Advances in Industrial Engineering, December 2025, 59(2): 401-424
DOI: 10.22059/aie.2025.388216.1932
RESEARCH PAPER

Crowd-Based Multi-Echelon Routing Problem with Roaming Delivery and Mobile Intermediate Transfer Locations

Sadjad Khalesi[1] and Mohammad Reza Akbari Jokar[2*]
[1]*Ph.D. Candidate, Industrial Engineering Department, Sharif University of Technology, Tehran, Iran.*
[2]*Professor, Industrial Engineering Department, Sharif University of Technology, Tehran, Iran.*

Received: 5 January 2025, Revised: 13 April 2025, Accepted: 29 April 2025

Abstract

The Vehicle Routing Problem with Roaming Delivery Locations (VRPRDL) represents a recent innovation in last-mile delivery, wherein a customer's order is delivered to the trunk of their vehicle, which may be parked at various locations across different time windows. In this Paper, we introduce a novel crowd-based multi-echelon variant of the vehicle routing problem with roaming delivery locations (crowd-based ME-VRPRDL). This model integrates a flexible multi-echelon logistics structure with hybrid intermediate transfer locations (satellites), which can be either mobile or stationary. The flexibility in our approach allows the optimal solution to dynamically adapt between single-echelon and multi-echelon configurations, depending on the specific problem parameters and constraints. In the proposed model, crowd shippers—individuals who assist with deliveries—are assigned to intermediate satellites based on their availability and time windows, enabling more efficient and dynamic resource allocation. To address the complexity of this problem, we develop an innovative heuristic algorithm that combines node classification with a greedy optimization approach. This algorithm is particularly tailored to handle the unique challenges posed by occasional crowd shippers and hybrid satellite configurations. Our findings demonstrate that the integration of multi-echelon logistics systems with crowd shipping and strategically placed satellites offers significant potential to optimize last-mile delivery operations. Specifically, it reduces delivery costs and travel times while leveraging underutilized resources in the logistics network. The study underscores the value of combining traditional and crowd-based delivery mechanisms in achieving more sustainable and cost-effective solutions for modern logistics challenges.

Keywords:
Crowd Logistics,
Last Mile Delivery,
Mobile Satellite,
Occasional Shippers,
Roaming Delivery.

Introduction

The modern business environment increasingly emphasizes flexibility in customer service, particularly within logistics, where accommodating customers' preferred delivery locations and times has become a cornerstone of competitive success. This operational flexibility enhances cost efficiency through improved planning while simultaneously fostering customer satisfaction by offering personalized delivery options (Tilk, Olkis and Irnich 2021). Urbanization has further amplified the demand for goods and services in cities, necessitating the expansion of

[*] Corresponding author: (Mohammad Reza Akbari Jokar)
Email: reza.akbari@sharif.edu

urban logistics (Sampaio Oliveira, et al. 2019). Recently, last-mile delivery has emerged as a critical focus area, significantly influencing urban residents' quality of life. Consequently, researchers have dedicated substantial efforts to developing innovative and sustainable solutions to optimize last-mile delivery systems.

A variety of advancements in last-mile logistics have been explored, including the use of unmanned aerial vehicles (drones) (Lemardelé, et al. 2021), autonomous delivery robots (Ostermeier, Heimfarth and Hübner 2022), mobile and stationary parcel lockers (Lin, et al. 2022), in-trunk delivery systems (Reyes, Savelsbergh and Toriello 2017), and crowd shipping (Mousavi, Bodur and Roorda 2022). These innovations aim to improve efficiency, reduce costs, and meet growing consumer expectations. Delivery modes generally fall into two categories: (1) home delivery and (2) delivery to secondary locations, such as parcel lockers in commercial centers or transit stations ((Janjevic, Winkenbach and Merchán 2019); (van Duin, et al. 2020)). While traditional parcel lockers are static, mobile lockers—capable of dynamic relocation—have introduced greater convenience and flexibility for customers (Schwerdfeger and Boysen 2020).

Advancements in web-based technologies have facilitated the integration of crowd logistics, enabling companies to utilize non-professional agents, or crowd shippers, who leverage their spare capacity to complete deliveries in exchange for compensation. Platforms like Uber and Airbnb exemplify this scalable and cost-effective model, though challenges such as irregular availability of crowd shippers remain ((Boysen, Fedtke and Schwerdfeger 2021); (Sampaio, et al. 2020)). Hybrid planning models that combine different transport agents and resources have been proposed to address these complexities. Additionally, the in-trunk delivery system, introduced as the Vehicle Routing Problem with Roaming Delivery Locations (VRPRDL) by Reyes et al. (2017) (Reyes, Savelsbergh and Toriello 2017), represents a novel approach where customers designate their vehicles as flexible, mobile delivery points.

This study builds upon the VRPRDL framework by proposing a flexible multi-echelon vehicle routing problem with roaming delivery locations (ME-VRPRDL). Key contributions include the development of a mathematical model that dynamically determines the number of echelons, integration of mobile parcel lockers and crowd shippers to enhance adaptability, consideration of crowd shipper spatial constraints, and a heuristic algorithm to solve large-scale instances. The remainder of this paper is structured as follows: Section 2 reviews relevant literature, Section 3 details the proposed model, Section 4 outlines the heuristic algorithm, Section 5 presents computational results, and Section 6 concludes the study with directions for future research.

Related Literature

The Vehicle Routing Problem (VRP), first introduced by Dantzig and Ramser (1959) (Dantzig and Ramser 1959), has been extensively studied and expanded over the years to address various real-world complexities. Among its well-known extensions are the Capacitated VRP (CVRP), VRP with Time Windows (VRPTW), VRP with Backhauls (VRPPB), Split Delivery VRP (SDVRP), Dynamic VRP (DVRP), VRP with Simultaneous Pickup and Delivery (VRPSPD), Open VRP (OVRP), and Time-Dependent VRP (TDVRP) (Braekers, Ramaekers and Nieuwenhuyse 2016). A notable variation, the Vehicle Routing Problem with Roaming Delivery Locations (VRPRDL), is a specialized case of the Generalized VRP with Time Windows (GVRPTW), first introduced by Moccia et al. (2012) (Moccia, Cordeau and Laporte 2012). This problem combines VRPTW and GVRP (Ozbaygin, et al. 2017).

Ozbaygin et al. (2017) (Ozbaygin, et al. 2017) approached the deterministic VRPRDL problem using a set coverage problem framework and proposed a branch-price algorithm to optimize it. They demonstrated that hybrid delivery strategies—combining home delivery and in-trunk delivery—can reduce costs by an average of 20%. However, real-world uncertainties, such as weather changes or vehicle breakdowns, can disrupt optimal routing plans, necessitating research into stochastic travel times. The VRPRDL with Stochastic Travel Times (VRPRDL-STT), first introduced by Lombard et al. (2018) (Lombard, Tamayo-Giraldo and Fontane 2018), has since been expanded with various modeling and solution approaches ((He, et al. 2020); (Sampaio Oliveira, et al. 2019). Another significant development is the Dynamic VRPRDL (D-VRPRDL), where customer locations may change during delivery execution. Ozbaygin and Savelsbergh (2019) (Ozbaygin and Savelsbergh 2019) addressed this problem using an iterative re-optimization approach based on branch-and-price algorithms. The Multi-Depot Vehicle Routing Problem with Roaming Delivery Locations Considering Hard Time Windows (MDVRPRDL-HTW), as introduced by Jolfaei and Alinaghian (2024) (Jolfaei and Alinaghian 2024), extends the VRPRDL by incorporating multiple depots, roaming delivery locations, and hard time windows.

In parallel, last-mile delivery optimization has emerged as a critical focus area, particularly with dual-service strategies like home delivery and parcel lockers (Zhou, He and Zhou 2019). Parcel lockers are classified into public/private, mechanical/electronic, and stationary/mobile categories (Zurel, et al. 2018). Research by Lemke et al. (2016) (Lemke, Iwan and Korczak 2016) highlights customer preferences for parcel lockers near homes or along commuting routes. Lachapelle et al. (2018) (Lachapelle, et al. 2018) explored the urban planning implications of locker placement, identifying commercial, suburban, postal, and shopping center locations as common sites. Bilik (2014) (Bilik 2014) demonstrated that parcel lockers can reduce delivery distances, increase daily delivery volumes, and lower CO2 emissions and fuel consumption.

A recent innovation in last-mile delivery is the adoption of crowd logistics, exemplified by Amazon Flex. In this model, occasional drivers, or "crowd shippers," use their vehicles to deliver parcels, enabling cost-effective and flexible distribution (Archetti, Guerriero and Macrina 2021). Crowd logistics relies on platforms to outsource delivery tasks, compensating drivers for completed deliveries without formal employment contracts (Ranard, et al. 2014). Studies show that crowd logistics can enhance economic, social, and environmental sustainability (Buldeo Rai, et al. 2017) and improve network efficiency (Klumpp 2017). However, challenges such as resource availability and planning complexity persist (Sampaio, et al. 2020). Multi-echelon distribution systems, incorporating intermediate transfer points or "satellites," have been proposed to address these limitations (Kafle, Zou and Lin 2017).

Recent research by Liao et al. (Liao, Dai and Ma 2025), published in 2025, introduces a Vehicle Routing Problem with Roaming Locations for Simultaneous Pickup and Delivery Services (VRPRL-SPD). This study formulates the problem as a mixed integer linear programming (MILP) model aimed at minimizing total travel cost. To solve this complex problem, the authors developed a two-stage metaheuristic approach that combines a random selection greedy insertion algorithm with a large neighborhood search algorithm. Additionally, Saker et al. (2023) (Saker, Eltawil and Ali 2023) introduced the Capacitated Vehicle Routing Problem with Delivery Options (CVRPDO), which incorporates parcel lockers as an alternative delivery option to reduce last-mile delivery costs. Their study proposed an Adaptive Large Neighborhood Search (ALNS) metaheuristic to solve the problem efficiently, outperforming exact solutions provided by a Mixed Integer Programming (MIP) model. Their work highlights the potential of integrating delivery options into VRP frameworks to enhance flexibility and reduce operational costs.

In summary, advancements in VRP and last-mile delivery systems have focused on integrating hybrid strategies, stochastic factors, and crowd logistics to enhance efficiency and sustainability. These innovations continue to shape the logistics industry, balancing cost optimization with environmental and social considerations. Below, in Table 1, a summary of related articles and their comparison with our research is presented.

In general, logistics services play a key role in providing good living conditions in a city. At the same time, the expansion of urbanization also brings disadvantages such as high noise, widespread pollution, heavy traffic, and congestion. Therefore, the presence of intelligent and innovative solutions in urban logistics management will be needed so that cities can remain attractive to their residents. In this paper and in the first phase, the VRPRDL problem presented by Reyes et al. (2017) (Reyes, Savelsbergh and Toriello 2017) is considered as an initial last mile delivery system and an extended version of it considering (1) a flexible two-echelon network structure, (2) accessibility of crowd shippers at certain shifts and paths and (3) the simultaneous operation of stationary and mobile parcel lockers as intermediate satellites was mathematically modeled.

On the other hand, since the problem is a NP-hard model, in the next phase, we will introduce a heuristic approach and compare its results with exact algorithm in various instances. So, the main contributions of our research are summarized as follows:

- This study introduces a novel extension of the VRPRDL called Crowd-based ME-VRPRDL for the first time.
- We consider crowd shippers and mobile parcel lockers as two innovative solutions to improve the capabilities of our ME-VRPRDL initial model and make it more comprehensive. Hence, we develop a mathematical model for a crowd-based ME-VRPRDL with stationary and mobile parcel lockers.
- We consider crowd shippers' covering radius limitation for parcel assignment to each crowd shipper.
- We develop a heuristic algorithm based on node classification and greedy approach to solve a crowd-based multi-echelon vehicle routing problem with occasional crowd shippers and hybrid satellites (mobile and stationary).

Table 1. summary of related articles

Paper	Customers locations	Model Type	Itinerary	Algorithm Type	Process of Algorithm	Echelon	Satellites Type	Fleets Type
(Reyes, Savelsbergh and Toriello 2017)	Roaming	Deterministic	Fixed	Heuristic	Offline	Single	_	Ownership
(Ozbaygin, et al. 2017)	Roaming	Deterministic	Fixed	Exact	Offline	Single	_	Ownership
(Kafle, Zou and Lin 2017)	Fixed	Deterministic	Static	Metaheuristic	Offline	Multi	Stationary	Ownership + crowd shippers
(Lombard, Tamayo-Giraldo and Fontane 2018)	Roaming	Stochastic	Fixed	Metaheuristic	Offline	Single	_	Ownership
(Ozbaygin and Savelsbergh 2019)	Roaming	Deterministic	Dynamic	Exact	Online	Single	_	Ownership
(Sampaio Oliveira, et al. 2019)	Roaming	Stochastic	Fixed	Heuristic	Offline	Single	_	Ownership
(He, et al. 2020)	Roaming	Stochastic	Fixed	Metaheuristic	Offline	Single	_	Ownership
(Sampaio, et al. 2020)	Fixed	Deterministic	Static	Heuristic	Offline	Multi	Stationary	Crowd shippers
(Archetti, Guerriero and Macrina 2021)	Fixed	Deterministic	Static	Heuristic	Online	Single	_	Ownership + Crowd shippers
(Mousavi, Bodur and Roorda 2022)	Fixed	Stochastic	Static	Heuristic	Offline	Multi	Mobile	Ownership + Crowd shippers
This Work	Roaming	Deterministic	Fixed	Heuristic	Offline	Multi	Stationary + Mobile	Ownership + Crowd shippers

Problem Description

The Vehicle Routing Problem with Roaming Delivery Locations (VRPRDL) is a novel variant of the well-known Vehicle Routing Problem (VRP) that models an innovative last-mile delivery system where customer orders are delivered to the trunk of their cars, rather than to fixed home addresses. In traditional VRP, delivery locations for customers are fixed, but in VRPRDL, each customer has a set of potential delivery locations, which may vary based on their daily schedule (e.g., home, work, or other destinations). The service provider must decide not only the sequence of deliveries but also the exact location and time for each delivery based on the customer’s geographic profile and time windows. These time windows reflect when a customer’s car is present at each location, and they are non-overlapping due to the customer's travel itinerary. The objective is to minimize the total delivery cost, typically measured by the total distance traveled, while ensuring that all customer demands are met, vehicle capacities are not exceeded, and deliveries are made within the allowed time windows.

The VRPRDL introduces several key assumptions that distinguish it from traditional VRP. First, each customer has a predefined set of potential delivery locations, along with time windows indicating when their vehicle will be present at each location. The time windows are determined by the customer’s itinerary, and travel between locations is accounted for, making the problem time-dependent. Second, customers can only receive one delivery during the planning period, and the delivery vehicles must start and end their routes at a central depot. Third, the problem assumes that delivery locations satisfy the triangle inequality for travel times and costs, and vehicle capacities are homogeneous and limited. The VRPRDL can be viewed as a special case of the generalized VRP with time windows, where clusters of delivery locations (specific to each customer) have non-overlapping time windows. These unique characteristics make the VRPRDL a challenging optimization problem with practical applications in improving last-mile delivery efficiency, reducing costs, and mitigating environmental impacts.

The provision of flexible time windows and delivery locations can increase customer satisfaction by offering a wider range of delivery options. Additionally, it presents opportunities for service providers to enhance their last mile delivery operations. As a result, the VRPRDL model exhibits potential advantages when applied in real-world scenarios.

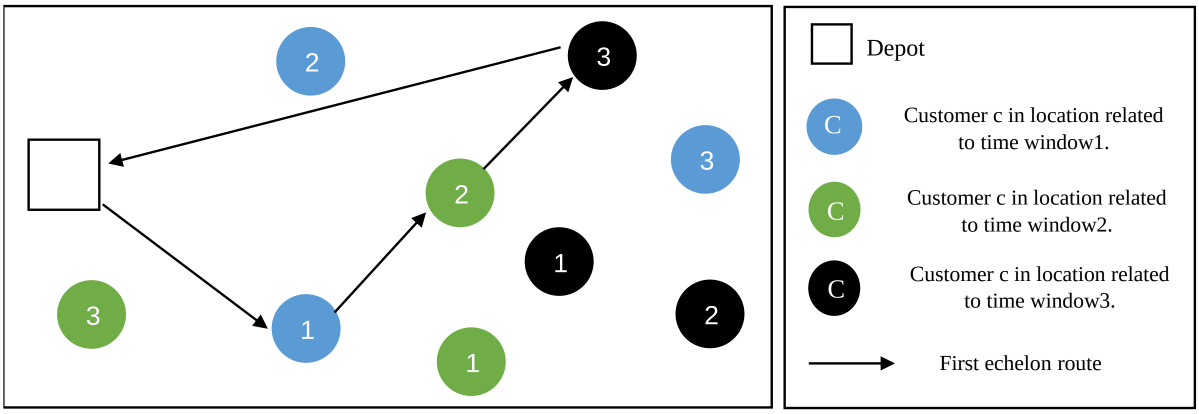


Figure 1. A sample VRPRDL problem

However, despite the advantages offered by these problem types, they also present certain operational challenges. One such challenge arises from the time window restrictions imposed by customers. Meeting the demands of all customers within their specified time windows may sometimes require increased total distance traveled or the use of additional delivery vehicles. Consequently, these situations can lead to higher operational costs. Therefore, it is crucial to seek solutions that not only provide flexible delivery options to customers but also consider the economic factors for service providers when implementing such models. Two potential solutions that show promise in improving the efficiency of the roaming delivery option under these circumstances are the utilization of crowd shippers and intermediate transfer locations called satellites. For example, by considering the availability of crowd shippers and satellites, the problem depicted in Figure 1 can be modified as described below.

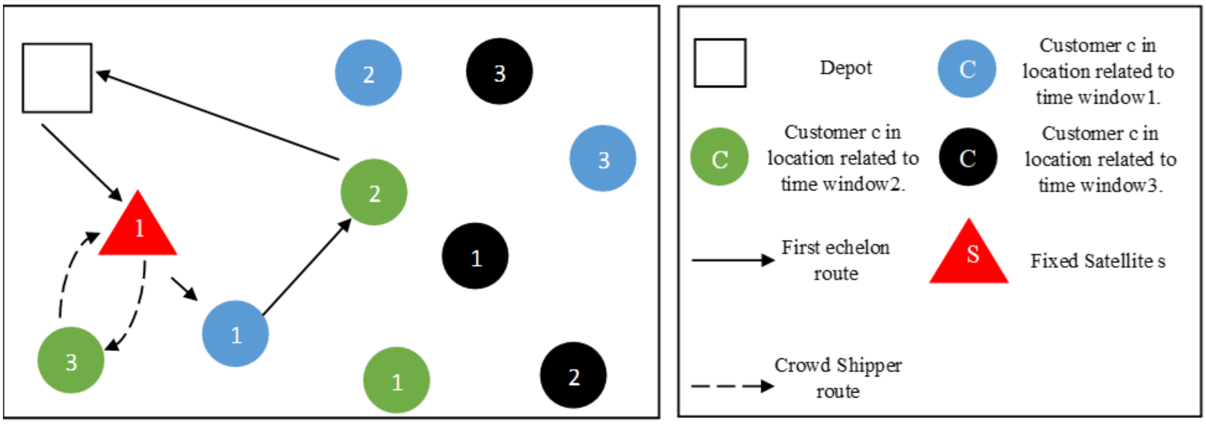


Figure 2. A sample VRPRDL problem with crowdshipper and fixed satellite

So in this section, according to basic VRPRDL modeling technologies which is provided by Reyes et al. (2017) (Reyes, Savelsbergh, & Toriello, 2017), we formulated a flexible multi-echelon extension of the VRPRDL problem (ME-VRPRDL), and then in the next step, we added crowd shippers to fulfill a part of customer orders, as well as stationary and mobile satellites to supply crowd shippers. Crowd shippers in this problem have a duty time window that will be announced by themselves in advance and in that period, they will be available only in a predetermined geographic radius. Hence, given the possibility of unavailability of crowd shippers (CSs) in certain scenarios, the first-echelon vehicles (FEVs) will possess the capability to directly deliver parcels to the final customers. Conversely, when CSs are available and their utilization is deemed optimal, parcels will be dispatched to satellites, where the assigned crowd shippers will collect and subsequently deliver them to the final customers. Furthermore, the main assumptions considered in mathematical modeling are as follows:

- The logistics system investigated in this research will focus on non-perishable parcels.
- We do not consider same day delivery policies.
- It is assumed that it is possible to receive a valid itinerary with time windows from customers as one of the input parameters.
- The scope of the research is focused on e-commerce logistics operations.
- Each crowd shipper is only assigned to one satellite, which will be the starting and ending point of its route.
- All customers' demands should be satisfied.

Crowed-Based Multi-Echelon VRPRDL Formulation

With the above-mentioned assumptions and the presented basic mathematical model in (Reyes, Savelsbergh and Toriello 2017) as a reference model, a MINLP model is proposed to solve the Crowed-based multi-echelon VRPRDL problem (C-MEVRPRDL). Let G = (V, A) denote a complete directed graph with node set V and arc set A, where node set V is a combination of the main depot, satellites and customers' locations that is shown in equation 1.

$$\boldsymbol{V} = \{\boldsymbol{0}\} \cup \boldsymbol{V^s} \cup \boldsymbol{V^c} = \{\boldsymbol{depot\ location}\} \cup \{\boldsymbol{satellites'locations}\} \cup \{\boldsymbol{customers'locations}\} \qquad (1)$$

The set 〖A=A〗_1 denotes the set of arcs in first echelon and the set A_2 is considered for second echelon arcs and determined by equations 2-3.

$$\boldsymbol{A_1} = \{(\boldsymbol{i},\boldsymbol{j})|\boldsymbol{i},\boldsymbol{j} \in \boldsymbol{V}, \boldsymbol{i} \neq \boldsymbol{j}\} \quad (2)$$
$$A_2 = \{(i,j) \in A|i,j \in V\backslash\{0\}, i \neq j\}\backslash\{(i,j) \in A|i,j \in V^s, i \neq j\} \quad (3)$$

To formulate the C-MEVRPRDL model, the following notations have been used:

Table 2. List of indices and sets

$\boldsymbol{indices}$:
i: $index\ of\ all\ nodes$
j: $index\ of\ first\ echelon\ vehicles$
l: $index\ of\ crowd\ shippers$
Cs: $crowd\ shippers\ set$
fev: $first\ echelon\ vehicles\ set$
s: $index\ of\ sattelites$
V^s: $satellites'\ locations\ set$
V^c: $customers'\ locations\ set$
c': $customers'\ index$
c: $customers'\ locations\ index$

Table 3. List of parameters

$\boldsymbol{Parameters}$
$\lvert S\rvert$: $number\ of\ sattelite$
$\lvert C\rvert$: $number\ of\ customers$
M^{fev}: $capacity\ of\ fev\ vehicle$
M^{cs_l}: $capacity\ of\ cs\ vehicle\ l$
T: $time\ horizon\ of\ planning\ period$
$GP_{c'}$: $geographic\ profile\ of\ customer\ c'$; $GP_{c'} \subseteq V^c$; $\lvert GP_{c'}\rvert = g\ for\ all\ customer\ c'$
$GR_l = (O_x^l, O_y^l, \alpha^l)$: $geographical\ region\ of\ Cs_l\ presence\ to\ serve\ customers$
O_x^l: $longitude\ coordination\ of\ Cs\ l$
O_y^l: $latitude\ coordination\ of\ Cs\ l$
α^l: $Covering\ radius\ of\ Cs\ l$
$long_i$: $longitude\ coordinate\ of\ node\ i \in V^s \cup V^c$
lat_i: $latitude\ coordinate\ of\ node\ i \in V^s \cup V^c$
BM^d: $big\ value\ (m)\ for\ distance$
g: $customer\ geographical\ profile\ size$
p_s: $handling\ cost\ of\ sattelite\ s$
d_i: $demand\ of\ node\ i$
a_i: $lower\ bound\ of\ time\ window\ for\ node\ i \in V^s \cup V^c$
b_i: $upper\ bound\ of\ time\ window\ for\ node\ i \in V^s \cup V^c$
e_l: $lower\ bound\ of\ available\ time\ for\ CS\ l$
f_l: $upper\ bound\ of\ available\ time\ for\ CS\ l$
$c_{i,j}$: $routing\ cost\ from\ i\ to\ j\ by\ first\ echelon\ vehicle (fev)$
$w_{i,j}$: $routing\ cost\ from\ i\ to\ j\ by\ crowd\ shipper (cs)$
$t_{i,j}^1$: $travel\ time\ which\ is\ related\ to\ (i,j) \in A_1\ by\ fev\ at\ first\ echelon$
$t_{i,j,l}^2$: $travel\ time\ which\ is\ related\ to\ (i,j) \in A_2\ by\ crowdshipper\ l\ at\ second\ echelon$

Table 4. List of variables

$\boldsymbol{Variables}$
$X_{i,j,k}^1 = \begin{cases} 1 & if\ k \in fev\ traverses\ arc\ (i,j) \in A_1 \\ 0 & O.W \end{cases}$
$X_{i,j,s,l}^2 = \begin{cases} 1 & if\ arc\ (i,j) \in A_2\ is\ traversed\ from\ satellite\ s \in V^s\ by\ Cs_l \\ 0 & O.W \end{cases}$
$q_{i,k} \in R^+$: $Number\ of\ parcels\ transported\ from\ the\ depot\ to\ the\ node\ i \in V^s \cup V^c\ by\ vehicle\ k \in fev$
$h_{i,j,s,l} \in R^+$: $The\ load\ of\ Cs_l\ dispatched\ from\ satellite\ s\ travels\ via\ arc(i,j) \in A_2$
$\tau_i^1 \in [0,T]$: $time\ of\ departure\ after\ visiting\ node\ i \in V^s \cup V^c\ by\ a\ fev$
$\tau_{i,l}^2 \in [0,T]$: $time\ of\ departure\ after\ visiting\ node\ i \in V^c\ by\ crowd\ shipper\ l\ at\ second\ echelon$
$u_{i,k} \in Z^+$: $auxiliary\ variable\ for\ eliminating\ subtours\ in\ the\ first\ echelon$

The mathematical formulation for C-MEVRPRDL is then given as follows:

$$Min\ Z = \sum_{k\in fev}\sum_{(i,j)\in A_1} c_{i,j}.X^1_{i,j,k} + \sum_{s\in V^S}\sum_{(i,j)\in A_2}\sum_{l\in Cs} w_{i,j}.X^2_{i,j,s,l} + \sum_{k\in fev}\sum_{s\in V^S} p_s.q_{s,k} \tag{4}$$

S.t.

$$\sum_{(i,j)\in A_1} X^1_{i,j,k} = \sum_{(j,i)\in A_1} X^1_{j,i,k} \qquad \forall i \in V^s \cup V^c \cup \{0\}, k \in fev \tag{5}$$

$$\sum_{(i,j)\in A_1} X^1_{i,j,k} \le 1 \qquad \forall i \in V^s \cup V^c \cup \{0\}, k \in fev \tag{6}$$

$$u_{i,k} - u_{j,k} + (|S|+|C|).X^1_{i,j,k} \le (|S|+|C|) - 1 \qquad \forall i,j \in V^s \cup V^c, i \ne j, k \in fev \tag{7}$$

$$q_{i,k} \le M^{fev}. \sum_{(j,i)\in A_1} X^1_{j,i,k} \qquad \forall i \in V^s \cup V^c, k \in fev \tag{8}$$

$$\sum_{j\in GP_{c'}}\sum_{i\in V}\sum_{k\in fev} X^1_{i,j,k} \le 1 \qquad \forall c' \in C' \tag{9}$$

$$\sum_{j\in GP_{c'}}\sum_{i\in V^S\cup V^C}\sum_{l\in Cs}\sum_{s\in V^S} X^2_{i,j,s,l} \le 1 \qquad \forall c' \in C' \tag{10}$$

$$\sum_{j\in GP_{c'}}\sum_{i\in V}\sum_{k\in fev} X^1_{i,j,k} + \sum_{j\in GP_{c'}}\sum_{i\in V^S\cup V^C}\sum_{l\in Cs}\sum_{s\in V^S} X^2_{i,j,s,l} = 1 \qquad \forall c' \in C' \tag{11}$$

$$\sum_{i\in V^S\cup V^C} q_{i,k} \le M^{fev} \qquad \forall\, k \in fev \tag{12}$$

$$\sum_{k\in fev} q_{s,k} = \sum_{j\in V^C}\sum_{l\in Cs} h_{s,j,s,l} \qquad \forall\, s \in V^s \tag{13}$$

$$\sum_{k\in fev} q_{c,k} = d_c.\sum_{j\in V}\sum_{k\in fev} X^1_{j,c,k} \qquad \forall\, c \in V^c \tag{14}$$

$$\sum_{(i,j)\in A_2} X^2_{i,j,s,l} = \sum_{(j,i)\in A_2} X^2_{j,i,s,l} \qquad \forall i \in V^s \cup V^c, s \in V^s, l \in Cs \tag{15}$$

$$\sum_{i\in GP_{c'}}\sum_{l\in Cs}\sum_{s\in V^S}\sum_{(j,i)\in A_2} h_{j,i,s,l} - \sum_{i\in GP_{c'}}\sum_{l\in Cs}\sum_{s\in V^S}\sum_{(i,j)\in A_2} h_{i,j,s,l} = {}^1/_g \sum_{i\in GP_{c'}} d_i.\left(1 - \sum_{i\in GP_{c'}}\sum_{j\in V}\sum_{k\in fev} X^1_{j,i,k}\right) \qquad \forall c' \in C' \tag{16}$$

$$h_{i,j,s,l} \le M^{Cs}_l.X^2_{i,j,s,l} \qquad \forall (i,j) \in A_2, s \in V^s, l \in Cs \tag{17}$$

$$X^2_{i,s,s',l} = 0 \qquad \forall i \in V^s \cup V^c, s,s' \in V^s, s \ne s', l \in Cs \tag{18}$$

$$X^2_{s,j,s',l} = 0 \qquad \forall j \in V^s \cup V^c, s,s' \in V^s, s \ne s', l \in Cs \tag{19}$$

$$X^2_{s,s,s,l} = 0 \qquad \forall s \in V^s, l \in Cs \tag{20}$$

$$\sum_{i\in GP_{c'}}\sum_{j\in GP_{c'}}\sum_{k\in fev} X^1_{i,j,k} = 0 \qquad \forall c' \in C' \tag{21}$$

$$\sum_{i\in GP_{c'}}\sum_{j\in GP_{c'}}\sum_{s\in V^S}\sum_{l\in Cs} X^2_{i,j,s,l} = 0 \qquad \forall c' \in C' \tag{22}$$

$$\sum_{j\in V^S\cup V^C} X^1_{0,j,k} \le 1 \qquad \forall\, k \in fev \tag{23}$$

$$\sum_{s\in V^S}\sum_{j\in V^S\cup V^C} X^2_{s,j,s,l} \le 1 \qquad \forall\, l \in Cs \tag{24}$$

$$\tau^1_i + \sum_{k\in fev} t^1_{i,j}.X^1_{i,j,k} \le \tau^1_j + T.\left(1 - \sum_{k\in fev} X^1_{i,j,k}\right) \qquad \forall i \in V^s \cup V^c, j \in V^s \cup V^c \backslash \{i\} \tag{25}$$

$$\sum_{k\in fev} t^1_{0,j}\,.\,X^1_{0,j,k} \le \tau^1_j + T.\left(1-\sum_{k\in fev} X^1_{0,j,k}\right) \qquad \forall j\in V^s\cup V^c \qquad (26)$$

$$\tau^2_{i,l} + \sum_{s\in V^s} t^2_{i,j,l}\,.\,X^2_{i,j,s,l} \le \tau^2_{j,l} + T\left(1-\sum_{s\in V^s} X^2_{i,j,s,l}\right) \qquad \forall i\in V^c, j\in V^c\backslash\{i\}, l\in Cs \qquad (27)$$

$$\tau^1_s + t^2_{s,j,l}.X^2_{s,j,s,l} \le \tau^2_{j,l} + T\left(1-X^2_{s,j,s,l}\right) \qquad \forall s\in V^s, j\in V^c, l\in Cs \qquad (28)$$

$$a_i.\sum_{j\in V^c\cup V^s\cup\{0\}\backslash\{i\}}\ \sum_{k\in fev} X^1_{i,j,k} \le \tau^1_i \le b_i.\sum_{j\in V^c\cup V^s\cup\{0\}\backslash\{i\}}\ \sum_{k\in fev} X^1_{i,j,k} \qquad \forall i\in V^s\cup V^c \qquad (29)$$

$$a_i.\sum_{j\in V^c\cup V^s\backslash\{i\}}\ \sum_{s\in V^s} X^2_{i,j,s,l} \le \tau^2_{i,l} \le b_i.\sum_{j\in V^c\cup V^s\backslash\{i\}}\ \sum_{s\in V^s} X^2_{i,j,s,l} \qquad \forall i\in V^c, l\in Cs \qquad (30)$$

$$\tau^2_{i,l} \ge e_l.X^2_{i,j,s,l} \qquad \forall i\in V^c, j\in V^c\cup V^s, l\in Cs, s\in V^s \qquad (31)$$

$$\tau^2_{j,l} \ge \sum_{s\in V^s}\left(e_l + t^2_{s,j,l}\right).X^2_{s,j,s,l} \qquad \forall j\in V^c, l\in Cs \qquad (32)$$

$$\tau^2_{j,l} \le f_l \qquad \forall j\in V^c, l\in Cs \qquad (33)$$

$$\sqrt{(O^l_x - long_i)^2 + \left(O^l_y - lat_i\right)^2} \le \alpha^l + BM^d.\left(1-\sum_{j\in V^c\cup V^s}\ \sum_{s\in V^s} X^2_{i,j,s,l}\right) \qquad \forall i\in V^c\cup V^s, l\in Cs \qquad (34)$$

$$h_{i,j,s,l} \ge 0 \qquad \forall (i,j)\in A_2, s\in V^s \qquad (35)$$

$$u_{i,k} \in Z^+ \qquad \forall i\in V^s\cup V^c, k\in fev \qquad (36)$$

$$q_{i,k} \ge 0 \qquad \forall i\in V^s\cup V^c, k\in fev \qquad (37)$$

$$X^1_{i,j,k} \in 0,1 \qquad \forall (i,j)\in A_1, k\in fev \qquad (38)$$

$$X^2_{i,j,s,l} \in 0,1 \qquad \forall (i,j)\in A_2, s\in V^s, l\in Cs \qquad (39)$$

The objective function seeks to minimize total routing costs, encompassing first-echelon vehicle (FEV) and second-echelon crowd shipper (CS) costs, as well as satellite handling costs. Constraints (5) ensure flow conservation for FEVs, ensuring vehicles leave nodes they enter, while constraints (6) limit each node to a single visit by an FEV. Subtour elimination is addressed in constraints (7), and constraints (8) govern parcel transfers to satellites or customers by FEVs. Constraints (9) - (11) ensure customer demands are satisfied by at most one FEV, and any unmet demands are served by exactly one CS and its assigned satellite. Capacity constraints for vehicles and satellites are imposed in constraints (12) and (13), while constraints (14) - (17) ensure demand satisfaction, flow conservation, and capacity limits in the second echelon. Constraints (18) - (24) prevent overlapping satellite assignments and limit vehicle usage. Time windows and travel times are enforced via constraints (25) - (33), while constraints (34) - (39) define geographic and variable domains.

Sensitivity Analysis for Small Instance

In this section, to be able to verify the developed mathematical model, we design a small instance that we know its optimal solution before solving it. Then this example is first solved by the VRPRDL model (provided by (Reyes, Savelsbergh and Toriello 2017)) coded in GAMS software and then solve the same example by the ME-VRPRDL model presented in this research.

The example used in this section is as follows:

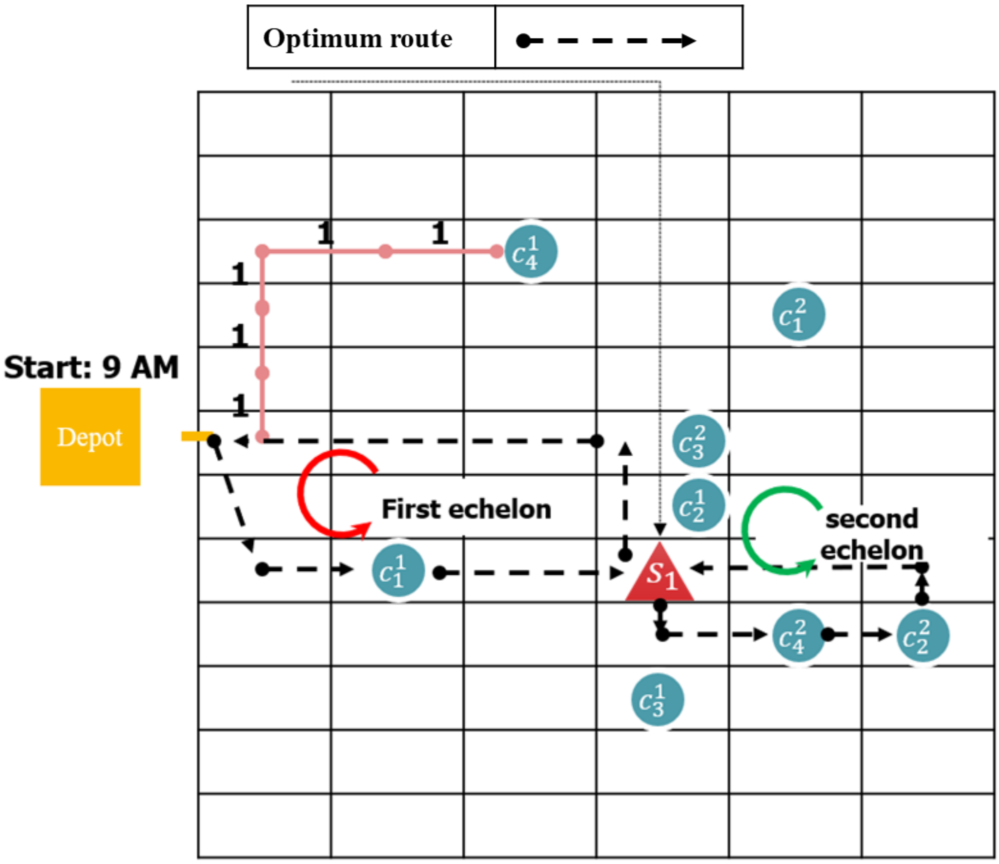


Figure 3. An example designed to verify the mathematical model

In the example above, the assumptions and parameters are as follows:

- Each block is equivalent to one distance unit.
- Each distance unit is considered equivalent to 15 minutes of travel time.
- Distances are calculated using Manhattan method.
- Demand for all customers is equal to 1.

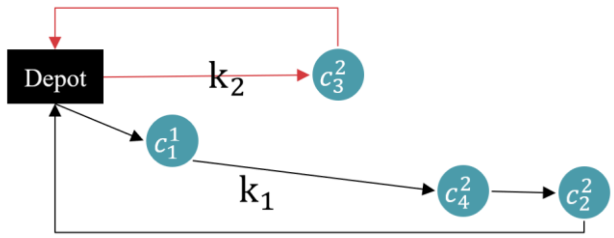
Table 5. List of instance parameters

parameters	
D	depot
c_1^1	Location 1 of customer 1: TW[9am-10am]
c_1^2	Location 2 of customer 1: TW [10am-12pm]
c_2^1	Location 1 of customer 2: TW[9am-10am]
c_2^2	Location 2 of customer 2: TW [10am-12pm]
c_3^1	Location 1 of customer 3: TW[9am-10am]
c_3^2	Location 2 of customer 3: TW [10am-12pm]
c_4^1	Location 1 of customer 4: TW[9am-10am]
c_4^2	Location 2 of customer 4: TW [10am-12pm]
s_1	Location 1 of mobile satellite: TW [10am-12pm]; ([9am-10pm] is moving to its destination and is unavailable)
Cs_1	Crowd shipper 1: shift[9am-10pm]; capacity=2

Cs_2	Crowd shipper 2: shift[10am-12pm]; capacity=2
k_1	FEV1: shift[8am-12pm]; capacity=4
k_2	FEV2: shift[8am-12pm]; capacity=4

If we want to solve this problem with the VRPRDL model, it means that we cannot use satellites and crowd shippers. In this case, the optimal solution of problem will be as follows.

Table 6. the optimal solution of VRPRDL model

Best solution according to VRPRDL	Objective function value
	66

But the optimal solution when we solve it with the ME-VRPRDL model presented in this article by considering the possibility of using satellites and crowd shippers will depend on the following two parameters:

1. The cost and time of travel with crowd shippers
2. The fixed cost of handling a satellite

Therefore, to make a more accurate comparison between the solutions obtained from the above two models, we take the cost and travel time of the crowd shippers as the same as the first echelon vehicles and change the fixed cost of handling the satellite in different iterations. In this case, the results will be as follows:

Heuristic Solution Approach

The VRPRDL problem, being NP-hard, extends its computational complexity to multi-echelon models, making large-scale, real-world applications highly time-intensive and often infeasible. Consequently, this section introduces a heuristic algorithm leveraging node classification and a greedy approach to address these challenges, with detailed procedural steps outlined for implementation.

Phase1: Node Classification

In this phase, we first consider each of the satellites as a class center, and then classify all nodes based on the distance from the class centers.

In the example of Fig 3, since there is only one location - satellite, as a result, there is no classification, and all nodes are assigned to the same satellite.

Phase2: Sorting Phase

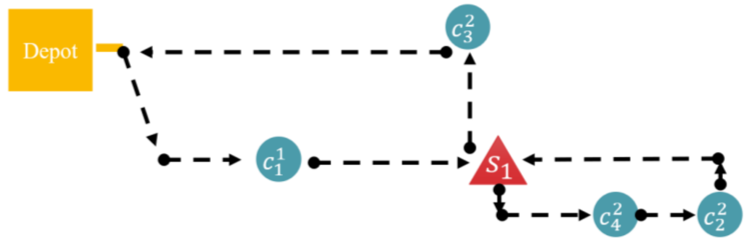
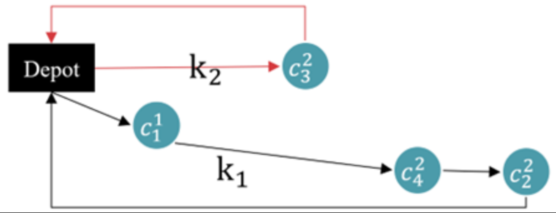
In the second phase to begin the greedy search, we sort all customer nodes based on distance criteria from the depot. For our example, we will consider:

Phase3: Greedy Constrained Solution

In this phase, we select the first node from the sorted list as the starting point and start the route with it. At each step, the next node is the closest node to the current customer. On the other hand, in each step, we check the three main constraints of the problem, and if any of them are not satisfied, we go to the next nearest node. These main constraints are:

- Customer time windows constraints
- Vehicle capacity constraints

Table 7. Sensitive analysis of p_s

p_s	Solution	Obj (cost)	Optimal structure
1		50	Multi echelon
2		52	
3		54	
4		56	
5		58	
6		60	
7		62	
8		64	
≥9		66	Single echelon

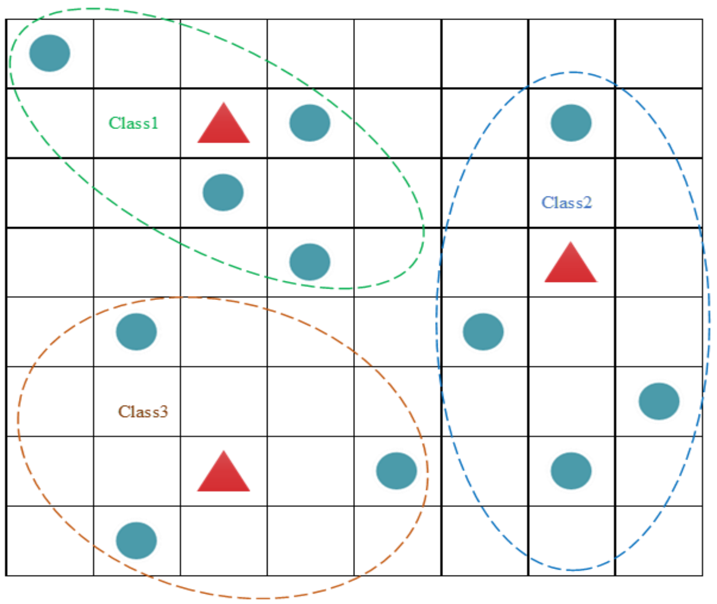


Figure 4. Node classification phase

Sorted nodes							
c_1^1	c_3^2	c_2^1	c_4^1	c_1^2	c_3^1	c_4^2	c_2^2

Figure 5. Node sorting phase

Table 8. final Route of vehicle 1 for selected sample

Selected Vehicle	Step	Route	Node time	Feasibility
	Step1	Depot $\rightarrow C_1^1$	$0 \rightarrow 0{:}45' < 1{:}00$	Yes
	Step2	$Depot \rightarrow C_1^1 \rightarrow C_3^2$	$0 \rightarrow 0{:}45' \rightarrow$1:45'<3:00	Yes
1	Step3	$Depot \rightarrow C_1^1 \rightarrow C_3^2 \rightarrow C_4^2$	$0 \rightarrow 0{:}45' \rightarrow 1{:}45' \rightarrow 2{:}45'$	Yes
	Step4	$Depot \rightarrow C_1^1 \rightarrow C_3^2 \rightarrow C_4^2 \rightarrow C_2^2$	$0 \rightarrow 0{:}45' \rightarrow 1{:}45' \rightarrow 2{:}45' \rightarrow 3$	No
	Step5	$Depot \rightarrow C_1^1 \rightarrow C_3^2 \rightarrow C_4^2 \rightarrow Depot$	$0 \rightarrow 0{:}45' \rightarrow 1{:}45' \rightarrow 2{:}45'$	Yes

Remaining customers = [C2]	final Route of vehicle 1 = [C1 to C3 to C4]	Objective function = 54

Table 9. final Route of vehicle 2 for selected sample

Selected Vehicle	Step	Route	Node time	Feasibility
2	Step1	$Depot \rightarrow C_2^1$	$0 \rightarrow 1:30' > 1:00$	No
	Step2	$Depot \rightarrow C_2^2$	$0 \rightarrow 2:00' < 3:00$	Yes
	Step3	$Depot \rightarrow C_2^2 \rightarrow Depot$	$0 \rightarrow 1:45' \rightarrow 3:30'$	Yes
Remaining customers = []		final Route of vehicle 2 = [C2]		Objective function = 24

The solution obtained above will be a feasible single-echelon solution for the sample problem whose total objective function is equal to 78.

Phase4: Greedy Unconstrained Solution

In this phase, operations are conducted similarly to Phase 3, with the key distinction that only capacity constraints are considered, while time window constraints are omitted. This approach is adopted because the solution generated in this phase serves as an initial input for constructing a two-echelon route in the subsequent phase. The time window constraints will be evaluated after obtaining the final solution in that phase. For the given example, where each customer's demand is assumed to be 1 unit and the capacity of a first-echelon vehicle (FEV) is 4 units, the demands of all customers are fulfilled by an FEV as described.

Initial greedy unconstrained solution for vehicle1

Figure 6. Initial greedy unconstrained solution for vehicle1

In the next step of this phase, we need to sort the nodes according to which type of the time windows they have. In this way, we first create matrices of the nodes of each type by keeping their current order, and then we merge these matrices together to get the final matrix.

Type 1 of time window:

Type 2 of time window:

- Matrix 1 = $[C_1^1, C_2^1, C_4^1]$
- Matrix 2 = $[C_3^2]$
- Final Matrix = $[C_1^1, C_2^1, C_4^1, C_3^2]$

So, final greedy unconstrained solution for vehicle1 is:

Final greedy unconstrained solution for vehicle1

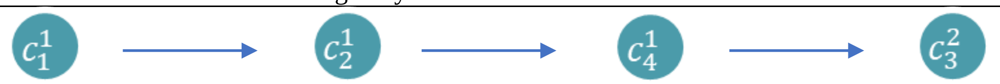


Figure 7. Final greedy unconstrained solution for vehicle1

Phase5: Create Multi-Echelon Solution

To start the process of generating the multi-echelon solution of our sample, the following steps will be taken:

1. **Start with first Crowd Shipper (CS1)**

2. **Satellite Selection**

Finding a satellite that can be covered by the crowd shipper and in the final solution obtained from phase 4, most of the nodes are of its class (according to the classification of nodes in phase 1)

Phase 4 solution	C_1^1 →	C_2^1 →	C_4^1 →	C_3^2
class center	S_1	S_1	S_1	S_1
Selected satellite : S_1				

Figure 8. Satellite selection

3. FEV Selection

Find FEV that has the most nodes of the selected satellite class (K1)

4. Extraction of Potential Customers for Allocation to the Satellite

To find potential customers of a satellite, if one of the locations of a customer is in the class of that satellite, in addition to that location, other locations of that customer are included in the list of potential customers of that satellite regardless of their time windows. So, for the example above, the list of potential customers - locations for Satellite 1 is: [C_1^1 , C_1^2, C_2^1 , C_2^2, $C_4^1, C_4^2, C_3^2, C_3^1$]

5. Check Compatibility of Time Windows

In this step, the nodes whose time windows are incompatible with the satellite time window are removed from the list of potential customers.

Final potential customer list: : [C_1^2, C_2^2, C_4^2, C_3^2]

6. Greedy Customers Selection

In this step, we select the first node from the potential customers list in such a way that:

- The location of the satellite should be between the customer and the depot.
- The customer that has the shortest distance to the satellite.
- The customer who has the greatest distance from the depot.

Therefore, to select the first node, we will seek to minimize the following equation:

$$\begin{aligned} \mathit{Distance\ Index} = {} & a.\sqrt{\big(customer(1) - satellite(1)\big)^2 + \big(customer(2) - satellite(2)\big)^2} \\ & - \frac{1}{b}.\sqrt{\big(customer(1) - depot(1)\big)^2 + \big(customer(2) - depot(2)\big)^2} \end{aligned} \tag{40}$$

$$\begin{cases} a > 1 & if\ \ \big(customer(1) - satellite(1)\big) + \big(customer(2) - satellite(2)\big) < 0 \\ a = 0; & otherwise \end{cases} \tag{41}$$

$$b > a \tag{42}$$

The second node will be the closest node to the first. We continue in the same way until either the capacity of the crowd shipper is less than the minimum available demand or all the nodes in the above list are checked.

Second echelon route	S_1 →	C_4^2 →	C_2^2 →	S_1
Demand	0	1	1	0
The remaining capacity of the crowd shipper	2	1	0	0

Figure 9. Second echelon route

7. Finalization of the first Echelon Route

In this step, we remove the customers assigned to the satellite from the first echelon route. Then we add the selected satellite to the remaining nodes and rearrange the existing nodes according to the greedy method described in phases 2-3.

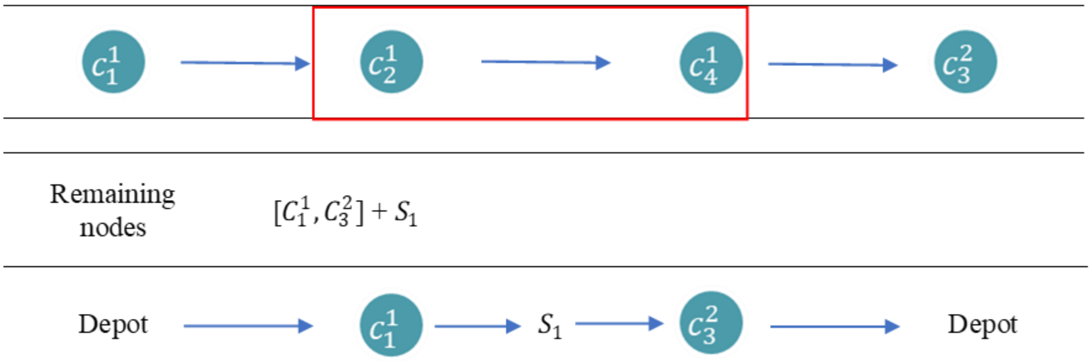


Figure 10. Final first echelon route

8. Feasibility Check

We examine the feasibility of the generated routes in terms of compliance with the time windows.

Table 10. Feasibility check of multi echelon route

Final first echelon route	Depot	C_1^1	S_1	C_3^2	Depot
Departure time of each nodes	0	0:45’	1:15’	1:45’	2:30’
status	feasible	feasible	feasible	feasible	feasible
Final second echelon route	S_1	C_4^2	C_2^2	S_1	
Departure time of each nodes	1:15’	1:45’	2:00’	2:45’	
status	feasible	feasible	feasible	feasible	

As can be seen, the solution obtained during the steps of the proposed heuristic algorithm is the optimal solution of the sample problem, which was also shown in Fig 3.

Phase6: Comparison between single-echelon and multi-echelon solutions and choosing the best one.

Phase7: Update sorted nodes list in phase 2.

remove the node used as the starting node from the sorted list in phase 2.

Phase8: Return to phase 3 and repeat the algorithm until the sorted nodes list is empty.

Algorithm 1 outlines the proposed heuristic in pseudo-code.

Algorithm 1. Classification-greedy heuristic approach

Inputs:
N^{FEV}: Number of FEVs
K_N: FEV capacity
N^{CS}: Number of crowd shippers
K_{CS}: CS capacity
C: Set of customers
g: Customer geographic profiles
D_C: Customer demands
C_{ij}, S_{ij}: Coordinates of customers and satellites
TW: Time windows for all nodes
Outputs:
FS^1: Single-echelon feasible routes

FS^2: Multi-echelon feasible routes

```
Customers ← classifyCustomers(C_ij, S_ij)
CustomersSorted ← sortCustomers(Customers)
C' ← CustomersSorted

for iter ← 1 to Length(C) do
   FEV ← 1
   StartPoint ← C'(iter)

   while FEV < N^FEV do
      UnservedCustomers ← C
      ServedCustomer[FEV] ← GreedyConstrainedProcess(FEV,K_N ,TW, D_C ,UnservedCustomers)
      GreedyRoute[FEV] ← ServedCustomer[FEV]

      if UnservedCustomers \ ServedCustomer[FEV] == Ø then
         GreedyConstrainedSolution[iter] ← GreedyRoute[1:FEV]
         FEV ← N^FEV + 1 (Break the Loop)
      else
         FEV ← FEV + 1
         UnservedCustomers ← UnservedCustomers \ ServedCustomer[FEV]
      end if
   end while

   FEV ← 1
   StartPoint ← C'(iter)

   while FEV < N^FEV do
      UnservedCustomers ← C
      ServedCustomer[FEV] ← GreedyUnconstrainedProcess(FEV, K_N , D_C, UnservedCustomers)
      GreedyRoute[FEV] ← ServedCustomer[FEV]

      if UnservedCustomers \ ServedCustomer[FEV] == Ø then
         GreedyUnconstrainedSolution[iter] ← GreedyRoute[1:FEV]
         FEV ← N^FEV + 1 (Break the Loop)
      else
         FEV ← FEV + 1
         UnservedCustomers ← UnservedCustomers \ ServedCustomer[FEV]
      end if
   end while
end for

for iter ← 1 to Length(C) do
   CS ← 1
   FirstEchelonRoute ← GreedyUnconstrainedSolution[iter]
   Mer ← 1
   BestCost ← ∞

   while CS < N^CS do
      (CS, SCS) ← SatelliteSelectionProcess(CS)
      (CS, SCS, FEVCS) ← FEVSelectionProcess(CS, SCS)
      PotentialCustomers ← ExtractPotentialCustomers(CS, SCS, FEVCS)
      CompatibleCustomers ← CheckTimeWindowCompatibility(PotentialCustomers, TW)
      SecondEchelonRoute[Mer] ← GreedyCustomerSelection(CompatibleCustomers, K_CS, D_C)

      FirstEchelonRoute[Mer] ← UpdateFirstEchelonRoute(FirstEchelonRoute, SecondEchelonRoute[1:Mer])
      MultiEchelonRoute[Mer] ← FeasibilityCheck(FirstEchelonRoute[Mer], SecondEchelonRoute[1:Mer], TW)
      RouteCost ← Cost(MultiEchelonRoute[Mer])
```

```
        if RouteCost < BestCost then
            FinalMultiEchelonRoute[iter] ← MultiEchelonRoute[Mer]
            BestCost ← RouteCost
            Mer ← Mer + 1
        end if

        CS ← CS + 1
    end while

    if BestCost < Cost(GreedyConstrainedSolution[iter]) then
        OptimumRoute[iter] ← FinalMultiEchelonRoute[iter]
    else
        OptimumRoute[iter] ← GreedyConstrainedSolution[iter]
    end if
end for

FinalOptimumRoute ← SelectMinCost(OptimumRoute[1:end])
```

Computational Results

To evaluate the proposed algorithm, scaled-up instances for n < 8 and new instances for n > 8 were solved using a heuristic algorithm, and results were compared with exact solutions from GAMS (Table 12). Additionally, Table 13 compares basic VRPRDL and crowd-based ME-VRPRDL models, analyzing the maximum acceptable level (MAL) for satellite handling costs. Both instances assume unrestricted crowd shipper allocation and availability across shifts. The details of these two instances are presented in the following table (Table 11).

Table 11. detailed information of instances

Instance	Node name	$long_i$	lat_i	d_i	Working shift
1	depot	1	7	0	1,2
	Customers	2	5	1	1
		5	9	1	2
		4	6	1	1
		6	4	1	2
		4	3	1	1
		4	7	1	2
		3	10	1	1
		5	4	1	2
		3	7	1	1
		6	2	1	2
		5	7	1	1
		7	8	1	2
		3	2	1	1
		6	6	1	2
	fixed satellite	4	5	0	1,2
	mobile satellite	4	11	0	1
		5	8	0	2
2	depot	1	7	0	1,2
	customers	2	9	1	1
		3	13	1	2
		3	9	1	1
		4	13	1	2
		5	9	1	1
		5	12	1	2
		5	6	1	1
		6	15	1	2
		6	4	1	1
		8	11	1	2
		4	1	1	1
		5	3	1	2
		2	1	1	1
		3	4	1	2

Instance	Node name	$long_i$	lat_i	d_i	Working shift
		11	8	1	1
		9	7	1	2
		11	5	1	1
		8	6	1	2
		2	12	1	1
		3	16	1	2
		1	12	1	1
		7	13	1	2
		1	18	1	1
		2	13	1	2
		2	19	1	1
		2	14	1	2
		3	20	1	1
		1	14	1	2
		4	21	1	1
		1	16	1	2
		5	21	1	1
		2	16	1	2
		8	17	1	1
		3	15	1	2
		9	18	1	1
		2	15	1	2
		7	23	1	1
		4	15	1	2
		7	24	1	1
		5	15	1	2
	fixed satellite	11	1	0	1,2
	mobile satellite	7	6	0	1
		11	15	0	2
		9	22	0	1
		5	18	0	2
		2	24	0	1
		1	15	0	2

Table 12. Comparison between exact and heuristic algorithms for Crowd-based ME-VRPRDL problem

#	Instance	Number of entities								Exact algorithm (Solver: BONMIN)						Heuristic algorithm (HA)						GAP**
		C	MS	SS	|g|	FEV	CS	TH	Cap	Obj^1	Rt	AMS	ASS	AFEV	ACS	Obj^2	Rt	AMS	ASS	AFEV	ACS	
1	1	4	0	1	2	3	2	4	4	48	10	0	1	1	1	48	0.2	0	1	1	1	-
2		5	0	1	2	3	2	12	4	54	57	0	1	2	1	72	0.3	0	1	2	1	0.25
3		6	1	1	2	5	4	16	5	60	*	0	1	2	1	78	0.4	0	0	2	0	0.23
4		7	1	1	2	5	4	16	8	60	*	0	0	1	0	66	0.4	0	0	1	0	0.09
5	2	9	1	1	2	5	4	16	9	90	231	1	0	1	1	90	0.1	1	0	1	1	-
۶		11	2	1	2	5	4	16	9	-	*	-	-	-	-	126	0.3	1	0	2	1	-
7		14	2	1	2	5	4	16	9	-	*	-	-	-	-	138	1.5	1	0	2	1	-
8		16	3	1	2	5	4	16	9	-	*	-	-	-	-	156	1.9	2	0	2	2	-
9		18	3	1	2	5	4	16	9	-	*	-	-	-	-	168	1.7	2	0	2	2	-
10		20	3	1	2	5	4	16	9	-	*	-	-	-	-	240	1.7	2	0	3	2	-

C: Customer; MS: Mobile Satellite; SS: Stationary Satellite; TH: Time Horizon; Cap: FEV Capacity; Rt: Running Time in sec; AMS: Allocated Mobile Satellite; ASS: Allocated Stationary Satellite; AFEV: Allocated First Echelon Vehicle; ACS: Allocated Crowd Shipper

* Set 2000 seconds as an upper bound of running time. So, we consider last feasible solution after 2000 seconds as final solution.

- No feasible solution found after 2000 seconds.

** $GAP = \frac{Obj^2 - Obj^1}{Obj^2}$

Table 13. Comparison between VRPRDL & Crowd-based ME-VRPRDL

#	Instance	Number of entities								VRPRDL			Crowd-based ME-VRPRDL			Optimal model	MAL/ VRPRDL COST	GAP $\frac{Obj^1 - Obj^2}{Obj^1}$
		C	MS	SS	|g|	FEV	CS	TH	Cap	Solver	Obj^1	Rt	Solver	Obj^2 (without satellite handling cost)	Rt			
1	1	4	0	1	2	3	2	4	4	BONMIN	66	0.20	BONMIN	48	10.3	Crowd-based ME-VRPRDL	27.27 %	0.27
2		5	0	1	2	3	2	12	4	BONMIN	60	0.39	BONMIN	54	57.1	Crowd-based ME-VRPRDL	10%	0.10
3		6	1	1	2	5	4	16	5	BONMIN	66	068	BONMIN	60	*	Crowd-based ME-VRPRDL	9%	0.09
4		7	1	1	2	5	4	16	8	BONMIN	60	0.51	BONMIN	60	*	both	0%	-
5	2	9	1	1	2	5	4	16	9	BONMIN	102	0.92	HA	90	0.1	Crowd-based ME-VRPRDL	11.7%	0.12
6		11	2	1	2	5	4	16	9	BONMIN	138	9.94	HA	126	0.28	Crowd-based ME-VRPRDL	8.6%	0.09
7		14	2	1	2	5	4	16	9	BONMIN	150	29.1	HA	138	1.55	Crowd-based ME-VRPRDL	8%	0.08
8		16	3	1	2	5	4	16	9	BONMIN	168	*	HA	156	1.86	Crowd-based ME-VRPRDL	7.1%	0.07
9		18	3	1	2	5	4	16	9	BONMIN	174	*	HA	168	1.68	Crowd-based ME-VRPRDL	3.4%	0.03
10		20	3	1	2	5	4	16	9	BONMIN	252	*	HA	240	1.75	Crowd-based ME-VRPRDL	4.7%	0.05

MAL: maximum acceptable level for the handling costs of satellites.
* Set 2000 seconds as an upper bound of running time. So, in this case, we consider last feasible solution after 2000 seconds as final solution.

As can be seen in the above tables, the complexity of Crowd-based ME-VRPRDL is much higher than that of VRPRDL, and as a result, obtaining the globally optimal solution requires much more time. However, in some cases, due to the high execution time, the last feasible solution of Crowd-based ME-VRPRDL is reported after 2000 seconds, this feasible solution has a lower cost or, in the worst case, equal to the globally optimal solution of VRPRDL.

In addition, it should be considered that in these example problems, the costs and travel times of FEVs and crowd shippers are considered equal. While in many cases these parameters are lower in crowd shippers than FEVs. Therefore, it can be concluded that in the case of using crowd shippers, the reduction of total costs and travel times will be much more impressive. Also, in addition to economic aspects, it has been shown in past studies that in general, the use of the multi-echelon structure as well as crowd-based agents will have positive effects on social and environmental aspects of sustainability.

Since Crowd-based ME-VRPRDL is an extraordinarily complex problem, it will be difficult to find a feasible solution for it. As a result, in many meta-heuristic algorithms, after applying the algorithm operators on the initial solutions, mostly the new solutions were left out of the feasible space of the problem, and practically the algorithms could not provide suitable results. This issue was due to the limited solution space, which would violate the feasible space of the problem with the smallest change in the solution. So, there was a need for a heuristic approach to produce feasible and high-quality solutions for large-scale problems. But one of the main applications of this algorithm is to create an online approach to solve Crowd-based ME-VRPRDL in real time in future researches. Since in many cases the parameters related to the available time and location of crowd shippers are not known in advance and their values may be determined during the execution, we will practically face a dynamic programming problem that their solutions should be updated in different time slots.

Another feature of the heuristic algorithm presented in this research is the provision of both single-echelon and multi-echelon solutions for the problem. By providing both solutions, this algorithm will always provide an alternative way for the user in real world applications.

In the following, in order to be able to examine more and random samples, we will generate several random instances as described below (Table 14-15) and solve them with the help of two exact and heuristic methods. As seen in the examples above, the solution to the VRPRDL problem can be considered as an upper bound for the ME-VRPRDL problem. On the other hand, in cases where it is not possible to exactly solve the ME-VRPRDL problem up to 2000 seconds in the GAMS software, the first feasible solution obtained after 2000 seconds will be considered as an upper bound for the instance.

Table 14. Entity information for generating random instances

Number of FEV	5
FEV Capacity	9
Number of CS	4
CS Capacity	2
Depot location	[1, 3]
Number of stationary satellites	1
Number of mobile satellites	3

Table 15. General information for generating random instances

Time Horizon	16
Geographic profile size	2
Shift1	[0, 7.9]
Shift2	[8, 16]
Travel Time of a Unit Distance	15 Min
CS travel time = FEV travel time	
CS travel cost = FEV travel cost	

As can be seen in Table 16, in the last instance with 35 customers (and 75 customers' locations), it was not possible to get a feasible solution in both VRPRDL and ME-VRPRDL problems by BONMIN exact solver (GAMS software) after 2000 seconds, but the proposed heuristic algorithm produced a suitable feasible solution for it in only 1.36 seconds. In none of the above examples, it has not been possible to obtain the global optimal solution for Crowd-based ME-VRPRDL problem.

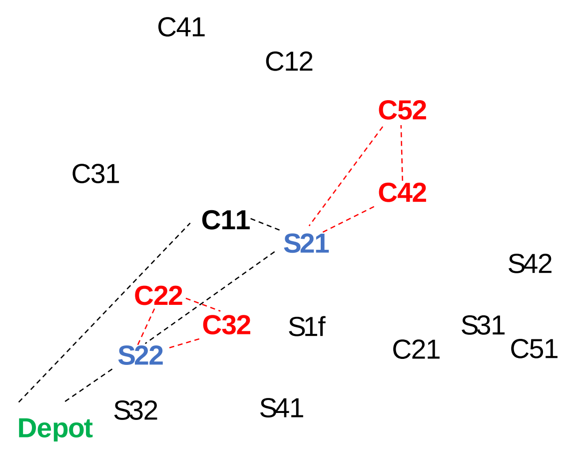
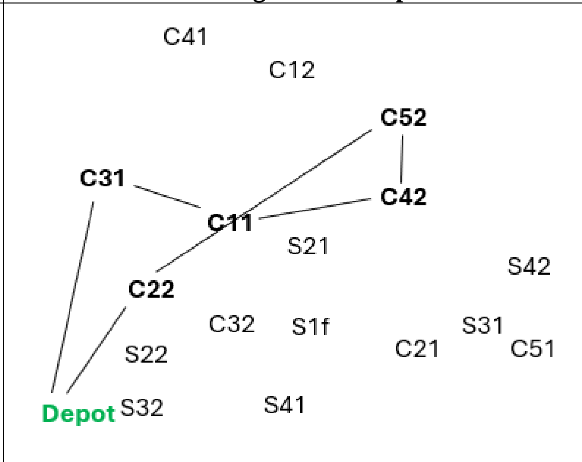
As a result, the comparison between the (1) optimal solution of VRPRDL problem as an upper bound of the objective value of ME-VRPRDL problem, (2) the best feasible solution resulting from the heuristic algorithm and (3) the best feasible solution of ME-VRPRDL problem after 2000 seconds by BONMIN algorithm was investigated.

According to the results obtained in Table 16, in small and medium-sized problems (less than 20 customers), if it is not possible to solve ME-VRPRDL problem accurately in a reasonable time, it is better to solve it as a VRPRDL problem (single echelon VRPRDL). But if the exact solution of ME-VRPRDL problem is not possible in a reasonable time, it is better to solve it by the provided heuristic algorithm. Also, large instances (with more than 20 customers), solving the problem by our heuristic algorithm will lead to better results in a much shorter time.

In the following, the schematic solutions of exact and heuristic algorithms for instances with less than 10 customers are shown in Figure 11.

Table 16. Comparison between the BONMIN solver and the heuristic algorithm in solving randomly generated instances of the crowd-based ME-VRPRDL problem

#)	Number of random customers	Random Range	Exact algorithm (BONMIN)					Heuristic algorithm (HA)			GAP
			Crowd-based ME-VRPRDL			VRPRDL		Crowd-based ME-VRPRDL			
			Obj^1	Rt	Number of echelons of solution	Obj	Rt	Obj^2	Rt	Number of echelons of solution	$\left(\frac{Obj^2-Obj^1}{Obj^2}\right)$
1	5	[3,10]	78	2000	1	78	0.419	84	0.235	1,2	0.07
2	7	[3,10]	60	2000	1	60	0.903	66	0.103	2	0.09
3	9	[3,12]	186	2000	2	90	4.45	102	0.231	2	-0.82
4	11	[3,12]	-	2000	-	114	57.667	114	0.395	1	-
5	15	[3,15]	-	2000	-	222	2000	216	0.559	1	-
6	19	[3,15]	-	2000	-	264	2000	294	0.677	1,2	-
7	23	[3,15]	-	2000	-	426	2000	336	0.983	2	-
8	27	[3,15]	-	2000	-	384	2000	342	1.561	1	-
9	31	[3,15]	-	2000	-	588	2000	402	2.898	2	-
10	35	[3,18]	-	2000	-	-	2000	474	1.369	2	-

#	HA Output	Exact Algorithm Output
1		
	Obj Function = 84	Obj Function = 78

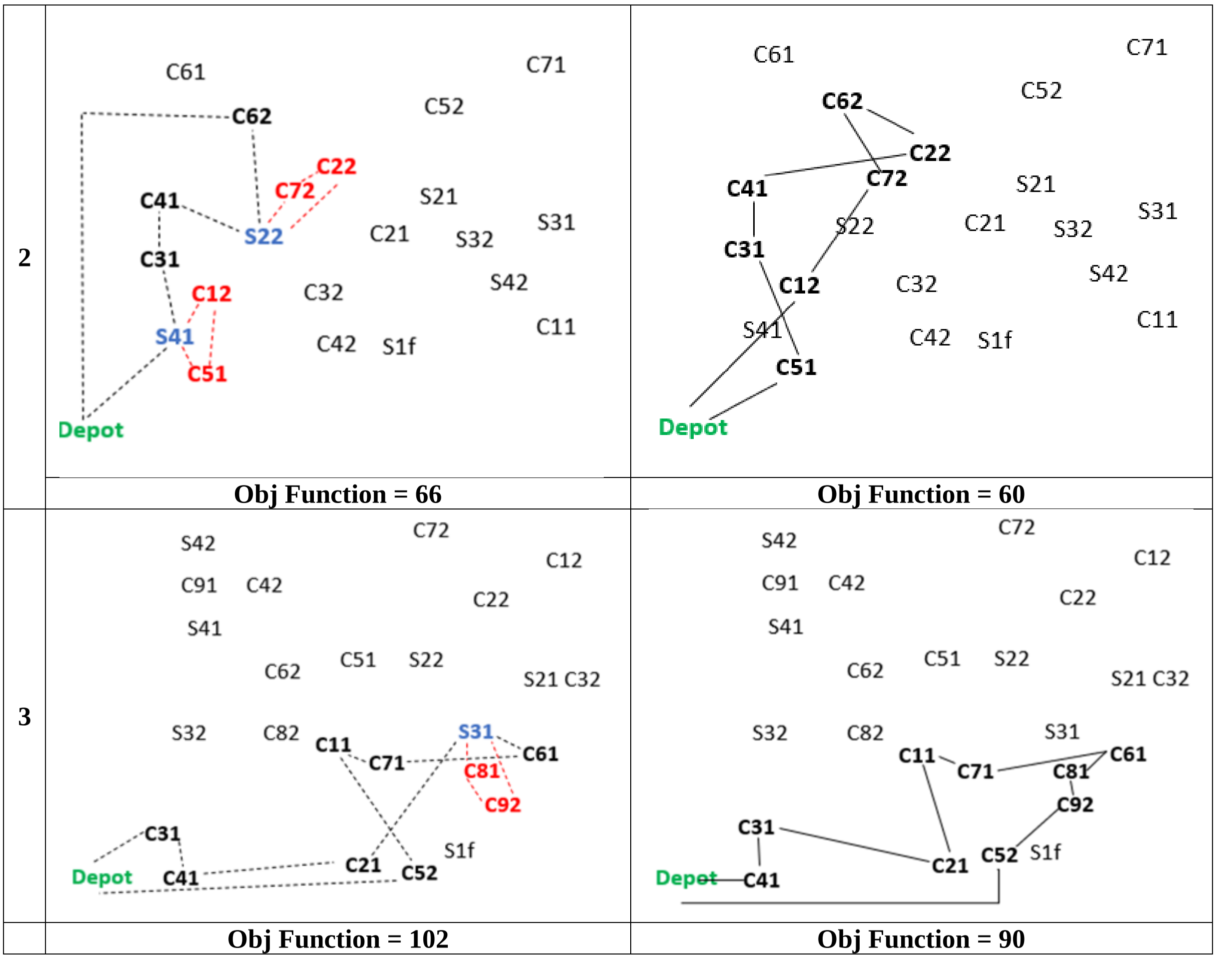


Figure 11. Solution networks for customers < 10

Finally, to evaluate the performance of our proposed heuristic algorithm in solving the basic VRPRDL problem, we tested six instances of varying sizes using both the BONMIN solver and our heuristic approach. The results are presented in Table 17. As shown in the table, for Instance 6 (with 20 customers), our proposed heuristic algorithm achieves superior results in just 1 second compared to those obtained by the BONMIN solver after 2,000 seconds.

Table 17. Comparison between the BONMIN solver and the proposed heuristic algorithm for VRPRDL problems

#	C	MS	SS	|g|	FEV	CS	TH	Cap	Solver	Obj^{BONMIN}	Solver	Obj^{HA}	Gap**
1	9	1	1	2	5	4	16	9	BONMIN	102	HA	126	0.19
2	11	2	1	2	5	4	16	9	BONMIN	138	HA	168	0.18
3	14	2	1	2	5	4	16	9	BONMIN	150	HA	180	0.17
4	16	3	1	2	5	4	16	9	BONMIN	168*	HA	198	0.15
5	18	3	1	2	5	4	16	9	BONMIN	174*	HA	228	0.24
6	20	3	1	2	5	4	16	9	BONMIN	252*	HA	240	-0.05

*Solution after 2000 seconds

** (ObjHA- ObjBONMIN)/ObjHA

Conclusions

This study presents an innovative mathematical model for a crowd-based, flexible multi-echelon vehicle routing problem with roaming delivery locations, extending the basic VRPRDL model. By integrating crowd shippers and mobile parcel lockers, the model improves delivery capabilities. The results indicate that the optimal solution derived from the crowd-based ME-

VRPRDL model is always superior or equal to the basic VRPRDL model. However, the complexity of the crowd-based model is significantly higher, requiring attention to problem dimensions and solution accuracy. Key factors such as handling costs, travel time, and transportation costs significantly influence the number of optimal echelons. Since the problem is NP-hard, solving it for large-scale real-world scenarios is computationally intensive, necessitating the development of a heuristic algorithm. The introduction of multiple time windows for crowd shippers, satellites, and customers further complicates the problem. Future research could explore online algorithms for real-time routing and mobile satellite routing to better reflect real-world conditions.

References

Archetti, Claudia, Francesca Guerriero, and Giusy Macrina. 2021. "The online vehicle routing problem with occasional drivers." Computers & Operations Research 127: 105144.

Bilik, J. 2014. "Parcel machines-green solution for green cities." 1st International Conference „Green Logistics for Greener Cities. Szczecin.

Boysen, Nils, Stefan Fedtke, and Stefan Schwerdfeger. 2021. "Last-mile delivery concepts: a survey from an operational research perspective." Or Spectrum 43 (1): 1-58.

Braekers, Kris, Katrien Ramaekers, and Inneke Van Nieuwenhuyse. 2016. "The vehicle routing problem: State of the art classification and review." Computers & Industrial Engineering 99: 300-313.

Buldeo Rai, Heleen, Sara Verlinde, Jan Merckx, and Cathy Macharis. 2017. "Crowd logistics: an opportunity for more sustainable urban freight transport?" European Transport Research Review 9 (3): 1-13.

Dantzig, George B., and John H. Ramser. 1959. "The truck dispatching problem." Management science 6 (1): 80-91.

He, Yandong, Mingyao Qi, Fuli Zhou, and Jiafu Su. 2020. "An effective metaheuristic for the last mile delivery with roaming delivery locations and stochastic travel times." Computers & Industrial Engineering 145: 106513.

Janjevic, Milena, Matthias Winkenbach, and Daniel Merchán. 2019. "Integrating collection-and-delivery points in the strategic design of urban last-mile e-commerce distribution networks." Transportation Research Part E: Logistics and Transportation Review 131: 37-67.

Jolfaei, Ali Aghadavoudi, and Mahdi Alinaghian. 2024. "Multi-depot vehicle routing problem with roaming delivery locations considering hard time windows: Solved by a hybrid ELS-LNS algorithm." Expert Systems with Applications 124608.

Kafle, Nabin, Bo Zou, and Jane Lin. 2017. "Design and modeling of a crowdsource-enabled system for urban parcel relay and delivery." Transportation research part B: methodological 99: 62-82.

Klumpp, Matthias. 2017. "Crowdsourcing in logistics: An evaluation scheme." Dynamics in Logistics (Springer, Cham) 401-411.

Lachapelle, Ugo, Matthew Burke, Aiden Brotherton, and Abraham Leung. 2018. "Parcel locker systems in a car dominant city: Location, characterisation and potential impacts on city planning and consumer travel access." Journal of Transport Geography 71: 1-14.

Lemardelé, Clément, Miquel Estrada, Laia Pagès, and Mónika Bachofner. 2021. "Potentialities of drones and ground autonomous delivery devices for last-mile logistics." Transportation Research Part E: Logistics and Transportation Review 149: 102325.

Lemke, Justyna, Stanisław Iwan, and Jerzy Korczak. 2016. "Usability of the parcel lockers from the customer perspective–the research in Polish Cities." Transportation Research Procedia 272-287.

Liao, Shirong, Ying Dai, and Zujun Ma. 2025. "Vehicle routing problem with roaming locations for simultaneous pickup and delivery services." OPSEARCH 1-21.

Lin, Yunhui, Yuan Wang, Loo Hay Lee, and Ek Peng Chew. 2022. "Profit-maximizing parcel locker location problem under threshold Luce model." Transportation Research Part E: Logistics and Transportation Review 157: 102541.

Lombard, Augustin, Simon Tamayo-Giraldo, and Frédéric Fontane. 2018. "Vehicle routing problem with roaming delivery locations and stochastic travel times (VRPRDL-S)." Transportation research procedia 30: 167-177.

Moccia, Luigi, Jean-François Cordeau, and Gilbert Laporte. 2012. "An incremental tabu search heuristic for the generalized vehicle routing problem with time windows." Journal of the Operational Research Society 63 (2): 232-244.

Mousavi, Kianoush, Merve Bodur, and Matthew J. Roorda. 2022. "Stochastic last-mile delivery with crowd-shipping and mobile depots." Transportation Science 56 (3): 612-630.

Ostermeier, Manuel, Andreas Heimfarth, and Alexander Hübner. 2022. "Cost-optimal truck-and-robot routing for last-mile delivery." Networks 79.3: 364-389.

Ozbaygin, gizem, and Martin Savelsbergh. 2019. "An iterative re-optimization framework for the dynamic vehicle routing problem with roaming delivery locations." Transportation Research Part B: Methodological 128 207-235.

Ozbaygin, Gizem, Oya Ekin Karasan, Martin Savelsbergh, and Hande Yaman. 2017. "A branch-and-price algorithm for the vehicle routing problem with roaming delivery locations." Transportation Research Part B: Methodological 100: 115-137.

Ranard, Benjamin L., Yoonhee P. Ha, Zachary F. Meisel, David A. Asch, Shawndra S. Hill, Lance B. Becker, Anne K. Seymour, and Raina M. Merchant. 2014. "Crowdsourcing—harnessing the masses to advance health and medicine, a systematic review." Journal of general internal medicine 187-203.

Reyes, Damián, Martin Savelsbergh, and Alejandro Toriello. 2017. "Vehicle routing with roaming delivery locations." Transportation Research Part C: Emerging Technologies (80): 71-91.

Saker, Amira, Amr Eltawil, and Islam Ali. 2023. "Adaptive large neighborhood search metaheuristic for the capacitated vehicle routing problem with parcel lockers." Logistics 7.

Sampaio Oliveira, A, J Kinable, L Veelenturf, and T van Woensel. 2019. "A scenario-based approach for the vehicle routing problem with roaming delivery locations under stochastic travel times." Optimization Online.

Sampaio, Afonso, Martin Savelsbergh, Lucas P. Veelenturf, and Tom Van Woensel. 2020. "Delivery systems with crowd-sourced drivers: A pickup and delivery problem with transfers." Networks (76): 232-255.

Schwerdfeger, Stefan, and Nils Boysen. 2020. "Optimizing the changing locations of mobile parcel lockers in last-mile distribution." European Journal of Operational Research 285.3: 1077-1094.

Tilk, Christian, Katharina Olkis, and Stefan Irnich. 2021. "The last-mile vehicle routing problem with delivery options." OR Spectrum 43 (4): 877-904.

van Duin, JH Ron, Bart W. Wiegmans, Bart van Arem, and Yorick van Amstel. 2020. "From home delivery to parcel lockers: A case study in Amsterdam." Transportation Research Procedia 46: 37-44.

Zhou, Fuli, Yandong He, and Lin Zhou. 2019. "Last mile delivery with stochastic travel times considering dual services." IEEE Access 7: 159013-159021.

Zurel, Özhan, Laurent Van Hoyweghen, Stijn Braes, and Aurelie Seghers. 2018. "Parcel lockers, an answer to the pressure on the last mile delivery?" In New business and regulatory strategies in the postal sector (Springer, Cham) 299-312.