a parameter is the one leading to the lowest overall objective value, which is calculated from the summation of the best objective value of all tested instances. Table A2 shows the final values for all parameters of the ALNS heuristic after implementation of OFAT.

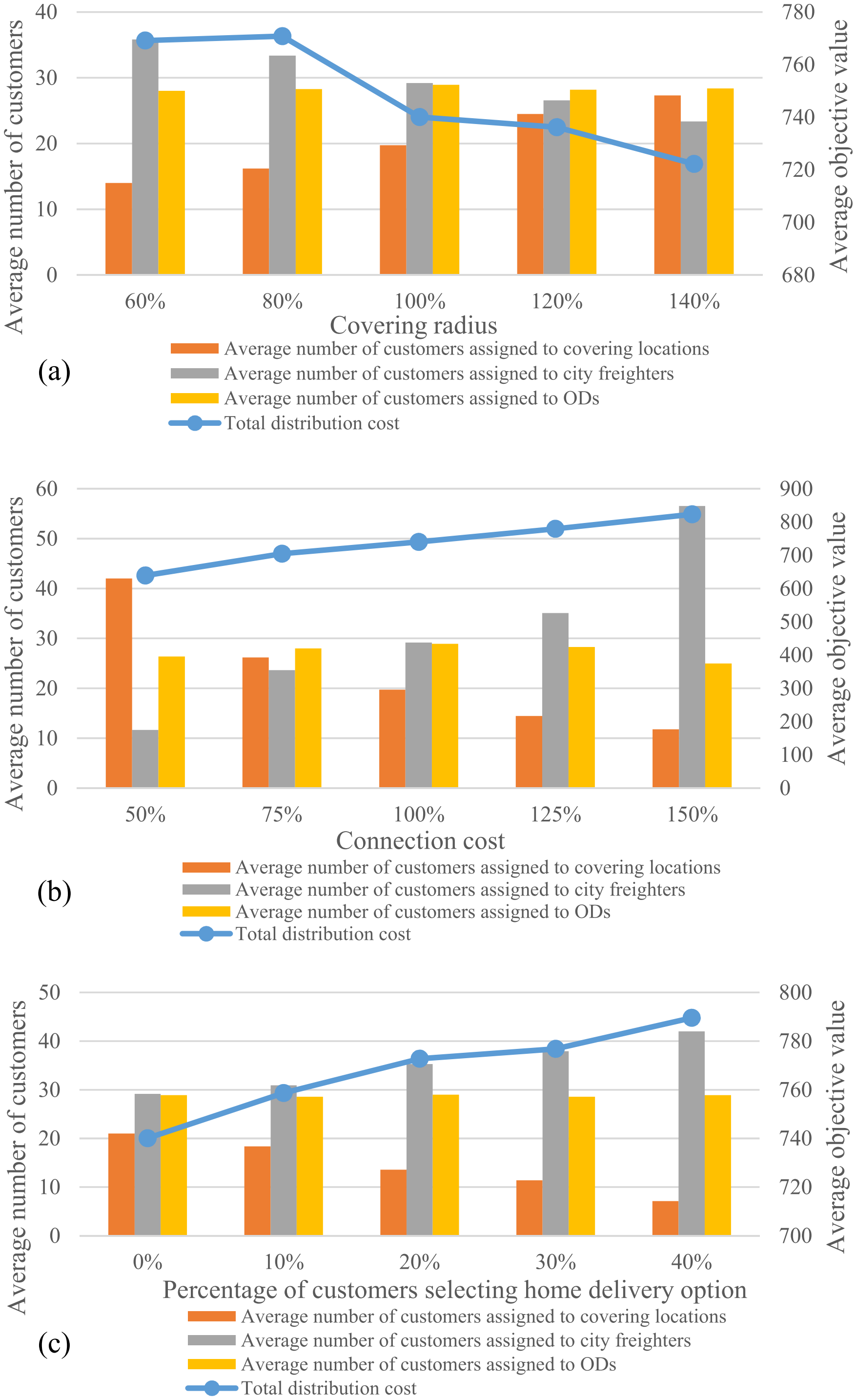

Fig. 4. Summary of the statistics under three varying conditions: (a) covering radius, (b) connection cost, and (c) percentage of customers selecting the home delivery option.

5.3. ALNS performance for solving 2E-VRP benchmark instances

Table 2 summarizes the results of the proposed ALNS heuristic for solving the 2E-VRP benchmark instances. In addition, the detailed results appear in Tables A3–A8 in the appendix. The benchmark instances obtained from Hemmelmayr et al. (2012) and Breunig et al. (2016) consist of four sets. The results from previous literature are listed for the purpose of comparison: (1) HCC-500K

(Hemmelmayr et al., 2012), (2) BSHV (Breunig et al., 2016), (3) EJRBS-ST (Enthoven et al., 2020), and (4) EJRBS-4T (Enthoven et al., 2020). For more details, we provide the average gaps between the best solutions of each method and BKS. Based on the aforementioned literature, the best solution is obtained from 5 replications of solving each benchmark instance. Thus, we apply the same condition to our ALNS to obtain the average gap. The BKS solutions are all obtained from Breunig et al. (2016).

The proposed ALNS successfully obtains zero gaps for all sets in Set 2 and two of three sets in Set 3. The remaining average gaps vary from 0.04% to 0.73%. Among the aforementioned literature, only two of four methods are applied to solve instances in Set 4. The quality of average gap resulting from our ALNS for Set 4 is between HCC-500K and BSHV. The highest average gap produced by ALNS is the one from solving Set 5. However, the average gap outperforms the average gap of three other algorithms: HCC-500K, EJRBS-ST, and EJRBS-4T. In other words, BSHV is the only algorithm that outperforms our results in Set 5 in terms of average gap. We additionally run ALNS by using a time limit (i.e., 1200 s) as the only stopping criterion.

The results show that the average gap improves to 0.67%. This indicates that the proposed ALNS needs longer computational time to produce better solutions for larger 2E-VRP benchmark instances. This behavior is similar to HCC-500K. For Set 5, HCC-500K changed the stopping criterion to 5000 K and observed that the average gap drops to 0.85%.

Table A9 presents the computational environments of state-of-the-art algorithms and their associated single-thread rating (i.e., P_{score}) obtained from PassMark Software (2021). Table A10 shows the average computational time required by state-of-the-art algorithms and ALNS. HCC-500K, EJRBS-ST, and EJRBS-4T employ maximum iterations, while BSHV utilizes a time limit as a stopping criterion. In addition, we present the scaled average computational time according to the processor of Breunig et al. (2016) in Table A10. The proposed ALNS utilizes both criteria as the stopping conditions. In other words, ALNS is terminated whenever one of the criteria is achieved. Based on Table A10, the average computational times required by ALNS are rather long for solving large-scale instances. The underlying reason for this phenomena is that the local search procedure is implemented in every iteration of ALNS.

5.4. Results of solving small 2E-VRPTW-CO-OD instances

In order to evaluate the performance of ALNS heuristic on 2E-VRPTW-CO-OD small instances, we solve the MILP model presented in section 4 using an exact solver, CPLEX. We refer small instances to Sets 2 and 3 of the generated instances. In total, there are 54 small instances. The maximum CPU time of CPLEX is set to 2 h, while ALNS runs for 5 replications by utilizing both t_{max} and $iteration^{max}$ as the stopping criterion. Table A11 summarizes the results of CPLEX and ALNS for solving Sets 2 and 3 of 2E-VRPTW-CO-OD instances.

Based on Table A11, all instances with 21 customers, 6 instances with 32 customers, and 1 instance with 50 customers are solved to optimality by CPLEX. The results show that the performance of CPLEX becomes worse when the size of problems increases. Among 19 optimal solutions provided by CPLEX, the proposed ALNS heuristic can obtain 16 optimal solutions, while the gap between the three remaining solutions' objective values and their associated optimal solutions is not larger than 0.38%. Moreover, the heuristic provides better solutions for the remaining instances that cannot be solved to optimality by CPLEX. Overall, the best solution obtained by the heuristic improves the results obtained by CPLEX for Sets 2 and 3 instances on average by 1.30%. Moreover, the average objective value of solutions obtained by the heuristic improves the results of CPLEX on average by 1.01%. This information shows that the developed ALNS is reasonably robust for solving small 2E-VRPTW-CO-OD instances. Based on Table A11, the computational time of CPLEX also increases significantly when the size of problems increases. On average, CPLEX required 5003.066 s to solve an instance while ALNS only required 29.521 s.

The ALNS heuristic can provide reasonably good results for small instances with significantly lower computational time. Therefore, we utilize ALNS heuristic to solve larger instances in the following section.

5.5. Performance of ALNS for solving large 2E-VRPTW-CO-OD instances

Table A12 summarizes the results of large-scale instances obtained by the proposed ALNS. For more details, we run the ALNS under two conditions: without and with local search. In particular, we perform five replications of ALNS on each instance with the same stopping criteria presented in Section 5.4. The performances of ALNS without and with local search are shown in columns "ALNS-noLS" and "ALNS-LS", respectively. In addition, we mention ALNS without LS as ALNS-noLS and ALNS with LS as ALNS-LS hereafter. Table A12′s analyses are provided as follows.

1. Both ALNS-noLS and ALNS-LS improve the quality of initial solution. These results show that ALNS heuristic is necessary to solve 2E-VRPTW-CO-OD.
2. ALNS-LS outperforms ALNS-noLS in terms of average gap to BKS solutions. As shown in Table A12, the average values for best solutions obtained by ALNS-LS and ALNS-noLS are 966.865 and 1008.145, respectively. More precisely, ALNS-LS is able to obtain 17 better solutions out of 18 large-scale instances when compared to that of ALNS-noLS.
3. The computational time required by ALNS-LS is higher than ALNS-noLS. More precisely, the average computational times of ALNS-LS and ALNS-noLS are respectively 564.143 s and 229.987 s. The local search procedure is performed in every iteration of ALNS in ALNS-LS. Consequently, the computational time of ALNS-LS is significantly higher than ALNS-noLS.

We conduct additional experiments on ALNS-LS and ALNS-noLS by only using the stopping criterion of maximum time limit. The purpose is to confirm the impact of the local search procedure to ALNS. The results are described in the appendix. As shown in Table A13, ALNS-LS also outperforms ALNS-noLS in terms of average solution quality. These results confirm that the local search procedure on average improves the quality of 2E-VRPTW-CO-OD solutions.

Table A14 shows the contribution of each ALNS operator for improving the best found solutions when solving the large-scale instances. The contribution is calculated by dividing the number of iterations that an operator successfully improves the best found solutions by the total number of iterations of all operators that improve the best found solutions. The contribution is expressed in terms of percentage and categorized based on the type of operator: small destroy, large destroy, and small repair operators.

There are four dominating small destroy operators in ALNS-LS: random, worst, probabilistic worst, and probabilistic related. In ALNS-noLS, only three of the aforementioned four operators are dominant: random, worst, and probabilistic worst. The contribution of these three small destroy operators for improving the best found solutions adds up to 95.86% in ALNS-noLS. For large destroy operators, satellite removal contributes the most for improving the best found solutions. Lastly, both versions of greedy insertion are the repair operators that improve the best found solutions the most. Similar to the behaviors of random, worst, and probabilistic worst removal operators, both versions of greedy insertion even perform better in ALNS-noLS with a total contribution of 98.37%.

5.6. Managerial insights

We now present sensitivity analyses on ODs and covering locations. We analyze two aspects of ODs: (1) compensation paid to the employed ODs and (2) number of available ODs. We then present three aspects related to covering location: (1) connection cost for assigning a customer to a covering location, (2) number of customers preferring the alternative delivery option, and (3) the radius of a covering location.

The analyses are conducted based on the solutions provided by the ALNS heuristic over randomly selected original instances. The instances used in this section are the same ones with the instances used for parameter tuning. In each section, one parameter is varied while the other parameters defined in the instances remain the same as the ones in the original instances. Three types of statistics are measured to reflect the decisions of 2E-VRPTW-CO-OD: (1) Number of customers assigned to city freighters; (2) Number of customers assigned to ODs; and (3) Number of customers assigned to covering locations. We also measure the average objective value to show the changes in total operational costs. In order to derive the insights, we collect the statistics from the best obtained solution of each instance and take the average value of those aforementioned statistics.

5.6.1. Sensitivity analyses on the characteristics of occasional drivers

Fig. 3(a) and (b) show the summary of statistics under varying OD compensation factors and varying number of available ODs. Five scenarios are generated for each of them. The compensation factor of OD ranges from 0.5 to 1.5, and the available number of ODs ranges from 80% to 0% of the original number of ODs.

Several changes occur when the compensation factor of OD increases. First, the average number of employed ODs drops, and the average number of customers assigned to city freighters increases. In other words, it is more beneficial for the system to shift the assignment of customers to city freighters when the cost of employing ODs is getting more expensive. The change in the compensation factor of OD also influences total distribution costs. In particular, the costs consistently rise when the compensation factor of OD increases.

Fig. 3(b) emphasizes the changes that occur when the number of available ODs varies. Similar to the compensation factor of OD, the average number of customers assigned to city freighters increases, while the average number of customers assigned to ODs decreases. Moreover, the total distribution costs are higher in the scenario with a lower number of available ODs. This implies that a higher number of available ODs brings higher cost savings to the system.

5.6.2. Sensitivity analyses on the characteristics of covering locations

Fig. 4(a) describes the influence of changing the covering radius toward the decisions in 2E-VRPTW-CO-OD and the total distribution cost. Increasing the covering radius of a covering location implies that a higher number of customers can be assigned to the covering location. In other words, it provides a higher degree of consolidation. This results in a lower distribution cost and a higher number of customers assigned to covering locations.

Fig. 4(b) shows the impact of varying connection costs. When the connection cost increases, it means that the compensation of assigning customers to covering locations also increases. Consequently, the number of customers assigned to city freighters increases, while the number of customers assigned to covering locations decreases. In terms of total objective cost, a higher connection cost leads to greater total distribution costs. Fig. 4(b) implies that reducing the connection cost as low as possible can achieve the lowest distribution cost. However, there is still a further consideration; i.e., the number of customers preferring a particular delivery option. Changing the connection cost may influence the number of customers selecting a particular delivery option.

Fig. 4(c) depicts the change in decisions of 2E-VRPTW-CO-OD and the total distribution cost when changes occur in the number of customers requiring the home delivery option. In the original instances, we assume that 100% of customers are flexible, and so the demands can be sent to either their home or covering locations. As shown in Fig. 4(c), the total distribution cost increases when the percentage of customers requiring home delivery increases. In addition, the utilization of covering locations is lower since there are customers who only require the home delivery option. Therefore, the connection costs need to be determined accordingly by considering the possible shift of flexible customers to home-delivery customers.

6. Conclusions and future research

This research introduces the Two-echelon Vehicle Routing Problem with Time Windows, Covering Options, and Occasional Drivers. The focus of this problem is on two-echelon distribution networks. In the second echelon, each customer can be served either at his/her location within his/her pre-determined time windows, or one of the available covering locations at any time during the planning period. When customers are served at their location, the demands of the customers are delivered from one of the available satellites by either city freighters or ODs. In the first echelon, satellites and covering locations are served by first-echelon trucks originating from the depot.

A MILP is formulated and an ALNS is proposed to deal with 2E-VRPTW-CO-OD. Since this problem is new, no benchmark instances exist. Therefore, we generate a set of benchmark instances from 2E – VRP benchmark instances. We assess the performance of the proposed ALNS by solving both 2E-VRPTW-CO-OD and 2E-VRP benchmark instances. The average gap between the best solutions obtained by ALNS and CPLEX for small-scale 2E-VRPTW-CO-OD instances is -1.30%, showing that our ALNS outperforms CPLEX in terms of average solution quality. In addition, we implement the proposed ALNS to solve 2E-VRP benchmark instances, and the maximum average gap to BKS is within 0.73%. These results confirm that the proposed ALNS produces high-quality solutions for 2E-VRPTW-CO-OD and comparable results for 2E-VRP.

We perform analyses on the ALNS components for solving the 2E-VRPTW-CO-OD instances. Results show that the local search procedure on average improves the quality of solutions for large-scale instances with the main trade-off; i.e., computational time. Indeed, ALNS-LS produced 17 better solutions out of 18 large-scale instances when compared to ALNS-noLS. The average computational time for ALNS-LS and ALNS-noLS are 564.143 and 229.987 s, respectively.

Based on the analyses of ALNS for solving the large-scale 2E-VRPTW-CO-OD instances, the contribution of small destroy and small repair operators becomes more evenly distributed when the local search procedure is embedded into ALNS. We also provide sensitivity analyses regarding the characteristics of ODs and covering locations, resulting in several interesting managerial insights for designing such a delivery system. Finally, several extensions that can be considered as future works are described as follows: (1) heterogeneous ODs; (2) the existence of transshipment nodes serving as meeting points between ODs and first-echelon trucks; (3) rich objective function incorporating various types of costs; and (4) multi-product delivery.

CRediT authorship contribution statement

Vincent F. Yu: Conceptualization, Methodology, Supervision, Resources, Writing – review & editing. **Panca Jodiawan:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Visualization, Writing – original draft. **Ming-Lu Hou:** Software, Validation, Formal analysis. **Aldy Gunawan:** Methodology, Formal analysis, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This research was partially supported by the Ministry of Science and Technology of the Republic of China (Taiwan) under Grant MOST 109-2410-H-011-010-MY3 and the Center for Cyber–Physical System Innovation from The Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan.

Appendix A

A.1. General structure of a local search operator

Algorithm 2

```
    noImprovement ←true
1   do
2   |   noImprovement ←true
3   |   for every neighborhood resulted from applying a local search operator to
    |   currentSolution do
4   |   |   if the neighborhood is better than currentSolution then
5   |   |   |   Replace currentSolution with neighborhood
6   |   |   |   noImprovement ←false
7   |   |   |   Go back to step 2
8   while noImprovement = false
```

A.2. An illustrative example

Fig. 2 describes a solution example of 2E-VRPTW-CO-OD. Fig. A1(a) to A1(c) show the related routes with arrival time at each visited node. The customers are numbered from 1 to 21, the satellites are numbered from 22 to 23, the covering locations are numbered from 24 to 25, the origin nodes of available occasional drivers are numbered from 26 to 46, and the destination nodes of available occasional drivers are numbered from 47 to 67. To sum up, there are 21 customers, 2 satellites, 2 covering locations, and 21 available occasional drivers.

Based on Fig. 2, two first-echelon trucks are utilized to serve two satellites (i.e., satellites 22 and 23) and one covering location (i.e., covering location 24). The first first-echelon truck arrives at satellite 23 at time point 9.85. By adding the service time incurred at the satellite, the first first-echelon truck finishes the service at satellite 23 at time point 99.85. By using a similar mechanism, the second first-echelon truck first serves satellite 22 at time point 11.18 and finishes the service there at time point 101.18. It then travels to covering location 24 and finishes serving the covering location at time point 208.18.

0	S23	0
0	9.85	109.7

0	S22	L24	0
0	11.18	118.18	225.443

(a)

S23	17	20	18	14	S23
99.85	184	225.31	267.35	410	445.38

S22	7	2	1	3	4	8	10	S22
101.18	126.67	175.78	214.32	269	301.83	350.93	390.15	437.15

S23	16	S23
99.85	99.85	129.85

S22	12	S22
101.18	101.18	131.18

(b)

O1	22	15	D1
85	108.77	125	167.65

O3	23	19	D3
98	137.05	157.86	193.24

O5	23	21	D5
109	146.01	170.10	211.14

O18	23	13	D18
96	112.4	136	198.70

(c)

Fig. A1. (a) the solution representation for first-echelon routes, (b) the solution representation for city freighter routes, and (c) the solution representation for occasional driver routes.

Four city freighters are utilized to serve customers in the second-echelon network. Two of them start their journey from satellite 22, and the remaining ones originate from satellite 23. The earliest allowable times for the city freighters starting from satellite 22 and satellite 23 are 99.85 and 101.18, respectively. The first city freighter has route {22, 7, 2, 1, 3, 4, 8, 10, 22}, the second one has route {22, 12, 22}, the third one has route {23, 16, 23}, and the fourth one has route {23, 17, 20, 18, 14, 23}. Moreover, 4 out of 21 occasional drivers are employed to serve four customers. Similar to city freighters, the earliest allowable times for occasional drivers to pick up the assigned demands at satellites 22 and 23 are 99.85 and 101.18, respectively. Occasional driver 1 serves customer 15 by picking up the demand at satellite 22. Occasional drivers 3, 5, and 18 serve customers 19, 21, and 13, respectively. All these three occasional drivers pick up the demands at satellite 23. Each vehicle, either city freighter or occasional driver, needs to arrive at each customer assigned to the vehicle no later than the latest time window of the customer. Lastly, 4 customers (i.e., customers 5, 6, 9, and 11) are served by covering location 24 (see Tables A1–A15).

Table A1
The values of ALNS parameters for parameter tuning purposes.

Symbol	Final Value
ω_{grace}	160, 180, 200, **220**, 240, 260, 280
p	1, 2, **3**, 4, 5, 6, 7
$\tau_{perturb}$	0.05, 0.1, 0.15, **0.2**, 0.25, 0.3, 0.35
σ_1	30, 40, 50, **60**, 70, 80, 90
σ_2	10, 20, **30**, 40, 50, 60, 70
β	0.4, 0.45, 0.5, **0.55**, 0.6, 0.65, 0.7
$\eta_{restart}$	6,000, 8,000, **10,000**, 12,000, 14,000, 16,000, 18,000
Ω	$\min(\lvert V_C\rvert, 10)$, $\min(\lvert V_C\rvert, 15)$, $\min(\lvert V_C\rvert, 20)$, $\mathbf{\min(\lvert V_C\rvert, 25)}$, $\min(\lvert V_C\rvert, 30)$, $\min(\lvert V_C\rvert, 35)$, $\min(\lvert V_C\rvert, 40)$
φ_1	**1**, 3, 5, 7, 9, 11, 13
φ_2	**1**, 3, 5, 7, 9, 11, 13
φ_3	**1**, 3, 5, 7, 9, 11, 13
φ_4	**1**, 3, 5, 7, 9, 11, 13
η_s	60, 80, **100**, 120, 140, 160, 180

Bold values indicate the initial values employed in the parameter tuning process.

Table A2
Final parameters in the proposed ALNS algorithm.

Symbol	Explanation	Final Value
t_{max}*	Maximum computational time	1,200 s
$iteration^{max}$*	Maximum number of Iterations	300 K
ω_{grace}	Grace period	160
p	Randomness factor	5
$\tau_{perturb}$	Perturbation factor	0.05
σ_1	Global solution score	30
σ_2	Local solution score	30
β	Control parameter score	0.55
$\eta_{restart}$	Restart period	10,000
Ω	Number of nodes per local search	$\min(\lvert V_C\rvert, 25)$
φ_1	First Shaw parameter	7
φ_2	Second Shaw parameter	1
φ_3	Third Shaw parameter	1
φ_4	Fourth Shaw parameter	1
η_s	Necessary iterations to update operators' probability	100

*The parameter is determined before the process of parameter tuning.

Table A3
Results for set 2 of 2E-VRP benchmark instances.

Instance	$\|V^C\|$	$\|V^S\|$	$\|V^T\|$	m^2	BKS	Best	Average	t(s)	Gap to BKS
Set 2a									
E-n22-k4-s6-17	21	2	3	4	417.07	417.07	417.07	27	0.00
E-n22-k4-s8-14	21	2	3	4	384.96	384.96	384.96	10	0.00
E-n22-k4-s9-19	21	2	3	4	470.6	470.6	470.6	48	0.00
E-n22-k4-s10-14	21	2	3	4	371.5	371.5	371.5	10	0.00
E-n22-k4-s11-12	21	2	3	4	427.22	427.22	427.22	19	0.00
E-n22-k4-s12-16	21	2	3	4	392.78	392.78	392.78	13	0.00
E-n33-k4-s1-9	32	2	3	4	730.16	730.16	730.16	79	0.00
E-n33-k4-s2-13	32	2	3	4	714.63	714.63	714.63	38	0.00
E-n33-k4-s3-17	32	2	3	4	707.48	707.48	707.48	24	0.00
E-n33-k4-s4-5	32	2	3	4	778.74	778.74	779.12	28	0.00
E-n33-k4-s7-25	32	2	3	4	756.85	756.85	756.85	42	0.00
E-n33-k4-s14-22	32	2	3	4	779.05	779.05	779.05	21	0.00
Set 2b									
Set2b_E-n51-k5-s2-4-17-46	50	4	4	5	530.76	530.76	530.76	54	0.00
Set2b_E-n51-k5-s2-17	50	2	3	5	597.49	597.49	597.49	61	0.00
Set2b_E-n51-k5-s4-46	50	2	3	5	530.76	530.76	530.76	58	0.00
Set2b_E-n51-k5-s6-12	50	2	3	5	554.81	554.81	555.03	56	0.00
Set2b_E-n51-k5-s6-12-32-37	50	4	4	5	531.92	531.92	531.92	58	0.00
Set2b_E-n51-k5-s11-19	50	2	3	5	581.64	581.64	581.64	107	0.00
Set2b_E-n51-k5-s11-19-27-47	50	4	4	5	527.63	527.63	527.63	59	0.00
Set2b_E-n51-k5-s27-47	50	2	3	5	538.22	538.22	538.22	193	0.00
Set2b_E-n51-k5-s32-37	50	2	3	5	552.28	552.28	552.28	70	0.00
Set 2c									
Set2c_E-n51-k5-s2-4-17-46	50	4	4	5	601.39	601.39	601.39	58	0.00
Set2c_E-n51-k5-s2-17	50	2	3	5	601.39	601.39	601.39	59	0.00
Set2c_E-n51-k5-s4-46	50	2	3	5	702.33	702.33	702.33	70	0.00
Set2c_E-n51-k5-s6-12	50	2	3	5	567.42	567.42	567.42	66	0.00
Set2c_E-n51-k5-s6-12-32-37	50	4	4	5	567.42	567.42	567.42	60	0.00
Set2c_E-n51-k5-s11-19	50	2	3	5	617.42	617.42	617.42	60	0.00
Set2c_E-n51-k5-s11-19-27-47	50	4	4	5	530.76	530.76	530.76	55	0.00
Set2c_E-n51-k5-s27-47	50	2	3	5	530.76	530.76	530.76	64	0.00
Set2c_E-n51-k5-s32-37	50	2	3	5	752.59	752.59	752.59	73	0.00
Average					**578.27**	**578.27**	**578.29**	**55**	**0.00**

Table A4
Results for set 3 of 2E-VRP benchmark instances.

Instance	$\|V^C\|$	$\|V^S\|$	$\|V^T\|$	m^2	BKS	Best	Average	t(s)	Gap to BKS
Set 3a									
Set3_E-n22-k4-s13-14	21	2	3	4	526.15	526.15	526.15	11	0.00
Set3_E-n22-k4-s13-16	21	2	3	4	521.09	521.09	521.09	18	0.00
Set3_E-n22-k4-s13-17	21	2	3	4	496.38	496.38	496.38	25	0.00
Set3_E-n22-k4-s14-19	21	2	3	4	498.8	498.8	498.80	11	0.00
Set3_E-n22-k4-s17-19	21	2	3	4	512.81	512.81	512.81	13	0.00
Set3_E-n22-k4-s19-21	21	2	3	4	520.42	520.42	520.42	43	0.00
Set3_E-n33-k4-s16-22	32	2	3	4	672.17	672.17	672.17	24	0.00
Set3_E-n33-k4-s16-24	32	2	3	4	666.02	666.02	666.02	26	0.00
Set3_E-n33-k4-s19-26	32	2	3	4	680.37	680.37	680.37	21	0.00
Set3_E-n33-k4-s22-26	32	2	3	4	680.37	680.37	680.37	87	0.00
Set3_E-n33-k4-s24-28	32	2	3	4	670.43	670.43	670.43	40	0.00
Set3_E-n33-k4-s25-28	32	2	3	4	650.58	650.58	650.58	28	0.00
Set 3b									
Set3_E-n51-k5-s12-18	50	2	3	5	690.59	690.59	695.12	59	0.00
Set3_E-n51-k5-s12-41	50	2	3	5	683.05	683.05	694.68	110	0.00
Set3_E-n51-k5-s12-43	50	2	3	5	710.41	710.41	710.41	99	0.00
Set3_E-n51-k5-s39-41	50	2	3	5	728.54	728.54	728.54	67	0.00
Set3_E-n51-k5-s40-41	50	2	3	5	723.75	723.75	726.50	199	0.00
Set3_E-n51-k5-s40-43					752.15	752.15	754.08	69	0.00
Set 3c									
Set3_E-n51-k5-s13-19	50	2	3	5	560.73	560.73	563.53	56	0.00
Set3_E-n51-k5-s13-42	50	2	3	5	564.45	564.45	564.45	65	0.00
Set3_E-n51-k5-s13-44	50	2	3	5	564.45	564.45	564.45	59	0.00
Set3_E-n51-k5-s40-42	50	2	3	5	746.31	746.31	751.36	68	0.00
Set3_E-n51-k5-s41-42	50	2	3	5	771.56	771.56	771.56	76	0.00
Set3_E-n51-k5-s41-44	50	2	3	5	802.91	808.78	812.61	72	0.73
Average					**641.44**	**641.68**	**643.04**	**56**	**0.03**

Table A5
Results for set 4a of 2E-VRP benchmark instances.

Instance	$\|V^C\|$	$\|V^S\|$	$\|V^T\|$	m^2	BKS	Best	Average	t(s)	Gap to BKS
1	50	2	3	6	1569.42	1569.42	1573.70	236	0.00
2	50	2	3	6	1438.32	1438.32	1455.92	175	0.00
3	50	2	3	6	1570.43	1570.43	1570.43	280	0.00
4	50	2	3	6	1424.04	1424.04	1433.19	217	0.00
5	50	2	3	6	2193.52	2194.11	2199.72	298	0.03
6	50	2	3	6	1279.89	1279.89	1279.89	94	0.00
7	50	2	3	6	1458.60	1458.60	1458.60	152	0.00
8	50	2	3	6	1363.76	1364.21	1372.61	332	0.03
9	50	2	3	6	1450.25	1450.25	1450.25	197	0.00
10	50	2	3	6	1407.65	1409.83	1414.04	282	0.15
11	50	2	3	6	2047.43	2059.38	2061.76	306	0.58
12	50	2	3	6	1209.46	1209.46	1215.28	194	0.00
13	50	2	3	6	1481.80	1481.80	1486.67	124	0.00
14	50	2	3	6	1393.64	1393.64	1393.64	123	0.00
15	50	2	3	6	1489.92	1489.92	1491.02	305	0.00
16	50	2	3	6	1389.20	1389.20	1408.67	78	0.00
17	50	2	3	6	2088.48	2089.96	2094.91	208	0.07
18	50	2	3	6	1227.68	1227.68	1227.68	390	0.00
19	50	3	3	6	1564.66	1564.66	1565.14	193	0.00
20	50	3	3	6	1272.98	1272.98	1272.98	47	0.00
21	50	3	3	6	1577.82	1577.82	1578.32	61	0.00
22	50	3	3	6	1281.83	1281.83	1281.83	50	0.00
23	50	3	3	6	1807.35	1807.35	1807.86	86	0.00
24	50	3	3	6	1282.69	1282.69	1282.70	149	0.00
25	50	3	3	6	1522.40	1534.18	1534.18	193	0.77
26	50	3	3	6	1167.47	1167.47	1167.47	46	0.00
27	50	3	3	6	1481.56	1490.07	1502.83	61	0.57
28	50	3	3	6	1210.46	1210.46	1210.46	49	0.00
29	50	3	3	6	1722.00	1729.04	1732.03	62	0.41
30	50	3	3	6	1211.63	1211.63	1211.63	50	0.00
31	50	3	3	6	1490.32	1490.32	1491.39	63	0.00
32	50	3	3	6	1199.05	1199.05	1199.05	57	0.00
33	50	3	3	6	1508.32	1511.02	1513.09	62	0.18
34	50	3	3	6	1233.96	1233.96	1233.96	51	0.00
35	50	3	3	6	1718.42	1718.42	1718.42	104	0.00
36	50	3	3	6	1228.95	1228.95	1228.95	55	0.00
37	50	5	3	6	1528.73	1528.73	1528.81	190	0.00
38	50	5	3	6	1169.20	1172.13	1173.78	53	0.25
39	50	5	3	6	1520.92	1520.92	1520.92	91	0.00
40	50	5	3	6	1199.42	1199.42	1199.42	49	0.00
41	50	5	3	6	1667.96	1667.96	1668.38	107	0.00
42	50	5	3	6	1194.54	1194.54	1195.84	46	0.00
43	50	5	3	6	1439.67	1439.67	1439.67	58	0.00
44	50	5	3	6	1045.14	1045.14	1045.68	45	0.00
45	50	5	3	6	1450.95	1469.32	1469.32	58	1.27
46	50	5	3	6	1088.79	1088.79	1088.79	42	0.00
47	50	5	3	6	1587.29	1587.29	1598.23	63	0.00
48	50	5	3	6	1082.21	1082.21	1082.21	40	0.00
49	50	5	3	6	1434.88	1434.88	1434.88	107	0.00
50	50	5	3	6	1083.16	1083.16	1089.20	48	0.00
51	50	5	3	6	1398.03	1398.03	1398.04	54	0.00
52	50	5	3	6	1125.69	1125.69	1125.69	84	0.00
53	50	5	3	6	1567.79	1567.79	1568.41	330	0.00
54	50	5	3	6	1127.66	1127.66	1127.91	47	0.00
Average					**1419.95**	**1421.21**	**1423.62**	**129**	**0.08**

Table A6
Results for set 4b of 2E-VRP benchmark instances.

Instance	$\|V^C\|$	$\|V^S\|$	$\|V^T\|$	m^2	BKS	Best	Average	t(s)	Gap to BKS
1	50	2	3	6	1569.42	1569.42	1574.29	80	0.00
2	50	2	3	6	1438.32	1445.05	1445.73	62	0.47
3	50	2	3	6	1570.43	1570.43	1571.10	93	0.00
4	50	2	3	6	1424.04	1424.04	1437.95	116	0.00
5	50	2	3	6	2193.52	2199.77	2199.77	87	0.28
6	50	2	3	6	1279.89	1279.89	1279.89	63	0.00
7	50	2	3	6	1408.58	1408.58	1410.77	79	0.00
8	50	2	3	6	1360.32	1360.32	1360.32	61	0.00
9	50	2	3	6	1403.53	1403.53	1403.53	76	0.00
10	50	2	3	6	1360.54	1360.54	1360.54	58	0.00
11	50	2	3	6	2047.43	2058.08	2061.81	308	0.52
12	50	2	3	6	1209.46	1209.46	1209.89	63	0.00
13	50	2	3	6	1450.94	1450.94	1450.94	93	0.00
14	50	2	3	6	1393.64	1393.64	1399.22	240	0.00
15	50	2	3	6	1466.84	1466.84	1466.84	125	0.00
16	50	2	3	6	1387.85	1387.85	1387.85	109	0.00
17	50	2	3	6	2088.48	2089.48	2092.36	88	0.05
18	50	2	3	6	1227.68	1227.68	1227.68	68	0.00
19	50	3	3	6	1546.28	1550	1551.33	264	0.24
20	50	3	3	6	1272.98	1272.98	1272.98	62	0.00
21	50	3	3	6	1577.82	1577.82	1579.17	245	0.00
22	50	3	3	6	1281.83	1281.83	1281.83	60	0.00
23	50	3	3	6	1652.98	1652.98	1652.98	76	0.00
24	50	3	3	6	1282.69	1282.7	1282.70	55	0.00
25	50	3	3	6	1408.58	1408.58	1410.07	291	0.00
26	50	3	3	6	1167.47	1167.47	1167.47	53	0.00
27	50	3	3	6	1444.49	1444.49	1457.75	75	0.00
28	50	3	3	6	1210.46	1210.46	1210.46	59	0.00
29	50	3	3	6	1552.66	1552.66	1552.66	74	0.00
30	50	3	3	6	1211.63	1211.63	1211.63	56	0.00
31	50	3	3	6	1440.85	1440.85	1441.17	78	0.00
32	50	3	3	6	1199.05	1199.05	1199.05	98	0.00
33	50	3	3	6	1478.87	1478.87	1478.87	77	0.00
34	50	3	3	6	1233.96	1233.96	1233.96	63	0.00
35	50	3	3	6	1570.73	1570.73	1570.73	82	0.00
36	50	3	3	6	1228.95	1228.95	1228.95	63	0.00
37	50	5	3	6	1528.73	1528.73	1528.73	64	0.00
38	50	5	3	6	1163.07	1163.07	1163.07	53	0.00
39	50	5	3	6	1520.92	1520.92	1521.64	293	0.00
40	50	5	3	6	1163.04	1163.04	1166.48	54	0.00
41	50	5	3	6	1652.98	1652.98	1652.98	68	0.00
42	50	5	3	6	1190.17	1190.17	1190.57	91	0.00
43	50	5	3	6	1406.1	1406.1	1410.35	71	0.00
44	50	5	3	6	1035.05	1035.05	1035.05	55	0.00
45	50	5	3	6	1401.87	1406.89	1414.41	203	0.36
46	50	5	3	6	1058.1	1058.1	1059.09	49	0.00
47	50	5	3	6	1552.66	1552.66	1553.12	113	0.00
48	50	5	3	6	1074.51	1074.51	1074.51	53	0.00
49	50	5	3	6	1434.88	1434.88	1438.62	106	0.00
50	50	5	3	6	1065.3	1065.3	1065.30	52	0.00
51	50	5	3	6	1387.51	1387.51	1387.51	71	0.00
52	50	5	3	6	1103.47	1106.96	1107.48	50	0.32
53	50	5	3	6	1545.76	1545.76	1546.85	76	0.00
54	50	5	3	6	1113.66	1113.66	1113.66	54	0.00
Average					**1397.06**	**1397.74**	**1399.14**	**98**	**0.04**

Table A7
Results for set 5 of 2E-VRP benchmark instances.

Instance	$\|V^C\|$	$\|V^S\|$	$\|V^T\|$	m^2	BKS	Best	Average	t(s)	Gap to BKS
Set 5									
100-5-1	100	5	5	32	1564.46	1565.44	1569.36	1079	0.06
100-5-1b	100	5	5	15	1108.62	1121.6	1129.49	789	1.17
100-5-2	100	5	5	32	1016.32	1020.54	1021.65	980	0.42
100-5-2b	100	5	5	15	782.25	782.29	782.75	259	0.01
100-5-3	100	5	5	30	1045.29	1045.29	1045.81	790	0.00
100-5-3b	100	5	5	16	828.54	828.99	829.58	216	0.05
100-10-1	100	10	5	35	1124.93	1124.94	1125.20	618	0.00
100-10-1b	100	10	5	18	916.25	924.49	924.49	336	0.90
100-10-2	100	10	5	33	990.58	1012.17	1015.99	461	2.18
100-10-2b	100	10	5	18	768.61	781.28	792.73	201	1.65
100-10-3	100	10	5	32	1043.25	1049.82	1055.52	639	0.63
100-10-3b	100	10	5	17	850.92	859.89	863.17	202	1.05
200-10-1	200	10	5	62	1556.79	1558.74	1565.61	1200	0.13
200-10-1b	200	10	5	30	1187.62	1188.67	1200.13	1067	0.09
200-10-2	200	10	5	63	1365.74	1391.93	1401.82	1201	1.92
200-10-2b	200	10	5	30	1002.85	1007.08	1019.45	1040	0.42
200-10-3	200	10	5	63	1787.73	1814.01	1833.43	1200	1.47
200-10-3b	200	10	5	30	1197.9	1209.27	1214.15	1200	0.95
Average					**1118.81**	**1127.02**	**1132.80**	**749**	**0.73**

Table A8
Results for set 5 of 2E-VRP benchmark instances obtained from ALNS with a time limit of 1200 s.

Instance	$\|V^C\|$	$\|V^S\|$	$\|V^T\|$	m^2	BKS	Best	Average	t(s)	Gap to BKS
Set 5									
100-5-1	100	5	5	32	1564.46	1565.33	1571.33	1200	0.06
100-5-1b	100	5	5	15	1108.62	1117.24	1127.07	1200	0.77
100-5-2	100	5	5	32	1016.32	1020.54	1020.64	1200	0.41
100-5-2b	100	5	5	15	782.25	782.29	782.39	1200	0.01
100-5-3	100	5	5	30	1045.29	1045.29	1045.29	1200	0.00
100-5-3b	100	5	5	16	828.54	828.55	828.77	1200	0.00
100-10-1	100	10	5	35	1124.93	1124.94	1129.94	1200	0.00
100-10-1b	100	10	5	18	916.25	924.49	924.49	1200	0.89
100-10-2	100	10	5	33	990.58	1013.31	1016.75	1200	2.24
100-10-2b	100	10	5	18	768.61	776.04	779.25	1200	0.96
100-10-3	100	10	5	32	1043.25	1049.82	1051.54	1200	0.63
100-10-3b	100	10	5	17	850.92	859.27	861.93	1200	0.97
200-10-1	200	10	5	62	1556.79	1559.19	1568.57	1200	0.15
200-10-1b	200	10	5	30	1187.62	1187.59	1199.87	1200	0.00
200-10-2	200	10	5	63	1365.74	1396.21	1406.71	1200	2.18
200-10-2b	200	10	5	30	1002.85	1006.95	1012.30	1200	0.41
200-10-3	200	10	5	63	1787.73	1810.36	1817.99	1200	1.25
200-10-3b	200	10	5	30	1197.9	1211.01	1218.32	1200	1.08
Average					**1118.81**	**1126.58**	**1131.29**	**1200**	**0.67**

Table A9
The computational environment of the state-of-the-art algorithms.

Algorithm	Computational Environment	P_{score}
HCC-500K	AMD Opteron 275 @2.2 GHz	445
BSHV	Intel E5-2670v2 @ 2.5 GHz	1592
EJRBS – ST	Intel Xeon E5 2680v3 @ 2.5 GHz	1766
EJRBS – 4T	Intel Xeon E5 2680v3 @ 2.5 GHz	1766

Table A10
The average computational time of state-of-the-art algorithms and ALNS for solving 2E-VRP benchmark instances.

Instances	HCC − 500 K	BSHV	EJRBS - ST	EJRBS − 4T	ALNS
Set 2					
2a	51 (14)	60	33 (37)	90 (100)	29 (49)
2b	148 (41)	60	69 (77)	178 (197)	79 (133)
2c			66 (73)	149 (165)	62 (104)
Set 3					
3a	59 (16)	60	32 (35)	85 (94)	28 (47)
3b	161 (45)	60	63 (70)	175 (194)	100 (168)
3c			66 (73)	151 (168)	66 (111)
Set 4					
4a		60			129 (217)
4b	169 (47)	60			97 (163)
Set 5	589* (165)	900	236 (262)	769 (853)	748/ 1200 (1259/2019)

*The reported average computational time is for the 500 K stopping criterion. We do not provide the average computational time for the 5000 K stopping criterion.

Table A11
Computational results for the 2E-VRPTW-CO-OD small instances.

Instance	$\|V^C\|$	$\|V^S\|$	$\|V^L\|$	$\|V^{OD}\|$	CPLEX				ALNS			Gap (Best) %	Gap (Average) %
					UB	LB	Gap$^+$ (%)	CPU time (s)	Obj (Best)	Obj (Average)	Average CPU time (s)		
Set 2a													
E-n22-k4-s6-17*	21	2	2	21	365.143	365.131	0.00	0.922	365.143	365.143	4.682	0.00	0.00
E-n22-k4-s8-14*	21	2	2	21	314.846	314.846	0.00	1.703	314.847	314.916	3.827	0.00	0.02
E-n22-k4-s9-19*	21	2	2	21	412.789	412.789	0.00	4.985	412.789	412.789	6.086	0.00	0.00
E-n22-k4-s10-14*	21	2	2	21	299.474	299.474	0.00	1.687	299.476	300.672	5.778	0.00	0.40
E-n22-k4-s11-12*	21	2	2	21	390.53	390.53	0.00	9.032	390.530	390.530	8.837	0.00	0.00
E-n22-k4-s12-16*	21	2	2	21	333.524	333.524	0.00	2.187	333.526	333.590	6.574	0.00	0.02
E-n33-k4-s1-9*	32	2	2	32	589.95	589.95	0.00	935.844	592.198	594.468	12.554	0.38	0.77
E-n33-k4-s2-13*	32	2	2	32	631.695	631.695	0.00	963.641	631.694	631.694	7.173	0.00	0.00
E-n33-k4-s3-17*	32	2	2	32	608.788	608.788	0.00	2784.92	608.788	608.788	10.275	0.00	0.00
E-n33-k4-s4-5	32	2	2	32	713.792	691.384	3.24	7200	713.791	713.791	8.243	0.00	0.00
E-n33-k4-s7-25*	32	2	2	32	637.585	637.585	0.00	1209.38	637.583	637.584	7.476	0.00	0.00
E-n33-k4-s14-22	32	2	2	32	676.741	672.544	0.62	7200	676.738	676.738	8.085	0.00	0.00
Set 2b													
E-n51-k5-s2-4-17-46	50	4	4	50	480.625	461.579	4.13	7200	474.426	474.455	16.895	−1.29	−1.28
E-n51-k5-s2-17*	50	2	2	50	488.304	488.304	0.00	6280.62	488.306	488.306	19.483	0.00	0.00
E-n51-k5-s4-46	50	2	2	50	510.321	462.611	10.31	7200	504.527	504.527	29.241	−1.14	−1.14
E-n51-k5-s6-12	50	2	2	50	519.569	491.018	5.81	7200	515.257	520.221	23.354	−0.83	0.13
E-n51-k5-s6-12-32-37	50	4	4	50	456.337	405.379	12.57	7200	436.337	436.857	19.409	−4.38	−4.27
E-n51-k5-s11-19	50	2	2	50	543.892	517.454	5.11	7200	542.802	543.455	18.464	−0.20	−0.08
E-n51-k5-s11-19-27-47	50	4	4	50	447.452	406.751	10.01	7200	444.263	444.263	18.942	−0.71	−0.71
E-n51-k5-s27-47	50	2	2	50	461.705	436.685	5.73	7200	457.378	460.155	19.809	−0.94	−0.34
E-n51-k5-s32-37	50	2	2	50	454.805	435.334	4.47	7200	450.223	452.081	16.738	−1.01	−0.60
Set 2c													
E-n51-k5-s2-4-17-46	50	4	4	50	556	497.245	11.82	7200	538.268	539.285	18.457	−3.19	−3.01
E-n51-k5-s2-17	50	2	2	50	580.361	537.28	8.02	7200	580.324	580.324	21.354	−0.01	−0.01
E-n51-k5-s4-46	50	2	2	50	669.015	611.539	9.40	7200	661.224	670.604	38.411	−1.16	0.24
E-n51-k5-s6-12	50	2	2	50	521.929	514.255	1.49	7200	521.935	527.501	122.169	0.00	1.07
E-n51-k5-s6-12-32-37	50	4	4	50	495.515	453.456	9.28	7200	486.338	493.738	18.211	−1.85	−0.36
E-n51-k5-s11-19	50	2	2	50	517.249	482.958	7.10	7200	510.097	510.097	19.061	−1.38	−1.38
E-n51-k5-s11-19-27-47	50	4	4	50	513.52	407.902	25.89	7200	458.780	460.321	17.485	−10.66	−10.36
E-n51-k5-s27-47	50	2	2	50	461.802	437.563	5.54	7200	455.011	455.011	16.693	−1.47	−1.47
E-n51-k5-s32-37	50	2	2	50	801.475	686.721	16.71	7200	736.259	736.259	25.679	−8.14	−8.14
Set 3													
E-n22-k4-s13-14*	21	2	2	21	473.404	473.404	0.00	7.515	473.401	474.227	5.431	0.00	0.17
E-n22-k4-s13-16*	21	2	2	21	492.774	492.774	0.00	22.266	492.773	492.773	5.862	0.00	0.00
E-n22-k4-s13-17*	21	2	2	21	439.181	439.181	0.00	52.328	439.664	439.664	5.147	0.11	0.11
E-n22-k4-s14-19*	21	2	2	21	414.53	414.53	0.00	76.219	414.530	414.530	4.923	0.00	0.00
E-n22-k4-s17-19*	21	2	2	21	447.544	447.544	0.00	110.297	447.545	447.545	6.204	0.00	0.00
E-n22-k4-s19-21*	21	2	2	21	494.252	494.252	0.00	158.203	495.682	495.682	5.688	0.29	0.29

(*continued on next page*)

Table A11 (*continued*)

Instance	$\|V^C\|$	$\|V^S\|$	$\|V^L\|$	$\|V^{OD}\|$	CPLEX				ALNS			Gap (Best) %	Gap (Average) %
					UB	LB	Gap$^+$ (%)	CPU time (s)	Obj (Best)	Obj (Average)	Average CPU time (s)		
E-n33-k4-s16-22*	32	2	2	32	531.457	531.457	0.00	991.125	531.455	541.046	12.526	0.00	1.80
E-n33-k4-s16-24	32	2	2	32	620.337	566.662	9.47	7200	604.646	608.025	15.622	−2.53	−1.98
E-n33-k4-s19-26*	32	2	2	32	510.421	510.421	0.00	4552.69	510.418	510.418	10.596	0.00	0.00
E-n33-k4-s22-26	32	2	2	32	552.383	539.277	2.43	7200	552.385	552.385	10.966	0.00	0.00
E-n33-k4-s24-28	32	2	2	32	632.38	585.905	7.93	7200	626.056	627.133	74.995	−1.00	−0.83
E-n33-k4-s25-28	32	2	2	32	596.472	563.508	5.85	7200	596.471	602.113	32.801	0.00	0.95
E-n51-k5-s12-18	50	2	2	50	611.934	570.562	7.25	7200	608.092	620.919	55.756	−0.63	1.47
E-n51-k5-s12-41	50	2	2	50	604.088	549.811	9.87	7200	592.503	592.503	28.997	−1.92	−1.92
E-n51-k5-s12-43	50	2	2	50	640.138	621.24	3.04	7200	638.189	638.190	55.576	−0.30	−0.30
E-n51-k5-s13-19	50	2	2	50	521.067	507.595	2.65	7200	521.067	523.683	30.230	0.00	0.50
E-n51-k5-s13-42	50	2	2	50	531.63	497.806	6.79	7200	519.480	519.480	23.568	−2.29	−2.29
E-n51-k5-s13-44	50	2	2	50	501.479	475.148	5.54	7200	501.476	501.476	21.549	0.00	0.00
E-n51-k5-s39-41	50	2	2	50	690.676	631.067	9.45	7200	679.230	680.511	64.512	−1.66	−1.47
E-n51-k5-s40-41	50	2	2	50	770.644	662.399	16.34	7200	714.665	715.509	36.852	−7.26	−7.15
E-n51-k5-s40-42	50	2	2	50	665.878	632.587	5.26	7200	657.842	658.345	62.741	−1.21	−1.13
E-n51-k5-s40-43	50	2	2	50	785.952	692.71	13.46	7200	742.832	743.085	42.498	−5.49	−5.45
E-n51-k5-s41-42	50	2	2	50	816.291	704.264	15.91	7200	750.362	750.362	33.635	−8.08	−8.08
E-n51-k5-s41-44	50	2	2	50	695.765	649.548	7.12	7200	692.686	704.253	378.027	−0.44	1.22
Average							**5.289**	**5003.066**			**29.521**	**−1.30**	**−1.01**

* Denotes that the solution obtained by CPLEX is optimal.

$$Gap(Best)\% = \frac{obj(Best)_{ALNS} - obj_{CPLEX}}{obj_{CPLEX}} \times 100\%.$$

$$Gap(Average)\% = \frac{obj(Average)_{ALNS} - obj_{CPLEX}}{obj_{CPLEX}} \times 100\%.$$

Table A12

The results of large instances of ALNS-noLS and ALNS-LS.

Instance	Initial Solution	ALNS-noLS			ALNS-LS		
		Best Obj	Ave Obj	Average CPU time (s)	Best Obj	Ave Obj	Average CPU time (s)
100-5-1	1664.78	1209.150	1246.920	71.661	1169.550	1177.552	86.732
100-5-1b	1487.52	1202.970	1214.368	115.600	**1117.120**	1135.384	136.688
100-5-2	1268.44	862.825	909.455	116.564	**843.145**	878.579	236.568
100-5-2b	1202.25	692.541	751.949	95.204	**681.024**	706.322	503.282
100-5-3	1136.45	887.849	912.379	115.680	**846.848**	855.121	740.789
100-5-3b	1106.38	739.176	778.610	174.820	**715.729**	723.452	534.779
100-10-1	1741.72	983.231	1008.076	126.287	**958.004**	996.842	69.071
100-10-1b	1127.89	822.545	848.809	136.950	**770.380**	816.550	1092.675
100-10-2	1418.67	924.043	928.312	133.386	**911.894**	932.232	932.069
100-10-2b	1285.07	763.713	784.258	105.326	**708.357**	736.662	577.364
100-10-3	1766.82	994.652	1015.371	131.048	**961.557**	967.845	340.558
100-10-3b	1533.06	823.207	872.645	84.586	**789.964**	841.329	353.610
200-10-1	2200.94	1400.240	1458.526	417.333	**1323.240**	1353.784	671.069
200-10-1b	1763.31	1067.870	1074.316	648.825	**910.954**	981.962	1102.318
200-10-2	1897.51	1220.680	1290.766	221.474	**1172.310**	1258.580	923.121
200-10-2b	1459.01	980.422	1086.350	265.494	**962.829**	1015.397	922.114
200-10-3	2007.13	1402.860	1444.154	364.106	1429.140	1468.244	434.796
200-10-3b	1801.92	1168.630	1236.120	815.423	**1131.520**	1140.506	496.969
Average	**1548.27**	**1008.145**	**1047.855**	**229.987**	**966.865**	**999.241**	**564.143**

Bold values represent the best obtained objective value between ALNS-noLS and ALNS-LS.

Table A13
Comparison between ALNS-noLS and ALNS-LS under the 1200-second time limit criterion.

Instance	Initial Solution	ALNS-noLS		ALNS-LS	
		Best Obj	Ave Obj	Best Obj	Ave Obj
100-5-1	1664.78	1209.17	1244.218	**1148.810**	1170.903
100-5-1b	1487.52	1170.05	1196.372	**1117.120**	1128.786
100-5-2	1268.44	855.624	888.280	**835.773**	845.829
100-5-2b	1202.25	**670.203**	702.209	672.466	689.328
100-5-3	1136.45	845.599	880.317	**841.112**	859.762
100-5-3b	1106.38	708.916	748.077	**699.423**	707.671
100-10-1	1741.72	996.552	1021.330	**895.906**	946.661
100-10-1b	1127.89	840.817	879.122	**779.989**	822.159
100-10-2	1418.67	911.477	919.584	**849.860**	876.367
100-10-2b	1285.07	**692.842**	754.548	700.692	747.242
100-10-3	1766.82	983.17	996.704	**943.540**	981.365
100-10-3b	1533.06	830.152	852.064	**804.474**	836.720
200-10-1	2200.94	1393.78	1418.490	**1305.140**	1372.272
200-10-1b	1763.31	1082.69	1129.536	**1061.870**	1098.278
200-10-2	1897.51	1230.12	1369.614	**1154.890**	1179.892
200-10-2b	1459.01	951.016	1020.933	**923.049**	979.953
200-10-3	2007.13	**1389.03**	1475.238	1412.950	1436.696
200-10-3b	1801.92	1162.59	1199.672	**1073.580**	1130.114
Average	**1548.27**	**995.767**	**1038.684**	**956.702**	**989.444**

Bold values represent the best obtained objective value between ALNS-noLS and ALNS-LS.

Table A14
Contribution of each operator on improving the best found solutions for large-scale instances.

ALNS Operator	ALNS-LS (%)	ALNS-noLS (%)
Small Destroy		
Random	16.17	32.38
Worst	23.16	25.83
Probabilistic Worst	29.06	37.65
Related	3.92	0.36
Probabilistic Related	17.53	0.49
Route	4.57	0.20
Neighborhood	5.59	0.47
Large Destroy		
Satellite Removal	48.57	50.20
Satellite Open	1.43	5.71
Satellite Swap	21.43	9.39
Covering Removal	17.14	20.00
Covering Open	4.29	3.67
Covering Swap	7.14	11.02
Small Repair		
Greedy	39.36	54.64
Perturbed Greedy	21.53	43.73
Regret	19.52	0.46
Perturbed Regret	19.59	1.16

Table A15
Literature on 2E-VRP and crowd-shipping delivery problem variants.

Literature	1D/ MD	1E/ 2E	ADO	HDO	TW	OD	AV	Highlight(s)	Objective	Approach
Archetti et al. (2016)	1D	1E	–	✓	–	✓	–	–	Routing costs and total compensation to ODs	Iterated local search
Kafle et al. (2017)	1D	1E	–	✓	✓	✓	–	The existence of relay points, synchronization, pick-up and delivery customers, soft time windows, and a bidding system for OD selection	Routing costs, OD compensation, and penalty costs for violation of time windows	Tabu search-based matheuristic
Huang and Ardiansyah (2019)	1D	1E	–	✓	–	✓	–	The existence of relay points	Routing costs, OD compensation costs, and vehicle fixed costs	Tabu search
Macrina et al. (2020)	1D	1E	–	✓	✓	✓	–	The existence of transshipment nodes and synchronization	Routing costs and total compensation to ODs	Variable neighborhood search
Orenstein et al. (2019)	1D	1E	✓	–	–	–	–	Route duration, possibility of postponing the delivery of some parcels to subsequent shifts, multiple products, and multi-size parcel lockers	Routing costs, undelivered parcels penalty costs, and vehicle fixed costs	Petal heuristic and Tabu search
Jiang et al. (2019)	1D	1E	✓	✓	✓	–	–	–	Routing costs, connection costs, and parcel lockers' opening costs	General variable neighborhood search
Perboli et al. (2011)	1D	2E	–	✓	–	–	–	–	Routing costs and handling operations costs	Matheuristics
Anderluh et al. (2017)	1D	2E	–	✓	–	–	–	Route duration, exact synchronization, and maximum permitted waiting time at satellites	Routing costs, penalty costs for 1st-echelon vehicles crossing the inner area, and vehicle fixed costs	Greedy randomized adaptive search with path relinking
Grangier et al. (2016)	1D	2E	–	✓	✓	–	–	Exact synchronization and multiple trips at the second level	Number of fleets (first objective) and routing costs (second objective)	Adaptive large neighborhood Search
Li et al. (2020)	1D	2E	–	✓	✓	–	✓	A van (1st-echelon vehicle) can carry several UAVs and route duration	Routing costs, waiting costs, and UAV operating costs	Adaptive large neighborhood search
Dellaert et al. (2019)	1D & MD	2E	–	✓	✓	–	–	Synchronization	Routing costs and vehicle fixed costs	Branch-and-price algorithms
Li et al. (2016)	1D	2E	–	✓	–	–	–	Route duration and exact synchronization	Routing costs, Handling operations costs and waiting costs	Two-stage heuristic algorithm
Liu et al. (2018)	1D	2E	–	✓	–	–	–	Grouping constraint	Routing costs and handling operations costs	Branch-and-cut algorithm
Soysal et al. (2015)	1D	2E	–	✓	–	–	–	Time-dependent speed for second-echelon vehicles	Fuel costs, driver costs, and handling operations costs	Off-shelf optimization package (CPLEX)
Wang et al. (2017)	1D	2E	–	✓	–	–	–	Time-dependent speed for second-echelon vehicles	Fuel costs, driver costs, and handling operations costs	VNS-based Matheuristic
Yu et al. (2020a)	1D	2E	–	✓	✓	–	✓	An LV (1st-echelon vehicle) can carry several SAVs and route duration	Number of LVs (first objective) and routing costs (second objective)	Hybrid adaptive large neighborhood search and iterated local search
Zhou et al. (2018)	MD	2E	✓	✓	–	–	–	Route duration	Routing costs, handling operations costs, and connection costs	Hybrid genetic algorithm
Enthoven et al. (2020)	1D	2E	✓	✓	–	–	–	–	Routing costs and connection costs	Adaptive large neighborhood search
This work	1D	2E	✓	✓	✓	✓	–	Synchronization	Routing costs, connection costs, and OD compensation costs	Adaptive large neighborhood search

Abbreviati ons: 1D: one depot; MD: multi-depot; 1E: One-echelon; 2E: Two-echelon; ADO: Alternative delivery option; HDO: Home delivery option; TW: Time windows; OD: Occasional driver; AV: Autonomous vehicle; UAV: Unmanned aerial vehicle; SAV: Smart fully automated ground vehicle.

References

Ali, O., Côté, J.-F., Coelho, L.C., 2020. Models and algorithms for the delivery and installation routing problem. Eur. J. Oper. Res.

Anderluh, A., Hemmelmayr, V.C., Nolz, P.C., 2017. Synchronizing vans and cargo bikes in a city distribution network. CEJOR 25 (2), 345–376.

Archetti, C., Savelsbergh, M., Speranza, M.G., 2016. The vehicle routing problem with occasional drivers. Eur. J. Oper. Res. 254 (2), 472–480.

Arnold, F., Cardenas, I., Sörensen, K., Dewulf, W., 2018. Simulation of B2C e-commerce distribution in Antwerp using cargo bikes and delivery points. Eur. Transport Res. Rev. 10 (1), 2.

Boccia, M., Crainic, T.G., Sforza, A., Sterle, C., 2010. A metaheuristic for a two echelon location-routing problem. In: International Symposium on Experimental Algorithms. Springer, pp. 288–301.

Boysen, N., Fedtke, S., Schwerdfeger, S., 2020. Last-mile delivery concepts: a survey from an operational research perspective. OR Spectrum 1–58.

Breunig, U., Schmid, V., Hartl, R.F., Vidal, T., 2016. A large neighbourhood based heuristic for two-echelon routing problems. Comput. Oper. Res. 76, 208–225.

Dellaert, N., Dashty Saridarq, F., Van Woensel, T., Crainic, T.G., 2019. Branch-and-price–based algorithms for the two-echelon vehicle routing problem with time windows. Transport. Sci. 53 (2), 463–479.

Demir, E., Bektaş, T., Laporte, G., 2012. An adaptive large neighborhood search heuristic for the pollution-routing problem. Eur. J. Oper. Res. 223 (2), 346–359.

Deutsch, Y., Golany, B., 2018. A parcel locker network as a solution to the logistics last mile problem. Int. J. Prod. Res. 56 (1–2), 251–261.

Drexl, M., Schneider, M., 2015. A survey of variants and extensions of the location-routing problem. Eur. J. Oper. Res. 241 (2), 283–308.

Enthoven, D.L., Jargalsaikhan, B., Roodbergen, K.J., uit het Broek, M.A., Schrotenboer, A.H., 2020. The two-echelon vehicle routing problem with covering options: City logistics with cargo bikes and parcel lockers. Comput. Oper. Res. 118, 104919.

Gonzalez-Feliu, J., Perboli, G., Tadei, R., Vigo, D., 2008. The two-echelon capacitated vehicle routing problem.

Grangier, P., Gendreau, M., Lehuédé, F., Rousseau, L.-M., 2016. An adaptive large neighborhood search for the two-echelon multiple-trip vehicle routing problem with satellite synchronization. Eur. J. Oper. Res. 254 (1), 80–91.

Guerrero-Lorente, J., Gabor, A.F., Ponce-Cueto, E., 2020. Omnichannel logistics network design with integrated customer preference for deliveries and returns. Comput. Ind. Eng. 144, 106433.

Guerrero-Lorente, J., Ponce-Cueto, E., Blanco, E.E., 2017. A model that integrates direct and reverse flows in omnichannel logistics networks. In: Engineering Systems and Networks. Springer, pp. 89–97.

Gunawan, A., Widjaja, A.T., Vansteenwegen, P., Yu, V.F., 2020. Adaptive large neighborhood search for vehicle routing problem with cross-docking. In: 2020 IEEE Congress on Evolutionary Computation (CEC). IEEE, pp. 1–8.

Hemmelmayr, V.C., Cordeau, J.-F., Crainic, T.G., 2012. An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics. Comput. Oper. Res. 39 (12), 3215–3228.

Huang, K., Ardiansyah, M.N., 2019. A decision model for last-mile delivery planning with crowdsourcing integration. Comput. Ind. Eng. 135, 898–912.

Janjevic, M., Winkenbach, M., Merchán, D., 2019. Integrating collection-and-delivery points in the strategic design of urban last-mile e-commerce distribution networks. Transport. Res. Part E: Logist. Transport. Review 131, 37–67.

Jiang, L., Dhiaf, M., Dong, J., Liang, C., Zhao, S., 2019. A traveling salesman problem with time windows for the last mile delivery in online shopping. Int. J. Prod. Res. 1–12.

Kafle, N., Zou, B., Lin, J., 2017. Design and modeling of a crowdsource-enabled system for urban parcel relay and delivery. Transport. Res. Part B: Methodol. 99, 62–82.

Li, H., Wang, H., Chen, J., Bai, M., 2020. Two-echelon vehicle routing problem with time windows and mobile satellites. Transport. Res. Part B: Methodol. 138, 179–201.

Li, H., Zhang, L., Lv, T., Chang, X., 2016. The two-echelon time-constrained vehicle routing problem in linehaul-delivery systems. Transport. Res. Part B: Methodol. 94, 169–188.

Liu, T., Luo, Z., Qin, H., Lim, A., 2018. A branch-and-cut algorithm for the two-echelon capacitated vehicle routing problem with grouping constraints. Eur. J. Oper. Res. 266 (2), 487–497.

Macrina, G., Pugliese, L.D.P., Guerriero, F., Laganà, D., 2017. The vehicle routing problem with occasional drivers and time windows. In: International Conference on Optimization and Decision Science. Springer, pp. 577–587.

Macrina, G., Pugliese, L.D.P., Guerriero, F., Laporte, G., 2020. Crowd-shipping with time windows and transshipment nodes. Comput. Oper. Res. 113, 104806.

Mara, S.T.W., Kuo, R., Asih, A.M.S., 2021. Location-routing problem: a classification of recent research. Int. Trans. Oper. Res.

Nagata, Y., Bräysy, O., Dullaert, W., 2010. A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows. Comput. Oper. Res. 37 (4), 724–737.

Orenstein, I., Raviv, T., Sadan, E., 2019. Flexible parcel delivery to automated parcel lockers: models, solution methods and analysis. EURO J. Transport. Logist. 8 (5), 683–711.

PassMark Software, 2021. Professional CPU benchmarks.

Perboli, G., Tadei, R., 2010. New families of valid inequalities for the two-echelon vehicle routing problem. Electronic Notes Discr. Mathe. 36, 639–646.

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

Pisinger, D., Ropke, S., 2007. A general heuristic for vehicle routing problems. Comput. Oper. Res. 34 (8), 2403–2435.

Prodhon, C., Prins, C., 2014. A survey of recent research on location-routing problems. Eur. J. Oper. Res. 238 (1), 1–17.

Rai, H.B., Verlinde, S., Merckx, J., Macharis, C., 2017. Crowd logistics: an opportunity for more sustainable urban freight transport? Eur. Transport Res. Rev. 9 (3), 39.

Rohmer, S., Gendron, B., 2020. A Guide to Parcel Lockers in Last Mile Distribution–Highlighting Challenges and Opportunities from an OR Perspective.

Salhi, S., Nagy, G., 1999. A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling. J. Oper. Res. Soc. 50 (10), 1034–1042.

Salhi, S., Rand, G.K., 1989. The effect of ignoring routes when locating depots. Eur. J. Oper. Res. 39 (2), 150–156.

Sampaio, A., Savelsbergh, M., Veelenturf, L., Van Woensel, T., 2019. Crowd-based city logistics. In: Sustainable Transportation and Smart Logistics. Elsevier, pp. 381–400.

Sampaio, A., Savelsbergh, M., Veelenturf, L.P., Van Woensel, T., 2020. Delivery systems with crowd-sourced drivers: A pickup and delivery problem with transfers. Networks 76 (2), 232–255.

Savelsbergh, M., Van Woensel, T., 2016. 50th anniversary invited article—city logistics: Challenges and opportunities. Transport. Sci. 50 (2), 579–590.

Schliwa, G., Armitage, R., Aziz, S., Evans, J., Rhoades, J., 2015. Sustainable city logistics—Making cargo cycles viable for urban freight transport. Res. Transport. Bus. Manage. 15, 50–57.

Schneider, M., Drexl, M., 2017. A survey of the standard location-routing problem. Ann. Oper. Res. 259 (1), 389–414.

Schneider, M., Sand, B., Stenger, A., 2013. A note on the time travel approach for handling time windows in vehicle routing problems. Comput. Oper. Res. 40 (10), 2564–2568.

Simoni, M.D., Marcucci, E., Gatta, V., Claudel, C.G., 2019. Potential last-mile impacts of crowdshipping services: a simulation-based evaluation. Transportation 1–22.

Smilowitz, K.R., Daganzo, C.F., 2007. Continuum approximation techniques for the design of integrated package distribution systems. Networks: An Int. J. 50 (3), 183–196.

Snoeck, A., Winkenbach, M., 2020. The value of physical distribution flexibility in serving dense and uncertain urban markets. Transport. Res. Part A: Policy Practice 136, 151–177.

Soysal, M., Bloemhof-Ruwaard, J.M., Bektaş, T., 2015. The time-dependent two-echelon capacitated vehicle routing problem with environmental considerations. Int. J. Prod. Econ. 164, 366–378.

Sun, P., Veelenturf, L.P., Hewitt, M., Van Woensel, T., 2020. Adaptive large neighborhood search for the time-dependent profitable pickup and delivery problem with time windows. Transport. Res. Part E: Logist. Transport. Rev. 138, 101942.

Van Duin, J., Wiegmans, B., van Arem, B., van Amstel, Y., 2020. From home delivery to parcel lockers: A case study in Amsterdam. Transp. Res. Procedia 46, 37–44.

Vidal, T., Crainic, T.G., Gendreau, M., Prins, C., 2013. Heuristics for multi-attribute vehicle routing problems: A survey and synthesis. Eur. J. Oper. Res. 231 (1), 1–21.
Wang, K., Shao, Y., Zhou, W., 2017. Matheuristic for a two-echelon capacitated vehicle routing problem with environmental considerations in city logistics service. Transport. Res. Part D: Transport Environ. 57, 262–276.
Winkenbach, M., Kleindorfer, P.R., Spinler, S., 2016. Enabling urban logistics services at La Poste through multi-echelon location-routing. Transport. Sci. 50 (2), 520–540.
Yu, S., Puchinger, J., Sun, S., 2020a. Two-echelon urban deliveries using autonomous vehicles. Transport. Res. Part E: Logist. Transport. Rev. 141, 102018.
Yu, V.F., Jewpanya, P., Redi, A.P., Tsao, Y.-C., 2021. Adaptive Neighborhood Simulated Annealing for the Heterogeneous Fleet Vehicle Routing Problem with Multiple Cross-docks. Comput. Oper. Res. 105205.
Yu, V.F., Lin, S.-W., Gunawan, A., 2020b. Design of a two-echelon freight distribution system in an urban area considering third-party logistics and loading–unloading zones. Appl. Soft Comput. 97, 106707.
Yu, V.F., Redi, A.P., Halim, C., Jewpanya, P., 2020c. The path cover problem: Formulation and a hybrid metaheuristic. Expert Syst. Appl. 146, 113107.
Zhao, Q., Wang, W., De Souza, R., 2018. A heterogeneous fleet two-echelon capacitated location-routing model for joint delivery arising in city logistics. Int. J. Prod. Res. 56 (15), 5062–5080.
Zhou, L., Baldacci, R., Vigo, D., Wang, X., 2018. A multi-depot two-echelon vehicle routing problem with delivery options arising in the last mile distribution. Eur. J. Oper. Res. 265 (2), 765–778.