



HAL
open science

Electric van-based robot deliveries with en-route charging

Shaohua Yu, Jakob Puchinger, Shudong Sun

► **To cite this version:**

Shaohua Yu, Jakob Puchinger, Shudong Sun. Electric van-based robot deliveries with en-route charging. European Journal of Operational Research, inPress, 10.1016/j.ejor.2022.06.056 . hal-03706482

HAL Id: hal-03706482

<https://hal.science/hal-03706482>

Submitted on 1 Jul 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Electric van-based robot deliveries with en-route charging

Shaohua Yu^{*a,b,d}, Jakob Puchinger^{b,c}, Shudong Sun^{d,e}

^a*School of Intelligent Manufacturing, Nanjing University of Science and Technology, Nanjing 210094, China*

^b*Laboratoire Genie Industriel, CentraleSupélec, Université Paris-Saclay, Gif-sur-Yvette, France*

^c*Institut de Recherche Technologique SystemX, Palaiseau, France*

^d*Department of Industrial Engineering, Northwestern Polytechnical University, Xi'an 710072, China*

^e*Key Laboratory of Industrial Engineering and Intelligent Manufacturing, Ministry of Industry and Information Technology, Xi'an 710072, PR China*

Abstract

We present a two-echelon electric van-based robot delivery system with en-route charging for last-mile delivery in logistics operations. Each of the vans is equipped with a single robot, and the robots can visit areas with van access restrictions, such as pedestrianized areas or university campuses. The time during which electric vans are carrying robots can be used to recharge the robots, thereby increasing the efficiency of the distribution system.

To model the proposed system, we present a mixed integer program. We note that the energy transfer from a van to its robot needs time and will cause the available travel distance of a van to decrease and that of a robot to increase. Focusing on the new time-distance-energy trade-off problem, which increases the difficulty checking the feasibility of any given route, we further propose a greedy route evaluation approach and a linear programming-based route evaluation method. An adaptive large neighborhood search algorithm is presented for solving larger instances. A sensitivity analysis for vehicle charging modes, charging rates, and maximum battery capacities shows that using en-route charging, while appropriately increasing battery level and charging rate can have useful effects on cost.

Keywords: Logistics, Innovative last-mile distribution, Van-based robot deliveries, Electric vehicle routing, Adaptive large neighborhood search

1. Introduction

A recent study by the World Economic Forum Forum (2020) predicts that growing e-commerce activity will generate 36% more delivery vehicles in inner city areas and last-mile services in urban areas are expected to grow by 78% by 2030. Last-mile delivery describes the final step of the retail supply chain (Ostermeier et al. 2021), and is the only link with direct customer contact, which is a crucial challenge for logistics companies. Besides, last-mile delivery usually accounts for more than 50% of logistics costs (Kuhn and Sternbeck 2013), hence, logistics service providers have to address this problem.

In 2015, global CO_2 emissions from transport are 9000 billion tons, and these are expected to grow by 60% until 2050 (Liimatainen et al. 2019). Freight transport currently accounts for slightly less than half of emissions from transport, but the share is expected to grow significantly as emissions from road freight transport are expected to grow by 56%-70% (Mulholland et al. 2018). Cities are forced to regulate access to their most populous central districts by limiting access to various means of transport in an effort to address these transport and environmental issues, which have brought challenges to the application of traditional fuel vans in urban logistics. Compared with traditional fuel vans, electric vans stand out because of their low local carbon emissions, environmental protection, and fewer restrictions on urban traffic. Hence electric vans have gradually become an important means of transportation for logistics and distribution.

Email addresses: shaohua.yu@njjust.edu.cn (Shaohua Yu*), jakob.puchinger@irt-systemx.fr (Jakob Puchinger), sdsun@nwpu.edu.cn (Shudong Sun)

In recent years, the rapid development of 5G, sensors, the internet of things, and artificial intelligence has promoted the technological progress and application promotion of robots in logistics and distribution. At present, robot-based distribution has become one of the important topics in the transformation of logistics companies, which will significantly change the design of distribution operations. In the past five years, robot delivery has been drawing increased attention. For example, Starship Technologies (Andrew 2019) and JD.COM (Gu 2018) implement autonomous distribution robots on campuses and in some residential areas. Most robots are electrically powered in real-world applications due to electric robots without local pollutant emissions (Bektaş et al. 2019). However, delivery robots will be very limited in range for the foreseeable future, due to their limited battery capacity. These robots move at walking speed for safety reasons, and so their application for long-distance delivery is not efficient (Daimler 2018).

However, separate delivery by electric vans and robots remains a challenge. The electric van is limited in distribution to streets and campuses due to its large size. And the robot is limited by travel speed, distance, and load capacity, so it is challenging to complete medium and long-distance logistics distribution alone.

Current research tends to use traditional vehicles with robots (classical two-echelon concept or mothership concept) for collaborative distribution. For the former, Alfandari et al. (2022) and Bakach et al. (2021) investigate how to properly use robots in two-tier networks and considered different objective functions. For the latter, Yu et al. (2020) present a two-echelon vehicle routing problem with robots for *2nd-level route* delivery, the van carrying the robot along a *1st-level route* for dropping off and picking up at parking nodes. The robot still needs to be recharged and replenished if it is to be used multiple times, especially in mothership delivery mode. With the development of energy storage and charging technology, charging modes tend to diversify. For example, NIO is courting Tesla owners with mobile charging stations inside electric vans (Fred 2018). Mathew et al. (2015) and Yu et al. (2019) consider and test joint planning of mobile recharging platforms and drone routes to enable the drone to visit specified nodes successfully. This prior work suggests that a van, stationary or moving, can act as a mobile charging station to charge its robots en route.

Incorporating the latest electric vans and taking van-based robot delivery research further, we present two-echelon electric van-based robot delivery with en-route charging (2E-VREC) as prototypical problems, in which vans and robots are all battery-powered, and the time during which the vans are carrying the robots can be used effectively to recharge the robots, thereby increasing the efficiency of the distribution system. In 2E-VREC, larger electric vans carry small electric robots (an electric van can only carry an electric robot) along the *1st-level route*. The robots travel along the *2nd-level open route*. The van and robot can serve customers directly, but some constrained customers have to be visited only by the delivery robots. The van stops at parking nodes to drop off and/or pick up its robot, and recharge and replenish its robot if needed. The van can charge its onboard robot on the move or at nodes, and the van can be recharged at the parking nodes.

Our 2E-VREC model presents the following differences to the van-based robot delivery model (Heimfarth et al. 2022): (i) we extend the models to cover specific electric vehicle characteristics, (ii) the 2E-VREC model brings two new technologies (en-route charging and reverse charging) into the electric van-based robot delivery model, and (iii) the 2E-VREC model makes full use of the time during which vans are carrying robots on the road to charge them, shortening the extra waiting time spent by robots at parking stations for recharging, and thereby improving the effectiveness of the distribution system.

The new model adds a new time-distance-energy trade-off to electric vehicle routing. In the classical electric vehicle routing problem (EVRP), the decrease in a vehicle's energy is caused only by the increase in the distance the vehicle travels, and these two variables are usually negatively correlated. By contrast, in the 2E-VREC model, the decrease in the energy of a van can be caused by the increase in the distance the van has traveled, but also by the amount of energy it transfers to its robot. The energy transfer is negatively correlated with the available travel distance of a van, but positively correlated with the available travel distance of its robot (i.e. the energy transfer will cause the available travel distance of a van to decrease and that of a robot to increase).

In addition, energy transfer needs time. Generally speaking, charging a robot en-route can reduce extra waiting time for charging, but it reduces the available travel distance of the van. Whether to charge the robot en-route and how much energy to transfer is therefore a difficult choice. To the best of our knowledge, these characteristics have not been considered, modeled, and solved in research to date. The work reported here makes the following contributions:

(1) We describe a new two-echelon van-based robot routing problem with en-route charging, which has the potential to improve the efficiency of the distribution system.

(2) We introduce the 2E-VREC model and propose a mathematical formulation.

(3) We provide a high-accuracy heuristic for single route feasibility checking of the 2E-VREC problem with time-distance-energy trade-off.

(4) We propose an adaptive large neighborhood search (ALNS) algorithm for the newly presented problem.

(5) We conduct a sensitivity analysis for vehicle charging modes, charging rates, and maximum battery capacities.

In what follows, a brief literature review is given in Section 2, Section 3 describes the problem setting, Section 4 introduces the methodology, a numerical analysis is presented in Section 5, and Section 6 concludes.

2. Related literature

This section presents a literature review related to the proposed 2E-VREC model. We give the latest research status for van-based drone delivery problems, van-based robot delivery problems, and two-echelon (electric) vehicle routing problems. Then, we identify the gaps in the literature and introduce our contribution.

2.1. Van-based drone delivery

Hybrid van and drone deliveries have drawn much attention (Otto et al. 2018). Murray and Chu (2015) first introduced the van and drone routing problems, and farther-reaching problems have since been studied. The advantage of van-based drone deliveries is that the van can carry drones, and along the route, it can drop off and pick up the drone at different locations. Goods can be delivered in parallel by the van and its drone. The drone can also play the role of the street postman to access restricted areas that a van cannot reach. These advantages improve the efficiency of the distribution system compared to more traditional distribution systems.

The van-based drone delivery problem can be divided into four types according to the number of vans, and the number of drones a van can carry. These are problems with (i) a single drone carried by a single van (Murray and Chu 2015, Carlsson and Song 2018, Agatz et al. 2018, Bouman et al. 2018, Poikonen and Golden 2020a, Dayarian et al. 2020, Gonzalez-R et al. 2020, Roberti and Ruthmair 2021); (ii) multiple drones are carried by a single van (Karak and Abdelghany 2019, Murray and Raj 2020, Poikonen and Golden 2020b, Moshref-Javadi et al. 2020a,b, Salama and Srinivas 2020, Kang and Lee 2021); (iii) multiple vans where each van carries a single drone (Luo et al. 2017, Sacramento et al. 2019); (iv) multiple vans where each van carries multiple drones (Poikonen et al. 2017, Pugliese and Guerriero 2017, Wang et al. 2017, Schermer et al. 2019a,b, Wang and Sheu 2019, Kitjacharoenchai et al. 2020, Chen et al. 2021).

Researchers use exact algorithms to solve smaller instances of van-based drone delivery problems. For example, Pugliese and Guerriero (2017) use mixed integer linear programming (MILP) to model and solve the problem with customer time window constraints. Tamke and Buscher (2021) propose a new MILP formulation for the problem and introduce new valid inequalities and a separation routine for extended subtour elimination constraints. Bouman et al. (2018) present exact solution approaches for the van-based drone delivery problem with one van and one carried drone, based on dynamic programming. Their numerical experiments show that their dynamic programming approach can solve larger instances than the mathematical programming approaches. Wang and Sheu (2019) design a branch and price algorithm for solving the van-based drone delivery with multiple vans carrying multiple drones. Its algorithm can address up to 12 coordinate points.

Researchers use (meta-)heuristics or artificial intelligence approaches to solve medium or larger instances. Luo et al. (2021) propose a multi-start tabu search algorithm that can efficiently solve the van-based robot delivery problem with 100 coordinate points. Schermer et al. (2019a) formulate the vehicle routing problem with drones and en route operations as a MILP, and introduce some valid inequalities that enhance the performance of the MILP solvers. Furthermore, they propose a variable neighborhood search and tabu search-based algorithm to solve 50 vertices instances. Moshref-Javadi et al. (2020b) model a van-based drone delivery problem with a van can carry one or multiple drones. They conduct several theoretical bound analyses on maximum possible savings and then develop a hybrid metaheuristic approach based on simulated annealing and tabu search to solve the problem. Their algorithm can solve 159 vertices instances. Sacramento et al. (2019) design an adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones with maximum duration time for all routes cost-minimization objective. Their algorithm can solve 200 vertices instances. Wu et al. (2021) created an encoder-decoder framework combined with reinforcement learning to solve the routing problem in the van-based drone delivery problem with 200 vertices instances.

We give a detailed discussion to compare the similarities and differences between the most recent treatments of van-based drone delivery problems, drawn from Yu et al. (2020) (Table 1). The discussion focuses on how many vans and drones are considered and the objective of the problem, the number of customers a drone can visit in the course of one trip, whether there are time window constraints for customers, whether charging is allowed in the model, and the contribution of the work. More details can be found in the literature review paper of Li et al. (2021), Moshref-Javadi and Winkenbach (2021).

Table 1: Similarities and differences between van-based drone delivery problems

Reference	Vans	Drones	Objective	VMC	CTW	Charging	Contribution
Murray and Chu (2015)	1	1	time	1	no	no	MILP, Heuristic
Poikonen et al. (2017), Wang et al. (2017)	n	m	time	1	no	no	Theoretical insights
Pugliese and Guerriero (2017)	n	m	cost	1	yes	no	MILP
Luo et al. (2017)	n	n	time	m	no	no	MILP, Heuristic
Carlsson and Song (2018)	1	1	time	1	no	no	Heuristic
Agatz et al. (2018)	1	1	time	1	no	no	IP, Heuristic
Bouman et al. (2018)	1	1	cost	1	no	no	DP
Schermer et al. (2019a,b)	n	m	time	1	no	no	MILP, VNS, Matheuristic
Karak and Abdelghany (2019)	1	m	cost	m	no	no	MILP, Heuristic
Sacramento et al. (2019)	n	1	cost	1	no	no	MILP, ALNS
Wang and Sheu (2019)	n	m	cost	m	no	no	MIP, Branch-and-price
Poikonen et al. (2019)	1	1	time	1	no	no	Branch-and-bound
Murray and Raj (2020)	1	m	time	1	no	no	MILP, Heuristic
Poikonen and Golden (2020a)	1	1	time	1	no	no	Branch-and-bound, heuristic
Poikonen and Golden (2020b)	1	m	time	m	no	no	ILP, Heuristic
Kitjacharoenchai et al. (2020)	n	m	time	m	yes	no	MIP, Heuristic, LNS
Moshref-Javadi et al. (2020a)	1	m	waiting time	1	no	no	MIP, Heuristic
Moshref-Javadi et al. (2020b)	1	m	waiting time	1	no	no	MIP, Hybrid TS-SA
Dayarian et al. (2020)	1	1	maximize orders	m	no	no	Heuristics
Gonzalez-R et al. (2020)	1	1	time	m	no	no	MIP, Iterated greedy heuristic
Salama and Srinivas (2020)	1	m	time, cost	1	no	no	MIP, Heuristic
Roberti and Ruthmair (2021)	1	1	time	1	no	no	MILP, Branch-and-price
Kang and Lee (2021)	1	m	time	1	no	no	MILP, Benders decomposition
Luo et al. (2021)	1	m	time	m	no	no	MILP, TS
Moshref-Javadi et al. (2021)	1	m	time	no	no	no	MILP, Metaheuristic
Tamke and Buscher (2021)	n	m	time	1	no	no	MILP, Branch-and-cut
Wu et al. (2021)	1	1	time	m	no	no	MILP, Reinforcement learning

VMC*: the number of customers a drone can visit in the course of one trip

CTW*: whether there are time window constraints for customers

2.2. Van-based robot delivery

Many innovative concepts for intelligent delivery have recently been developed to reduce the negative impact of excessive traffic in urban areas. One of these concepts is van-based robot delivery. From the modeling, planning, and mathematical perspective, van-based drone delivery and van-based robot delivery share high similarities (Alfandari et al. 2022). For example, the basic van-based robot delivery and van-based drone delivery model are similar. However, when we calculate the travel distance, the robot usually uses the Manhattan distance, and the drone usually uses the Euclidean distance. The travel speed of drones is generally faster than vans, but the robots are not.

Boysen et al. (2018) introduce an innovative van-based robot delivery concept, in which a van loads the goods dedicated to a set of customers in a central depot and moves into the city center. Also on board are small robots that can be loaded with the goods dedicated to a single customer and launched from the van. Then, the robots move to their dedicated customers and, after delivery, the robots return to some robot depots in the city center. The van can replenish robots at these decentralized depots to launch further of them until all its customers are served. They prove the computational complexity of this problem and introduce exact and heuristic solution methods to solve it.

Chen et al. (2021) design a variant of vehicle routing problem with time window arising in the coordination of vans and robots in the last-mile delivery in populated areas. Each van carries several robots. While the van serves a customer, multiple robots could be dropped off to serve multiple customers at the customer's parking node. The van cannot leave before serving its customers and picking up all its robots. They improve an ALNS algorithm to handle instances with a large number of customers.

Simoni et al. (2020) investigate an integrated van and robot delivery model. They present a heuristic that efficiently identifies solutions based on initial van tours and corresponding joint robot operations. Numerical experiments show

that van-based robot delivery systems are efficient if robots are employed in heavily congested areas and appropriately retrofitted to accommodate several compartments in the robot's storage.

Ostermeier et al. (2021) consider the conventional van-and-robot concept in which vans do not serve customers directly, their numerical experiments show that the conventional van-and-robot approach can reduce last-mile delivery costs by up to 68% compared to van-only delivery. Heimfarth et al. (2022) consider the mixed van-and-robot concept in which both the van and its robots can serve customers directly. They propose a general variable neighborhood search method to solve this problem, which can reduce runtime by up to 94% compared to existing algorithms.

Yu et al. (2020) consider a new two-echelon urban delivery concept with time windows relying on second-level open route robot delivery. Their model does not allow a van to serve customers directly but allows a van to carry multiple robots. Numerical experiments show the bottleneck of this delivery system is robot speed. Yu et al. (2022) propose a van-based robot pickup and delivery model, where vans or vans carrying robots move along at the first-level route (a van can carry a single robot), serve van customers, or drop off/pick up, and replenish or swap their robots' batteries at parking nodes. Robots handle customer services along second-level open routes. For hybrid pickup and delivery operations, vans and robots can load goods from a depot and deliver them to a customer or pick up goods from a customer (supplier) and deliver them to another customer or to a depot. Their results show hybrid pickup and delivery operations can increase delivery system efficiency.

2.3. *Two-echelon (electric) vehicle routing problem*

In recent years, researchers have begun to study the two-echelon (electric) vehicle routing problem for application to city scenarios.

The two-echelon vehicle routing problem (2EVRP) involves a two-tier distribution network. In the first level, vans perform deliveries from distribution centers to satellites. In the second level, small vehicles load goods from satellites and can travel along any street in the city center area to serve final customers (Mühlbauer and Fontaine 2021, Sluijk et al. 2022).

The two-echelon electric vehicle routing problem adds electric vehicles to the 2EVRP. Breunig et al. (2019) extend the two-echelon vehicle routing problem where electric vehicles are used on the second echelon with full charging technology. They propose a large neighborhood search and an exact mathematical programming algorithm, which uses decomposition techniques to enumerate promising first-level solutions in conjunction with bounding functions and route enumeration for the second-level routes. Jie et al. (2019) and Wang et al. (2019) study the two-echelon vehicle routing problem with battery swap technology. Jie et al. (2019) consider a two-echelon capacitated electric vehicle routing problem with battery swapping stations. The first and second echelon vehicles could all swap their batteries in the battery swapping stations. They propose an integer programming formulation and a hybrid algorithm that combines a column generation and an adaptive large neighborhood search to solve the problem. Wang et al. (2019) address a two-echelon vehicle routing problem involving electric vehicles considering time windows and battery swapping stations. They formulate the mathematical model and minimize the total of shipping cost, handling cost, battery swapping cost, fixed cost of vehicles, and penalty cost due to tardiness.

Scholars have begun to consider the use of robots for second-level delivery. Bakach et al. (2021) study a two-echelon urban delivery network with robot-based deliveries. They determine the optimal number of local robot hubs and the optimal number of robots to service all customers and compare the resulting operational cost to conventional van-based deliveries. Their numerical experiments show that their system can save about 60% operational costs, for instance, with customer time windows. Alfandari et al. (2022) consider the objective of minimizing tardiness indicators in a two-echelon delivery network with robot-based deliveries. They provide a better understanding of how three major tardiness indicators (maximum tardiness, total tardiness, and the number of late deliveries) can be used to improve the quality of service. They propose an efficient branch-and-cut scheme to address instances of realistic size.

2.4. *Research gap and contribution*

From Sections 2.1-2.2 we conclude that existing van-based drone (robot) delivery problems consider various collaboration scenarios for vans and drones (robots). However, they mainly consider using the drone (robot) once or adopting battery swapping technology for multiple use. Existing research does not consider making full use of the time the van carries the drone (robot) en route for recharging, not even considering recharging scenarios. We note that research on classical two-echelon vehicle routing problems has started to consider electric vehicles. Still, the two-echelon electric vehicle problem does not involve the scenario of electric vans carrying robots, and the collaboration

is more straightforward compared to the van-based drone (robot) delivery problems. Going beyond the state of the art, we consider an electric van-based robot delivery with an en-route model. Our research allows for en-route charging, making full use of the time the van carries the robot, to improve system charging efficiency, which might improve overall system efficiency.

Compared with Yu et al. (2020) and Yu et al. (2022), which focuses on the difficulty in vehicle flow coordination and freight flow coordination, this paper focuses on the difficulty in energy flow coordination, in other words, electric van-based robot delivery with en-route charging.

3. Problem Setting

In 2E-VREC, vans or vans carrying robots move along a *1st-level route*, serve van customers, or drop off/pick up, and replenish their robots at parking nodes. Robots handle customer services along *2nd-level routes*, and are not required to return to the parking node where they left their van but they can be picked up at any parking node. The van can charge its onboard robot on the move or at nodes, and the van can be recharged at the parking nodes.

We define five types of route in the 2E-VREC model for the sake of illustration. The van route is the *1st-level-route*. The robot route is the *2nd-level-route*. In other words, the robot route is the route where a robot moves under its own power. The route along which the van carries its robot is termed the van-robot route, the route where the van moves independently is termed the independent-van route, and the route traveled by the robot is called the whole-robot route. A simple example together with a short description to describe the proposed model is shown in Appendix A.

In what follows, we first introduce the related technology on which the 2E-VREC problem is based. Then, we describe the critical assumptions used in the 2E-VREC problem. Finally, we define the fundamental notations used in this paper.

3.1. Technical details

Current robot delivery technology is mainly driven by the development of automated driving technology, which includes the controller, controller area network, global positioning system, inertial measurement unit, perception sensor (vision sensors, laser sensors, radar sensors, etc.), and automated driving algorithms.

Delivery robots navigate autonomously on sidewalks and bike lanes but can be remotely controlled in the event of problems (Heimfarth et al. 2022). Robot delivery is usually used in last-mile delivery scenarios. In practical applications, when a robot arrives at a delivery node, customers are notified by phone (Marble 2019). The customer enters a code to unlock the corresponding compartment of the robot and then take away the goods (Yu et al. 2022).

Small robots travel slowly and can carry light loads for shorter journeys. For example, the Starship Technology delivery robot (Starship 2017), only contains a single compartment. The maximum speed and average effective service speed of this robot are 6 km/h and 3 km/h, respectively, the maximum cargo weight is 10 kg, and the maximum travel distance is 6 km. Larger robots, such as JD.COM delivery robots (JD.COM 2022), have multiple compartments. The maximum speed and average effective service speed of these robots are 25 km/h and 10 km/h, respectively, the weight of the whole robot is 400kg, and the maximum travel distance is 50 km .

Given the potential limitations of pure robot deliveries, Mercedes-Benz and Starship technologies present the idea of a mothership approach in their Future Transportation unit (Daimler 2018, Technologies 2016).

En-route charging has already been implemented and used by mobile charging platforms to charge drones (Yu et al. 2019), and mobile charging stations to charge vehicles (Huang et al. 2014, Cui et al. 2018).

3.2. Critical assumptions

In the real world, there are customers located in narrow streets, on campuses, or in other communities where the entry of vans is restricted. We therefore distinguish two kinds of customers. One kind can be visited by either the van or the robot: van customers. The other kind can only be visited by the robot: robot customers. This implies that robots can serve all the customers, but the van can only serve a subset.

We assume there are specific parking nodes for the vehicles, used for the van to meet, replenish and recharge its robot, rather than performing these operations at customer nodes. This setting is reasonable, especially for electric vehicles, since it is often impossible for vans/robots to be recharged at customer nodes, and parking nodes can be visited multiple times by the vans/robots.

The freight carried by the van must be loaded at the beginning of the tours at the depot. There are maximum capacities for both the van and the robot. We assume that when the robot is on board the van, the van and its robot's total load cannot exceed the van's maximum capacity. We also allow a robot to visit multiple customers during a tour. In practical applications, there are several independent containers in a robot. After a customer enters the pickup code in the robot, the corresponding container will open, and the customer can take out the goods.

We consider partial recharging, in which the electric vehicle may not be completely, as described in Keskin and Çatay (2016). We assume the van can be recharged at parking nodes. The robot can be recharged by the van while moving or at parking nodes and at customer nodes if it is on board the van. Recharging takes time for both the van and the robot, depending on the recharging rate and level. In addition, there are maximum battery capacities for both the van and the robot. We note that the battery capacity of the robot corresponds to its maximum individual travel distance. The van's battery not only provides its own power, but is also needed to recharge the robots. When the van charges the robot, the power of the van decreases, while the power of the robot increases.

We assume that each customer node must be visited by exactly one van/robot once. Customer nodes and the depot have time windows. Our model allows costless waiting at all locations.

We make the following further assumptions: (i) each van carries a single robot, (ii) the robot cannot leave the depot to serve customers directly, (iii) a robot dropped off by a van must be picked up by the same van, (iv) the operating time spent dropping off, picking up, replenishing and preparing for recharging is ignored, (v) the charging function is linear as most EVRP study assumed, and (vi) there are enough vans in the depot to meet the total demand.

To illustrate which route type is allowed and which is not allowed, we show the 2E-VREC route cases in Figure 1. For the customer nodes, cases (1)–(3) show that customers can only be reached and then left by van, robot, or van carrying robot, respectively. Cases (4)–(6) forbid customers being reached by both van and robot independently or left by both van and robot independently. For the parking nodes, case (7) allows a van to arrive and depart from a parking node. Case (8) forbids a robot from accessing the parking node independently unless there is a van also accessing this node – cases (9)–(12). For the depot, only the van carrying the robot can arrive and leave the depot – case (15) and case (19), while cases (13), (14), (16), (17), (18), (20) are forbidden.

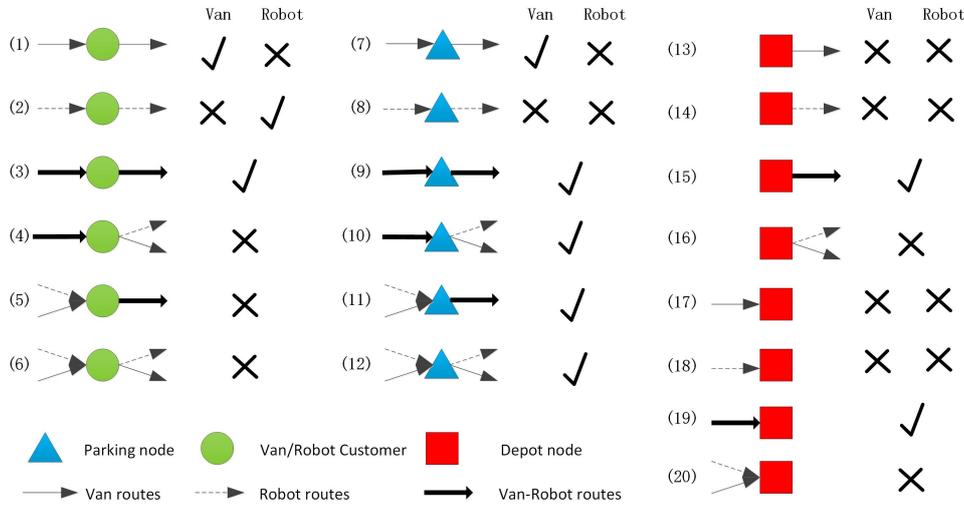


Figure 1: 2E-VREC route cases of permission and prohibition

3.3. Fundamental notation

The 2E-VREC problem is defined on a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, where the depot \mathcal{V}_0 is represented by two nodes 0 and $0'$. Every van route starts at node 0 and ends at node $0'$. Let \mathcal{V}_s be the set of parking nodes where vans can drop off, pick up, replenish and swap the battery for their robots. A parking node can be visited more than once, and we introduce dummy nodes of the parking node allowing to model these multiple visits. Up to $n + 1$ dummy stations are required in the worst case, where n is the number of customer nodes.

Let \mathcal{V}_{c1} denote the van customer node set and \mathcal{V}_c the customer node set, $\mathcal{V}_{c1} \subseteq \mathcal{V}_c$. The sets $\mathcal{V}_\alpha^0, \mathcal{V}_\alpha', \mathcal{V}_\alpha''$ denote the node set \mathcal{V}_α union the depot $0, 0',$ and $0 \cup 0'$ respectively. Where α stands for different combinations of customer and parking node sets. For example we let $\mathcal{V}_{sc1} = \mathcal{V}_s \cup \mathcal{V}_{c1}$. Let $\mathcal{A}_1 = \{(i, j) \mid i \in \{0\}; j \in \mathcal{V}_{sc1}\} \cup \{(i, j) \mid i, j \in \mathcal{V}_{sc1}, i \neq j\} \cup \{(i, j) \mid i \in \mathcal{V}_{sc1}; j \in \{0'\}\}$ be the arc set corresponding to the van routes and let $\mathcal{A}_2 = \{(i, j) \mid i \in \mathcal{V}_s; j \in \mathcal{V}_c\} \cup \{(i, j) \mid i, j \in \mathcal{V}_c, i \neq j\} \cup \{(i, j) \mid i \in \mathcal{V}_c; j \in \mathcal{V}_s\}$ be the arc set corresponding to the robot routes. We also let $\mathcal{A}_3 = \mathcal{A}_1 \cup \mathcal{A}_2$ be the arc set corresponding to complete possible robot routes and $\mathcal{A}_4 = \mathcal{A}_3 \setminus \mathcal{A}_1$ be the arc set corresponding to routes that the van cannot reach.

We define c_1 and c_2 as the unit travel cost of vans and robots, and v_1 and v_2 the travel speed of vans and robots, respectively. For each edge, $d_{i,j}$ is the associated travel distance, $c_1 d_{i,j}$ ($c_2 d_{i,j}$) is the associated travel cost and $d_{i,j}/v_1$ ($d_{i,j}/v_2$) is the associated travel time for the van (robot). The freight must be delivered from the depot $\{0\}$ to customer nodes i with the demand q_i and serving time s_i . The time window of the customer nodes $i \in \mathcal{V}_c$ is $[a_i, b_i]$, is the time interval where the service at node i is permitted to start. We also let a_0 represent the earliest possible departure time from the depot 0 , and $b_{0'}$ represent the latest possible arrival times at the depot $0'$. These time windows are hard. $\mathcal{K} = \{1, 2, \dots, k, \dots, K\}$ is the set of vans, where K is the number of vans, and k represents the k th van. Because vans and robots are in one-to-one correspondence, \mathcal{K} also corresponds to the set of robots. M refers to an arbitrary large constant number in the model.

For configuration of vans and robots, we assume C_1 and C_2 represent their load capacity, Q_1 and Q_2 represent their battery capacity, g_1 and g_2 represent their the recharging rates, and h_1 and h_2 represent their energy consumption rate. Note that when the robot is on board the van, the total load of the van and its robot cannot exceed C_1 .

The following decision variables are required for the formulation of the 2E-VREC. Binary variables $x_{i,j,k}$ take value 1 if and only if arc (i, j) in \mathcal{A}_1 is traveled by the k th – van. Binary variables $y_{i,j,k}$ are equal to 1 if and only if arc (i, j) in \mathcal{A}_3 is traveled by the k th – robot. Binary variables $z_{i,j,k}$ take value 1 if arc (i, j) in \mathcal{A}_1 is traveled by the k th – van with its robot on board, 0 otherwise. Variables o_i^k and O_i^k refer to the recharging amount of the robot and van at the parking node $i, i \in \mathcal{V}_s$. Variables $o_{i,j}^k$ represent the recharging amount of the robot by the van during arc $(i, j), (i, j) \in \mathcal{A}_1$. Variables E_i^k and e_i^k refer to the remaining battery amount of the van and the robot at node i on arrival. Variables W_i^k and w_i^k refer to the arrival time for the van and robot at node i . Variables $f_{i,j,k}$ represent the freight flow of the robot in arc (i, j) in \mathcal{A}_2 . Variables u_i^k represent count variable for node $i \in \mathcal{V}_s$ of vehicle k .

The sets, variables and parameters of our model are summarized in the table of Appendix B.

4. Methodology

In this section, we first propose the mathematical model of the proposed problem, it can provide exact benchmark solutions for smaller instances to evaluate the adaptive large neighborhood search algorithm. The adaptive large neighborhood search algorithm is proposed to approximately solve the larger instances. Since evaluating the feasibility of a single given route is an essential part of the adaptive large neighborhood algorithm, we finally detailed introduce the single route evaluation and feasibility approaches.

4.1. Model formulation

In this section, we present a mixed integer programming model for the electric van-based robot delivery with en-route charging problem.

Objective:

$$\min \left(\sum_{k \in \mathcal{K}} \sum_{(i,j) \in \mathcal{A}_1} c_1 d_{i,j} x_{i,j,k} + \sum_{k \in \mathcal{K}} \sum_{(i,j) \in \mathcal{A}_3} c_2 d_{i,j} y_{i,j,k} - \sum_{k \in \mathcal{K}} \sum_{(i,j) \in \mathcal{A}_1} c_2 d_{i,j} z_{i,j,k} \right) \quad (1)$$

Subject to:

$$\begin{aligned} \sum_{(i,j) \in \mathcal{A}_1} x_{i,j,k} &\leq 1, & \forall j \in \mathcal{V}_{sc1}, k \in \mathcal{K}, & (2) \\ \sum_{(i,j) \in \mathcal{A}_1} x_{i,j,k} - \sum_{(j,i) \in \mathcal{A}_1} x_{j,i,k} &= 0, & \forall j \in \mathcal{V}_{sc1}, k \in \mathcal{K}, & (3) \\ \sum_{(i,j) \in \mathcal{A}_3} y_{i,j,k} &\leq 1, & \forall j \in \mathcal{V}_{sc}, k \in \mathcal{K}, & (4) \\ \sum_{(i,j) \in \mathcal{A}_3} y_{i,j,k} - \sum_{(j,i) \in \mathcal{A}_3} y_{j,i,k} &= 0, & \forall j \in \mathcal{V}_{sc}, k \in \mathcal{K}, & (5) \\ \sum_{i \in \mathcal{V}_{sc1}} x_{i,0',k} = \sum_{j \in \mathcal{V}_{sc1}} x_{0,j,k} = \sum_{i \in \mathcal{V}_{sc1}} y_{i,0',k} = \sum_{j \in \mathcal{V}_{sc1}} y_{0,j,k} &\leq 1, & \forall k \in \mathcal{K}, & (6) \\ \sum_{k \in \mathcal{K}} \left(\sum_{(i,j) \in \mathcal{A}_3} y_{i,j,k} + \sum_{(i,j) \in \mathcal{A}_3} x_{i,j,k} - \sum_{(i,j) \in \mathcal{A}_3} z_{i,j,k} \right) &= 1, & \forall j \in \mathcal{V}_c, & (7) \\ \sum_{(i,j) \in \mathcal{A}_3} y_{i,j,k} + \sum_{(i,j) \in \mathcal{A}_1} x_{i,j,k} - \sum_{(i,j) \in \mathcal{A}_1} z_{i,j,k} &\leq 1, & \forall i \in \mathcal{V}_{c1}^0, k \in \mathcal{K}, & (8) \\ \sum_{(i,j) \in \mathcal{A}_3} y_{i,j,k} + \sum_{(i,j) \in \mathcal{A}_1} x_{i,j,k} - \sum_{(i,j) \in \mathcal{A}_1} z_{i,j,k} &\leq 1, & \forall j \in \mathcal{V}'_{c1}, k \in \mathcal{K}, & (9) \\ \sum_{(i,j) \in \mathcal{A}_3} y_{i,j,k} &\leq \sum_{(i,j) \in \mathcal{A}_1} x_{i,j,k}, & \forall i \in \mathcal{V}_s, k \in \mathcal{K}, & (10) \\ 2 \times z_{i,j,k} &\leq x_{i,j,k} + y_{i,j,k} \leq 2 \times z_{i,j,k} + 1, & \forall (i,j) \in \mathcal{A}_1, k \in \mathcal{K}, & (11) \\ x_{i,j,k}, y_{i,j,k}, z_{i,j,k} &\in \{0, 1\}, & \forall (i,j) \in \mathcal{A}_3, k \in \mathcal{K}, & (12) \\ z_{i,j,k} &= 0, & \forall (i,j) \in \mathcal{A}_4, k \in \mathcal{K}, & (13) \\ x_{i,j,k} &= 0, & \forall (i,j) \in \mathcal{A}_4, k \in \mathcal{K}, & (14) \\ \max(W_i^k + O_i^k/g_1, W_i^k + o_i^k/g_2, w_i^k + o_i^k/g_2) + d_{i,j}/v_1 - W_j^k &\leq M(1 - x_{i,j,k}), & \forall i \in \mathcal{V}_s^0(i,j) \in \mathcal{A}_1, k \in \mathcal{K}, & (15) \\ \max(W_i^k, w_i^k) + o_i^k/g_2 + d_{i,j}/v_2 - w_j^k &\leq M(1 - y_{i,j,k} + x_{i,j,k}), & \forall i \in \mathcal{V}_s(i,j) \in \mathcal{A}_2, k \in \mathcal{K}, & (16) \\ W_i^k + \max(s_i, o_i^k/g_2) + d_{i,j}/v_1 - W_j^k &\leq M(1 - x_{i,j,k}), & \forall i \in \mathcal{V}_{c1}(i,j) \in \mathcal{A}_1, k \in \mathcal{K}, & (17) \\ w_i^k + s_i + d_{i,j}/v_2 - w_j^k &\leq M(1 - y_{i,j,k} + x_{i,j,k}), & \forall i \in \mathcal{V}_c(i,j) \in \mathcal{A}_2, k \in \mathcal{K}, & (18) \\ |W_j^k - w_j^k| &\leq M \left(1 - \sum_{(i,j) \in \mathcal{A}_1} z_{i,j,k} \right), & \forall j \in \mathcal{V}'_{sc1}, k \in \mathcal{K}, & (19) \\ a_i &\leq W_i^k, w_i^k \leq b_i, & \forall i \in \mathcal{V}'_c, k \in \mathcal{K}, & (20) \\ E_j^k + h_1 d_{i,j} - O_i^k + o_i^k + o_{i,j}^k - E_i^k &\leq M(1 - x_{i,j,k}), & \forall (i,j) \in \mathcal{A}_1 | i \in \mathcal{V}_s^0, k \in \mathcal{K}, & (21) \\ E_j^k + h_1 d_{i,j} + o_i^k + o_{i,j}^k - E_i^k &\leq M(1 - x_{i,j,k}), & \forall (i,j) \in \mathcal{A}_1 | i \in \mathcal{V}_{c1}, k \in \mathcal{K}, & (22) \\ e_j^k - o_i^k - o_{i,j}^k - e_i^k &\leq M(1 - z_{i,j,k}), & \forall (i,j) \in \mathcal{A}_1 | i \in \mathcal{V}_{sc1}^0, k \in \mathcal{K}, & (23) \\ e_j^k + h_2 d_{i,j} - o_i^k - e_i^k &\leq M(1 - y_{i,j,k} + x_{i,j,k}), & \forall (i,j) \in \mathcal{A}_2 | i \in \mathcal{V}_s, k \in \mathcal{K}, & (24) \\ e_j^k + h_2 d_{i,j} - e_i^k &\leq M(1 - y_{i,j,k} + x_{i,j,k}), & \forall (i,j) \in \mathcal{A}_2 | i \in \mathcal{V}_c, k \in \mathcal{K}, & (25) \\ 0 \leq o_i^k &\leq Q_2 \sum_{(i,j) \in \mathcal{A}_1} y_{i,j,k}, & \forall i \in \mathcal{V}_s, k \in \mathcal{K}, & (26) \\ 0 \leq o_i^k &\leq Q_2 \sum_{(i,j) \in \mathcal{A}_1} z_{i,j,k}, & \forall i \in \mathcal{V}_s, k \in \mathcal{K}, & (27) \\ 0 \leq O_i^k &\leq (Q_1 + Q_2) \sum_{(i,j) \in \mathcal{A}_1} x_{i,j,k}, & \forall i \in \mathcal{V}_s^0, k \in \mathcal{K}, & (28) \\ 0 \leq o_{i,j}^k &\leq Q_2 z_{i,j,k}, & \forall (i,j) \in \mathcal{A}_1, k \in \mathcal{K}, & (29) \\ 0 \leq E_i^k + O_i^k - o_i^k &\leq Q_1, & \forall i \in \mathcal{V}_s^0, k \in \mathcal{K}, & (30) \\ 0 \leq e_i^k + o_i^k &\leq Q_2, & \forall i \in \mathcal{V}_s^0, k \in \mathcal{K}, & (31) \\ 0 \leq E_i^k &\leq Q_1, & \forall i \in \mathcal{V}'_{sc1}, k \in \mathcal{K}, & (32) \\ 0 \leq e_i^k &\leq Q_2, & \forall i \in \mathcal{V}'_{sc}, k \in \mathcal{K}, & (33) \end{aligned}$$

$$\sum_{(i,j) \in \mathcal{A}_2} f_{i,j,k} - \sum_{(j,i) \in \mathcal{A}_2} f_{j,i,k} = q_j \sum_{(i,j) \in \mathcal{A}_2} (y_{i,j,k} - z_{i,j,k}), \quad \forall j \in \mathcal{V}_c, k \in \mathcal{K}, \quad (34)$$

$$\sum_{i \in \mathcal{V}_c} q_i \sum_{j \in \mathcal{V}_{sc}^0(i,j) \in \mathcal{A}_2} (y_{i,j,k} - z_{i,j,k}) + \sum_{i \in \mathcal{V}_{c1}} q_i \sum_{j \in \mathcal{V}_{sc1}^0(i,j) \in \mathcal{A}_1} x_{i,j,k} \leq C_1, \quad \forall k \in \mathcal{K}, \quad (35)$$

$$0 \leq f_{i,j,k} \leq C_2(y_{i,j,k} - z_{i,j,k}), \quad \forall (i,j) \in \mathcal{A}_2, k \in \mathcal{K}, \quad (36)$$

$$u_i^k - u_j^k + 1 \leq M(1 - x_{i,j,k}), \quad \forall i \in \mathcal{V}_s, j \in \mathcal{V}_s, (i,j) \in \mathcal{A}_1, k \in \mathcal{K}, \quad (37)$$

$$0 \leq u_i^k \leq M, \quad \forall i \in \mathcal{V}_s, k \in \mathcal{K}. \quad (38)$$

The objective function (1) minimizes the total travel cost. It corresponds to the van route cost plus the whole-robot route cost, minus the van-robot route cost.

Constraints (2)-(14) are van/robot arc constraints used to plan and limit the routes of vans and robots. Constraints (2)-(6) ensure that every node is visited by a van/robot at most once, and the numbers of departures are equal to the numbers of arrivals. Constraints (2) ensure a van can visit a parking node or a van customer node no more than once. Constraints (3) ensure that each van entering a parking or customer node is required to leave it. Constraints (4) enforce that robots do not visit parking or customer nodes more than once. Constraints (5) formulate similar to constraint (3). Constraints (6) force the number of a van leaving the depot to be equal to that of its robot leaving the depot, and equal to the number of van/robot coming back to the depot. Constraints (7) ensure that every customer node is visited by vans or robots exactly once. Constraints (8)-(9) specify for customer nodes and depot, that a van and its robot cannot visit the same van customer node or depot unless the robot is on board the van. Constraints (8) ensure a van and its robot cannot leave a van customer node or the depot separately. Constraints (9) state that a van and its robot cannot arrive at a van customer node or the depot separately. Constraints (10) represent that a robot cannot visit a parking node unless its van visited this node. Constraints (11) let $z_{i,j,k}$ be equal to 1 only if $x_{i,j,k}$ and $y_{i,j,k}$ are all equal to 1, which means arc (i,j) in \mathcal{A}_1 is traveled by the k th - van with its robot on board. Constraints (12) are the binary variable constraints. Constraints (13)-(14) force the arc variable to be equal to 0 where they are not allowed to visit.

Constraints (15)-(31) are the time and energy constraints. Constraints (15)-(16) are time flow constraints describing vehicle departure from the parking node or depot. Constraints (15) are the time flow of the van routes, and constraints (16) are the time flow of robot routes. Note that constraints (15)-(16) ensure the feasible rendezvous of a van and its robot even in the complex scenario: case (12) in Figure 1. In constraints (15), we have $w_i^k + o_i^k/g_2 + d_{i,j}/v_1 - W_j^k \leq M(1 - x_{i,j,k}), \forall i \in \mathcal{V}_s^0(i,j) \in \mathcal{A}_1, k \in \mathcal{K}$. When $x_{i,j,k} = 1$, we have $w_i^k + o_i^k/g_2 \leq W_j^k - d_{i,j}/v_1$. Since $o_i^k/g_2 \geq 0$, inequality $w_i^k \leq W_j^k - d_{i,j}/v_1$ holds, refers to the departure time of a van ($W_j^k - d_{i,j}/v_1$) at node i is larger than the arrival time of its robot (w_i^k) at node i , ensuring only when a van's robot arrive at a parking node can the van leave this parking node. In constraints (16), we have $W_i^k + o_i^k/g_2 + d_{i,j}/v_2 - w_j^k \leq M(1 - y_{i,j,k} + x_{i,j,k}), \forall i \in \mathcal{V}_s(i,j) \in \mathcal{A}_2, k \in \mathcal{K}$. When $y_{i,j,k} = 1$ and $x_{i,j,k} = 0$, we have $W_i^k + o_i^k/g_2 \leq w_j^k - d_{i,j}/v_2$. Since $o_i^k/g_2 \geq 0$, inequality $W_i^k \leq w_j^k - d_{i,j}/v_2$ holds, represents the departure time of a robot ($w_j^k - d_{i,j}/v_1$) at node i is larger than the arrival time of its corresponding van (W_i^k) at node i , ensuring only when a van arrive at a parking node can its robot leave this parking node. Constraints (17)-(18) are time flow constraints, ensuring that the van and robot depart from the customer node. Constraints (17) are the time flow of the van routes, and constraints (18) are the time flow of the robot routes. Note in constraints (15)-(17) we consider the charging time of the van and the robot in the time flow constraint. Constraints (19) force the arrival time of the van at a node to be equal to that of the robot if its van is carrying the robot. Constraints (20) are time window constraints.

Constraints (21)-(22) define the battery level (energy flow conservation) of van route in \mathcal{A}_1 . Constraints (21) enforce energy flow conservation when a van departs from a parking node or depot. The energy when the van leaves the parking node or depot ($E_i^k + O_i^k - o_i^k$) minus the energy consumed by the van en route ($h_1 d_{i,j} + o_{i,j}^k$) is greater than the energy when the van reaches the arrival node (E_j^k). Constraints (22) enforce energy flow conservation when a van departs from a van customer node. The energy when the van leaves the van customer node ($E_i^k - o_i^k$) minus the energy consumed by the van en route ($h_1 d_{i,j} + o_{i,j}^k$) is greater than the energy when the van reaches the arrival node (E_j^k). Constraints (23)-(25) define the energy flow of the robot. Constraints (23) ensure the robot energy flow conservation when it is carried. The energy when the robot leaves the parking node, depot, or van customer node ($e_i^k + o_i^k$) plus the energy charged by the van en route ($o_{i,j}^k$) is greater than the energy when the robot reaches the arrival node (e_j^k). Constraints (24)-(25) ensure the energy flow conservation when the robot moves on its own. Constraints (26)-(33)

represent the variable constraints. Constraints (26)-(27) define that only when both the van and the robot reach the same node, can the charging amount of robot be greater than 0. Constraints (26) represent the case where the van and robot rendezvous at the parking node while constraints (27) are the case where the van carries the robot to the parking node. Constraints (28) ensure that only when a van reaches a parking node/depot can the charging amount of the van be greater than 0, and the total recharging amount should be less than the total battery capacity of van and its robot. Constraints (29) ensure that only when a van carries its robot can the charging amount of the robot en-route be greater than 0. Constraints (30)-(31) ensure that when a van or a robot leaves a parking node, the battery amount of van/robot is below its maximum battery capacity. Constraints (32)-(33) are variable constraints.

Constraints (34)-(36) are freight flow constraints. Constraints (34) are the freight flow conservation along *2nd-level routes* (robot routes). Constraints (35) guarantee that the total load of a van and its robot does not exceed the van's capacity. Constraints (36) are the variable constraints, ensuring the freight flow does not exceed robot capacity on the *2nd-level route*.

Constraints (37)-(38) are Miller-Tucker-Zemlin constraints. Constraints (37)-(38) eliminate subtours in dummy stations that correspond to the same physical node (the distance and travel time between these dummy stations are 0). Furthermore, these constraints speed up solving the MIP model, since these constraints eliminate a lot of meaningless paths.

Note: Some constraints are used to allow or forbid the cases in Figure 1. Constraints (6) forbid cases (13), (14), (17), (18). Constraints (8)-(9) forbid cases (4), (5), (6), (16), (20). Constraints (10) forbid case (8). Besides, constraints (15)-(16) can ensure the feasible rendezvous of a van and its robot in case (12). In addition, nonlinear constraints in the MIP model can be linearized as follows.

Constraints (15)-(17) can be linearized as follows:

$$W_i^k + o_i^k/g_1 + d_{i,j}/v_1 - W_j^k \leq M(1 - x_{i,j,k}), \quad \forall \{i \in \mathcal{V}_s^0 | (i, j) \in \mathcal{A}_1\}, k \in \mathcal{K}, \quad (39)$$

$$W_i^k + o_i^k/g_2 + d_{i,j}/v_1 - W_j^k \leq M(1 - x_{i,j,k}), \quad \forall \{i \in \mathcal{V}_s^0 | (i, j) \in \mathcal{A}_1\}, k \in \mathcal{K}, \quad (40)$$

$$w_i^k + o_i^k/g_2 + d_{i,j}/v_1 - W_j^k \leq M(1 - x_{i,j,k}), \quad \forall \{i \in \mathcal{V}_s^0 | (i, j) \in \mathcal{A}_1\}, k \in \mathcal{K}, \quad (41)$$

$$W_i^k + o_i^k/g_2 + d_{i,j}/v_2 - W_j^k \leq M(1 - y_{i,j,k} + x_{i,j,k}), \quad \forall \{i \in \mathcal{V}_s | (i, j) \in \mathcal{A}_2\}, k \in \mathcal{K}, \quad (42)$$

$$w_i^k + o_i^k/g_2 + d_{i,j}/v_2 - W_j^k \leq M(1 - y_{i,j,k} + x_{i,j,k}), \quad \forall \{i \in \mathcal{V}_s | (i, j) \in \mathcal{A}_2\}, k \in \mathcal{K}, \quad (43)$$

$$W_i^k + s_i + d_{i,j}/v_1 - W_j^k \leq M(1 - x_{i,j,k}), \quad \forall \{i \in \mathcal{V}_{c1} | (i, j) \in \mathcal{A}_1\}, k \in \mathcal{K}, \quad (44)$$

$$W_i^k + o_i^k/g_2 + d_{i,j}/v_1 - W_j^k \leq M(1 - x_{i,j,k}), \quad \forall \{i \in \mathcal{V}_{c1} | (i, j) \in \mathcal{A}_1\}, k \in \mathcal{K}. \quad (45)$$

Constraints (19) can be linearized as follows:

$$-M \left(1 - \sum_{(i,j) \in \mathcal{A}_1} z_{i,j,k} \right) \leq W_j^k - w_j^k \leq M \left(1 - \sum_{(i,j) \in \mathcal{A}_1} z_{i,j,k} \right), \quad \forall \{j \in \mathcal{V}'_{sc1}\}, k \in \mathcal{K}. \quad (46)$$

4.2. Adaptive large neighborhood search

In this section, we propose an ALNS algorithm to solve the 2E-VREC problem. Since our problem considers electric vehicles in a two-echelon concept, we draw on some operations proposed in Hemmelmayr et al. (2012), Keskin and Çatay (2016), and Sacramento et al. (2019). The specific structure of our problem lies in parking stations that have a variety of functions, unlike other similar situations that have been solved by ALNS. For example, in our problem, parking stations can be used for dropping off and picking up robots (vehicle transshipment), and can also be used as charging stations or stations where a van can top up its robot. The parking station is an independent station and can be visited multiple times. Table 2 gives the characteristics of stations in different problem types. Our ALNS algorithm framework is the same as the framework in Yu et al. (2022), which can be found in Appendix C.

4.2.1. Initial solution

At the beginning of the algorithm, all customers are inserted into an un-assigned customer list in random order. We then repeat the following two steps until all the customer nodes are assigned. In the first step, we randomly choose a customer node in the un-assigned customer list. If the chosen node is a robot customer node, then we construct a 2E-VREC route with van routes and robot routes. Otherwise, we decide whether to construct a 2E-VREC route with van routes and robot routes or to construct a 2E-VREC route only including van routes, according to roulette-wheel

Table 2: Characteristics of station in different problems

	Charging	Vehicle transshipment	Freight transshipment	Multi-visit station	Independent station
Electric vehicle routing problem	yes	no	no	yes	yes
Two-echelon vehicle routing problem	yes	no	yes	no	yes
Van-based drone delivery problem	no	yes	yes	no	no
This work	yes	yes	yes	yes	yes

selection. The second step randomly sorts the order of the un-assigned customer list, and then chooses a customer to insert into the random-feasible position of the constructed route, until no customer node can be inserted.

4.2.2. Destroy moves and operations

A destroy operation destroys a chosen 2E-VREC route by removing nodes or routes. We define four destroy operations: customer removal, station removal, route closure, and route destruction. The ALNS algorithm destroys a part of the current solution in each iteration. We mainly adopt a random and a greedy destroy strategy.

In our problem setting, we denote a parking station used for charging only a charging station. Otherwise, it is denoted a connection station. We distinguish between removing two kinds of stations in station removal operations, since removing a connection station dramatically changes the solution, while removing a charging station does not.

The four destroy operations and the moves used are as follows:

1. Customer removal operation: The customer removal operation removes customers from a 2E-VREC route with a probability $\beta \in [0, 1]$. We adopt *random customer removal* and *greedy customer removal* moves.
 - *Random customer removal* randomly removes a customer node.
 - *Greedy customer removal* removes the customer that can yield the largest cost reduction for a given route.
2. Station removal operation: The station removal operation removes a parking station from a 2E-VREC route. Here we adopt *station-route removal*, *charging station removal* and *redundant charging station removal* moves. After we remove a station, the route may become battery-unfeasible. We remove a customer before or after the removed station in the van route, or we insert a station different from the removed station in the van route to make the route feasible if needed, since visiting fewer customers or visiting more stations could make a battery-unfeasible route feasible.
 - *Station-route removal* randomly removes a connection station and the robot routes depart from and arrive at this station.
 - *Charging station removal* randomly removes a charging station.
 - *Redundant charging station removal* removes a redundant charging station if one exists.
3. Route closure operation: The route closure operation closes a van or robot route in a 2E-VREC route. We use *van/robot route closure* moves, and we adopt *random/greedy* strategies.
 - *Random van/robot route closure* removes all customers from a randomly chosen van/robot route.
 - *Greedy van/robot route closure* removes the route that can yield the largest cost reduction.
4. Route destruction operation: The route destruction operation chooses a 2E-VREC route and then destroys it. All customer nodes in this 2E-VREC route are put in the un-assigned customer list. We adopt *random/greedy route destruction* moves.
 - *Random route destruction* randomly chooses a 2E-VREC route in the solution to destroy.
 - *Greedy route destruction* chooses a 2E-VREC route, with a minimum number of customer nodes to destroy.

4.2.3. Repair moves and operations

Route repair operations repair the existing routes or construct a new route if needed, using three kinds of repair operations: route reconstruction, customer insertion, and route structure change first / customer insert second. The repair operations do not allow unfeasible solutions but allow worse solutions through simulated annealing evaluation and calibration. Note that in the repair process, we consider opening a new van/robot route from a connection station, or changing the role of charging stations to connection stations and then opening a robot route to considerably change the solution.

The three kinds of repair operation are as follows.

1. Route reconstruction operation: The route reconstruction operation ensures that we always obtain a viable solution. We adopt the same method as we used to obtain the initial solution in Section 4.2.1.
2. Customer insertion operation: The customer insertion operation is used to insert customers in a 2E-VREC route. First, we use *random/greedy customer insertion* moves to insert customer nodes. If a customer cannot be inserted because of battery outage, we try to use *nearest station insertion* moves to make the route feasible. If all the possible positions have been tried and the customer still cannot be inserted, we try to open a new robot route, where the start node and end node of the route are the same, and then insert the customer node. Finally, if there still are unassigned customer nodes, we use route reconstruction-based repair to reconstruct a new 2E-VREC route.
 - *Random customer insertion* sorts the order of the un-served customers list randomly and chooses a customer to insert into the random-feasible position of the 2E-VREC route until all customers have been tried.
 - *Greedy customer insertion* sorts the order of un-served customers list randomly, and then chooses a customer to insert into the 2E-VREC route with the least cost increases.
 - *Nearest station insertion* is used after we have inserted a customer node and found the route is unfeasible: we try to insert a station before or after the inserted customer position.
3. Route structure change first / customer insert second operation: We present two kinds of route structure change approach: (i) we use *van/robot route open* moves to begin a new van/robot route from a station; the van/robot route includes one customer node, and the ending station can be the same one or a different one; we then perform a customer insertion operation, (ii) we first use *random/greedy station insertion* moves to insert a station into a 2E-VREC route and then follow route open moves at this station if the station is not inserted in an independent van route; next, we conduct customer insertion- based repair moves.
 - *Van route open* to begin a new van route from a connection station.
 - *Robot route open* to begin a new robot route from a station; *robot route open* can convert a charging station into a connection station or increase the robot route from a connection station. The ending station can be the same as or different from the starting station.
 - *Random station insertion* randomly chooses a position in the 2E-VREC route to insert a station.
 - *Greedy station insertion* chooses the best insertion position with feasible and least total travel cost increases to insert a station.

4.3. Single route evaluation and feasibility

We propose a local search based metaheuristic for solving the problem presented. Before developing the metaheuristic, we need to introduce methods required for evaluating and checking the feasibility of a given 2E-VREC route.

Move procedures in metaheuristics for solving vehicle routing problems aim to determine which node to visit in what order. However, other essential decisions need to be made during move procedures. For example, we need to decide the departure time of a vehicle from a node (Savelsbergh and Martin 1992), and the charging time at a charging station in EVRP (Hiermann et al. 2016, 2019).

In our problem, vans and robots are all-electric vehicles. The van can charge the robot when the van is carrying its robot and charging needs time. The energy transfer is negatively correlated with the available travel distance of a van,

but positively correlated with the available travel distance of its robot, which makes the time-distance-energy trade-off between vans and robots complex, leading to an unconventional charging function. We need to make decisions about: (i) who charges, (ii) how much, (iii) where, and (iv) when. We also need to consider synchronization in the mothership system, such as pickup and delivery, time windows, and other constraints. Hence, making a suitable decision involves a broad variety of time-based constraints and complex trade-offs that are not straightforward.

Because of the increasing difficulty met when making complex decisions for a given route, sometimes even checking the feasibility of a given single route is a hard, possibly even an NP-hard problem. Using heuristic methods (Masson et al. 2017, Hunsaker and Savelsbergh 2002) to solve such cases have been considered.

In this section, we first propose a greedy route checking approach to solving our route feasibility checking problem. We then present a linear programming (LP) checking approach to accurately check the feasibility of a given single route. We also compare the two route checking approaches in Section 5. Checking the freight of a given route is straightforward, and so will not be dealt with here.

4.3.1. Greedy route checking approach

We draw on the idea of a greedy route evaluation approach (Hiermann et al. 2019) to solve our 2E-VREC route feasibility checking problem. Here we apply a greedy recharging policy with a simple rule, which is to give priority to charging the van battery over the robot battery: the robot needs the van to carry and charge it, so prioritizing the van battery is reasonable.

We define the time warp to represent how much time can be used for recharging at the departure station without increasing the time to reach the arrival station. The time warp between two parking stations i to j can be calculated as follows. First, we calculate the vehicle’s earliest arrival time at node j . Second, we keep the vehicle’s earliest arrival time at node j unchanged, calculate the vehicle’s latest starting time from node i . Third, we subtract the vehicle’s latest start time at node i to the earliest start time at node i to get the time warp. The pseudocode for calculating the time warp is given in Appendix D.

Based on this greedy recharging policy, we propose a four-step heuristic procedure to charge the vehicles. We introduce our heuristic with an example of a 2E-VREC route (Figure 2). In the four-step heuristic, the robot-battery may be recharged to low-level, high-level, or max-level. Low-level charging means the robot battery is charged to the level that makes sure the robot can independently finish its next trip, e.g. tour 4-5-4 in Figure 2. High-level charging means the robot battery is charged to the level that ensures the robot can independently finish the tours before its van leaves the parking station, e.g. tour 4-5-4 and 4-6-8 in Figure 2. Max-level charging means the robot battery is charged to its maximum level. The van battery may be recharged to low-level or max-level. Low-level charging of a van means the van battery is charged to the level that ensures that the van can reach the next parking station. Max-level charging of a van means the van battery is charged to its maximum level.

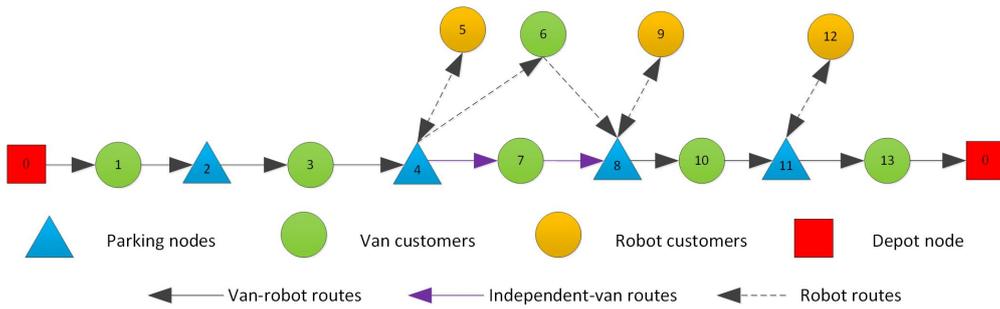


Figure 2: A 2E-VREC route example

Algorithm 1 is the four-step (priority) heuristic algorithm. Step 1 (Line 2-7), we ensure the current van battery level is sufficient for reaching the next parking station ($van \leftarrow Charge(low - level)$). For example, we must ensure the van battery is charged enough to travel from node 2 to node 4 in Figure 2, even if the charging time is longer than the calculated time warp. Note if the route is to violate the time window constraint due to charging, the route is not feasible. Step 2 (Line 8-26), we charge the current robot battery to high-level ($robot \leftarrow Charge(high -$

level)). For example, in station node 2, we charge the robot battery so it can finish route 4-5-4 and route 4-6-8 in Figure 2. If we used the whole calculated time warp to charge the robot, but the robot battery is still at less than its high-level, we conduct en-route charging on arc 2-3-4 subject to the van battery being charged enough to reach station 4. In station node 4, we must ensure that the robot can finish tour 4-5-4 even if the charging time is longer than the calculated time warp. Step 3 (Line 27-32), we charge the current battery level of the van to its max-level ($van \leftarrow Charge(max - level)$). Step 4 (Line 33-38), we charge the current battery level of the robot to its max-level ($robot \leftarrow Charge(max - level)$).

Algorithm 1 *Four – step heuristic algorithm*

```

1: Input: Given_Route, Time_warp
2: Step 1: Charge the van battery to low-level
3: if the van battery could be recharged to low-level without violate the time window then
4:    $van \leftarrow Charge(low - level)$ 
5: else
6:   output false and break
7: end if
8: Step 2: Charge the robot battery to high-level subject to Step 1
9: if the robot battery could be recharged to high-level in time warp then
10:   $robot \leftarrow Charge(high - level)$ 
11: else
12:  if parking station where the vehicles are now is at the node where this robot is to be dropped off then
13:    if the robot battery could be recharged to low-level in time warp then
14:       $robot \leftarrow Charge(time\_warp)$ 
15:    else
16:      if the robot battery could be recharged to low-level without violate the time window then
17:         $robot \leftarrow Charge(low - level)$ 
18:      else
19:        output false and break
20:      end if
21:    end if
22:  else
23:     $robot \leftarrow Charge(time\_warp)$ 
24:     $robot \leftarrow Enroute\_Charge(high - level)$  subject to the van battery at its low-level
25:  end if
26: end if
27: Step 3: Charge the van battery to max-level subject to Step 1,2
28: if the van battery could be recharged to max-level in time warp then
29:   $van \leftarrow Charge(max - level)$ 
30: else
31:   $van \leftarrow Charge(time\_warp)$ 
32: end if
33: Step 4: Charge the robot battery to max-level subject to Step 1,2,3
34: if the van battery could be recharged to max-level in time warp then
35:   $robot \leftarrow Charge(max - level)$ 
36: else
37:   $robot \leftarrow Charge(time\_warp)$ 
38: end if

```

After we have defined the priority charging strategy, we can further optimize the charging by appropriately setting charging time. The target for the further optimal charging approach is to minimize the recharging end time of vehicles. An example of further optimizing the charging is shown in Figure 3. First, we charge the van battery to its low-level in case (A). Second, we charge the robot-battery to its high-level in case (B). We can then optimize the charging by properly setting the charging time of the van and the robot, by bringing forward the charging time of the robot to the starting recharging time of the van, as in case (C). We note that if the robot charging time is longer than the charging time of the van, we charge the van battery until its charging time reaches the robot charging time or charge

the van-battery up to its max-level. Third, we charge the van battery to its max-level in case (D), and then charge the robot-battery to its max-level in the fourth step in (E). We can further optimize the charging by bringing forward the charging time of the robot to the recharging start time of the van again, as in case (F).

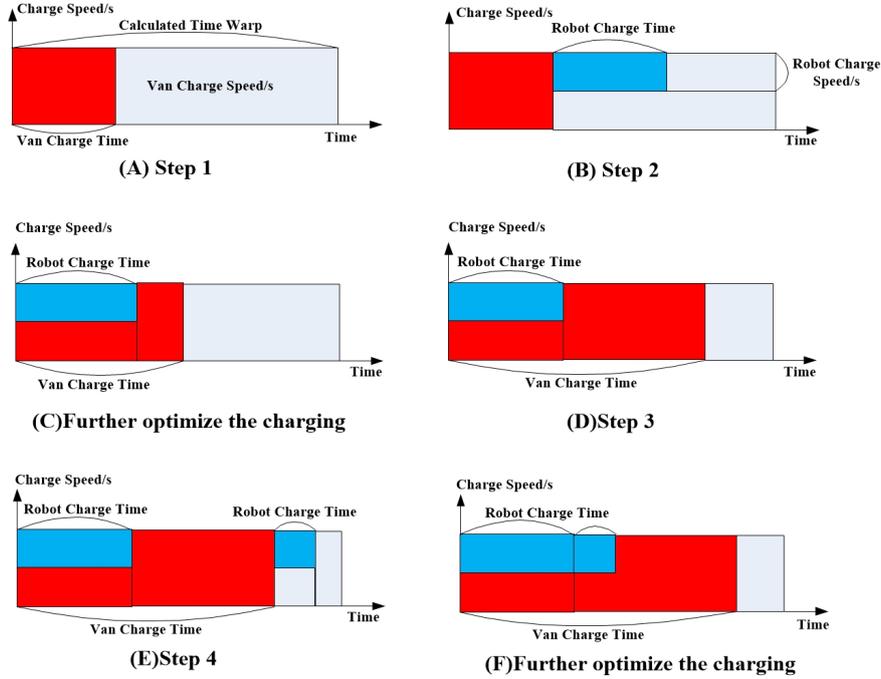


Figure 3: Example of further optimizing the charging

4.3.2. Linear programming model for route checking

We present an LP model for route evaluation. It is used to accurately check the feasibility of time, distance, and energy for a given single route. The input of the LP model is a 2E-VREC route, and the output is the feasibility of the given route. The formulation of the LP model is similar to constraints (15)-(33). The only difference is that the binary variables on right side are set to 1 (due to the fixed route). The details of the formulation of the LP model are given in Appendix E.

5. Numerical Analysis

We perform four types of numerical experiments. First, we use parameter tuning to determine the parameters. Second, we evaluate the performance of two proposed route evaluation approaches. Third, we conduct the ALNS experiments to see the overall performance of the proposed ALNS algorithm. Fourth, we perform a sensitivity analysis to determine the impact of related charging modes, charging rate, and maximum battery capacities. In the four types of numerical experiments, we run our algorithms ten times for every instance.

The mathematical programming model is coded in OPL. CPLEX 12.8 is used to solve the model. The ALNS is coded in python version 3.6.8. Both CPLEX and python are executed on an Intel(R) Core(TM) 2.9 GHz processor with 16 GB memory running under Windows 10. Python is run with single-threading.

5.1. Instance generation

We draw on Bakach et al. (2021) to set instances used in numerical analysis experiments. We use Manhattan distances between all the network nodes and set 20 tiny-scale instances, 20 small-scale instances, 20 medium-scale instances, and 20 large-scale instances, respectively.

For tiny-scale and small-scale instances, we consider instances with $10 \times 10 \text{ km}^2$ and $100 \times 100 \text{ m}^2$ blocks as well as 3 and 7 possible parking nodes, and 5 and 15 customer nodes, respectively. For medium-scale and large-scale instances, we consider instances with $20 \times 20 \text{ km}^2$ and $100 \times 100 \text{ m}^2$ blocks as well as 15 and 30 possible parking nodes, and 30 and 60 customer nodes, respectively. The customers are randomly distributed but are located at corner points of the blocks without duplicates. The service time of each customer is 0.1 hours, the width of the customer time window is 2 hours, and the start time window is an integer. All instances have the same depot and its time window is $[0,8]$. Each customer’s demand is randomly selected from $[10 \text{ kg}, 20 \text{ kg}, 30 \text{ kg}, 40 \text{ kg}, 50 \text{ kg}]$, and we set vans that could access two-thirds of the total customer nodes in each instance. We round up if the calculated number of van customer nodes was fractional. The first two-thirds of customers in the instance are the van customers.

We also refer to the literature of Ostermeier et al. (2021) and Bakach et al. (2021), to design parameters suitable for electric vans and electric robots, which are in Table 3.

Table 3: Equipment parameters

	Van	Robot
Average travel speed	25km/h	5km/h
Battery capacity	100kwh	1.5kwh
Capacity	1000kg	50kg
Travel cost rate	RMB 2/km	RMB 0.5/km
Energy consumption rate	0.5kwh/km	0.1kwh/km
Recharging rate	25kwh/h	2.5kwh/h

Section 5.2 and Section 5.3 use small-scale instances for parameters tuning, greed route evaluation and LP-based evaluation comparing experiments. Section 5.4 uses all types of instances for ALNS algorithm experiments. Section 5.5 uses medium-scale instances for 2E-VREC distribution systems performance analysis experiments.

The instances can be downloaded at https://www.researchgate.net/publication/361407582_2E-VREC_instances.

5.2. Parameter tuning

We draw on the parameter tuning concept (Jie et al. 2019) to determine the parameters. The parameter tuning approach starts with a set of initial parameter values. Parameter tuning conducts the ALNS algorithm with the parameter values in the search interval. The parameter value with the best solution is chosen to perform the following calculation. The process is repeated until all the parameters in the parameter set have been tuned. We select the first eight small-scale instances to conduct the parameter tuning procedure. The parameter notations, descriptions, and values used here can be found in Appendix F.

5.3. Comparison of greedy route evaluation and LP-based evaluation

Since the greedy route evaluation (GE) approach presented is an approximate method, we compare it with the LP-based evaluation approach to assess its accuracy and speed. We also conduct additional experiments to evaluate an approach combining the GE-based and LP-based evaluation (GE-LP-based evaluation) to exactly check the feasibility of a given route. The GE-LP-based evaluation first uses a GE-based evaluation to test the feasibility of the route. If the GE-based evaluation finds the route feasible, it is deemed feasible, and the evaluation procedure ends. Otherwise, the evaluation procedure takes the LP-based evaluation approach to finally check the feasibility of the route.

The instances used in this experiment are the same as those in the parameter turning experiments. Table 4 gives the results for the comparison of GE-based, LP-based, and GE-LP-based evaluations. Column 1 indicates the instances tested. Our num1 is the average number of times the output of the GE-based route evaluation differs from the output of the LP-based route evaluation approach. Our num2 is the average total number of times that the route evaluation procedure is performed. The LP time, GE time, and GE-LP time are the average evaluation times of LP-based, GE-based, and GE-LP-based evaluation approaches in 10 experiments, respectively. The runtime (LP time, GE time, GE-LP time) is the time for the ALNS to solve the instance using different evaluation approaches.

It is obvious from the results that the estimation error (num1/num2) of the GE-based evaluation approach is less than 10^{-3} . The GE-based evaluation runs 47 times faster than the LP-based evaluation method. The GE-LP-based evaluation also runs faster than the LP-based evaluation approach. However, the runtime reduction is only about

Table 4: Comparison of different route evaluation approaches

Instances	num1/num2	LP time(s)	GE time(s)	GE-LP time(s)
sc1	17/108279	1872.0	37.3	1631.9
sc2	8/90761	1561.0	36.9	1267.7
sc3	56/92968	1790.3	36.6	1795.5
sc4	348/117509	1718.4	41.6	1756.2
sc5	37/113092	1739.7	37.8	1584.2
sc6	35/137428	2437.4	47.4	2100.1
sc7	136/95829	2009.3	37.1	1753.2
sc8	32/112191	1864.6	46.0	1896.5
AVER	84/108507	1874.1	40.1	1723.2

8%. This may be because many unfeasible routes that need to be checked using the LP-based evaluation method are generated in the process of running the ALNS algorithm. In any case, if we need to check the feasibility of a route exactly, the GE-LP-based evaluation approach is faster than the evaluation method using only the LP-based approach.

5.4. ALNS experiment

First, we compare the GE-based ALNS results with the GE-LP-based ALNS results to analyze the GE-based ALNS algorithm’s performance. Second, we compare the ALNS results with the CPLEX results in tiny-scale instances. Third, we use the ALNS algorithm to solve larger instances and see the overall performance of the proposed ALNS algorithm.

5.4.1. Comparison of the GE-based with the GE-LP-based ALNS results

In Section 5.3, we analyze the estimation error of the GE-based evaluation approach, which is less than 10^{-3} . Here we run the GE-based ALNS algorithm and the GE-LP-based ALNS algorithm on small-scale instances to see whether an error of less than one thousandth greatly changes the results.

Table 5 gives a comparison of the GE-based ALNS and GE-LP-based ALNS results. AC and BC are the averages and best objective values. SDC is the standard deviation of the objective value, and AT(s) is the average runtime of the GE-based(GE-LP-based) ALNS algorithm.

Table 5 shows that the average performance difference between the two algorithms is not significant, and the average solving time of the GE-based ALNS is much shorter than that of the GE-LP-based ALNS algorithm. We therefore use the GE-based ALNS algorithm in the subsequent ALNS numerical experiments.

5.4.2. Comparison of the ALNS results with the CPLEX results

The van-based robot delivery and van-based drone delivery model is hard to solve for general-purpose MIP solvers. For example, Wang and Sheu (2019) show the commercial solver can only address the van-based drone delivery problem with 10 nodes (contains parking nodes and customer nodes), and even the branch and price algorithm they provided can only solve the problem with 12 points. In our problem, we replicate the parking nodes to allow for multiple visits. Dummy nodes increase the number of nodes in the model, making the problem more difficult to solve, and reducing the problem size we can address.

We test 20 tiny-scale instances with five customer nodes and three parking nodes. Theoretically, each parking node can be reached up to 5+1 times. However, using too many dummy parking nodes usually makes the problem difficult to solve. We therefore increase the number of dummy parking nodes iteratively and solve the resulting models using CPLEX: The increase in the number of dummy nodes stops when no improvement in the solution cost is found. However, if the solution is still worse than the ALNS solution, we continue to increase the number of dummy nodes until the CPLEX solution is better than or equal to the ALNS solution. The iterative process will also stop if the procedure runs out of solving time or out of memory.

For each instance, CPLEX 12.8 runs with default settings until it finds an optimal solution, exhausting the predetermined maximum computation time (7200s), or until the program runs out of memory. The computational results for all instances are presented in Table 6, in which K represents the number of times a vehicle can visit a parking node, and UB and LB are the upper and lower bounds of CPLEX solutions. E1 is the CPLEX gap between CPLEX upper

Table 5: GE-based and GE-LP-based ALNS algorithm comparison

Instances	GE-based ALNS				GE-LP-base ALNS			
	AC	BC	SDC	AT	AC	BC	SDC	AT
sc0	104.2	99.9	2.1	37.3	101.9	98.1	3.2	1631.9
sc1	76.5	74.6	1.6	36.9	75.8	74.6	1.2	1267.7
sc2	111.4	106.8	4.4	36.6	111.4	106.8	3.1	1795.5
sc3	100.1	95.8	2.4	41.6	100.2	95.8	1.6	1756.2
sc4	108.3	101.8	5.3	37.8	109.4	101.7	5.5	1584.2
sc5	89.3	85.7	4.5	47.4	87.7	82.2	2.5	2100.1
sc6	90.0	87.5	2.9	37.1	93.2	87.5	3.6	1753.2
sc7	96.4	88.5	3.8	46.0	93.4	88.5	4.0	1896.5
sc8	120.4	117.4	1.9	44.0	119.8	116.9	1.9	1798.6
sc9	106.1	102.5	2.5	45.2	107.7	104.0	2.0	1717.9
sc10	86.5	78.8	4.2	49.8	86.2	78.8	4.0	1447.2
sc11	117.3	114.3	2.5	43.0	116.6	114.3	2.0	1538.8
sc12	69.5	64.1	3.2	42.3	69.9	67.1	2.0	1867.3
sc13	103.4	101.0	2.2	44.7	103.7	102.4	2.0	1744.8
sc14	86.7	86.3	0.6	37.7	86.9	86.3	0.9	1664.3
sc15	114.7	114.4	0.4	42.2	113.6	111.4	1.5	2117.7
sc16	108.5	107.3	1.4	42.0	108.0	107.3	0.6	1882.2
sc17	84.1	82.6	0.8	46.5	85.1	83.4	1.7	1522.9
sc18	93.6	89.6	2.5	44.2	95.1	89.8	3.0	1712.9
sc19	98.8	97.3	3.0	44.2	98.9	97.3	2.4	1733.4
AVER	98.3	94.8	2.6	42.3	98.2	94.7	2.4	1726.7

bound (baseline) and CPLEX lower bound, and TIME is the CPLEX solving time. BC and AT are the best solutions for ALNS in 10 tests and average solving time of ALNS. SDC and VK are the standard deviation of the objective value and the dummy parking nodes used in the ALNS best solution. E2 is the gap between the best ALNS solution and the best ($K=2, 3, \text{ or } 4$) CPLEX upper bound (baseline).

Table 6 shows that our ALNS can always reach or improve the upper bounds of CPLEX solutions within a limited time (7200s) in tiny-scale instances.

5.4.3. ALNS to solve larger instances

We use medium-scale and large-scale instances to perform ALNS algorithm performance analysis. The ALNS experiment results of medium-scale and large-scale instances are shown in Table 7. AC and BC are the averages and best objective values. The SDC represents the standard deviation of the objective value, and the AT(s) is the average runtime of the ALNS.

The results show that the average solving time for the hardest instances is less than 800s. It also shows that our algorithm has the potential to solve larger problems quickly.

5.5. Performance analysis of 2E-VREC model

We analyze the impact of the en-route-charge, static-charge, and no-charge modes of vehicles. The en-route-charge mode is the mode we are studying here. The static-charge mode does not allow a van to charge its robot en-route, and the van can only recharge its robot at charging stations. The no-charge mode allows the van and robot to be recharged at the depot only. The charging rate of vehicles and the maximum battery capacity of vehicles may also affect the efficiency of the distribution system. Hence, we also analyze the impact of charging rate and maximum battery capacity on the results.

5.5.1. Sensitivity analysis on impact of charging modes

First, we perform a comparison experiment between different charging modes. We set the en-route-charge mode as baseline. We also try to set the width of the time window to $1/4$, $1/2$, 2 , and 4 times the original to see how the width of the time window affects the model's output. Table 8 reports the comparison of charge modes with different

Table 6: CPLEX and ALNS results comparison (tiny-scale instances)

Instances	$K=2$				$K=3$				$K=4$				ALNS					
	UB	LB	E1(%)	TIME	UB	LB	E1(%)	TIME	UB	LB	E1(%)	TIME	BC	AC	SDC	AT	VK	E2(%)
ts1	43.7	43.7	0.0	36.6	41.3	25.3	38.7	7200.0	/	/	/	/	32.7	32.7	0.0	7.8	4	-20.7
ts2	67.6	67.6	0.0	31.8	67.6	55.0	18.6	7200.0	/	/	/	/	67.6	67.6	0.0	7.4	2	0.0
ts3	48.1	48.1	0.0	47.3	46.5	20.3	56.3	7200.0	/	/	/	/	43.8	43.8	0.0	7.3	3	-5.8
ts4	67.9	67.9	0.0	41.8	69.6	26.7	61.7	7200.0	/	/	/	/	67.9	67.9	0.0	5.9	2	0.0
ts5	63.6	63.6	0.0	11.2	63.6	42.6	33.0	7200.0	/	/	/	/	59.4	59.4	0.0	7.5	5	-6.5
ts6	65.1	65.1	0.0	27.1	59.1	34.6	41.5	7200.0	/	/	/	/	51.7	51.7	0.0	8.4	4	-12.5
ts7	47.4	47.4	0.0	23.8	45.4	30.0	33.9	7200.0	/	/	/	/	39.5	39.5	0.0	6.8	4	-13.0
ts8	67.6	67.6	0.0	79.6	67.6	41.6	38.5	7200.0	/	/	/	/	67.6	67.6	0.0	8.0	2	0.0
ts9	42.3	42.3	0.0	11.9	42.3	17.7	58.2	7200.0	/	/	/	/	42.3	47.3	2.4	5.8	2	0.0
ts10	50.2	50.2	0.0	18.4	50.2	26.1	47.9	7200.0	/	/	/	/	38.8	38.8	0.0	16.3	3	-22.6
ts11	41.8	41.8	0.0	4.5	33.0	33.0	0.0	567.4	32.0	28.2	11.8	7200.0	32.0	32.0	0.0	7.6	4	0.0
ts12	72.6	72.6	0.0	76.1	72.6	38.1	47.5	7200.0	/	/	/	/	72.6	73.3	0.9	10.5	2	0.0
ts13	68.3	68.3	0.0	218.6	68.3	33.8	50.5	7200.0	/	/	/	/	68.3	69.8	1.7	7.0	2	0.0
ts14	51.0	51.0	0.0	26.3	50.5	30.1	40.4	7200.0	/	/	/	/	50.5	50.5	0.0	6.9	3	0.0
ts15	49.6	49.6	0.0	505.4	40.0	25.4	36.6	7200.0	/	/	/	/	40.0	40.0	0.0	10.5	3	0.0
ts16	48.5	48.5	0.0	282.5	43.7	26.7	39.0	7200.0	/	/	/	/	38.1	38.1	0.0	11.4	5	-12.8
ts17	63.1	63.1	0.0	25.8	69.1	39.9	42.2	7200.0	/	/	/	/	62.0	62.0	0.0	7.9	3	-1.7
ts18	51.5	51.5	0.0	16.4	46.1	32.6	29.3	7200.0	/	/	/	/	46.1	46.1	0.0	6.1	3	0.0
ts19	70.7	70.7	0.0	17.0	70.7	50.7	28.3	7200.0	/	/	/	/	70.7	70.7	0.0	6.3	2	0.0
ts20	52.3	52.3	0.0	5.4	52.3	34.7	33.6	7200.0	/	/	/	/	49.2	49.2	0.0	7.5	3	-5.8

Table 7: ALNS experiment results of medium-scale and large-scale instances

Medium-scale instances					Large-scale instances				
Instances	AC	BC	SDC	AT	Instances	AC	BC	SDC	AT
ms0	403.5	393.3	7.1	128.9	ls0	617.5	595.9	12.0	513.2
ms1	330.2	326.5	2.9	135.6	ls1	671.5	625.6	17.8	867.8
ms2	403.6	381.5	10.4	153.5	ls2	585.8	555.2	15.9	1140.7
ms3	367.1	361.9	3.8	363.6	ls3	649.4	608.7	17.8	859.0
ms4	376.9	361.8	10.2	188.9	ls4	557.1	534.7	14.8	710.1
ms5	400.5	389.3	7.0	152.2	ls5	564.0	497.9	29.0	858.6
ms6	356.0	349.1	5.7	238.8	ls6	574.4	543.6	19.5	432.0
ms7	405.4	382.5	13.2	139.4	ls7	595.4	566.8	20.6	856.6
ms8	362.7	356.6	3.9	152.5	ls8	634.5	601.7	14.7	600.7
ms9	431.0	424.6	5.1	253.4	ls9	602.4	570.0	17.9	1143.6
ms10	399.0	382.1	15.1	157.6	ls10	577.8	558.9	11.4	565.1
ms11	338.2	331.3	5.2	125.1	ls11	613.1	584.1	16.3	782.4
ms12	368.9	365.3	3.1	144.8	ls12	597.9	579.4	16.9	655.6
ms13	366.2	360.3	3.4	144.9	ls13	567.9	554.2	9.2	413.1
ms14	366.6	348.4	11.2	120.7	ls14	590.2	579.3	9.1	580.2
ms15	383.4	376.0	5.5	131.4	ls15	554.7	533.4	13.4	677.4
ms16	364.1	357.0	5.0	207.0	ls16	606.0	570.7	13.9	630.6
ms17	467.8	451.9	8.8	184.5	ls17	619.3	599.4	11.0	1097.7
ms18	368.9	360.9	4.0	243.5	ls18	563.0	544.8	13.4	959.7
ms19	337.6	325.3	8.2	174.2	ls19	627.4	605.8	15.9	751.8
AVER	379.9	369.3	6.9	177.0	AVER	598.5	570.5	15.5	754.8

time window widths. Column 1 represents different widths of the time window. Column 2 - Column 4 represent the average model output for the en-route charge, no-charge, and static-charge mode in different time window widths. Column 5 - Column 7 represent the average percentage gap between different charge modes in different time window widths. E3 represents the average percentage gap between the average cost of the no-charge mode and the en-route charge mode (baseline). E4 represents the average percentage gap between the average cost of the static-charge mode and the en-route charge mode (baseline). E5 represents the average percentage gap between the average cost of the no-charge mode and the static-charge mode (baseline).

Table 8: Comparison of charge modes with different time windows

Different time windows	En-route charge	No-charge	Static-charge	E3 (%)	E4 (%)	E5 (%)
1/4 times time window	453.5	466.8	458.8	2.9	1.1	1.7
1/2 times time window	418.9	440.8	422.6	5.2	0.9	4.3
1 times time window (baseline)	379.9	406.1	382.2	6.9	0.6	6.3
2 times time window	336.6	369.7	338.5	9.8	0.6	9.2
4 times time window	300.8	340.9	302.3	13.3	0.5	12.8

The experimental results in Table 8 show that en-route charge under the baseline time window width can effectively reduce the cost of the 2E-VREC model by about 6.9%, compared with the no-charge mode, and it can be reduced by up to 13.3% under 4 times time window width setting. The en-route charge mode can reduce the average cost by about 0.6% compared with the static-charge mode under the baseline time window width setting, and it can be reduced by up to 0.5%-1.1% under different time window width settings.

5.5.2. Sensitivity analysis on impact of charging rates

For the sensitivity analysis on the impact of charging rate experiments under en-route charge mode, we set the charging rate used in the en-route-charge mode as the baseline. We then set the charging rate at 1/4, 1/2, 1, 2 and 4 times the baseline. The comparison of the charging rate under the en-route charge mode setting is shown in Figure 4.

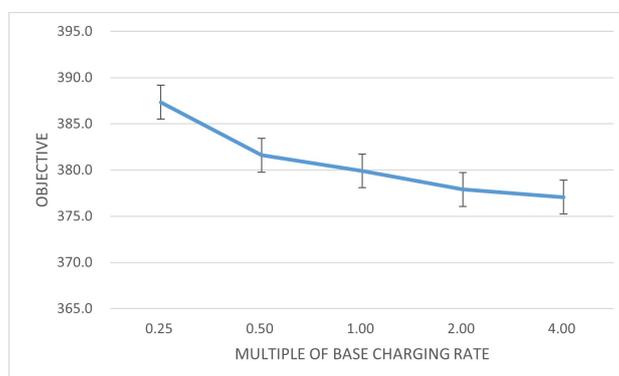


Figure 4: Comparison of charging rate in en-route charge mode setting

Figure 4 shows that increasing the charging rate can significantly reduce cost. However, the yield curve on increasing the charging rate tends to flatten as the charging rate increases.

Next, we conduct a sensitivity analysis on the impact of charging rates experiments under static-charge mode to see how the charging rate influences the comparison of en-route and static-charge modes.

Table 9 reports the comparison of en-route charge mode and static-charge mode at different charging rates. Row 1 represents the charging rates compared with the baseline. Row 2 shows the average cost in en-route charge mode, and row 3 shows the average cost in static-charge mode. Row 4 (E6(%)) shows the average percentage gap between the en-route charge mode (baseline) and static-charge mode at different charging rates.

Experimental results show that the en-route charge mode can reduce the average cost by about 0.1%-1.1% compared with the static-charge mode under different charging rates. The en-route charge mode is more advantageous when the charging rate is low.

Table 9: Comparison of en-route charge mode and static-charge mode in different charging rate

Charging rate	1/4	1/2	1 (baseline)	2	4
Aver en-route charge cost	387.3	381.6	379.9	377.9	377.1
Aver static-charge cost	391.5	384.5	382.2	378.7	377.3
E6(%)	1.1	0.8	0.6	0.2	0.1

From sensitivity analysis on the impact of charging modes and charging rates, we observe that en-route-charge mode can reduce cost by nearly 1.1% on average compared with using the static-charge mode in an optimistic situation. We recommend logistics companies to use en-route-charge mode if the fixed cost of the new technology is controllable since the reduction of 1.1% in cost is a valuable saving in a distribution system.

5.5.3. Sensitivity analysis on impact of battery capacities

For the sensitivity analysis on the impact of maximum battery capacity, we generate the different van/robot maximum capacity combinations by multiplying the van/robot’s maximum battery capacity by a different multiplication factor. Table 10 reports the comparison of maximum battery capacities. Row 1 shows the different maximum battery capacity combinations, in which θ_1/θ_2 is the multiplication factor. Row 2 shows the average cost for the instances under en-route charge modes. Row 3 shows the average percentage gap (E7(%)) for the instances under the en-route charge setting. The control group solution (baseline) and the experimental group solution are reported as an average percentage gap.

Table 10: Comparison of maximum battery capacities

θ_1/θ_2	1.0/1.0(baseline)	1.0/1.1	1.0/1.2	1.1/1.0	1.1/1.1	1.1/1.2	1.2/1.0	1.2/1.1	1.2/1.2
Average cost	379.9	375.2	370.5	379.5	374.9	370.1	379.2	374.9	369.9
E7(%)	0.0	-1.2	-2.5	-0.1	-1.3	-2.6	-0.2	-1.3	-2.6

Table 10 shows that increasing the maximum battery capacity of the robot 1.1 and 1.2 times can reduce the cost by 1.2% and 2.5%, respectively. Increasing the maximum battery capacity of the van 1.1 and 1.2 times produces only 0.1% and 0.2% increase in efficiency, respectively.

6. Conclusion and Future Research Directions

This paper notes that the time during which vans are carrying robots can be used effectively to recharge the robots, thereby increasing the efficiency of distribution systems. Here we present a novel transportation delivery model that incorporates en-route-charging for autonomous electric vehicle logistics in cities. The model extends the van-based robot delivery model to electric vehicle aspects and brings en-route-charging technologies into the electric van-based robot delivery model, which provides a new option for logistics operators.

In the 2E-VREC model, the vans carry small robots along the *1st-level route*, and the robot itself travels along the *2nd-level open route*. The van and robot can both serve customers directly, but some constrained customer sets can be visited only by robots. The van stops at parking nodes for dropping off and/or picking up its robot, and for recharging and replenishing its robot if needed. The van can charge its robot (if it is on board the van) during its trip. The van can be recharged at the parking nodes.

To model the proposed problem, we introduce a mixed integer program. The 2E-VREC model brings a new time-distance-energy trade-off to electric vehicle routing. The energy transfer is negatively correlated with the van’s available travel distance, but positively correlated with the robot’s available travel distance. Energy transfer also needs time.

An adaptive large neighborhood search algorithm is presented for solving larger instances. Owing to the difficulty processing the trade-off for checking the feasibility of a given 2E-VREC route, we further propose a greedy route evaluation approach and an linear programming route evaluation method. A comparison of GE-based evaluation and LP-based evaluation shows that the GE-based evaluation runs 47 times faster than the LP-based evaluation method, and the estimation error of the GE-based evaluation approach is less than 10^{-3} . Moreover, the average performance

difference between the GE-based ALNS and GE-LP-based ALNS algorithm is not significant, but the former runs much faster.

A sensitivity analysis for vehicle charging modes, charging rate, and maximum battery capacities reveals that using en-route-charging technologies, with appropriate increases in charging rate and maximum battery capacity, has useful effects on cost. In particular, the en-route charge mode can reduce the average cost by about 0.5%-1.1% compared with the static-charge mode under 1/4 - 4 times time-window width settings, and can reduce the average cost by about 0.1%-1.1% compared with the static-charge mode under 1/4 - 4 times charging rate settings. Increasing the maximum battery capacity of the robot 1.2 times can reduce the cost by up to 2.5%. We recommend that logistics companies use en-route-charge technology if the fixed cost of the new technology is controllable.

Future research could try to design a fast and accurate algorithm for checking the feasibility of a given route. The application environments or model settings in which the en-route charging can significantly improve distribution system efficiency could be explored. For example, if the 2E-VREC model's objective is to minimize travel duration, and there is no time window on customers, the en-route-charge mode may significantly reduce travel duration. Besides, if we allow a van to carry multiple robots in the 2E-VREC model, it may further improve delivery efficiency. From a more realistic perspective, vehicle power consumption often has a complex nonlinear relationship with multiple factors, so it is also promising to study the 2E-VREC problem under a nonlinear charging and discharging constraints. Moreover, distribution scenarios where customers do not want to be serviced by robots but by vans (humans) are also very forward research directions.

Acknowledgements The research is supported by the National Natural Science Foundation of China [grant no. 51975482] and Shaanxi Provincial Key R&D Program of China [grant no. 2019ZDLGY14-10]. This work also supported by the China Scholarship Council, by public funding within the scope of the French Program "Investissements d'Avenir" and by the European Union's Horizon 2020 research and innovation programme project LEAD under grant agreement No 861598.

References

- Agatz, N., Bouman, P., and Schmidt, M. (2018). Optimization approaches for the traveling salesman problem with drone. *Transportation Science*, 52(4):965–981.
- Alfandari, L., Ljubić, I., and da Silva, M. D. M. (2022). A tailored benders decomposition approach for last-mile delivery with autonomous robots. *European Journal of Operational Research*, 299(2):510–525.
- Andrew, J. H. (2019). Thousands of autonomous delivery robots are about to descend on us college campuses. <https://www.theverge.com/2019/8/20/20812184/starship-delivery-robot-expansion-college-campus>. Accessed August 20, 2019.
- Bakach, I., Campbell, A. M., and Ehmke, J. F. (2021). A two-tier urban delivery network with robot-based deliveries. *Networks*, 78(4):461–483.
- Bektaş, T., Ehmke, J. F., Psaraftis, H. N., and Puchinger, J. (2019). The role of operational research in green freight transportation. *European Journal of Operational Research*, 274(3):807–823.
- Bouman, P., Agatz, N., and Schmidt, M. (2018). Dynamic programming approaches for the traveling salesman problem with drone. *Networks*, 72(4):528–542.
- Boysen, N., Schwerdfeger, S., and Weidinger, F. (2018). Scheduling last-mile deliveries with truck-based autonomous robots. *European Journal of Operational Research*, 271(3):1085–1099.
- Breunig, U., Baldacci, R., Hartl, R. F., and Vidal, T. (2019). The electric two-echelon vehicle routing problem. *Computers & Operations Research*, 103:198–210.
- Carlsson, J. G. and Song, S. (2018). Coordinated logistics with a truck and a drone. *Management Science*, 64(9):4052–4069.
- Chen, C., Demir, E., and Huang, Y. (2021). An adaptive large neighborhood search heuristic for the vehicle routing problem with time windows and delivery robots. *European Journal of Operational Research*, 3(294):1164–1180.
- Cui, S., Zhao, H., and Zhang, C. (2018). Multiple types of plug-in charging facilities' location-routing problem with time windows for mobile charging vehicles. *Sustainability*, 10(8):2855.
- Daimler (2018). Vans and robots paketbote 2.0. <https://www.daimler.com/innovation/specials/future-transportation-vans/paketbote-2-0.html>. Accessed Dec 25, 2018.

- Dayarian, I., Savelsbergh, M. W. P., and Clarke, J.-P. (2020). Same-day delivery with drone resupply. *Transportation Science*, 54:229–249.
- Forum, W. E. (2020). The future of the last-mile ecosystem. <https://www.weforum.org/reports/the-future-of-the-last-mile-ecosystem/>. Accessed Jun 7, 2022.
- Fred, L. (2018). Nio is courting tesla owners with mobile charging stations inside electric vans. <https://electrek.co/2018/07/26/nio-courting-tesla-owners-mobile-charging-stations-electric-vans/>. Accessed Dec 25, 2018.
- Gonzalez-R, P. L., Canca, D., Andrade-Pineda, J. L., Calle, M., and Leon-Blanco, J. M. (2020). Truck-drone team logistics: A heuristic approach to multi-drop route planning. *Transportation Research Part C-emerging Technologies*, 114:657–680.
- Gu, L. (2018). Delivery robots hit the road in beijing. <http://www.ecns.cn/news/sci-tech/2018-06-20/detail-ifyvmiee7350792.shtml>. Accessed June 20, 2018.
- Heimfarth, A., Ostermeier, M., and Hübner, A. (2022). A mixed truck and robot delivery approach for the daily supply of customers. *European Journal of Operational Research*.
- Hemmelmayr, V. C., Cordeau, J.-F., and Crainic, T. G. (2012). An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics. *Computers & operations research*, 39(12):3215–3228.
- Hiermann, G., Hartl, R. F., Puchinger, J., and Vidal, T. (2019). Routing a mix of conventional, plug-in hybrid, and electric vehicles. *European Journal of Operational Research*, 272(1):235–248.
- Hiermann, G., Puchinger, J., Ropke, S., and Hartl, R. F. (2016). The electric fleet size and mix vehicle routing problem with time windows and recharging stations. *European Journal of Operational Research*, 252(3):995–1018.
- Huang, S., He, L., Gu, Y., Wood, K., and Benjaafar, S. (2014). Design of a mobile charging service for electric vehicles in an urban environment. *IEEE Transactions on Intelligent Transportation Systems*, 16(2):787–798.
- Hunsaker, B. and Savelsbergh, M. (2002). Efficient feasibility testing for dial-a-ride problems. *Operations Research Letters*, 30(3):169–173.
- JD.COM (2022). Company website: Delivery robots. https://x.jdwl.com/unmannedCar/distributeRobot?locale=en_US. Accessed March 31, 2022.
- Jie, W., Yang, J., Zhang, M., and Huang, Y. (2019). The two-echelon capacitated electric vehicle routing problem with battery swapping stations: Formulation and efficient methodology. *European Journal of Operational Research*, 272(3):879–904.
- Kang, M. and Lee, C. (2021). An exact algorithm for heterogeneous drone-truck routing problem. *Transportation Science*.
- Karak, A. and Abdelghany, K. (2019). The hybrid vehicle-drone routing problem for pick-up and delivery services. *Transportation Research Part C: Emerging Technologies*, 102:427–449.
- Keskin, M. and Çatay, B. (2016). Partial recharge strategies for the electric vehicle routing problem with time windows. *Transportation Research Part C: Emerging Technologies*, 65:111–127.
- Kitjacharoenchai, P., Min, B.-C., and Lee, S. (2020). Two echelon vehicle routing problem with drones in last mile delivery. *International Journal of Production Economics*, 225:107598.
- Kuhn, H. and Sternbeck, M. G. (2013). Integrative retail logistics: An exploratory study. *Operations Management Research*, 6(1):2–18.
- Li, H., Chen, J., Wang, F., and Bai, M. (2021). Ground-vehicle and unmanned-aerial-vehicle routing problems from two-echelon scheme perspective: A review. *European Journal of Operational Research*, 294(3):1078–1095.
- Liimatainen, H., van Vliet, O., and Aplyn, D. (2019). The potential of electric trucks—an international commodity-level analysis. *Applied energy*, 236:804–814.
- Luo, Z., Liu, Z., and Shi, J. (2017). A two-echelon cooperated routing problem for a ground vehicle and its carried unmanned aerial vehicle. *Sensors*, 17(5):1144.
- Luo, Z., Poon, M., Zhang, Z., Liu, Z., and Lim, A. (2021). The multi-visit traveling salesman problem with multi-drones. *Transportation Research Part C: Emerging Technologies*, 128:103172.
- Marble (2019). Company website. <https://www.marble.io/>. Accessed Sept 28, 2019.
- Masson, R., Trentini, A., Lehuédé, F., Malhéné, N., Péton, O., and Tlahig, H. (2017). Optimization of a city logistics transportation system with mixed passengers and goods. *Euro Journal on Transportation & Logistics*, 6(1):81–109.
- Mathew, N., Smith, S. L., and Waslander, S. L. (2015). Multirobot rendezvous planning for recharging in persistent tasks. *IEEE Transactions on Robotics*, 31(1):128–142.
- Moshref-Javadi, M., Hemmati, A., and Winkenbach, M. (2020a). A truck and drones model for last-mile delivery: A mathematical model and heuristic approach. *Applied Mathematical Modelling*, 80:290–318.

- Moshref-Javadi, M., Hemmati, A., and Winkenbach, M. (2021). A comparative analysis of synchronized truck-and-drone delivery models. *Computers & Industrial Engineering*, 162:107648.
- Moshref-Javadi, M., Lee, S., and Winkenbach, M. (2020b). Design and evaluation of a multi-trip delivery model with truck and drones. *Transportation Research Part E-logistics and Transportation Review*, 136:101887.
- Moshref-Javadi, M. and Winkenbach, M. (2021). Applications and research avenues for drone-based models in logistics: A classification and review. *Expert Systems with Applications*, 177:114854.
- Mühlbauer, F. and Fontaine, P. (2021). 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*, 289(2):742–757.
- Mulholland, E., Teter, J., Cazzola, P., McDonald, Z., and Gallachóir, B. P. Ó. (2018). The long haul towards decarbonising road freight—a global assessment to 2050. *Applied energy*, 216:678–693.
- Murray, C. C. and Chu, A. G. (2015). The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C: Emerging Technologies*, 54:86–109.
- Murray, C. C. and Raj, R. (2020). The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones. *Transportation Research Part C: Emerging Technologies*, 110:368–398.
- Ostermeier, M., Heimfarth, A., and Hübner, A. (2021). Cost-optimal truck-and-robot routing for last-mile delivery. *Networks*.
- Otto, A., Agatz, N., Campbell, J., Golden, B., and Pesch, E. (2018). Optimization approaches for civil applications of unmanned aerial vehicles (uavs) or aerial drones: A survey. *Networks*, 72(4):411–458.
- Pisinger, D. and Ropke, S. (2019). Large neighborhood search. In *Handbook of Metaheuristics*, pages 99–127. Springer.
- Poikonen, S. and Golden, B. (2020a). The mothership and drone routing problem. *INFORMS Journal on Computing*, 32:249–262.
- Poikonen, S. and Golden, B. (2020b). Multi-visit drone routing problem. *Computers & Operations Research*, 113:104802.
- Poikonen, S., Golden, B., and Wasil, E. A. (2019). A branch-and-bound approach to the traveling salesman problem with a drone. *INFORMS Journal on Computing*, 31(2):335–346.
- Poikonen, S., Wang, X., and Golden, B. (2017). The vehicle routing problem with drones: Extended models and connections. *Networks*, 70(1):34–43.
- Pugliese, L. D. P. and Guerriero, F. (2017). Last-mile deliveries by using drones and classical vehicles. In *International Conference on Optimization and Decision Science*, pages 557–565. Springer.
- Roberti, R. and Ruthmair, M. (2021). Exact methods for the traveling salesman problem with drone. *Transportation Science*, 55(2):315–335.
- Sacramento, D., Pisinger, D., and Ropke, S. (2019). An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones. *Transportation Research Part C: Emerging Technologies*, 102:289–315.
- Salama, M. and Srinivas, S. (2020). Joint optimization of customer location clustering and drone-based routing for last-mile deliveries. *Transportation Research Part C-emerging Technologies*, 114:620–642.
- Savelsbergh and Martin, W. P. (1992). The vehicle routing problem with time windows: Minimizing route duration. *Informatics Journal on Computing*, 4(2):146–154.
- Schermer, D., Moeini, M., and Wendt, O. (2019a). A hybrid vns/tabu search algorithm for solving the vehicle routing problem with drones and en route operations. *Computers & Operations Research*, 109:134–158.
- Schermer, D., Moeini, M., and Wendt, O. (2019b). A matheuristic for the vehicle routing problem with drones and its variants. *Transportation Research Part C: Emerging Technologies*, 106:166–204.
- Simoni, M. D., Kutanoglu, E., and Claudel, C. G. (2020). Optimization and analysis of a robot-assisted last mile delivery system. *Transportation Research Part E: Logistics and Transportation Review*, 142:102049.
- Sluijk, N., Florio, A. M., Kinable, J., Dellaert, N., and Van Woensel, T. (2022). Two-echelon vehicle routing problems: A literature review. *European Journal of Operational Research*.
- Starship (2017). Factsheet starship delivery robot. <https://www.post.ch/-/media/post/ueber-uns/medienmitteilungen/2017/factsheet-lieferroboter.pdf?la=en>. Accessed March 31, 2022.
- Tamke, F. and Buscher, U. (2021). A branch-and-cut algorithm for the vehicle routing problem with drones. *Transportation Research Part B: Methodological*, 144:174–203.
- Technologies, S. (2016). Robovan - future proof local delivery. <https://www.youtube.com/watch?v=lzww1UsxYdk>. Accessed Sep 7, 2016.
- Wang, D., Zhou, H., and Feng, R. (2019). A two-echelon vehicle routing problem involving electric vehicles with time windows. In *Journal of Physics: Conference Series*, volume 1324, page 012071. IOP Publishing.

- Wang, X., Poikonen, S., and Golden, B. (2017). The vehicle routing problem with drones: several worst-case results. *Optimization Letters*, 11(4):679–697.
- Wang, Z. and Sheu, J.-B. (2019). Vehicle routing problem with drones. *Transportation research part B: methodological*, 122:350–364.
- Wu, G., Fan, M., Shi, J., and Feng, Y. (2021). Reinforcement learning based truck-and-drone coordinated delivery. *IEEE Transactions on Artificial Intelligence*.
- Yu, K., Budhiraja, A. K., Buebel, S., and Tokekar, P. (2019). Algorithms and experiments on routing of unmanned aerial vehicles with mobile recharging stations. *Journal of Field Robotics*, 36(3):602–616.
- Yu, S., Puchinger, J., and Sun, S. (2020). Two-echelon urban deliveries using autonomous vehicles. *Transportation Research Part E: Logistics and Transportation Review*, 141(doi.org/10.1016/j.tre.2020.102018).
- Yu, S., Puchinger, J., and Sun, S. (2022). Van-based robot hybrid pickup and delivery routing problem. *European Journal of Operational Research*, 298(3):894–914.

Appendix A. A simple example for 2E-VREC model

We show a simple example together with a short description to describe the proposed problem. Figure A.5 shows a simple example of the 2E-VREC model. Triangles represent parking nodes, the square represents the depot, and circles correspond to robot customer nodes/van customer nodes. Solid lines correspond to *1st-level routes* and dotted lines to *2nd-level routes*. Distances are displayed on each arc, and time windows are displayed around the nodes.

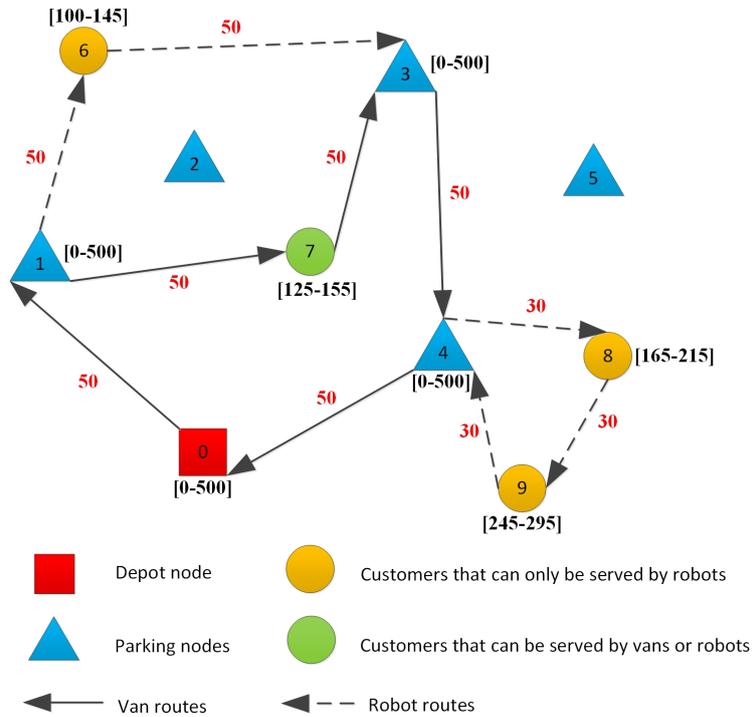


Figure A.5: A simple example of the 2E-VREC model

The demand and serving time for each customer node is 10, respectively. Table A.11 lists the parameters of vans and robots. Table A.12 gives the route, arrival time, leaving time, energy on arrival, energy when leaving, charging time, and the en-route charging time of the van and robot for the simple example in Figure A.5. In this feasible solution example, the van route is 0-1-7-3-4-0, whole-robot route is 0-1-6-3-4-8-9-4-0, and robot routes is 1-6-3 and 4-8-9-4. Note that the van recharges its robot en route 3-4 for 17.5 units of time.

	Van	Robot
Average travel speed	2	1
Battery capacity	120	400
Capacity	50	200
Travel cost rate	2	1
Energy consumption rate	2	1
Recharging rate	10	4

Table A.12: Feasible solution of the 2E-VREC simple example

	Van route						Whole-robot route								
Node	0	1	7	3	4	0	0	1	6	3	4	8	9	4	0
Arrival time	/	25	60	160	185	320	/	25	75	160	185	215	255	295	320
Leaving time	0	35	135	160	295	/	0	25	110	160	185	225	265	295	/
Energy on arrival	/	300	300	200	30	300	/	120	70	20	90	60	30	0	0
Energy when leaving	400	400	300	200	400	/	120	120	70	20	90	60	30	0	/
Charging time	0	10	0	0	37	0	/	/	/	/	/	/	/	/	/
En-route charging time	/	/	/	17.5	/	/	/	/	17.5	/	/	/	/	/	/

Appendix B. Sets, variables and parameters used in mathematical model

The sets, variables and parameters of our model are summarized in Table B.13.

Table B.13: Sets, variables and parameters

\mathcal{V}_0	Depot
\mathcal{V}_s	Set of parking nodes
\mathcal{V}_c	Set of customers
\mathcal{V}_{c1}	Set of van customers, $\mathcal{V}_{c1} \in \mathcal{V}_c$
\mathcal{V}_{sc1}	Set of parking nodes and van customer nodes, $\mathcal{V}_{sc1} = \mathcal{V}_s \cup \mathcal{V}_{c1}$
\mathcal{A}_1	1st-level route (van routes), $\mathcal{A}_1 = \{(i, j) \mid i \in \{0\}; j \in \mathcal{V}_{sc1}\} \cup \{(i, j) \mid i, j \in \mathcal{V}_{sc1}, i \neq j\} \cup \{(i, j) \mid i \in \mathcal{V}_{sc1}; j \in \{0\}\}$
\mathcal{A}_2	2nd-level route (robot routes), $\mathcal{A}_2 = \{(i, j) \mid i \in \mathcal{V}_s; j \in \mathcal{V}_c\} \cup \{(i, j) \mid i, j \in \mathcal{V}_c, i \neq j\} \cup \{(i, j) \mid i \in \mathcal{V}_c; j \in \mathcal{V}_s\}$
\mathcal{A}_3	Complete possible robot routes, $\mathcal{A}_3 = \mathcal{A}_1 \cup \mathcal{A}_2$
\mathcal{A}_4	Routes that the van cannot reach, $\mathcal{A}_4 = \mathcal{A}_3 \setminus \mathcal{A}_1$
\mathcal{K}	The set of vans, $\mathcal{K} = \{1, 2, \dots, k, \dots, K\}$, where K is the number of vans, and $k \in \mathcal{K}$ represents the k th van. Because vans and robots are in one-to-one correspondence, it also corresponds to the set of robots
$d_{i,j}$	Distance between vertices i and j
M	An arbitrary large constant number
g_1	Recharging rate of the van
g_2	Recharging rate of the robot
h_1	Charge consumption rate of the van
h_2	Charge consumption rate of the robot
c_1	Travel cost rate of the van
c_2	Travel cost rate of the robot
Q_1	Battery capacity of the van
Q_2	Battery capacity of the robot
C_1	Capacity of the van
C_2	Capacity of the robot
v_1	Travel speed of the van
v_2	Travel speed of the robot
q_i	Demand of vertex i
a_i	Earliest start of service at vertex i
b_i	Latest start of service at vertex i
s_i	Service time at vertex i
$x_{i,j,k}$	Decision variable equals to 1 if arc (i, j) in \mathcal{A}_1 is traveled by the k th – van, 0 otherwise
$y_{i,j,k}$	Decision variable equals to 1 if arc (i, j) in \mathcal{A}_3 is traveled by the k th – robot, 0 otherwise
$z_{i,j,k}$	Decision variable equals to 1 if arc (i, j) in \mathcal{A}_1 is traveled by the k th – van with its robot on board, 0 otherwise
W_i^k	Arrival time for the van at node i
w_i^k	Arrival time for the robot at node i
E_i^k	Remaining battery amount of the van at node i on arrival
e_i^k	Remaining battery amount of the robot at node i on arrival
O_i^k	Recharging amount of the van at the parking node i , $i \in \mathcal{V}_s$
o_i^k	Recharging amount of the robot at the parking node i , $i \in \mathcal{V}_s$
$o_{i,j}^k$	Recharging amount of the robot by the van during arc (i, j) , $(i, j) \in \mathcal{A}_1$
$f_{i,j,k}$	Freight flow of the robot in arc (i, j) in \mathcal{A}_2
u_i^k	Count variable in node \mathcal{V}_s for vehicle k

Appendix C. ALNS algorithm

After the initial solution has been constructed, the proposed ALNS algorithm seeks to improve it iteratively until a stopping condition is satisfied. We adopt a maximum iteration number ($MaxIteNum$) limit to terminate the algorithm. At each iteration, the existing feasible solution is destroyed by removing some nodes or routes by a destroy algorithm. The resulting partial solution is then repaired using a repair algorithm which heuristically repairs the existing route or constructs a new route if needed, for the purpose of obtaining a better solution than the previous one. We also set a number of non-improving iterations ($MaxNonImp$) before restarting the ALNS algorithm from a new initial solution as in Breunig et al. (2019).

During repair operations, the route evaluation method is used to check the feasibility of the routes. We adopt simulated annealing as acceptance criterion, which allows the algorithm to accept poor solutions during iterations. In addition, each of our destroy-and-repair operations is assigned a weight that controls how often the operation is selected. The weights are adjusted dynamically as the search progresses by a weight-adjusting method. The algorithm thus adapts to the instance at hand and to the state of the search. We draw on simulated annealing and weight-adjusting proposed by Pisinger and Ropke (2019).

Our proposed ALNS algorithm is described in Algorithm 2. The best solution is recorded in *recordset* (Line 1). The algorithm starts with an initial solution x and it is also the current best solution x^b (Line 2). The value of *NumNunImp* is then initialized to 0 (Line 3), where *NumNunImp* records the number of iterations until *NumNunImp* becomes larger than *MaxNunImp*. Next, a destroy-and-repair operation is performed until the stopping criterion (*MaxIteNum*) is met (Line 4-19). The *restart()* function is used to restart the ALNS from Line 2 but keeps the iteration number unchanged.

Algorithm 2 *Adapt_Large_Neighborhood_Search()*

```
1: recordset = {}
2: Initialization: use RouteReconstruction operation to get an initial solution  $x$ , let  $x^b = x$ .
3: NumNunImp = 0
4: while stopping criterion is not met do
5:   select destroy and repair methods from set.
6:    $x^t = r(d(x))$ 
7:   if simulated annealing criterion accepts the solution then
8:      $x = x^t$ 
9:   end if
10:  if  $c(x) < c(x^b)$  then
11:     $x^b = x$ , NumNunImp = 0
12:  else
13:    NumNunImp+ = 1
14:    if restart conditions is met then
15:      recordset.append( $x^b$ ) and then restart()
16:    end if
17:  end if
18:  update weights and selection parameters
19: end while
20: return: output best  $x^b$  from recordset
```

Appendix D. Time warp calculating algorithm

Algorithm 3 is used to calculate the time warp. The input of the algorithm (Line 1) contains the *Path* from station *n* to station *m*. *DepartTime* is the time a vehicle leaves the parking station. *Distance* is the distance matrix. *StartTimeWindow*, *EndTimeWindow* and *ServiceTime* are the time window list and service time list for nodes. The algorithm starts with time *DepartTime* (Line 2). For the sequence from node *n* to node *m*, we calculate the earliest arrival time at the node *m* (Line 3-11). Next, we reverse the *Path*, and we then calculate the latest starting time from node *n* subject to the earliest arrival time at the node *m* not changing (Line 13-20). The time warp is equal to the latest starting time at node *n* minus the earliest start time at node *n* (Line 21).

Algorithm 3 *Time_warp_calculating*

```
1: Input: Path, DepartTime, Distance, StartTimeWindow, EndTimeWindow, ServiceTime
2:  $T = \text{DepartTime}$ 
3: for i in range(len(Path)-1) do
4:    $T = T + \text{Distance}[\text{Path}[i]][\text{Path}[i+1]]/v$ 
5:    $T = \max(\text{StartTimeWindow}[\text{Path}[i+1]], T)$ 
6:   if  $T > \text{EndTimeWindow}[\text{Path}[i+1]]$  then
7:     return: (False)
8:   else
9:      $T = T + \text{ServiceTime}[\text{Path}[i+1]]$ 
10:  end if
11: end for
12: reverse(Path)
13: for i in range(len(Path)-1) do
14:    $T = T - \text{Distance}[\text{Path}[i]][\text{Path}[i+1]]/v - \text{ServiceTime}[\text{Path}[i+1]]$ 
15:    $T = \max(\text{StartTimeWindow}[\text{Path}[i+1]], T)$ 
16:    $T = \min(T, \text{EndTimeWindow}[\text{Path}[i+1]])$ 
17:   if  $\text{StartTimeWindow}[\text{Path}[i+1]] > T$  then
18:     return: (False)
19:   end if
20: end for
21: Time Warp =  $T - \text{DepartTime}$ 
22: return: Time Warp
```

Appendix E. Formulation of LP model

The LP model is used to check the feasibility of a given 2E-VREC route.

Explanations of symbol definitions are the same as in Section 3.3 for our 2E-VREC model, except without symbol k . This means we consider a van, not multi-vans. For the sake of simplicity, no further explanation is given. We add the following new symbols: \mathcal{R}_1 : van route; \mathcal{R}_2 : robot route; \mathcal{R}_3 : van-robot route and independent-van route and robot route; \mathcal{R}_4 : independent-van route; \mathcal{R}_5 : van-robot route. The LP model is as follows.

Objective:

Checking the feasibility of a given route

Subject to:

$$\begin{aligned}
 W_i + O_i/g_1 + d_{i,j}/v_1 - W_j &\leq 0, & \forall (i \in \mathcal{V}_s^0 | (i, j) \in \mathcal{R}_1), & \quad (\text{E.1}) \\
 W_i + o_i/g_2 + d_{i,j}/v_1 - W_j &\leq 0, & \forall (i \in \mathcal{V}_s^0 | (i, j) \in \mathcal{R}_1), & \quad (\text{E.2}) \\
 w_i + o_i/g_2 + d_{i,j}/v_1 - W_j &\leq 0, & \forall (i \in \mathcal{V}_s^0 | (i, j) \in \mathcal{R}_1), & \quad (\text{E.3}) \\
 W_i + o_i/g_2 + d_{i,j}/v_2 - w_j &\leq 0, & \forall (i \in \mathcal{V}_s | (i, j) \in \mathcal{R}_2), & \quad (\text{E.4}) \\
 w_i + o_i/g_2 + d_{i,j}/v_2 - w_j &\leq 0, & \forall (i \in \mathcal{V}_s | (i, j) \in \mathcal{R}_2), & \quad (\text{E.5}) \\
 W_i + s_i + d_{i,j}/v_1 - W_j &\leq 0, & \forall (i \in \mathcal{V}_{c1} | (i, j) \in \mathcal{R}_1), & \quad (\text{E.6}) \\
 W_i + o_i/g_2 + d_{i,j}/v_1 - W_j &\leq 0, & \forall (i \in \mathcal{V}_{c1} | (i, j) \in \mathcal{R}_1), & \quad (\text{E.7}) \\
 w_i + s_i + d_{i,j}/v_2 - w_j &\leq 0, & \forall (i \in \mathcal{V}_c | (i, j) \in \mathcal{R}_2), & \quad (\text{E.8}) \\
 W_j = w_j, & & \forall (i, j) \in \mathcal{R}_5 | j \in \mathcal{V}'_{sc1}, & \quad (\text{E.9}) \\
 a_i \leq W_i, w_i \leq b_i, & & \forall i \in \mathcal{V}_c'', & \quad (\text{E.10}) \\
 E_j + h_1 d_{i,j} - O_i + o_i + o_{i,j} - E_i &\leq 0, & \forall (i, j) \in \mathcal{R}_5 | i \in \mathcal{V}_s^0, & \quad (\text{E.11}) \\
 E_j + h_1 d_{i,j} - O_i + o_i - E_i &\leq 0, & \forall (i, j) \in \mathcal{R}_4 | i \in \mathcal{V}_s^0, & \quad (\text{E.12}) \\
 E_j + h_1 d_{i,j} + o_i + o_{i,j} - E_i &\leq 0, & \forall (i, j) \in \mathcal{R}_5 | i \in \mathcal{V}_{c1}, & \quad (\text{E.13}) \\
 E_j + h_1 d_{i,j} - E_i &\leq 0, & \forall (i, j) \in \mathcal{R}_4 | i \in \mathcal{V}_{c1}, & \quad (\text{E.14}) \\
 e_j - o_i - o_{i,j} - e_i &\leq 0, & \forall (i, j) \in \mathcal{R}_5 | i \in \mathcal{V}'_{sc1}, & \quad (\text{E.15}) \\
 e_j + h_2 d_{i,j} - o_i - e_i &\leq 0, & \forall (i, j) \in \mathcal{R}_2 | i \in \mathcal{V}_s, & \quad (\text{E.16}) \\
 e_j + h_2 d_{i,j} - e_i &\leq 0, & \forall (i, j) \in \mathcal{R}_2 | i \in \mathcal{V}_c, & \quad (\text{E.17}) \\
 0 \leq o_i \leq Q_2, & & \forall i \in \mathcal{V}'_{sc1}, & \quad (\text{E.18}) \\
 0 \leq O_i \leq (Q_1 + Q_2), & & \forall i \in \mathcal{V}_s^0, & \quad (\text{E.19}) \\
 0 \leq o_{i,j} \leq Q_2, & & \forall (i, j) \in \mathcal{R}_1, & \quad (\text{E.20}) \\
 0 \leq E_i + O_i - o_i \leq Q_1, & & \forall i \in \mathcal{V}_s^0, & \quad (\text{E.21}) \\
 0 \leq e_i + o_i \leq Q_2, & & \forall i \in \mathcal{V}_s^0, & \quad (\text{E.22}) \\
 0 \leq E_i \leq Q_1, & & \forall i \in \mathcal{V}'_{sc1}, & \quad (\text{E.23}) \\
 0 \leq e_i \leq Q_2, & & \forall i \in \mathcal{V}''_{sc}, & \quad (\text{E.24})
 \end{aligned}$$

Constraints (E.1)-(E.10) are the time flow constraints for the given routes. Constraints (E.1)-(E.3) and constraints (E.6)-(E.7) enforce the time flow of the van for \mathcal{R}_1 . Constraints (E.4)-(E.5) and constraints (E.8) enforce the time flow of the robot for \mathcal{R}_2 . Constraints (E.9) ensure that when a van carries its robot, their arrival time for each node is equal. Constraints (E.10) are the time window constraints for each node.

Constraints (E.11)-(E.24) are the energy constraints for the given routes. Constraints (E.11) and constraints (E.13) are the energy flow of the van for the route along which the van carries its robot, and constraints (E.15) are the energy flow of the robot for the routes when the robot is carried in its van. Constraints (E.12) and constraints (E.14) are the energy flow of the van for the independent-van routes. Constraints (E.16) and constraints (E.17) are the energy flow of the robot for the robot routes. Constraints (E.18)-(E.24) are the energy variable constraints.

Appendix F. Parameter notations, descriptions, and values

Table F.14 lists and describes the parameters investigated in our experiment. Column 1 gives the parameter name, and column 2 its corresponding notation and description. Table F.15 lists the parameter notations, the initial value of the parameters, the allowed range for each parameter, and the final values found by the parameter tuning process. Final values are used for the rest of the experiments.

Table F.14: Parameter notation and description

Parameter	Notation and description
MaxIteNum	Maximum iteration number of ALNS
MaxNunImp	Maximum iteration number with no improved solution
β	Destruction rate
λ	Decay parameter that controls how sensitive the weights are to changes
T_{st}	Start temperature per customer of SA
α	Cooling rate of SA
w_1	If the new solution is a new global best solution
w_2	If the new solution is better than the current one
w_3	If the new solution is accepted
w_4	If the new solution is rejected

Table F.15: Parameter value

Parameter	Initial value	Search interval	Final value
MaxIteNum	10000	5000-50000 (5000)	10000
MaxNunImp	500	100-2000 (100)	500
β	0.15	0.05-0.5 (0.05)	0.30
λ	0.9	0.1-1.0 (0.01)	0.99
T_{st}	2000	1000-10000 (1000)	9000
α	0.95	0.90-0.99 (0.01)	0.93
w_1	3	3-30	27
w_2	2	2-27	7
w_3	1	1-12	4
w_4	0	0	0