Hindawi
Journal of Advanced Transportation
Volume 2022, Article ID 5603956, 23 pages
https://doi.org/10.1155/2022/5603956

Research Article

The Two-Echelon Vehicle Routing Problem with Transshipment Nodes and Occasional Drivers: Formulation and Adaptive Large Neighborhood Search Heuristic

Vincent F. Yu,[1,2] **Minh P. K. Nguyen**,[1] **Kuza Putra**,[1,3] **Aldy Gunawan**,[4] **and I Gusti Bagus Budi Dharma**[3]

[1]*Department of Industrial Management, National Taiwan University of Science and Technology, Taipei, Taiwan*
[2]*Center for Cyber-Physical System Innovation, National Taiwan University of Science and Technology, Taipei, Taiwan*
[3]*Department of Mechanical and Industrial Engineering, Universitas Gadjah Mada, Daerah Istimewa Yogyakarta, Indonesia*
[4]*School of Computing and Information Systems, Singapore Management University, Singapore*

Correspondence should be addressed to Minh P. K. Nguyen; d10901807@gapps.ntust.edu.tw

Received 18 April 2022; Revised 6 December 2022; Accepted 9 December 2022; Published 22 December 2022

Academic Editor: Michele Simoni

This research introduces a new variant of the two-echelon vehicle routing problem (2EVRP) called the two-echelon vehicle routing problem with transshipment nodes and occasional drivers (2EVRP-TN-OD). In addition to city freighters in the second-echelon network, a set of occasional drivers (ODs) is available to serve customers. ODs are the basis of a crowd-shipping system in which crowds with planned trips are willing to take detours to deliver packages in exchange for some compensation. To serve customers, ODs collect the assigned packages at either satellite served by first-echelon trucks or transshipment nodes served by city freighters. We formulate this problem as a mixed-integer nonlinear programming model and develop an adaptive large neighborhood search (ALNS) to solve it. New problem-specific destroy and repair operators and a tailored local search procedure are embedded into ALNS to deal with the problem's unique characteristics. The experiments show that the proposed ALNS effectively solves 2EVRP-TN-OD by outperforming Gurobi in terms of both solution quality and computational time. Moreover, the experiments confirm that employing occasional drivers leads to lower operational costs. Sensitivity analyses on the characteristics of occasional drivers and the impact of transshipment nodes are presented as interesting managerial insights from 2EVRP-TN-OD.

1. Introduction

City logistics has recently become an emerging branch of supply chain management due to rapid population growth in urban areas around the world. Consequently, rising consumer demand to fulfill the needs of this population has become unavoidable, leading to an increasing number of e-commerce industries. In 2019, e-commerce sales globally reached a value of US$3.53 trillion and were projected to grow to US$6.54 trillion in 2022 [1]. This market's sales have led to new challenges in delivery operations. While raising the number of operational vehicles is one direct solution, it results in many freight transportation issues, such as traffic congestion, higher emissions and air pollution, and noise, to mention a few.

Recent technological advancements called crowd-shipping have resulted in a sharing economy-based delivery concept for increasing delivery effectiveness and reducing operational costs [2]. Crowd-shipping makes use of idle resources to perform the delivery task that would otherwise be performed by delivery companies. Several large retailers, such as Walmart and Amazon, have started to develop and implement crowd-shipping platforms [3]. These recent business practices performed by well-known enterprises show that crowd-shipping is growing in popularity as a new and promising delivery system.

In order to improve the living quality in a city, a two-echelon distribution strategy is commonly implemented to reduce the volume of vehicles travelling around the urban area by setting up intermediate facilities called satellites [4]. A satellite represents a secondary facility located nearby the city center, where the freights delivered by municipal trucks from the central depots are transferred and consolidated to smaller vehicles, commonly called city freighters [5, 6]. This strategy leads to an interesting topic in vehicle routing problem variants called the two-echelon vehicle routing problem (2EVRP).

Due to the rise of 2EVRP implementations and the crowd-shipping concept, one of the main aims of this research is to introduce a new variant of 2EVRP called 2EVRP with transshipment nodes and occasional drivers (2EVRP-TN-OD). The first-echelon network connects the central depot with satellites, and the second-echelon network connects satellites with customers and transshipment nodes (TNs). In this research, every occasional driver (OD) has his/her origin and destination nodes, available capacity, and flexibility. ODs are able to utilize their available capacity and collect the assigned packages from either a satellite or a TN. Moreover, every employed OD receives a particular amount of compensation, depending on the total distances travelled. The contributions of this research work are summarized as follows:

(1) Introduce a new variant of 2EVRP called the two-echelon vehicle routing problem with transshipment nodes and occasional drivers (2EVRP-TN-OD).

(2) Formulate a mixed-integer nonlinear programming (MINLP) for 2EVRP-TN-OD.

(3) Develop an adaptive large neighborhood search algorithm and show the effectiveness of the proposed algorithm.

(4) Perform sensitivity analyses on important characteristics related to the crowd-shipping concept to better understand the impact of the integration into the two-echelon distribution network.

The remaining parts of this article are described as follows: Section 2 provides a description of previous related works. Section 3 explains the problem definition and formulation. Section 4 describes the detailed procedure of our developed ALNS. Section 5 shows the generation of benchmark instances, the computational results of ALNS, and the sensitivity analyses related to 2EVRP-TN-OD. Lastly, Section 6 concludes the findings of this work and possible future direction.

2. Literature Review

2.1. Two Echelon Vehicle Routing Problem. Crainic et al. [7] first introduced 2EVRP by considering the setting up of a two-tier distribution network. The model aims to mainly address the day-before-planning problem with several key decisions related to freight assignment to the satellites as well as the routing and scheduling of vehicles that operate in both first- and second-echelon networks. Perboli et al. [8] pioneered the method's development by addressing the basic problem of 2EVRP, where time windows and satellite synchronizations are not considered. This problem is called a two-echelon capacitated vehicle routing problem (2E-CVRP). Recent work has focused on integrating trends and realistic scenarios into the 2EVRP structure. Wang et al. [9] considered the environmental impact of 2EVRP. In addition, Belgin et al. [10] developed an extension of 2EVRP by incorporating not only delivery but also pickup demand.

Due to the rise in the utilization of electric vehicles, several works tackled similar networks by considering electric vehicles [11, 12]. Anderluh et al. [13] explicitly integrated emission minimization by formulating 2EVRP into a multiobjective problem in which both operational and emission costs are considered. In light of rapid technological development, several helpful innovative alternatives have become available, such as parcel lockers, crowd-shipping, delivery robots, and drones. These technology-enabled options led to the fruitful developments of 2EVRP [14–18]. Mühlbauer and Fontaine [19] adopted swap containers for solving 2E-CVRP, where the first echelon's requests are fulfilled by a set of vans and the second echelon's requests are serviced by a set of cargo bicycles or city freighters provided by the logistic companies. Various studies were collected and summarized in the review of 2EVRP variants by Sluijk et al. [20]. It can be seen that the existing literature has rarely considered heterogeneous vehicles, and so far, only Yu et al. [17] have introduced occasional drivers to 2EVRP with covering locations.

2.2. Location-Routing Problem. The location-routing problem (LRP) is an extension of VRP that involves two key decisions: the selection of depots and the routing of vehicles originating from a particular depot [21]. The network setting of LRP is similar to the second-echelon network of 2EVRP. The main differences between LRP and 2EVRP can be found in their objective functions and network structures. Various works have proposed solution methods such as heuristics and exact algorithms [22–26]. Schneider and Löffler [27] constructed a three-phase tree-based search algorithm for solving CLRP. Their numerical analysis showed the effectiveness of the algorithm in solving LRPs. New variants of LRP were also proposed to cope with emerging challenges and trends. Karaoglan et al. [28] set up a model and a heuristic to deal with LRP under simultaneous pickup and delivery to handle the existence of returned products from customers. Yu and Lin [29] tackled LRP under the condition that all operational fleets are supplied by a third-party logistics company. Nowadays, due to the efforts to reduce the number of operating vehicles within cities, LRP has been extended to two-tier distribution networks, namely, 2ELRP [4]. Several recent works have addressed the case of 2ELRP under specific real-world characteristics [30, 31]. Arnold and Sörensen [32] proposed a progressive filtering heuristic adopting the regret Clarke-Wright and knowledge-guided local search heuristic for solving 2ELRP and its special case, the single truck and trailer routing problem with satellite depots (STTRPSD), which have a configuration similar to the settings of 2EVRP.

2.3. Crowdshipping. The trend of providing crowd-shipping models and analyzing the positive impacts of considering this concept gives rise to a new VRP variant. Archetti et al. [33] pioneered this strand of research by introducing the vehicle routing problem with occasional drivers (VRPOD), where deliveries are conducted by two types of fleets: company vehicles and ODs. Macrina et al. [34] introduced three different characteristics to the VRPOD problem: time window constraints, multiple deliveries, and split delivery that can be performed by ODs. Chen et al. [35] proposed a multihop problem where parcels can be delivered using multiple consecutive tours to reach their destination within a designed time window and a maximum rate of detour compared to the drivers' shortest paths.

Based on the definition of crowdsourcing, Sampaio et al. [36] presented a system with a large number of drivers, a uniquely owned depot, and short working shifts when performing the pickup and delivery activities. These drivers do not have a desired destination, and they will return to the depot after delivering the requests. Moreover, they limit the number of transfers to once per request only, which distinguishes their work from the proposed study of Chen et al. [35]. These configurations have turned their problem into the classic pickup and delivery problem with time windows and transfer (PDPTW-T). Voigt and Kuhn [37] constructed their model based on the work of Sampaio et al. [36] with a modification in the shipping fleet team by categorizing them into occasional drivers and regular drivers. In their settings, the latter team will be the main force handling the delivery operation, and the former ones will accept the tasks based on their convenience. To sum up, recent works on the crowd-shipping system consider the existence of transshipment nodes so that ODs have more options for picking up the assigned demand [38–40]. The main advantage of a crowd-shipping system, as mentioned in all the aforementioned works, is savings on operational costs.

2.4. Transshipment Nodes. Transshipment nodes (TNs) are considered important factors when handling the vehicle routing problem with crowd shipping and are usually referred to as intermediate depots [37, 40]. The benefits TNs bring to the context of ODs have been proposed by various scholars. Macrina et al. [40] believed the adoption of these intermediate transfer points would gain interest and relax the barriers for more ODs to performing delivery tasks due to the advantages in the geography context, with a total savings cost of up to 31% compared to the nontransshipment nodes model in VRPOD. This rate will significantly increase in proportion to the number of available TNs and their capacities. In addition, by allowing the ODs to visit the depot, a saving of 6% in the consumed operation costs has been found in their study. Voigt and Kuhn [37] proposed a similar model to our work with the involvement of both regular drivers and ODs in their study and performed some experiments with several scenarios: VRPOD without TNs (T0), VRPOD with random TNs (TR), and VRPOD with fixed assigned TNs (T4). Though there is no significant difference in the outcomes when comparing T0 and TR in scenarios with 100 requests and 100 occasional drivers, other computational results show that the adoption of TNs usually has huge advantages in the consumed cost when compared with the reverse scenarios in their study. Yu et al. [41] also confirmed the benefit to total operation cost with the setup of TNs in the crowd shipping delivery system with four designed scenarios: alternative delivery options are applied/not applied, and TNs are applied/not applied. The mentioned works have proven the effectiveness of TNs in advancing savings in VRP in the crowd-shipping context and are worth considering when working with the crowd-shipping system.

2.5. Main Contributions. Our work contributes to the 2EVRP literature with the adoption of crowdsourcing and heterogeneous vehicles, which cover occasional drivers, in picking up and delivering customers' requests. For a prior contribution, the problem settings of our work and those of Mühlbauer and Fontaine [19] are quite similar; however, we support the delivery activities of the second echelon by integrating a group of ODs. Although 2EVRPTW-OD was introduced by Yu et al. [17], they dealt with the covering locations while we handled the issue of transshipment nodes. Our latter contribution is addressed through the involvement of occasional drivers and city freighters for delivery purposes in our work's second echelon.

To our knowledge, existing studies on the crowd-shipping problem have also not yet dealt with 2EVRP. Both Huang and Ardiansyah [38] and Macrina et al. [40] handled VRP-CS with transfer locations, where regular drivers mainly serve both transshipment nodes and customers, and customers are serviced by both types of drivers: regular and occasional. Those works match our second-echelon work; however, the first-echelon activities are not analyzed in their works. Moreover, our work distinguishes between the depot-satellite delivery trucks and the satellite-transshipment node city freighters, which also have not been analyzed in previous studies. This is our addition to existing studies related to crowd-shipping problems. For more details, our paper refers to regular drivers as city freighters and first-echelon trucks for better discrimination in description and modelling. A table is given in the appendix to summarize and analyze the difference between our work and existing studies.

3. Problem Description

Similar to the classical 2EVRP addressed by Hemmelmayr et al. [24] and Breunig et al. [42], we model 2EVRP-TN-OD where the first-echelon network connects the central depot with satellites and the second-echelon network connects the satellites with customers. In 2EVRP-TN-OD, a set of TNs exists in the second-echelon network, serving as places for transferring goods from second-echelon vehicles (city freighters) to ODs. Figure 1 illustrates a possible solution for 2EVRP-TN-OD.

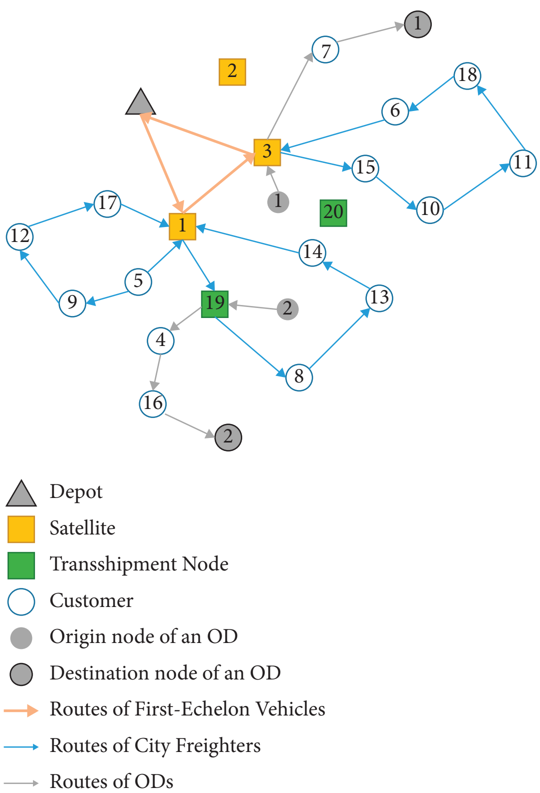


FIGURE 1: An illustration of 2EVRP-TN-OD.

In the first echelon of the distribution network in 2EVRP-TN-OD, there exists a set of homogeneous first-echelon trucks serving satellites. The demands of a satellite are the total demands of the customers assigned to the satellite. In the second echelon, there are two types of fleets for serving customers: a set of homogeneous city freighters and a set of heterogeneous ODs. Each OD has the willingness to serve one or more customers depending on his/her available capacity. The willingness is defined as a maximum extra distance that an OD willingly travels.

Each customer has an amount of demand that must be fulfilled by one visit by a vehicle, either a city freighter or an OD. While no split delivery is allowed in the second echelon, 2EVRP-TN-OD adopts the split delivery of the first echelon of 2EVRP, allowing a particular satellite to be visited by more than one first-echelon truck. The first-echelon trucks start and finish at the depot, while city freighters start to visit customers or TNs from a particular satellite and end at the same satellite. Each OD starts at his/her origin, ends at his/her destination, and is able to collect demands either at a satellite or a TN before serving the assigned customers. In this work, our main objective is to minimize the total operation cost for delivery activities, including: (1) the travel costs of the first-echelon trucks, second-echelon city freighters, and ODs; and (2) the employment costs accounted for by ODs.

An overview of our proposed work is illustrated in Figure 1. There are 2 transshipment nodes, 3 satellites, 5 customer nodes, 1 vehicle serving the first echelon, 2 ODs and 3 city freighters fulfilling requests at the second echelon, and one depot. From the main depot, a large capacity vehicle serves two satellites (1 and 3), and then there are two city freighters that will come to the first satellite to pickup the requests, namely, freighters 1 and 2. At the same time, at the third satellite, an OD departs from origin node 1 and collects the request to deliver to the customer at node 7 before ending at destination node 1. In addition, another freighter 3 visits this satellite to pick up the requests and then deliver them to customers at Nodes 6, 10, 11, 15, and 18. Back at the first satellite, while city freighter 1 will normally deliver requests to customers at nodes 5, 9, 12, and 17, city freighter 2 is assigned to fulfill customers' requests at nodes 8, 13, and 14 in addition to delivering the remaining requests to transshipment node 19 (green square) for later pickup and delivery activities performed by OD 2. This OD, who departs from original location 2, visits transshipment node 19 to collect requests and then continues delivering them to customers at nodes 4 and 16 on his travel route to the respected destination node 2. The other delivery activities performed by other ODs or freighters are similar to those performed by OD2 and freighter 1 in this illustrated example. To sum up, in this illustrated example, not all satellites and transshipment nodes are used for delivering requests to customers; one vehicle is used for servicing requests at satellites in the first echelon, while 2 occasional drivers and 3 city freighters perform the requests' delivery to end-customers at the second echelon through the utilization of one transshipment node out of the two existing ones.

4. A Mixed-Integer Nonlinear Program Formulation

The problem is formally defined on a directed graph $G = (N, A)$, where N represents a set of vertices and A denotes a set of arcs. Here, N consists of a depot $\{0\}$, the set of satellites N_s, the set of customers N_c, the set of TNs N_r, and the set of origin and destination nodes of available ODs, N_O and N_D. There are two sets of arcs in A–that is, $A_1 := \{(i, j)|i, j \in N\{N_c \cup N_r \cup \cup N_O \cup N_D\}, i \neq j\}$ and $A_2 := A\backslash A_1$, consecutively representing the arcs in the first- and second-echelons. For all arcs (i, j) in A, d_{ij} is defined as the distance from vertex i to vertex j, where $t_{\text{ij}} = d_{\text{ij}}$. In addition, several additional sets are defined; i.e., $N_1 = 0 \cup N_s$, $N_2 = N_c \cup N_r \cup N_O \cup N_D$, $N_p = N_s \cup N_r$, and $N_v = N_c \cup N_r$, for the sake of simplicity of the formulation.

Each customer $i \in N_c$ has a nonnegative demand δ_i. A set of K_1 homogeneous first-echelon trucks is available at the depot, each having a maximum capacity of Q_1. In the second-echelon network, there exists a set of K_2 homogeneous city freighters that can be positioned at any existing satellite and a set of K_{OD} ODs. Each OD $k \in K_{\text{OD}}$ has his/her own origin n_k^o stored in N_O and his/her own destination n_k^D stored in N_D. Each city freighter $k \in K_2$ has a maximum capacity of Q_2, while each OD $k \in K_{\text{OD}}$ has a maximum capacity of Q_k^{OD}.

Let Δ_k be the original distance between the origin and destination nodes of an OD $k \in K_{\text{OD}}$. OD k is willing to serve the customers as long as the maximum extra-travelled distance is not greater than $\mu\Delta_k$. The cost of employing an

OD consists of two components: (1) a fixed cost H and (2) a distance-related variable cost ρ_{OD}. In addition, both first-echelon trucks and city freighters have distance-related variable costs c_d. Finally, the following MINLP describes 2EVRP-TN-OD.

4.1. Sets

N_s Set of satellites.

N_r Set of available TNs.

N_c Set of customers.

N_O Set of origin nodes of ODs.

N_D Set of destination nodes of ODs.

K_1 Set of first-echelon trucks.

K_2 Set of city freighters.

K_{OD} Set of ODs.

4.2. Additional Sets

N_1 Set of all first-echelon nodes, $N_1 = \{0\} \cup N_s$

N_2 Set of all second-echelon nodes, $N_2 = N_s \cup N_C \cup N_r$

N_p Set of all ODs' pickup point nodes, $N_p = N_s \cup N_r$

N_v Set of all nodes that are able to be visited by city freighters excluding the satellites, $N_v = N_r \cup N_c$

4.3. Parameters

d_{ij} Travel distance from node i to node j, where $(i, j) \in A$

δ_i Demand of customer i, where $i \in N_c$

Q_i^{OD} Maximum available capacity of OD i, where $i \in K_{\mathrm{OD}}$

Δ_i Distance between origin and destination nodes of OD i, where $i \in K_{\mathrm{OD}}$

Q^1 Capacity of a first-echelon truck.

Q^s Capacity of a satellite.

Q^2 Capacity of a city freighter.

Q^r Capacity of a TN.

H Fixed cost of employing an OD.

ρ_{OD} Compensation per distance unit travelled by an OD.

c_d Variable cost per distance unit for either a first-echelon truck or a city freighter.

μ A multiplier for calculating the extra-travelled distance of an OD.

4.4. Decision Variables

x_{ijk} Binary variables representing the selection of arc (i, j) to be traversed by first-echelon truck $k \in K_1$; 1 if first-echelon truck k traverses through arc (i, j) and 0 otherwise.

y_{ijk} Binary variables representing the selection of arc (i, j) to be traversed by city freighter $k \in K_2$; 1 if city freighter k traverses through arc (i, j) and 0 otherwise.

z_{ijk} Binary variables representing the selection of arc (i, j) to be traversed by OD $k \in K_{OD}$; 1 if OD k traverses through arc (i, j) and 0 otherwise.

α_{pk} Binary variable indicating the assignment of OD k to pickup point $p \in N_p$; 1 if OD k visits pickup point i and 0 otherwise.

β_{ik} Binary variable representing whether customer $i \in N_c$ is the last node visited by OD $k \in K_{\mathrm{OD}}$; 1 if OD k goes to the destination after visiting customer i and 0 otherwise.

σ_k^{OD} Total demands assigned to OD $k \in K_{\mathrm{OD}}$, $\sigma_k^{\mathrm{OD}} \geq 0$

σ_k^2 Total demands assigned to city freighter $k \in K_2$, $\sigma_k^2 \geq 0$

σ_k^1 Total demands assigned to first-echelon truck $k \in K_1$, $\sigma_k^1 \geq 0$

l_i^M Total demands assigned to TN $i \in N_r$, $l_i^M \geq 0$

l_i^S Total demands assigned to satellite $i \in N_S$, $l_i^S \geq 0$

q_{ik}^1 Remaining loads carried by first-echelon truck $k \in K_1$ when visiting node $i \in N_S$, $q_{\mathrm{ik}}^1 \geq 0$

q_{ik}^2 Remaining loads carried by city freighter $k \in K_2$ when visiting node $i \in N_v$, $q_{\mathrm{ik}}^2 \geq 0$

$q_{\mathrm{ik}}^{\mathrm{OD}}$ Remaining loads carried by OD $k \in K_{\mathrm{OD}}$ when visiting node $i \in N_c$, $q_{\mathrm{ik}}^{\mathrm{OD}} \geq 0$

4.5. Objective Function

$$\begin{aligned}\text{Min } Z = &\sum_{k\in K_1}\sum_{i,j\in N_1} c_d d_{\mathrm{ij}} x_{\mathrm{ijk}} + \sum_{k\in K_2}\sum_{i,j\in N_2} c_d d_{\mathrm{ij}} y_{\mathrm{ijk}} \\ &+ \sum_{k\in K_{OD}} \sum_{\substack{i\in\{n_k^o\cup N_2\}\\ j\in\{N_2\cup n_k^D\}}} \rho_{\mathrm{OD}} d_{\mathrm{ij}} z_{\mathrm{ijk}} + \sum_{k\in K_{\mathrm{OD}}}\sum_{p\in N_p} H\alpha_{\mathrm{pk}}. \end{aligned} \tag{1}$$

Subject to

$$\sum_{j\in N_1} x_{0\mathrm{jk}} \leq 1\,\forall k \in K_1, \tag{2}$$

$$\sum_{j\in N_1} x_{j0k} \leq 1\,\forall k \in K_1, \tag{3}$$

$$\sum_{i\in N_1} x_{\mathrm{ijk}} \leq 1\,\forall j \in N_S, \quad \forall k \in K_1, \tag{4}$$

$$\sum_{j\in N_1} x_{\mathrm{ijk}} - \sum_{j\in N_1} x_{\mathrm{jik}} = 0\, i \in N_1, \quad \forall k \in K_1, \tag{5}$$

$$\sum_{k\in K_2}\sum_{j\in N_v} y_{\text{pjk}}\sigma_k^2 + \sum_{k\in K_{\text{OD}}}\alpha_{\text{pk}}\sigma_k^{\text{OD}} \le l_p^S\ p \in N_S, \tag{6}$$

$$\sum_{i\in N_1}\sum_{j\in N_S} x_{\text{ijk}} l_j^S \le \sigma_k^1\ \forall k \in K_1, \tag{7}$$

$$\sigma_k^1 \le Q^1\ \forall k \in K_1, \tag{8}$$

$$\sigma_k^1 \ge q_{\text{jk}}^1 + M\left(1 - x_{0\text{jk}}\right) \forall j \in N_S, \quad \forall k \in K_1, \tag{9}$$

$$q_{\text{ik}}^1 - l_i^S \le q_{\text{jk}}^1 + M\left(1 - x_{\text{ijk}}\right) \forall i \in N_S, \forall j \in N_S, \quad \forall k \in K_1, \tag{10}$$

$$\sum_{j\in N_v} y_{\text{ijk}} \le \sum_{l\in K_1}\sum_{j\in N_1} x_{\text{jil}}\ i \in N_S, \quad \forall k \in K_2, \tag{11}$$

$$\sum_{j\in N_C} z_{\text{ijk}} \le \sum_{l\in K_1}\sum_{j\in N_1} x_{\text{jil}}\ i \in N_S, \quad \forall k \in K_{\text{OD}}. \tag{12}$$

The objective of 2EVRP-TN-OD in (1) is to minimize the total cost of the delivery operation. The first, second, and third terms consecutively address the travelling costs of first-echelon trucks, city freighters, and the employed ODs, while the last one represents the payments for employing these ODs. Constraints (2) to (12) focus on managing the characteristics of the first-echelon network. Constraints (2) to (4) ensure that for each first-echelon truck, every satellite can only be visited at most once. Constraint (5) represents the conservation-of-flow constraint of every first-echelon truck. Constraint (6) defines the total demands assigned to each satellite. Constraint (7) defines the total demands carried by each first-echelon truck, while constraint (8) limits the maximum number of demands that can be carried by each first-echelon truck. Constraints (9) and (10) track the remaining demands carried by each first-echelon truck while visiting a node. Constraint (11) ensures that a city freighter can originate from a satellite if and only if the satellite is served by first-echelon trucks. Similarly, constraint (12) guarantees that each OD can collect demands at a satellite if and only if the satellite is served by at least one first-echelon truck.

$$\sum_{k\in K_2}\sum_{i\in N_2} y_{\text{ijk}} + \sum_{k\in K_{\text{OD}}}\sum_{i\in N_2} z_{\text{ijk}} = 1\ j \in N_C, \tag{13}$$

$$\alpha_{\text{pk}} \le \sum_{l\in K_2}\sum_{i\in N_2} y_{\text{ipl}}\ p \in N_r, \quad k \in K_{\text{OD}}, \tag{14}$$

$$\alpha_{\text{pk}} \le \sum_{l\in K_2}\sum_{i\in N_1} x_{\text{ipl}}\ p \in N_S, \quad k \in K_{\text{OD}}, \tag{15}$$

$$\sum_{p\in N_p} \alpha_{\text{pk}} \le 1\ k \in K_{\text{OD}}, \tag{16}$$

$$\sum_{i\in N_C} \beta_{\text{ik}} = \sum_{p\in N_p} \alpha_{\text{pk}}\ k \in K_{\text{OD}}, \tag{17}$$

$$\sum_{i\in N_C} z_{\text{pik}} = \alpha_{\text{pk}}\ p \in N_p, \quad k \in K_{\text{OD}}, \tag{18}$$

$$\sum_{i\in n_k^o} z_{\text{ipk}} = \alpha_{\text{pk}}\ p \in N_p, \quad k \in K_{\text{OD}}, \tag{19}$$

$$\sum_{j\in n_k^D} z_{\text{ijk}} = \beta_{\text{ik}}\ i \in N_C, \quad k \in K_{\text{OD}}, \tag{20}$$

$$\sum_{j\in N_p\cup N_C} z_{\text{jik}} - \sum_{j\in N_C\cup n_k^D} z_{\text{ijk}} = 0\pi\ i \in N_C, \quad k \in K_{\text{OD}}, \tag{21}$$

$$\sigma_k^{\text{OD}} \le q_{\text{jk}}^{\text{OD}} + M\left(1 - z_{\text{pjk}}\right) j \in N_C,\ p \in N_p, \quad k \in K_{\text{OD}}, \tag{22}$$

$$q_{\text{ik}}^{\text{OD}} - \delta_i \le q_{\text{jk}}^{\text{OD}} + M\left(1 - z_{\text{ijk}}\right) i,\ j \in N_C, \quad k \in K_{\text{OD}}, \tag{23}$$

$$\sum_{i\in N_2}\sum_{j\in N_C} z_{\text{ijk}}\delta_j \le \sigma_k^{\text{OD}}\ k \in K_{\text{OD}}, \tag{24}$$

$$\sigma_k^{\text{OD}} \le Q_k^{\text{OD}}\ k \in K_{\text{OD}}, \tag{25}$$

$$\begin{aligned}\sum_{p\in N_p} d_{n_k^o p}\alpha_{\text{pk}} &+ \sum_{i\in N_p\cup N_C}\sum_{j\in N_C} d_{\text{ij}} z_{\text{ijk}} \\ &+ \sum_{j\in N_C} d_{\text{j}n_k^D}\beta_{\text{jk}} \le (1+\mu)\Delta_k\ k \in K_{\text{OD}}.\end{aligned} \tag{26}$$

Constraint (13) ensures that either an OD or a city freighter must serve each customer. Constraints (14) to (26) focus on managing the characteristics of ODs' routes. In particular, constraints (14) to (15) define the assignment of each OD to a pickup point, either a satellite or a TN. Constraint (16) ensures that each OD can only visit one pickup point to take the assigned demands. Constraint (17) ensures that each OD needs to visit a customer before reaching his/her destination if the OD is assigned a demands. Constraints (18) to (21) ensure the flow-in and flow-out of routes for an OD. Constraints (22) and (23) track the remaining demands carried by an OD while visiting customer nodes. Constraint (24) calculates the total demands carried by an OD, and constraint (25) defines the maximum capacity of an OD. Lastly, constraint (26) ensures that the total distance travelled by an OD is not greater than the maximum willingness of the OD.

$$\sum_{i\in N_S}\sum_{j\in N_v} y_{\text{ijk}} \le 1\ k \in K_2, \tag{27}$$

$$\sum_{i\in N_S\cup N_v} y_{\text{ijk}} - \sum_{i\in N_S\cup N_v} y_{\text{jik}} = 0\ j \in N_v, \quad k \in K_2, \tag{28}$$

$$\sum_{k\in K_2}\sum_{i\in N_2} y_{\text{ijk}} \le 1\ j \in N_r, \tag{29}$$

$$\sum_{k\in K_{OD}} \alpha_{\text{pk}}\sigma_k^{\text{OD}} \le l_p^M\ p \in N_r, \tag{30}$$

$$l_i^M \le Q^r \; i \in N_r, \tag{31}$$

$$\sum_{i\in N_2}\sum_{j\in N_C} y_{\mathrm{ijk}}\delta_j + \sum_{i\in N_2}\sum_{j\in N_r} y_{\mathrm{ijk}} l_j^M \le \sigma_k^2 \; k \in K_2, \tag{32}$$

$$\sigma_k^2 \le Q^2 \; k \in K_2, \tag{33}$$

$$\sigma_k^2 \le q_{\mathrm{jk}}^2 + M\left(1 - y_{\mathrm{ijk}}\right) i \in N_S,\; j \in N_v, \quad k \in K_2, \tag{34}$$

$$q_{\mathrm{ik}}^2 - \delta_i \le q_{\mathrm{jk}}^2 + M\left(1 - y_{\mathrm{ijk}}\right) i \in N_C,\; j \in N_v, \quad k \in K_2, \tag{35}$$

$$q_{\mathrm{ik}}^2 - l_i^M \le q_{\mathrm{jk}}^2 + M\left(1 - y_{\mathrm{ijk}}\right) \lim_{x\longrightarrow\infty} i \in N_r,\; j \in N_v, \quad k \in K_2. \tag{36}$$

Constraints (27) to (36) focus on the route characteristics of the city freighters. Constraint (27) ensures that each city freighter can only originate from one satellite. Constraint (28) ensures the flow-in and flow-out of each city freighter. Constraint (29) guarantees that each TN can only be visited at most once. Constraint (30) defines the number of demands delivered to a TN by a city freighter. Constraint (31) limits the maximum allowable demands delivered to a TN. Constraint (32) defines the total demand carried by a city freighter from a satellite, while the maximum allowable demands that can be carried by a city freighter are limited by constraint (33). Moreover, constraints (34) to (36) track the remaining demands carried by a city freighter while visiting a node, either a customer node or a TN. Finally, the domains of the decision variables are given by constraints (37)–(49).

$$x_{\mathrm{ijk}} \in \{0,1\}\, \forall i \in N_1,\, \forall j \in N_1, \quad \forall k \in K_1, \tag{37}$$

$$y_{\mathrm{ijk}} \in \{0,1\}\, \forall i \in N_2,\, \forall j \in N_2, \quad \forall k \in K_2, \tag{38}$$

$$z_{\mathrm{ijk}} \in \{0,1\}\, \forall i \in n_k^o \cup N_2,\, \forall j \in N_2 \cup n_k^D, \quad \forall k \in K_{\mathrm{OD}}, \tag{39}$$

$$\alpha_{\mathrm{pk}} \in \{0,1\}\, \forall p \in N_p, \quad \forall k \in K_{\mathrm{OD}}, \tag{40}$$

$$\beta_{\mathrm{ik}} \in \{0,1\}\, \forall i \in N_c, \quad \forall k \in K_{\mathrm{OD}}, \tag{41}$$

$$\sigma_k^1 \ge 0\, \forall k \in K_1, \tag{42}$$

$$\sigma_k^2 \ge 0\, \forall k \in K_2, \tag{43}$$

$$\sigma_k^{\mathrm{OD}} \ge 0\, \forall k \in K_{\mathrm{OD}}, \tag{44}$$

$$l_i^M \ge 0\, \forall i \in N_r, \tag{45}$$

$$l_i^S \ge 0\, \forall i \in N_S, \tag{46}$$

$$q_{ik}^1 \ge 0\, \forall i \in N_S, \quad \forall k \in K_1, \tag{47}$$

$$q_{\mathrm{ik}}^2 \ge 0\, \forall i \in N_v, \quad \forall k \in K_2, \tag{48}$$

$$q_{\mathrm{ik}}^{\mathrm{OD}} \ge 0\, \forall i \in N_C, \quad \forall k \in K_{\mathrm{OD}}. \tag{49}$$

The formulation belongs to a mixed-integer nonlinear program due to the multiplication of two types of decision variables in several constraints, i.e., constraints (6), (7), (30), and (32).

5. Proposed ALNS Algorithm for 2EVRP-TN-OD

This section presents the ALNS algorithm for solving 2EVRP-TN-OD. The following sections discuss the solution representation, the procedure for generating an initial solution, and the components of the proposed ALNS.

5.1. Adaptive Large Neighborhood Search. Due to the complexity of VRP variants, most of the aforementioned works opted for metaheuristics to solve the problems, which are summarized in Table 1. Among them, a large neighborhood search-based algorithm is one of the most effective methods for solving VRPs [12, 14, 42]. ALNS was first introduced by Ropke and Pisinger [44] to deal with the pickup and delivery problem with time windows (PDPTW). This approach searches for a new solution using various removal as well as insertion mechanisms and the computed costs for reinserting the removed nodes from the current solutions. Originally, the removal operators included Shaw Removal, Random Removal, and Worst Removal, while the insertion ones were Greedy heuristic and Regret heuristic. They are selected based on rewards using the roulette wheel principle. A new solution will be accepted if it is better than the current one.

Akpunar and Akpinar [45] proposed a hybrid adaptive large neighborhood (H-ALNS) search algorithm, which outperforms the existing ones in terms of computational time and solution quality. This is confirmed by Voigt et al. [43], who used H-ALNS to solve various instances of 2EVRP, LRP, and multidepot VRP (MDVRP). Despite the superiority in solving 2EVRP of HALNS, our work adopts ALNS to solve 2EVRP-TN-OD.

Although the proposed ALNS of our work is partly similar to those of Hemmelmayr et al. [24] and Breunig et al. [42], they did not consider the crowd-shipping scenario in their works. Our work has a similar destroy mechanism to the "satellite-related removal mechanism" of theirs, but the destroyed nodes in our ALNS can be either satellites or transshipment nodes. When a TN is removed, a set of customers related to this node will also be removed from the second echelon, which is the uniqueness of our study. Moreover, our work's Greedy insertion mechanism is slightly different from theirs due to the extra costs for servicing the TNs of the city freighters. The related insertion is also another difference between our work and that of the mentioned authors. To sum up, ALNS with destroy and repair mechanisms for solving the 2EVRP-TN-OD

Table 1: Summary of metaheuristics and heuristics used for solving various variants of 2EVRP, LRP, and crowd-shipping (CS) problems.

Reference	Proposed problem	Solution approach
Archetti et al. [33]	CS	Iterated local search
Chen et al. [35]$^{(**)}$	CS	Time compatibility-based heuristic and time-expanded graph-based heuristic (heuristic approach)
Macrina et al. [40]	CS	Variable neighborhood search
Sampaio et al. [36]$^{(***)}$	CS	Adaptive large neighborhood search
Voigt and Kuhn [37]	CS	Adaptive large neighborhood search
Yu et al. [26]	LRP	Simulated annealing
Karaoglan et al. [28]	LRP	Two-phase simulated annealing
Hemmelmayr et al. [24]	LRP (2EVRP based)	Adaptive large neighborhood search
Ting and Chen [25]$^{(*)}$	LRP	Multiple ant colony optimization
Yu and Lin [29]	LRP	Simulated annealing
Schneider and Löffler [27]	LRP	Tree-based search algorithm (heuristic approach)
Pichka et al. [30]	2ELRP	Simulated annealing
Yu et al. [16]	2ELRP	Simulated annealing
Huang and Ardiansyah [38]	2ELRP and CS	Tabu search
Arnold and Sörensen [32]$^{(****)}$	2ELRP	Progressive filtering algorithm (heuristic approach)
Wang et al. [9]	2EVRP	Variable neighborhood search and integer programming
Belgin et al. [10]	2EVRP	Hybrid variable neighborhood Descent and local search
Zhou et al. [18]	2EVRP	Hybrid Genetic algorithm
Breunig et al. [11]	2EVRP	Large neighborhood search
Jie et al. [12]	2EVRP	Column generation-adaptive large neighborhood search
Enthoven et al. [14]	2EVRP	Adaptive large neighborhood search
Kitjacharoenchai et al. [15]	2EVRP	Large neighborhood search and drone truck route construction
Yu et al. [31]	2EVRP	Hybrid multistart metaheuristic
Anderluh et al. [13]	2EVRP	Combined ϵ-constraints and large neighborhood search, and combined large neighborhood and the heuristic Rectangle Splitting method
Mühlbauer and Fontaine [19]	2EVRP	Parallelized large neighborhood search
Yu et al. [17]	2EVRP and CS	Adaptive large neighborhood search
Voigt et al. [43]$^{(**-**)}$	2EVRP and LRP	Hybrid adaptive large neighborhood search

The works of the mentioned authors also involve the following problem: (*): FLP and MDVRP (facility location problem and multidepot vehicle routing problem). (**): The authors addressed CS through MDMPMP (Multidriver, multiparcel matching problem). (***): The authors addressed CS through the vehicle settings of PDPTW-T (pickup and delivery problem with time windows and transfers). (****): STTRPSD (single truck and trailer routing problem with satellite depots). (**-**): MDVRP (multidepot vehicle routing problem).

distinguishes our study from the works of Hemmelmayr et al. [24] and Breunig et al. [42].

5.2. Solution Representation. The solution representation of the 2EVRP-TN-OD problem consists of three parts: (1) routes of first-echelon trucks, (2) routes of city freighters, and (3) routes of ODs. First, let $R_k^1 = \{R_k^1(0), R_k^1(1), \ldots, R_k^1(|R_k^1|)\}$ be the route of first-echelon truck *k*, consisting of depot positioned at $R_k^1(0)$ and $R_k^1(|R_k^1|)$, and the sequence of visited satellites. Second, let $R_i^2 = \{R_i^2(0), R_i^2(1), \ldots, R_i^2(|R_i^2|)\}$ denote the route of city freighter *i*. Herein, $R_i^2(0)$ represents the satellite from which city freighter *i* originates. Note that $R_i^2(0) = R_i^2(|R_i^2|)$ because city freighter *i* returns to the same satellite after visiting a sequence of nodes. The visited vertices by a city freighter can be customers and/or TNs. Third, let $\Omega_j = \{\Omega_j(0), \Omega_j(1), \ldots, \Omega_j(|\Omega_j|)\}$ be the route of OD *j* consisting of two parts: (1) a satellite or a TN where the OD picks up the assigned demands $\Omega_j(0)$, and (2) the visited customers $\{\Omega_j(1), \ldots, \Omega_j(|\Omega_j|)\}$.

Figure 2 illustrates the solution representation of Figure 1. For example, R_1^1 represents the route of first-echelon truck 1, visiting satellites 1 and 3 before returning to the depot. R_1^2 represents the route of city freighter 1, which originates from satellite 1, and performs a visit to a sequence of customers. Ω_1 represents the route of OD 1 that visits transshipment node 19 and serves customers 4 and 16.

5.3. Construction Heuristic. The construction heuristic generates a feasible initial solution for ALNS. It first assigns customers to available ODs using the nearest neighborhood insertion based on the capacity of ODs. Each OD is assigned to the nearest collection point, either a satellite or a TN, to collect demands. Afterward, unassigned customers are assigned to their nearest satellite. From each utilized satellite, city freighters' routes are then built to serve these customers

0	1	3	0
R_1^1			

1	5	9	12	17	1	
R_1^2						
1	19	8	13	14	1	
R_2^2						
3	15	10	11	18	6	3
R_3^2						

19	4	16
Ω_1		
3	7	
Ω_2		

Figure 2: The solution representation example of 2EVRP-TN-OD.

and utilize TNs. Finally, routes of first-echelon trucks are constructed to serve the demands assigned to each satellite.

5.4. Proposed Adaptive Large Neighborhood Search Heuristic. Algorithm 1 describes the general structure of the ALNS heuristic developed for solving 2EVRP-TN-OD. The construction algorithm described in Section 5.2 is executed to build an initial solution. The resulting solution is stored in F^c and copied to the working solution, F^w, and the best solution, F^*. ALNS begins with removing nodes from F^W with a destroy operator selected by a roulette wheel mechanism. The number of removed nodes from the solution is randomly chosen between $[1, \theta]$. To accommodate the reconfiguration of both echelons, two conditions are expressed in Lines 5–8. The main purpose of the first condition in Lines 5–6 is to remove nodes visited in the second-echelon network, while the main purpose of the second condition in Lines 7–8 is not only for dealing with nodes in the second-echelon network but also for changing the configuration of satellites.

After the removal phase is conducted, ChangeODPickupPoint( ) is performed to change pickup points visited by ODs. The repair phase then starts by applying a selected repair operator to F^W. The repair phase focuses on the second-echelon network. Thus, the first-echelon routes' optimization procedure follows to reconstruct a set of feasible routes for first-echelon trucks, as described in Section 5.3.4.

A local search procedure expressed in Lines 11–18 is performed for intensification purpose and therefore increases the possibility of obtaining a better solution. When F^W is better (in this case, has a lower objective value) than F^C, we replace F^C with F^W, as expressed in Lines 15–18. Furthermore, if F^W successfully improves the best solution so far, then F^* will be updated, and the selection probability of destroying and repairing operators is updated using UpdateOperatorScore(λ_r, λ_i), which only considers the performance of available operators after the previous score updates. Initially, every operator has a score of λ. In particular, an amount of score τ is added to an operator's score whenever the operator can achieve a new best solution.

The acceptance of F^W with worse quality is based on threshold acceptance; i.e., still accepting F^W as long as the gap between the objective value of F^W and F^C is within a particular threshold value. This type of acceptance is expressed using a function, i.e., $accept(F^w, F^c)$, in Line 26. The algorithm performs a restart strategy when a certain number of nonimproving iterations, β, is reached. Finally, ALNS runs until the maximum time limit, $T_{\max}$, is reached.

5.4.1. Destroy Operators. The developed destroy operators are mainly adopted from Ropke and Pisinger [44] and Hemmelmayr et al. [24]. In the solution, there exist two types of nodes, i.e., customer nodes and TNs, in the second-echelon network. In this section, we thus refer to a node as either a customer node or a TN. The removal operators are specifically designed to address the existence of the aforementioned types of nodes. Whenever applicable, the number of customers and/or TNs to remove, q, is randomly chosen within the range of available customers and TNs $[\delta, \theta]$. The description of all destroy operators is provided as follows:

(1) Pickup Point Removal. A pickup point in the solution space, either a TN or a satellite, is selected, and all customers associated with the point are put on the removal list. This operator only operates whenever n_{noupdate} reaches the maximum number of nonimprovements' iterations L. After the implementation of this operator, n_{noupdate} is set to 0.

(2) Random Removal. This operator will choose q nodes randomly and exclude them from the solution pool for reinserting purposes. If the selected removed node is a TN, then all associated customers with this TN are destroyed.

(3) Worst Removal. A number of q nodes with the highest normalized removal gains, which are the absolute value of the subtraction between the original solution's cost and the cost of the solution without the selected node, are removed using this operator. This value is then divided by the arcs' average cost where the respected node belongs for normalization purposes. By doing this, candidates who have an extreme distance from other customers can limit the rate at which they are selected over iterations.

(4) Node Neighborhood Removal. This operator randomly retrieves a node as a seed, and then $(q-1)$ nodes closest to this node are removed. If a removed node is a TN, then all related customers with the TN are removed.

(5) Random Route Removal. A route in the solution pool is randomly selected and excluded from the pool along with its candidates. It is strictly prohibited for destroyed nodes using this operator to be reinserted back into their most recent satellite.

(6) Route Redistribution. For each satellite, we remove a route containing one or more nodes that are actually located nearer to another satellites . The distances of nodes' pairs are multiplied with a perturbation value $p \epsilon$ [0.8, 1.2] for randomization purposes.

5.4.2. Repair Operators. Previously removed customer nodes are stored in the removal list and are reinserted to generate a new solution from the existing ones in this section using the following operators:

(1) Greedy Insertion. Customer nodes from the removal list are retrieved based on the ascending order of their insertion costs at all available locations in the searching space first. Those nodes are then ranked and reinserted at the location with the lowest cost among the currently available routes. A new route is also considered whenever applicable. An additional inserting cost is also accounted for by the city freighter to service unopened TNs that are launched for inserting customers to be served by unemployed ODs. Two other variants of greedy insertion are also proposed: Greedy insertion with noise, where the insertion cost is modified using the noise factor $p \in [0.8, 1.2]$ for randomization purposes, and Greedy insertion with a prohibition, where nodes are not allowed to be reinserted into routes that they were recently excluded from.

(2) Regret Insertion. Nodes are inserted based on their regret values, which are the absolute subtraction value between the insertion position of the node's best cost and its second-best ones among existing routes. Similar to the greedy operator, the insertion costs also consider the scenario where a new route is opened. In addition, the regret value for each removed node holds its value until the current iteration finishes.

(3) Related Insertion. This operator first randomly chooses a removed customer node as a seed and selects the following inserted candidates based on the distance between the seed and the remaining nodes in the removal pool in ascending order.

5.4.3. Local Search. This study implements local search (LS) for the purpose of intensification. Five operators are considered in this procedure: insertion, interroute swap, intraroute swap, 2-opt, and 2-opt*. For each local search operator, a solution is accepted when its objective value surpasses the current one. When a better solution is achieved, it is immediately implemented by the procedure, just prior to the termination of the search process, and the operator is restarted. This loop continues until no more improvements are achieved. In this case, the next local search operator using the same mechanism is implemented. Each move is explained as follows:

(1) Insertion. One node is randomly excluded from its current route and then assigned to a different place within the same route that extracted it. If a TN is excluded for wrapping itself into a new position in the routing sequence of the second-level route, then there are no changes to the customers' sequences served using this TN.

(2) Inter-Route Swap. A pair of nodes is removed from their current route, and their origin position in the route sequence is switched with the other ones within the same route. If a TN is chosen for switching its position with another node, then no changes happen to all customers in the servicing sequence from this TN.

(3) Intra-Route Swap. A pair of routes is chosen, and within themselves, a node is taken out and switched positions in the route sequence with the ones that are extracted from the other route. If a TN is chosen for swapping positions with nodes from other routes, then all customers in the servicing sequence using this TN are also delivered to the new route and are still served by this TN. It must be noted that the swapping method also checks the validity of the respected constraints when swapping nodes; hence, the demands of all TNs are determined, bringing no violations to the whole solving procedure.

(4) 2-Opt. This option selects 2 consecutive pairs of nodes $(i, i+1)$ and $(j, j+1)$ in a random route through the solution pool, and then each candidate in a pair swaps with the respected candidate on the other pair. The two new pairs (i, j) and $(i+1, j+1)$ are then formed from the previous ones on the same route. Similar to the interroute swap, customers' orders that are fulfilled by a selected TN are still handled by this TN.

*(5) 2-opt**. This option is similar to the 2-opt, but instead of doing the swapping on the same route, a pair of random routes is selected for initiating the swapping process, and at each route, there is only one consecutive pair excluded from the existing route. Just like the intraroute swap, if a TN is chosen for this swapping operator, then its validity on the demand constraints is checked to ensure there is no violation of the route capacity, and all customers whose demands are delivered from this TN are still distributed using it.

5.4.4. First-Echelon Routes' Optimization. After the repair phase of the second-echelon network, a procedure for building first-echelon trucks' routes is implemented to complete the solution. The procedure consists of two phases: the route construction phase and the route improvement phase. The first phase constructs first-echelon trucks' routes following the algorithm proposed by Breunig et al. [42]. The second phase applies the improvement heuristic shown in Algorithm 2 to improve the solution obtained in Phase 1.

Let F^C_{1E} and F^W_{1E} be the routes of the first-echelon network in F^C and F^W, respectively. For every iteration of Algorithm 2, a random number r is generated. One of the three local search operators (i.e., insertion, swap, and 2-opt) is then implemented. This procedure terminates when the number of iterations reaches μ_{1E}. In this work, μ_{1E} is set to 500 based on a trial-and-error experiment. In this procedure, there is no involvement with the routes that cross the TNs; thus, the cases when TNs are selected are not considered.

Input: $\mathbb{O}_D$, $\mathbb{O}_I$, λ, p, δ, θ, β, T, L and an instance of 2EVRP-TN-OD
Output: S^*
$F^c \longleftarrow InitialSolution$
$F^* \longleftarrow F^w \longleftarrow F^c$
$T \longleftarrow T_0$, $i \longleftarrow 0$, $n_{noupdate} \longleftarrow 0$
$q \longleftarrow RandInt\,[\delta, \theta]$
while $T \le T_{\max}$ **do**
  if $n_{noupdate} < L$ **then**
    Remove *and reinsert q customers based on removal and a repair operator* λ_r *and* λ_i *where* $r \in \mathbb{O}_D$ *and* $i \in \mathbb{O}_I$
  else
    Pick – *up* point *removal and a repair operator based on* λ_i *is Selected*
    $n_{noupdate} \longleftarrow 0$
  $F^w \longleftarrow$ Repair $(ChangeODPickupPoint\,(Destroy\,(F^c)))$
  $LocalSearch\,(F^w)$
  if $f\,(LocalSearch\,(F^w)) > f(F^w))$ **then**
    while $f\,(LocalSearch\,(F^w)) > f(F^w))$ **do**
      $LocalSearch\,(F^w)$
      if $f\,(LocalSearch\,(F^w)) \le f\,(F^w))$ **then**
        $F^w \longleftarrow LocalSearch\,(F^w)$
  else
    $F^w \longleftarrow LocalSearch\,(F^w)$
  if $f\,(F^w) < f\,(F^c)$ **then**
    $F^c \longleftarrow F^w$
    if $f\,(F^c) < f\,(F^*)$ **then**
      $F^* \longleftarrow F^c$
      $\lambda_r, \lambda_i \longleftarrow UpdateOperatorScore\,(\lambda_r, \lambda_i)$
      $n_{noupdate} \longleftarrow 0$
  else
    $F^c \longleftarrow accept\,(F^w, F^c)$
  if $n_{noupdate} \ge \beta$ **then**
    $F^c \longleftarrow F^*$
  $n_{noupdate} \longleftarrow n_{noupdate} + 1$
return F^*

Algorithm 1: The proposed ALNS heuristic.

for $i = 1$ to μ_{1E} **do**
  $r \longleftarrow U\,(0, 1)$
  if $(r < 1/3)$ **then**
    $1E\_Insertion\,(F_{1E}^W)$
  else if $(1/3 \le r < 2/3)$ **then**
    $1E\_Swap\,(F_{1E}^W)$
  else
    $1E\_2opt\,(F_{1E}^W)$
  if $(f\,(F_{1E}^W) < f\,(F_{1E}^C))$ **then**
    $F_{1E}^C \longleftarrow F_{1E}^W$
end for
return F_{1E}^C

Algorithm 2: The pseudocode of the first-echelon routes' optimization.

5.5. Our Contributions. Different from the existing 2EVRP, our 2EVRP-TN-DO included the pickup point operator for the removal process, in which the selected nodes can be either TNs or satellites, while Hemmelmayr et al. [24] addressed this as a satellite removal only. Our random and node neighborhood removals also include the retrieval of TNs, which is not mentioned in the work of the scholars mentioned.

In the insertion phase, we also account for the cost of opening new TNs that are served by unemployed ODs in addition to the related insertion operator. This has not been suggested by the studies of Hemmelmayr et al. [24] and Breunig et al. [42]. Instead of implementing the local search procedure on the first level of existing work, we apply it to both first- and second-level route optimizations in our work.

6. Computational Experiments

MINLP is solved using AMPL with the Gurobi solver, and the proposed ALNS is coded in Microsoft Visual Studio C++ 2019. Both of them were run on a computer with an Intel® Core i7-6700 CPU at 3.40 GHz and 8 GB of RAM under Windows 10 Professional.

6.1. Benchmark Instances. The 2EVRP-TN-OD datasets are generated based on the existing 2EVRP instances. There are four sets of benchmark instances. Set 1, originally proposed by Perboli et al. [8], consists of 1 depot, 2 satellites, and 12 customers. Sets 2 and 3 are larger instances, with 1 depot, 2–4 satellites, and 21–50 customers, respectively. A set of available ODs and a set of TNs are added to the 2EVRP instances. We follow Archetti et al. [33] by setting the number of ODs in the system to be equal to the number of customers. For a given instance with locations of customers (x_i, y_i), $\forall i \in N_c$, the origin and destination of ODs are randomly generated in the lower left-hand corner of the square $(\min_{i\in N_c} x_i \times 0.75, \min_{i\in N_c} y_i \times 0.75)$ and upper right-hand corner of the square $(\max_{i\in N_c} x_i \times 1.25, \max_{i\in N_c} y_i \times 1.25)$, respectively. Second, ODs' capacity is generated under a uniform distribution of $U[0.05, 0.25] \times Q^2$. Third, we make sure that every generated OD is able to at least serve one customer; otherwise, another OD is generated to replace the OD who is not able to serve any customers. The parameters of ODs, partly adopted from Archetti et al. [33], are provided as follows:

(1) The variable cost of an OD is set to 0.2 times the total travelled distance, and the fixed cost of an OD is set to 5.
(2) The maximum distance deviation that is willingly traversed by an OD is 0.5 times the distance between the origin and destination nodes of the OD.

TNs are generated by following these rules. First, the number of TNs in the system is set to 3. This is reasonable since the number of satellites in these instances is 2 or 4. Second, for the location of TNs, we randomly select a position in the lower left-hand corner of the customer zone $(\min_{i\in N_c} x_i, \min_{i\in N_c} y_i)$ and the upper right-hand corner of the customer zone $(\max_{i\in N_c} x_i, \max_{i\in N_c} y_i)$ and locate a TN in the selected position. After that, we ensure that every TN is not too close to each other and the satellites by defining a minimum distance requirement, γ. Here, γ is set by the following equation (50). In case a newly generated TN violates the distance requirement, another location is generated using the aforementioned mechanism. This procedure repeats until all generated TNs fulfill the minimum distance requirement.

$$\gamma = \mathrm{sqrt}\Big(\left(\max x_i - \min x_i\right)^2 + \left(\max y_i - \min y_i\right)^2 \Big) \times 0.25. \tag{50}$$

6.2. Parameter Tuning. We follow the parameter tuning procedure for the ALNS heuristic developed by Hemmelmayr et al. [24] and Ropke and Pisinger [44]. Initially, the parameters' values of the proposed ALNS follow the aforementioned references. By considering a set of candidate values for each parameter, the tuning procedure is conducted by only accepting the value leading to an improvement in terms of the average objective value over all selected testing instances. The aforementioned procedure is performed sequentially over all parameters being tuned. The final selected values for all parameters are presented in Table 2.

6.3. Performance of the Proposed ALNS for Solving 2EVRP Instances. Three sets of 2EVRP instances from Perboli et al. [8] and Hemmelmayr et al. [24] are used to test the effectiveness of the proposed ALNS. We compare the results obtained by ALNS with three state-of-the-art algorithms proposed by Hemmelmayr et al. [24]; Breunig et al. [42]; and Enthoven et al. [14]. Tables 3–5 present the comparative Table 6 results. In these tables, columns "Best" and "CPU (s)" represent the best value obtained and the average computational time used by each method, respectively.

Based on Tables 3–5, the proposed ALNS provides results comparable to those obtained from the state-of-the-art algorithms. In particular, ALNS achieves exactly the same best solutions for Sets 2 and 3 instances, while the average gap between the best solutions obtained by our ALNS for Set 5 and BKS is 1.40%. The average gaps to BKS are 0.25% for Set 2, 1.37% for Set 3, and 2.47% for Set 5, respectively.

6.4. Performance of ALNS for Solving 2EVRP-TN-OD Small Instances. Table 7 shows the results obtained by MINLP and the proposed ALNS. All of the instances of Set 1 are solved to optimality by the Gurobi solver, but it takes more than 5 hours on average to solve one small-scale instance. ALNS is run 5 times to solve each instance of Set 1. Based on Table 7, our ALNS can obtain results of equal quality to those produced by Gurobi's solver. Moreover, both the best solution and average solution values are the same, showing the robustness of the proposed ALNS for solving small-scale instances. The computational time required by ALNS to obtain those results is significantly smaller. To sum up, the proposed ALNS outperforms the Gurobi solver in terms of computational time.

6.5. Performance of ALNS for Solving 2EVRP-TN-OD Larger Instances. Tables 8 and 9 show the results of solving the larger instances; i.e., Sets 2 and 3 of 2EVRP-TN-OD instances. For solving each instance of Sets 2 and 3, we utilize both Gurobi and the proposed ALNS. We set the maximum running time to 3600 seconds for Gurobi. ALNS is run5

Table 2: Final parameters in the proposed ALNS algorithm.

Symbol	Explanation	Final value
λ	Initial score of destroy and repair operators	50
p	Perturbation factor	0.2
δ	Minimum number of customers and/or TNs to be removed	1
θ	Maximum number of customers and/or TNs to be removed	0.4 $\lvert N \rvert$
β	Number of iterations without improvement before returning to the best-found solution	1000
L	Number of iterations without improvement before implementing the satellite removal	100
T^{2E-VRP}	Time budget for 2EVRP	60 s (small instance), 900 s (large instance)
$T^{2E-VRP-TN-OD}$	Time budget for 2EVRP-TN-OD	60 s (small instance), 120 s (medium instance), 300 s (large instance)

Table 3: Comparative results for solving 2EVRP Set 2 instances

Instance	BKS	Hemmelmayr et al. [24]		Breunig et al. [42]		Enthoven et al. [14]		Proposed ALNS			Avg. gap to BKS (%)	Best gap to BKS (%)
		Best	CPU (s)	Best	CPU (s)	Best	CPU (s)	Avg. 5	Best	CPU (s)		
Set 2a												
E-n22-k4-s6-17	**417.07**	**417.07**	37	**417.07**	60	**417.07**	75	417.07	**417.07**	60	0.00	0.00
E-n22-k4-s8-14	**384.96**	**384.96**	34	**384.96**	60	**384.96**	103	384.96	**384.96**	60	0.00	0.00
E-n22-k4-s9-19	**470.6**	**470.6**	35	**470.6**	60	**470.6**	64	470.60	**470.60**	60	0.00	0.00
E-n22-k4-s10-14	**371.5**	**371.5**	37	**371.5**	60	**371.5**	75	371.50	**371.50**	60	0.00	0.00
E-n22-k4-s11-12	**427.22**	**427.22**	31	**427.22**	60	**427.22**	138	427.22	**427.22**	60	0.00	0.00
E-n22-k4-s12-16	**392.78**	**392.78**	36	**392.78**	60	**392.78**	67	392.78	**392.78**	60	0.00	0.00
E-n33-k4-s1-9	**730.16**	**730.16**	74	**730.16**	60	**730.16**	157	730.16	**730.16**	60	0.00	0.00
E-n33-k4-s2-13	**714.63**	**714.63**	64	**714.63**	60	**714.63**	83	714.63	**714.63**	60	0.00	0.00
E-n33-k4-s3-17	**707.48**	**707.48**	**58**	**707.48**	60	**707.48**	80	707.48	**707.48**	60	0.00	0.00
E-n33-k4-s4-5	**778.74**	**778.74**	77	**778.74**	60	**778.74**	79	778.74	**778.74**	60	0.00	0.00
E-n33-k4-s7-25	**756.85**	**756.85**	**53**	**756.85**	60	**756.85**	82	756.85	**756.85**	60	0.00	0.00
E-n33-k4-s14-22	**779.05**	**779.05**	85	**779.05**	60	**779.05**	82	779.05	**779.05**	60	0.00	0.00
Set 2b												
E-n51-k5-s2-4-17–46	**530.76**	**530.76**	154	**530.76**	60	**530.76**	145	530.76	**530.76**	60	0.00	0.00
E-n51-k5-s2-17	**597.49**	**597.49**	100	**597.49**	60	**597.49**	140	597.49	**597.49**	60	0.00	0.00
E-n51-k5-s4-46	**530.76**	**530.76**	173	**530.76**	60	**530.76**	211	534.76	**530.76**	60	0.75	0.00
E-n51-k5-s6-12	**554.81**	**554.81**	149	**554.81**	60	**554.81**	199	555.03	**554.81**	60	0.04	0.00
E-n51-k5-s6-12-32–37	**531.92**	**531.92**	150	**531.92**	60	**531.92**	141	537.78	**531.92**	60	1.10	0.00
E-n51-k5-s11-19	**581.64**	**581.64**	182	**581.64**	60	**581.64**	131	584.06	**581.64**	60	0.42	0.00
E-n51-k5-s11-19-27–47	**527.63**	**527.63**	147	**527.63**	60	**527.63**	137	533.96	**527.63**	60	1.20	0.00
E-n51-k5-s27-47	**538.22**	**538.22**	136	**538.22**	60	**538.22**	237	538.22	**538.22**	60	0.00	0.00
E-n51-k5-s32-37	**552.28**	**552.28**	141	**552.28**	60	**552.28**	263	552.28	**552.28**	60	0.00	0.00
Set 2c												
E-n51-k5-s2-4-17–46	**601.39**			**601.39**	60	**601.39**	209	606.75	**601.39**	60	0.89	0.00
E-n51-k5-s2-17	**601.39**			**601.39**	60	**601.39**	141	602.66	**601.39**	60	0.21	0.00
E-n51-k5-s4-46	**702.33**			**702.33**	60	**702.33**	196	706.78	**702.33**	60	0.63	0.00
E-n51-k5-s6-12	**567.42**			**567.42**	60	**567.42**	129	569.36	**567.42**	60	0.34	0.00
E-n51-k5-s6-12-32–37	**567.42**			**567.42**	60	**567.42**	136	569.97	**567.42**	60	0.45	0.00
E-n51-k5-s11-19	**617.42**			**617.42**	60	**617.42**	137	621.18	**617.42**	60	0.61	0.00
E-n51-k5-s11-19-27–47	**530.76**			**530.76**	60	**530.76**	135	532.10	**530.76**	60	0.25	0.00
E-n51-k5-s27-47	**530.76**			**530.76**	60	**530.76**	134	530.76	**530.76**	60	0.00	0.00
E-n51-k5-s32-37	**752.59**			**752.59**	60	**752.6**	132	757.74	**752.59**	60	0.68	0.00
Average	**578.27**	**565.55**	**93**	**578.27**	**60**	**578.27**	**134**	**579.76**	**578.27**	**60.00**	**0.25**	**0.00**

Table 4: Comparative results for solving 2EVRP set 3 instances.

Instance	BKS	Hemmelmayr et al. [24]		Breunig et al. [42]		Enthoven et al. [14]		Proposed ALNS			Avg. Gap to BKS (%)	Best gap to BKS (%)
		Best	CPU (s)	Best	CPU (s)	Best	CPU (s)	Avg. 5	Best	CPU (s)		
						Set 3a						
E-n22-k4-s13-14	**526.15**	**526.15**	43	**526.15**	60	**526.15**	70	532.25	**526.15**	60	1.16	0.00
E-n22-k4-s13-16	**521.09**	**521.09**	44	**521.09**	60	**521.09**	64	521.96	**521.09**	60	0.17	0.00
E-n22-k4-s13-17	**496.38**	**496.38**	49	**496.38**	60	**496.38**	65	511.59	**496.39**	60	3.06	0.00
E-n22-k4-s14-19	**498.8**	**498.8**	43	**498.8**	60	**498.8**	63	518.63	**498.80**	60	3.98	0.00
E-n22-k4-s17-19	**512.81**	**512.81**	26	**512.81**	60	**512.81**	72	526.15	**512.81**	60	2.60	0.00
E-n22-k4-s19-21	**520.42**	**520.42**	34	**520.42**	60	**520.42**	62	525.93	**520.42**	60	1.06	0.00
E-n33-k4-s16-22	**672.17**	**672.17**	76	**672.17**	60	**672.17**	114	677.53	**672.17**	60	0.80	0.00
E-n33-k4-s16-24	**666.02**	**666.02**	77	**666.02**	60	**666.02**	106	692.45	**666.02**	60	3.97	0.00
E-n33-k4-s19-26	**680.36**	**680.36**	84	**680.36**	60	**680.36**	79	681.16	**680.37**	60	0.12	0.00
E-n33-k4-s22-26	**680.37**	**680.37**	77	**680.37**	60	**680.37**	136	683.34	**680.37**	60	0.44	0.00
E-n33-k4-s24-28	**670.43**	**670.43**	88	**670.43**	60	**670.43**	77	679.16	**670.43**	60	1.30	0.00
E-n33-k4-s25-28	**650.58**	**650.58**	63	**650.58**	60	**650.58**	112	675.43	**650.58**	60	3.82	0.00
						Set 3b						
E-n51-k5-s12-18	**690.59**	**690.59**	147	**690.59**	60	**690.59**	125	693.62	**690.57**	60	0.44	0.00
E-n51-k5-s12-41	**683.05**	**683.05**	133	**683.05**	60	**683.05**	239	690.74	**683.05**	60	1.13	0.00
E-n51-k5-s12-43	**710.41**	**710.41**	217	**710.41**	60	**710.41**	137	710.83	**710.41**	60	0.06	0.00
E-n51-k5-s39-41	**728.54**	**728.54**	155	**728.54**	60	**728.54**	179	736.08	**728.54**	60	1.03	0.00
E-n51-k5-s40-41	**723.75**	**723.75**	154	**723.75**	60	**723.75**	129	736.46	**723.75**	60	1.76	0.00
E-n51-k5-s40-43	**752.15**	**752.15**	158	**752.15**	60	**752.15**	242	763.2	**752.15**	60	1.47	0.00
						Set 3c						
E-n51-k5-s13-19	**560.73**			**560.73**	60	**560.73**	119	566.35	**560.73**	60	1.00	0.00
E-n51-k5-s13-42	**564.45**			**564.45**	60	**564.45**	174	568.69	**564.45**	60	0.75	0.00
E-n51-k5-s13-44	**564.45**			**564.45**	60	**564.45**	121	571.14	**564.45**	60	1.19	0.00
E-n51-k5-s40-42	**746.31**			**746.31**	60	**746.31**	122	749.71	**746.31**	60	0.46	0.00
E-n51-k5-s41-42	**771.56**			**771.56**	60	**771.56**	203	775.65	**771.56**	60	0.53	0.00
E-n51-k5-s41-44	**802.91**			**802.91**	60	**802.91**	169	807.64	**802.91**	60	0.59	0.00
Average	**641.44**	**632.45**	**92**	**641.44**	**60**	**641.44**	**124**	**649.82**	**641.44**	**60.00**	**1.37**	**0.00**

Table 5: Comparative results for solving 2EVRP Set 5 instances.

Instance	BKS	Hemmelmayr et al. [24]		Breunig et al. [42]		Enthoven et al. [14]		Proposed ALNS			Avg. Gap to BKS (%)	Best gap to BKS (%)
		Best	CPU (s)	Best	CPU (s)	Best	CPU (s)	Avg. 5	Best	CPU (s)		
100-5-1	**1564.46**	1578.4	**429**	**1564.46**	900	1605.59	554	1583.75	1569.47	900	1.23	0.32
100-5-1b	**1108.62**	1118.95	**476**	**1108.62**	900	1121.7	699	1134.03	1112.88	900	2.29	0.38
100-5-2	**1016.32**	**1016.32**	**432**	**1016.32**	900	1022.7	647	1024.36	1022.34	900	0.79	0.59
100-5-2b	**782.25**	784.06	415	**782.25**	900	**782.25**	**345**	803.73	794.32	900	2.75	1.54
100-5-3	**1045.29**	1046.05	**418**	**1045.29**	900	1046.05	565	1047.76	**1045.29**	900	0.24	0.00
100-5-3b	**828.54**	828.99	391	**828.54**	900	828.99	**372**	829.63	**828.54**	900	0.13	0.00
100-10-1	**1124.93**	1133.17	**353**	1125.53	900	1125	643	1130.61	1127.06	900	0.51	0.19
100-10-1b	**916.25**	917.05	**397**	**916.25**	900	921.29	402	930.69	924.58	900	1.58	0.91
100-10-2	**990.58**	997.42	**406**	1002.15	900	1010.72	454	1014.79	1009.61	900	2.44	1.92
100-10-2b	**768.61**	770.7	**340**	774.11	900	775.72	384	795.48	779.41	900	3.50	1.41
100-10-3	**1043.25**	1047.05	**352**	1048.53	900	1053.02	810	1057.34	1049.48	900	1.35	0.60
100-1-3b	**850.92**	862.11	**391**	854.9	900	859.24	682	874.22	871.19	900	2.74	2.38
200-10-1	**1556.79**	1597.19	**888**	**1556.79**	900	1570.88	2375	1593.15	1572.30	900	2.34	1.00
200-10-1b	**1187.62**	1225.9	**692**	**1187.62**	900	1195.12	827	1253.27	1238.17	900	5.53	4.26
200-10-2	**1365.74**	1385.9	1072	**1365.74**	900	1389.11	1364	1413.51	1392.86	900	3.50	1.99
200-10-2b	**1002.85**	1016.14	1058	**1002.85**	900	**1002.63**	810	1055.70	1033.49	900	5.27	3.06
200-10-3	**1787.73**	1799.85	916	1793.99	900	1837.62	1096	1833.30	1806.15	900	2.55	1.03
200-10-3b	**1197.9**	1203.05	1217	**1197.9**	900	1219.92	825	1266.69	1241.07	900	5.74	3.60
Average	**1118.81**	**1129.35**	**591.38**	**1120.66**	**900**	**1131.53**	**769.67**	**1146.78**	**1134.35**	**900.00**	**2.47**	**1.40**

BKS: best-known solution from the state-of-the-art algorithms. Avg. gap to BKS: $(\text{ProposedALNS}_{\text{Avg. 5}} - \text{BestSolution}/\text{BestSolution}) \times 100\%$. Best gap to BKS: $(\text{ProposedALNS}_{\text{Best}} - \text{BestSolution}/\text{BestSolution}) \times 100\%$. Bold number indicates the same solution with BKS.

Table 6: Literature on 2EVRP, LRP, and CS variants.

Literature	1D/MD	1E/2E	EI	CS	TN/CV	IR	HV	Problem name	Highlights	Objective function	Solution approach
Anderluh et al. [13]	MD	2	☒	☐	CV	☐	☐	2eVRPSyn (two-echelon vehicle routing problem with vehicle synchronization)	The existence of a gray zone, environmental factors such as greenhouse gas (GHG), disturbance, a second-echelon depot for small electric vehicles (small EVs), synchronization, and cloned satellites for multivisit purposes	Routing cost in manner of time and distance, vehicle fixed cost, GHG, disturbance	Combined ϵ-constraints and large neighborhood search, and combined large neighborhood and the heuristic Rectangle Splitting method
Archetti et al. [33]	1D	1	☐	☒	—	☐	☒	VRPOD (vehicle routing problem with occasional drivers)	Introduction of occasional drivers	Routing cost, compensation paid to occasional drivers	Iterated local search
Arnold and Sörensen [32]	1D	2	☐	☐	—	☐	☐	2ELRP and STTRPSD (two-echelon location-routing problem and the single truck and trailer routing problem with satellite depots)	Proposed the single truck and trailer routing problem with satellite depots (STTRPSD) as the 2ELRP special case with configuration closed to 2EVRP, a new solution approach, limited capacity satellites	Routing cost, depot opening cost, and vehicle fixed costs	Progressive filtering Algorithm$^{(*)}$
Belgin et al. [10]	MD	2	☐	☐	—	☐	☐	2EVRPSPD (two-echelon vehicle routing problem with simultaneous pickup and delivery)	Customers (and satellites) with pickup and delivery demand, limited capacity satellites	Routing cost	Hybrid variable neighborhood Descent and local search
Breunig et al. [11]	1D	2	☐	☐	—	☐	☐	E2EVRP (electric two-echelon vehicle routing problem)	Limited capacity satellites, small EVs, charging stations for small EVs, and limited energy capacity for small Evs	Routing costs in the manner of time and energy, vehicle fixed costs	Large neighborhood search
Chen et al. [35]	MD	1	☐	☒	—	☐	☒	MDMPMP (multidriver multiparcel matching problem)	Safety constraints for securing parcels, synchronizations, ridesharing, relay visits at nodes.	Routing cost, travel cost of ODs, parcel transferring cost of ODs, waiting time cost of ODs, and extra cost for ODs' detours.	Time compatibility-based heuristic and time-expanded graph-based heuristic$^{(*)}$
Enthoven et al. [14]	1D	2	☐	☐	CV	☐	☐	2EVRP-CO (two-echelon vehicle routing problem with covering options)	Have both satellites and covering locations, delivery options	Routing cost and connection cost	Adaptive large neighborhood search
Hemmelmayr et al. [24]	1D	2	☐	☐	—	☐	☐	LRP (based on 2EVRP) (location-routing problem as special case of the two-echelon vehicle routing problem)	—	Routing cost, vehicle fixed cost, and satellites opening cost	Adaptive large neighborhood search

Table 6: Continued.

Literature	1D/MD	1E/2E	EI	CS	TN/CV	IR	HV	Problem name	Highlights	Objective function	Solution approach
Huang and Ardiansyah [38]	1D	2	☐	☒	—	☐	☒	2ELRP-TTRP (two-echelon location-routing problem with the truck and trailer routing problem)	Relay points	Routing cost, payments for hiring CS	Tabu search
Jie et al. [12]	1D	2	☐	☐	—	☐	☐	2E-EVRP-BSS (two-echelon capacitated electric vehicle routing problem with battery swapping stations)	Multiple type of EVs, charging (battery swapping) stations for EVs, limited energy capacity for EVs	Routing cost, recharging cost, handling cost	Column generation-adaptive large neighborhood search
Karaoglan et al. [28]	1D	1	☐	☐	—	☐	☐	LRPSPD (location-routing problem with simultaneous pickup and delivery)	—	Routing cost, depot opening cost, and vehicle usage cost	Two-phase simulated annealing
Kitjacharoenchai et al. [15]	1D	2	☐	☐	—	☐	☐	2EVRPD (two-echelon vehicle routing problem with drones)	Delivery truck carrying drones for deliveries at second-echelon, customers nodes are adopted as satellites	Routing cost (denoted as truck arrival time)	Large neighborhood search and drone truck route construction
Macrina et al. [40]	1D	1	☐	☒	TN	☐	☒	VRPTW-TN-OD (vehicle routing problem with time windows, transshipment nodes, and occasional drivers)	Capacitated transshipment nodes, synchronization	Routing cost, compensation cost for ODs activities	Variable neighborhood search
Mühlbauer and Fontaine [19]	1D	2	☐	☐	—	☐	☐	2E-CVRPSC (two-echelon capacitated vehicle routing problem with swap containers)	Swap containers	Routing cost	Parallelized large neighborhood search
Pichka et al. [30]	1D	2	☐	☐	—	☐	☐	2E-OLRP (two-echelon open location-routing problem)	Propose the two-echelon open location-routing problem where vehicles will not return to depot and satellites after delivering parcels, limited capacity satellites	Routing cost, vehicle cost, and satellite open cost	Simulated annealing
Sampaio et al. [36]	MD	1	☐	☒	TN	☒	☒	PDPTW-T (pickup and delivery problem with time windows and transfers)	Adopt the crowd-shipping characteristics in PDPTW-T	Routing cost	Adaptive large neighborhood search
Schneider and Löffler [27]	MD	1	☐	☐	—	☐	☐	CLRP (capacitated location-routing problem)	Proposed a tree-based search algorithm that is far more effective in dealing with CLRP compared to existing studies	Routing cost, depot cost, and vehicle cost	Tree-based search Algorithm$^{(*)}$
Ting and Chen [25]	MD	1	☐	☐	—	☒	☐	CLRP (capacitated location-routing problem)	—	Routing cost, depot opening cost, and vehicle cost	Multiple ant colony optimization

Table 6: Continued.

Literature	1D/MD	1E/2E	EI	CS	TN/CV	IR	HV	Problem name	Highlights	Objective function	Solution approach
Voigt and Kuhn [37]	MD	1	☐	☒	TN	☐	☒	PDPTOD (pickup and delivery problem with transshipments and occasional drivers)	Occasional drivers and transshipment points on multidepot scenarios	Routing cost, compensation cost on detours, and stops for ODs	Adaptive large neighborhood search
Voigt et al. [43]	Various experimental design		☐	☐	—	☐	☐	Various experimental design (two-echelon VRP (2E-VRP), location-routing problem (LRP), and multidepot VRP (MDVRP))	Proposed a hybrid ALNS that has higher impacts on solving 2EVRP, LRP, and MDVRP compared to existing studies	Operation cost	Hybrid adaptive large neighborhood search
Wang et al. [9]	1D	2	☒	☐	—	☐	☐	2E-CVRP-E (two-echelon capacitated vehicle routing problem with environmental considerations)	Consider environmental impact on 2E-CVRP	Routing cost, fuel cost, and satellites' handling cost	Variable neighborhood search and integer programming
Yu and Lin [29]	MD	1	☐	☐	—	☐	☐	OLRP (open location-routing problem)	—	Routing cost, depot opening cost, and vehicle cost	Simulated annealing
Yu et al. [26]	MD	1	☐	☐	—	☐	☐	LRP ((capacitated) location-routing problem)	—	Routing cost, depot opening cost, and vehicle cost	Simulated annealing
Yu et al. [16]	MD	1	☐	☐	—	☐	☐	2E-OLRP (two-echelon open location-routing problem)	Considering third-party logistics (TPL) and loading-unloading zones (LUZs)	Routing cost, satellites opening cost,and vehicle cost	Simulated annealing
Yu et al. [17]	1D	2	☐	☒	CV	☐	☒	2EVRPTW-CO-OD (two-echelon vehicle routing problem with time windows, covering options, and occasional drivers)	Covering locations with occasional drivers in the 2EVRP, delivery options	Routing costs, connection costs, and compensation for ODs	Adaptive large neighborhood search
Yu et al. [31]	1D	2	☐	☐	—	☐	☐	2EUDP-UV (two-echelon urban delivery problem with second level unmanned vehicles)	Adopt the "mothership approach" scenario; first-echelon will not deliver to customers nodes; rendezvous nodes	Routing cost, number of used large vehicles	Hybrid multi-start metaheuristic
Zhou et al. [18]	MD	2	☐	☐	—	☐	☐	MD-TEVRP-DO (multidepot two-echelon vehicle routing problem with delivery options)	Delivery options, multidepot 2EVRP, limited working time for vehicle	Routing cost, connection cost, handling cost, and vehicle cost	Hybrid Genetic algorithm
This work	1D	2	☐	☒	TN	☐	☒	2EVRP-TN-OD (two-echelon vehicle routing problem with transshipment nodes and occasional drivers)	Transshipment nodes in the 2EVRP framework with heterogeneous vehicles, ALNS for 2EVRP-TN	Routing costs, ODs employment cost	Adaptive large neighborhood search

Abbreviations: 1D, one depot; MD, multidepot; 1E, one-echelon; 2E, two-echelon; EI, environmental impact; CS, crowdsourcing; TN, transshipment nodes; CV, covering locations; IR, indirect related; HV, heterogeneous vehicles.

Table 7: Comparison between Gurobi and the proposed ALNS for solving Set 1 instances of 2EVRP-TN-OD.

Instance	Gurobi		Proposed ALNS			Avg. Gap (%)	Best gap (%)
	Cost	CPU (s)	Avg. 5	Best 5	Avg. CPU (s)		
ODE-n13-s2-m2-1	319.7	48,643	319.7	319.7	30	0	0
ODE-n13-s2-m2-2	291.19	7,692	291.19	291.19	30	0	0
ODE-n13-s2-m2-3	372.74	19,379	372.74	372.74	30	0	0
ODE-n13-s2-m2-4	321.58	11,940	321.58	321.58	30	0	0
ODE-n13-s2-m2-5	324.2	26,099	324.2	324.2	30	0	0
ODE-n13-s2-m2-6	273.83	3,489	273.83	273.83	30	0	0
ODE-n13-s2-m2-7	315.13	2,445	315.13	315.13	30	0	0
ODE-n13-s2-m2-8	300.54	5,933	300.54	300.54	30	0	0
ODE-n13-s2-m2-9	303.83	8,101	303.83	303.83	30	0	0
ODE-n13-s2-m2-10	334.28	58,091	334.27	334.27	30	0	0
Average	**315.7**	**19,181**	**315.7**	**315.7**	**30**	**0**	**0**

Table 8: Comparison between Gurobi and the proposed ALNS for solving Set 2 instances of 2EVRP-TN-OD.

Instance	BKS of 2EVRP	Gurobi	Proposed ALNS			Gap^+	Gap^*
			Best	Average	CPU (s)		
			Set 2a				
E-n22-k4-s6-17	417.07	329.64	328.18	328.21	60	−21.31	−0.44
E-n22-k4-s8-14	384.96	316.96	316.96	316.96	60	−17.66	0.00
E-n22-k4-s9-19	470.6	372.77	351.51	351.69	60	−25.31	−5.70
E-n22-k4-s10-14	371.5	277.09	277.09	277.09	60	−25.41	0.00
E-n22-k4-s11-12	427.22	367.62	350.58	350.58	60	−17.94	−4.64
E-n22-k4-s12-16	392.78	311.88	306.24	306.58	60	−22.03	−1.81
E-n33-k4-s1-9	730.16	606.01	511.93	513.07	120	−29.89	−15.52
E-n33-k4-s2-13	714.63	763.04	510.77	513.83	120	−28.53	−33.06
E-n33-k4-s3-17	707.48	—	512.48	512.95	120	−27.56	—
E-n33-k4-s4-5	778.74	761.61	555.73	557.81	120	−28.64	−27.03
E-n33-k4-s7-25	756.85	612.65	536.78	536.92	120	−29.08	−12.38
E-n33-k4-s14-22	779.05	713.55	573.87	583.34	120	−26.34	−19.58
			Set 2b				
E-n51-k5-s2-4-17-46	530.76	—	334.54	350.99	300	−36.97	—
E-n51-k5-s2-17	597.49	—	389.54	401.84	300	−34.80	—
E-n51-k5-s4-46	530.76	—	358.97	365.06	300	−32.37	—
E-n51-k5-s6-12	554.81	—	347.34	360.41	300	−37.39	—
E-n51-k5-s6-12-32-37	531.92	—	336.03	341.66	300	−36.83	—
E-n51-k5-s11-19	581.64	—	398.27	413.28	300	−31.53	—
E-n51-k5-s11-19-27-47	527.63	—	341.57	363.55	300	−35.26	—
E-n51-k5-s27-47	538.22	—	349.85	353.78	300	−35.00	—
E-n51-k5-s32-37	552.28	—	371.96	373.12	300	−32.65	—
			Set 2c				
E-n51-k5-s2-4-17-46	601.39	—	485.8	491.75	300	−19.22	—
E-n51-k5-s2-17	601.39	—	448.58	452.99	300	−25.41	—
E-n51-k5-s4-46	702.33	—	534.21	540.66	300	−23.94	—
E-n51-k5-s6-12	567.42	—	415.9	421.48	300	−26.70	—
E-n51-k5-s6-12-32-37	567.42	—	463.34	473.26	300	−18.34	—
E-n51-k5-s11-19	617.42	—	452.49	461.79	300	−26.71	—
E-n51-k5-s11-19-27-47	530.76	—	421.11	428	300	−20.66	—
E-n51-k5-s27-47	530.76	—	440.76	444.43	300	−16.96	—
E-n51-k5-s32-37	752.59	—	670.74	683.62	300	−10.88	—
Average			**423.10**	**429.02**		**−26.71**	

$Gap^+ = (ALNS^{Best} - \text{BKS of 2E} - VRP)/\text{BKS of 2E} - VRP \times 100\%$, $Gap^* = (ALNS^{Best} - Gurobi)/ALNS^{Best} \times 100\%$.

times. We provide BKS from solving the related 2EVRP instance for the purpose of analyzing the benefit of considering ODs and TNs.

Based on both Tables 8 and 9, Gurobi can only provide 11 and 10 feasible solutions for Sets 2 and 3, respectively. The results show that ALNS outperforms Gurobi in terms of both solution quality and computational time. While the time limit of Gurobi is 1 hour, the time limit of ALNS is set to 60–300 seconds, depending on the size of a particular instance.

Table 9: Comparison between Gurobi and the proposed ALNS for solving Set 3 instances of 2EVRP-TN-OD.

Instance	BKS of 2EVRP	Gurobi	Proposed ALNS			Gap^{+}	Gap*
			Best	Average	CPU (s)		
			Set 3a				
E-n22-k4-s13-14	526.15	556.14	457.51	457.65	60	−13.05	−17.74
E-n22-k4-s13-16	521.09	442.88	430.08	430.08	60	−17.47	−2.89
E-n22-k4-s13-17	496.38	485.54	438.44	438.44	60	−11.67	−9.70
E-n22-k4-s14-19	498.8	444.79	411.3	411.3	60	−17.54	−7.53
E-n22-k4-s17-19	512.81	491.22	443.2	443.37	60	−13.57	−9.78
E-n22-k4-s19-21	520.42	490.14	448.91	448.91	60	−13.74	−8.41
E-n33-k4-s16-22	672.17	744.65	469.01	469.88	120	−30.22	−37.02
E-n33-k4-s16-24	666.02	720.02	478.78	487.85	120	−28.11	−33.50
E-n33-k4-s19-26	680.37	503.14	450.44	450.44	120	−33.79	−10.47
E-n33-k4-s22-26	680.37	—	462.8	463.22	120	−31.98	—
E-n33-k4-s24-28	670.43	—	494.86	495.55	120	−26.19	—
E-n33-k4-s25-28	650.58	726.24	453.11	455.65	120	−30.35	−37.61
			Set 3b				
E-n51-k5-s12-18	690.59	—	494.18	506.34	300	−28.44	—
E-n51-k5-s12-41	683.05	—	506.84	520.78	300	−25.80	—
E-n51-k5-s12-43	710.41	—	538.31	548.97	300	−24.23	—
E-n51-k5-s39-41	728.54	—	606.12	611.77	300	−16.80	—
E-n51-k5-s40-41	723.75	—	633.6	640.05	300	−12.46	—
E-n51-k5-s40-43	752.15	—	631.78	643.05	300	−16.00	—
			Set 3c				
E-n51-k5-s13-19	560.73	—	359.91	372.4	300	−35.81	—
E-n51-k5-s13-42	564.45	—	421.6	425.68	300	−25.31	—
E-n51-k5-s13-44	564.45	—	461.58	468.5	300	−18.22	—
E-n51-k5-s40-42	746.31	—	594.25	603.82	300	−20.37	—
E-n51-k5-s41-42	771.56	—	682.28	698.77	300	−11.57	—
E-n51-k5-s41-44	802.91	—	664.06	693.47	300	−17.29	—
Average			**498.08**	**507.75**		**−21.50**	

The impacts of considering ODs and TNs are analyzed by comparing the BKS of 2EVRP and the best solution obtained by solving the related 2EVRP-TN-OD instance. Based on Tables 8 and 9, the existence of ODs and TNs on average reduces the total operational costs up to 26.71% and 21.50% for Sets 2 and 3, respectively. The higher cost resulting from 2EVRP occurs due to the longer distances that should be travelled by city freighters in order to serve all customers. In 2EVRP-TN-OD, some customers are more beneficially served by OD.

6.6. Sensitivity Analyses. We further elaborate on the impact of varying several characteristics of ODs on operational costs. We also analyze the impact of considering TNs in the system. The considered characteristics of ODs are described as follows:

(1) The variable cost (ρ_{OD}) and the fixed cost (H) of utilizing an OD.

(2) The number of available ODs, $|K_{\mathrm{OD}}|$.

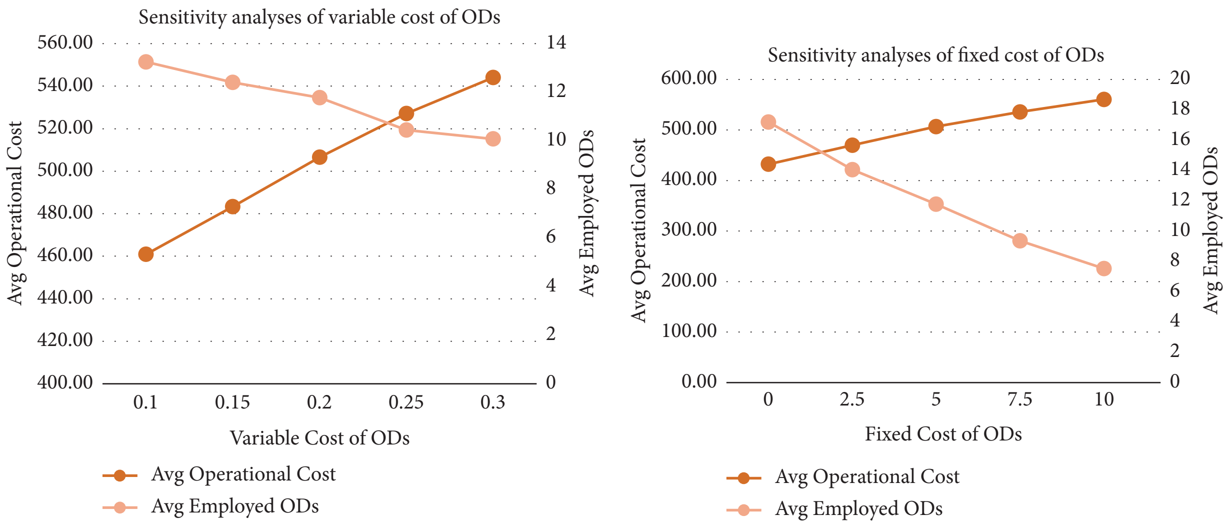
(3) The delivery flexibility of an OD, μ.

When the crowd-shipping system is implemented, the fixed and variable costs of an OD surely influence the total operational costs. Figures 3(a) and 3(b) show that changes in both fixed and variable costs lead to similar behavior. When the fixed/variable cost of an OD increases, the operational cost also increases while the number of employed ODs decreases.

Based on the observation in Figure 3, it is better to keep the fixed and variable costs as low as possible. However, from a practical perspective, fixed and variable costs will be one of the concerns when one implements such a system. Although the lower fixed and variable costs of an OD lead to lower operational costs, they also impact the number of available ODs. If the values are too low, then the number of available ODs will likely decrease.

Figure 4 shows the impact of varying the number of available ODs on the total operational costs. The lower the number of available ODs is, the higher the operational cost is. Based on the sensitivity analyses presented in both Figures 3 and 4, anyone who plans to implement the system needs to carefully design the fixed and variable costs of employing ODs so that the costs should be reasonable enough to attract ODs and to gain operational cost savings.

Figure 5 shows the sensitivity analyses of varying the flexibility of ODs. When the flexibility of ODs increases, the operational cost drops while the utilization of ODs increases. Similar behaviors are exhibited by varying the number of available ODs, as shown in Figure 4. However, both the marginal reduction of operational cost and the marginal improvement of the utilization of ODs are lower when the value of flexibility of ODs reaches 0.5 and the number of available ODs equals the number of customers. These



(a) (b)

Figure 3: Sensitivity analyses of the cost of ODs. (a) Sensitivity analyses of the variable cost of ODs. (b) Sensitivity analyses of the fixed cost of ODs.

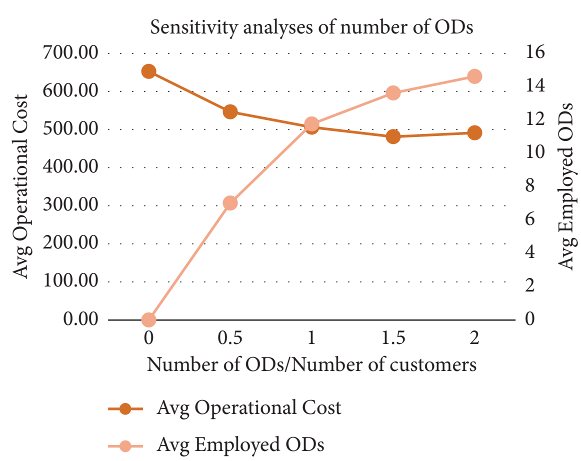


Figure 4: Sensitivity analyses of the number of ODs.

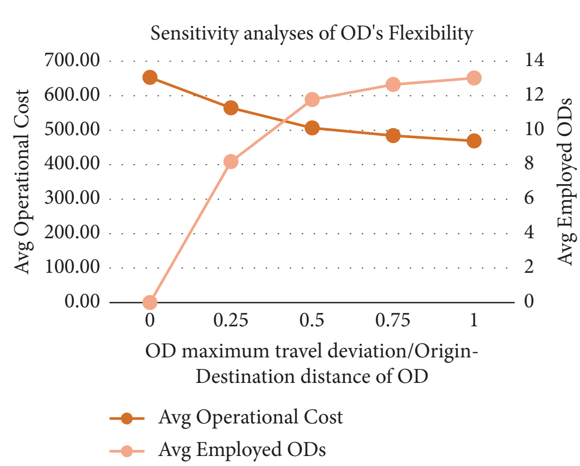


Figure 5: Sensitivity analyses of the flexibility of ODs.

Table 10: Comparison of the average operational costs of 2EVRP-TN-OD and 2EVRP-OD solutions.

	Average operational cost
Without TN	515.11
With TN	506.52
Difference (%)	1.67

phenomena indicate that the company that plans to implement this system needs to carefully design a compensation scheme to attract a reasonable number of ODs who offer a reasonable value of flexibility in order to leverage the crowd-shipping system.

The impact of TNs on operational costs is presented in Table 10. When TNs are introduced to the delivery system, there is an average cost reduction of 1.67%.

7. Conclusion

This research introduces a new variant of 2EVRP called 2EVRP-TN-OD by considering the existence of transshipment nodes (TNs) and occasional drivers (ODs). The available ODs are heterogeneous in terms of capacity. In addition, a set of TNs is introduced to leverage the ODs so that ODs may collect the assigned demands at either satellites or TNs. We developed MINLP and ALNS to deal with the problem. The experiments indicate that MINLP took a significantly large amount of computational time to solve small-scale instances; i.e., 19,181 seconds. The proposed ALNS combines the problem-specific destroy and repair operators with a local search procedure for the purpose of intensification. Our developed ALNS outperforms the performance of MINLP in terms of solution quality over a significantly shorter computational time. In addition, we benchmark our ALNS with the state-of-the-art algorithms for solving the 2EVRP benchmark instances. The results show that our ALNS provides a comparable result to BKS; i.e., 0.0% gap for both Sets 2 and 3 and 1.4% gap for Set 5.

The results show that considering TNs and ODs in a two-echelon distribution system is beneficial in terms of cost savings. By utilizing the developed 2EVRP-TN-OD instances, we compare the obtained results with the BKS of the associated 2EVRP instances. The results show that considering TNs and ODs can result in cost reductions of up to 26.71% and 21.50% for Sets 2 and 3, respectively.

To give a deeper understanding of how ODs affect the operational cost and decisions in a two-echelon distribution system, sensitivity analyses regarding three OD parameters are provided: (1) the cost of utilizing ODs, (2) the number of available ODs, and (3) the flexibility of ODs. As expected, the cost reduction can be further achieved by considering the TNs. The marginal improvements from adopting TNs in our research can be explained by various reasons (e.g., the generated instances for experimental designs); however, the benefit of adopting TNs has validated the correctness of our work compared to existing studies. In addition, other benefits of introducing TNs, as mentioned in Section 2.4, can be considered for future work.

Several interesting extensions can be deliberated due to the limitations of this study, resulting in a range of research opportunities. They include (1) designing various compensation schemes for ODs, (2) heterogeneous occasional drivers, (3) a multiobjective problem to analyze the impact of considering ODs toward both operational costs and environment-related objectives, (4) time windows of customers and ODs, and (5) the dynamic version to approach the realistic case of a system where only partial information is provided at the beginning of a planning period.

Data Availability

The detailed data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially supported by the Ministry of Science and Technology of the Republic of China (Taiwan) (Grant MOST 108-2221-E-011-051-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.

References

[1] J. Clement, "Global Retail E-Commerce Sales 2014-2023," 2019, https://www.statista.com/statistics/379046/worldwide-retail-e-commerce-sales/.

[2] S. Sampaio, M. Savelsbergh, L. P Veelenturf, and T. Van Woensel, "The benefits of transfers in crowdsourced pickup-and-delivery systems," *Optimization Online*, 2018, https://research.tue.nl/en/publications/the-benefits-of-transfers-in-crowdsourced-pickup-and-delivery-sys.

[3] A. Barr and j. Wohl, "Exclusive: Wal-Mart May Get Customers to Deliver Packages to Online Buyers," 2013, https://www.reuters.com/article/us-retail-walmart-delivery/exclusive-wal-mart-may-get-customers-to-deliver-packages-to-online-buyers-idUSBRE92R03820130328.

[4] M. Boccia, T. G. Crainic, A. Sforza, and C. Sterle, "Location-routing models for designing a two-echelon freight distribution system," *Rapport technique*, Université de Montréal, vol. 91, 2011.

[5] T. G. Crainic, "City logistics," in *State-of-the-art Decision-Making Tools in the Information-Intensive Age*, pp. 181–212, INFORMS, Catonsville, MD 21228, USA, 2008.

[6] T. G. Crainic, N. Ricciardi, and G. Storchi, "Advanced freight transportation systems for congested urban areas," *Transportation Research Part C: Emerging Technologies*, vol. 12, no. 2, pp. 119–137, 2004.

[7] T. G. Crainic, N. Ricciardi, and G. Storchi, "Models for evaluating and planning city logistics systems," *Transportation Science*, vol. 43, no. 4, pp. 432–454, 2009.

[8] G. Perboli, R. Tadei, and D. Vigo, "The two-echelon capacitated vehicle routing problem: models and math-based

heuristics," *Transportation Science*, vol. 45, no. 3, pp. 364–380, 2011.

[9] K. Wang, Y. Shao, and W. Zhou, "Matheuristic for a two-echelon capacitated vehicle routing problem with environmental considerations in city logistics service," *Transportation Research Part D: Transport and Environment*, vol. 57, pp. 262–276, 2017.

[10] O. Belgin, I. Karaoglan, and F. Altiparmak, "Two-echelon vehicle routing problem with simultaneous pickup and delivery: mathematical model and heuristic approach," *Computers & Industrial Engineering*, vol. 115, pp. 1–16, 2018.

[11] U. Breunig, R. Baldacci, R. F. Hartl, and T. Vidal, "The electric two-echelon vehicle routing problem," *Computers & Operations Research*, vol. 103, pp. 198–210, 2019.

[12] W. Jie, J. Yang, M. Zhang, and Y. Huang, "The two-echelon capacitated electric vehicle routing problem with battery swapping stations: formulation and efficient methodology," *European Journal of Operational Research*, vol. 272, no. 3, pp. 879–904, 2019.

[13] A. Anderluh, P. C. Nolz, V. C. Hemmelmayr, and T. G. Crainic, "Multi-objective optimization of a two-echelon vehicle routing problem with vehicle synchronization and 'grey zone'customers arising in urban logistics," *European Journal of Operational Research*, vol. 289, no. 3, pp. 940–958, 2021.

[14] D. L. Enthoven, B. Jargalsaikhan, K. J. Roodbergen, M. A. uit het Broek, and A. H. Schrotenboer, "The two-echelon vehicle routing problem with covering options: city logistics with cargo bikes and parcel lockers," *Computers & Operations Research*, vol. 118, Article ID 104919, 2020.

[15] P. Kitjacharoenchai, B.-C. Min, and S. Lee, "Two echelon vehicle routing problem with drones in last mile delivery," *International Journal of Production Economics*, vol. 225, Article ID 107598, 2020.

[16] S. Yu, J. Puchinger, and S. Sun, "Two-echelon urban deliveries using autonomous vehicles," *Transportation Research Part E: Logistics and Transportation Review*, vol. 141, Article ID 102018, 2020a.

[17] V. F. Yu, P. Jodiawan, M.-L. Hou, and A. Gunawan, "Design of a two-echelon freight distribution system in last-mile logistics considering covering locations and occasional drivers," *Transportation Research Part E: Logistics and Transportation Review*, vol. 154, Article ID 102461, 2021.

[18] L. Zhou, R. Baldacci, D. Vigo, and X. Wang, "A multi-depot two-echelon vehicle routing problem with delivery options arising in the last mile distribution," *European Journal of Operational Research*, vol. 265, no. 2, pp. 765–778, 2018.

[19] F. Mühlbauer and P. Fontaine, "A parallelised large neighbourhood search heuristic for the asymmetric two-echelon vehicle routing problem with swap containers for cargo-bicycles," *European Journal of Operational Research*, vol. 289, no. 2, pp. 742–757, 2021.

[20] N. Sluijk, A. M. Florio, J. Kinable, N. Dellaert, and T. Van Woensel, "Two-echelon vehicle routing problems: a literature review," *European Journal of Operational Research*, vol. 304, no. 3, pp. 865–886, 2023.

[21] G. Nagy and S. Salhi, "Location-routing: issues, models and methods," *European Journal of Operational Research*, vol. 177, no. 2, pp. 649–672, 2007.

[22] R. Baldacci, A. Mingozzi, and R. Wolfler Calvo, "An exact method for the capacitated location-routing problem," *Operations Research*, vol. 59, no. 5, pp. 1284–1296, 2011.

[23] C. Contardo, J.-F. Cordeau, and B. Gendron, "An exact algorithm based on cut-and-column generation for the capacitated location-routing problem," *INFORMS Journal on Computing*, vol. 26, no. 1, pp. 88–102, 2014.

[24] V. C. Hemmelmayr, J.-F. Cordeau, and T. G. Crainic, "An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics," *Computers & Operations Research*, vol. 39, no. 12, pp. 3215–3228, 2012.

[25] C.-J. Ting and C.-H. Chen, "A multiple ant colony optimization algorithm for the capacitated location routing problem," *International Journal of Production Economics*, vol. 141, no. 1, pp. 34–44, 2013.

[26] V. F. Yu, S.-W. Lin, W. Lee, and C.-J. Ting, "A simulated annealing heuristic for the capacitated location routing problem," *Computers & Industrial Engineering*, vol. 58, no. 2, pp. 288–299, 2010.

[27] M. Schneider and M. Löffler, "Large composite neighborhoods for the capacitated location-routing problem," *Transportation Science*, vol. 53, no. 1, pp. 301–318, 2019.

[28] I. Karaoglan, F. Altiparmak, I. Kara, and B. Dengiz, "The location-routing problem with simultaneous pickup and delivery: formulations and a heuristic approach," *Omega*, vol. 40, no. 4, pp. 465–477, 2012.

[29] V. F. Yu and S.-Y. Lin, "A simulated annealing heuristic for the open location-routing problem," *Computers & Operations Research*, vol. 62, pp. 184–196, 2015.

[30] K. Pichka, A. H. Bajgiran, M. E. Petering, J. Jang, and X. Yue, "The two echelon open location routing problem: mathematical model and hybrid heuristic," *Computers & Industrial Engineering*, vol. 121, pp. 97–112, 2018.

[31] V. F. Yu, Winarno, S.-W. Lin, S. W. Lin, and A. Gunawan, "Design of a two-echelon freight distribution system in an urban area considering third-party logistics and loading–unloading zones," *Applied Soft Computing*, vol. 97, Article ID 106707, 2020b.

[32] F. Arnold and K. Sörensen, "A progressive filtering heuristic for the location-routing problem and variants," *Computers & Operations Research*, vol. 129, Article ID 105166, 2021.

[33] C. Archetti, M. Savelsbergh, and M. G. Speranza, "The vehicle routing problem with occasional drivers," *European Journal of Operational Research*, vol. 254, no. 2, pp. 472–480, 2016.

[34] G. Macrina, L. D. P. Pugliese, F. Guerriero, and D. Laganà, "The vehicle routing problem with occasional drivers and time windows," in *Proceedings of the Paper presented at the International Conference on Optimization and Decision Science*, Sorrento, Italy, September 2017.

[35] W. Chen, M. Mes, and M. Schutten, "Multi-hop driver-parcel matching problem with time windows," *Flexible Services and Manufacturing Journal*, vol. 30, no. 3, pp. 517–553, 2018.

[36] A. Sampaio, M. Savelsbergh, L. P. Veelenturf, and T. J. N. Van Woensel, "Delivery systems with crowd-sourced drivers: a pickup and delivery problem with transfers," *Networks*, vol. 76, no. 2, pp. 232–255, 2020.

[37] S. Voigt and H. J. N. Kuhn, "Crowdsourced logistics: the pickup and delivery problem with transshipments and occasional drivers," *Networks*, vol. 79, no. 3, pp. 403–426, 2022.

[38] K. Huang and M. N. Ardiansyah, "A decision model for last-mile delivery planning with crowdsourcing integration," *Computers & Industrial Engineering*, vol. 135, pp. 898–912, 2019.

[39] N. Kafle, B. Zou, and J. Lin, "Design and modeling of a crowdsource-enabled system for urban parcel relay and delivery," *Transportation Research Part B: Methodological*, vol. 99, pp. 62–82, 2017.

[40] G. Macrina, L. Di Puglia Pugliese, F. Guerriero, and G. Laporte, "Crowd-shipping with time windows and transshipment nodes," *Computers & Operations Research*, vol. 113, Article ID 104806, 2020.

[41] V. F. Yu, P. Jodiawan, and A. P. Redi, "Crowd-shipping problem with time windows, transshipment nodes, and delivery options," *Transportation Research Part E: Logistics and Transportation Review*, vol. 157, Article ID 102545, 2022.

[42] U. Breunig, V. Schmid, R. F. Hartl, and T. Vidal, "A large neighbourhood based heuristic for two-echelon routing problems," *Computers & Operations Research*, vol. 76, pp. 208–225, 2016.

[43] S. Voigt, M. Frank, P. Fontaine, and H. Kuhn, "Hybrid adaptive large neighborhood search for vehicle routing problems with depot location decisions," *Computers & Operations Research*, vol. 146, Article ID 105856, 2022.

[44] S. Ropke and D. Pisinger, "An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows," *Transportation Science*, vol. 40, no. 4, pp. 455–472, 2006.

[45] S. Akpinar and Ş. Akpinar, "Hybrid large neighbourhood search algorithm for capacitated vehicle routing problem," *Expert Systems with Applications*, vol. 61, pp. 28–38, 2016.