Hindawi
Journal of Advanced Transportation
Volume 2022, Article ID 2920532, 18 pages
https://doi.org/10.1155/2022/2920532

Research Article

Pick-Up and Delivery Problem for Sequentially Consolidated Urban Transportation with Mixed and Multi-Puprope Vehicle Fleet

Haoye Chen,[1] **Jonas Hatzenbühler**,[2] **and Erik Jenelius**[2]

[1]*School of Architecture and the Built Environment, KTH Royal Institute of Technology, Teknikringen 74C, Stockholm, 100 44, Sweden*
[2]*Division of Transport Planning, KTH Royal Institute of Technology, Brinellvägen 23, Stockholm, 10044, Sweden*

Correspondence should be addressed to Haoye Chen; haoye@kth.se

Received 7 November 2021; Revised 13 December 2021; Accepted 14 December 2021; Published 11 March 2022

Academic Editor: Lei Wang

Different urban transportation flows (e.g., passenger journeys, freight distribution, and waste management) are conventionally separately handled by corresponding single-purpose vehicles (SVs). The multi-purpose vehicle (MV) is a novel vehicle concept that can enable the sequential sharing of different transportation flows by changing the so-called modules, thus theoretically improving the efficiency of urban transportation through the utilization of higher vehicles. In this study, a variant of the pick-up and delivery problem with time windows is established to describe the sequential sharing problem considering both MVs and SVs with features of multiple depots, partial recharging strategies, and fleet sizing. MVs can change their load modules to carry all item types that can also be carried by SVs. To solve the routing problem, an adaptive large neighborhood search (ALNS) algorithm is developed with new problem-specific heuristics. The proposed ALNS is tested on 15 small-size cases and evaluated using a commercial MIP solver. Results show that the proposed algorithm is time-efficient and able to generate robust and high-quality solutions. We investigate the performance of the ALNS algorithm by analyzing convergence and selection probabilities of the heuristic solution that destroy and repair operators. On 15 large-size instances, we compare results for pure SV, pure MV, and mixed fleets, showing that the introduction of MVs can allow smaller fleet sizes while approximately keeping the same total travel distance as for pure SVs.

1. Introduction

The growing urban transportation brings challenges. Taking Stockholm as an example, it faces the problem of congestion [1] and ensures reliable transport as the number of inhabitants increase [2]. At the same time, Sims et al. [3] pointed out that unless transport emissions can be strongly decoupled from GDP growth, the increasing transport activities could outweigh all mitigation measures to reduce global transport greenhouse gas emissions. As a potential solution, Savelsbergh and Woensel [4] and Los et al. [5] stressed the importance of integrated and/or collaborative transportation.

Transportation demand tends to substantially vary across space and over time. The demand profiles differ depending on the type of transportation. Figure 1 shows an example of unevenly distributed demand of multiple transportation types. Passenger flows peak in the morning and evening, while freight is concentrated during the daytime and recycling demand is solely distributed in the morning.

Passenger, freight, and recycling transportation operations are currently handled by separate fleets of vehicles of distinct types. These single-purpose vehicles (SVs) can transport items of only one type each. When one kind of demand is low, the utilization of corresponding vehicles becomes less efficient as they stand idle or run with lower fill rates, while the vehicle capacity for other demand types may be insufficient. Chen and Li [6] and Liu et al. [7] discussed the similar underutilization of nonmodular

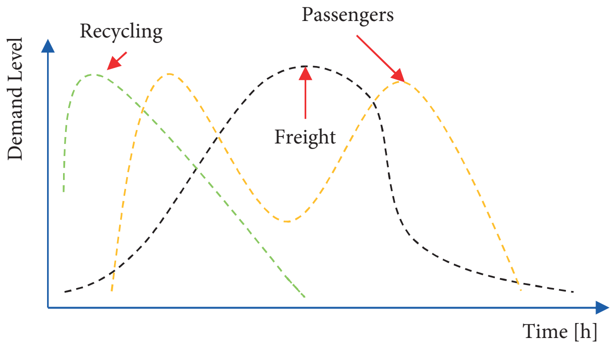


Figure 1: Example demand distribution for passengers, freight, and recycling in an urban context.

vehicles. Each kind of SV is specially designed for a specific demand type and thus cannot efficiently carry items of other types. For example, vehicles for passengers have plenty of seats, which only allow few parcels in the small storage room or on the seats. Some vehicles are not able to carry items of any other types (e.g., using vehicles for waste to carry people).

The concept of multi-purpose vehicles (MVs), which can change their load modules at dedicated change stations to transport items of multiple types while keeping the drive train and other parts, is schematically shown in Figure 2. Given the current rapid development of autonomous vehicles, future MVs may be automated (i.e., no drivers). If not, the vehicle cab for drivers should be located on the fixed component.

The overall cost for operating a fleet including MVs is the fixed cost of vehicles, the variable cost of the total travel distance, and the module change cost at change stations. MVs enable sequential sharing of vehicles, thereby potentially increasing the utilization of vehicles and achieving a lower overall cost by fewer vehicles for the fixed cost and more flexible choices of routes for the variable cost. In the example in Figure 1, parts of the MV fleet may transport passengers and waste, respectively, in the morning peak. Then, MVs can be changed for freight during the daytime. Before the evening peak, the MVs can be changed back to transport people. The MVs utilized for recycling in the morning peak can also transport passengers and freight later. As a result, fewer MVs than SVs are potentially required for urban transportation.

Future urban transportation should be sustainable to cope with the challenges of global warming. Great efforts in developing electric vehicles have been devoted to such ambitions. Therefore, we assume all vehicles are electric in the future. However, even if electric vehicles are environmentally friendly in operation, manufacturing electric vehicles will still cause many emissions. Higher utilization of MVs can theoretically solve this problem to a certain extent by using fewer vehicles.

From the view of MVs, the module change needs extra distance for visiting a change station. Fortunately, if the MVs are electric, this shortcoming will be overcome since increasing the driving distance of electric vehicles does not cause excessive emissions. Therefore, electric MVs can enjoy more flexible route choices and higher utilization without significantly increasing emissions.

Although various forms of shared mobility systems are widely studied (see [8]), little research has been devoted to the sequential sharing problem. In a recently published study by Hatzenbühler et al. [9], the authors propose a pick-up and delivery problem with time windows and sequential sharing. The authors state that the operation of sequentially sharing MVs leads to a reduction in fleet size, while the level of service for passengers and freight requests can be maintained. We will extend it by introducing the mixed fleet including both SVs and MVs to study the transition from the current pure SV fleet to the pure MV fleet or the mixed fleet. The performed numerical experiments will fill the aforementioned research gap by enhancing the understanding of the impacts of MVs in future urban transportation systems.

To understand the potential of MVs, there are two important questions: i) what are the most suitable vehicle fleet characteristics (i.e., a pure MV fleet or a mixed fleet containing both SVs and MVs)? and ii) how to plan routes and fleet configurations (i.e., how many vehicles of each type should be used)? To answer these questions, an adaptive large neighborhood search (ALNS) algorithm is developed, which is able to determine the best routes and fleet configurations. We then compare the best solutions for pure SV operations, pure MV operations, and operations with mixed fleets.

The contributions of this study can be summarized as follows:

(i) To the best of our knowledge, we are the first to consider mixed fleets and MVs in the pick-up and delivery problem and present the mathematical formulation of the problem.

(ii) We integrate the aspects of multiple depots, partial recharging strategies, fleet sizing, and mixed fleets including both MVs and SVs.

(iii) We propose an efficient ALNS algorithm for the proposed problem, which introduces new mechanisms to deal with MVs.

(iv) We validate the performance of the proposed heuristic by comparing its results to the solutions from an exact algorithm.

(v) We show that the mixed fleet may improve the solutions obtained with pure SVs in scenarios with unevenly distributed demand of several types. Both a pure MV fleet and a mixed fleet may lead to fewer vehicles, while the total travel distance is comparable to a SV fleet.

The remainder of the study is organized as follows. Section 2 reviews related work in the literature. Section 3 proposes the problem and gives it the mathematical formulation. Section 4 presents the proposed ALNS algorithm. The results and performances of the ALNS operators are displayed in Section 5 followed by discussions. Section 6

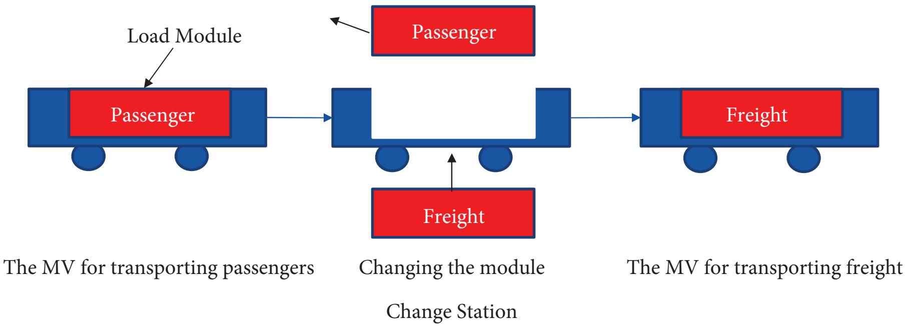


Figure 2: Schematic concept of multi-purpose vehicles.

contains conclusions, future possible applications, and research directions.

2. Literature Review

2.1. Pick-Up and Delivery Problem with Time Windows. One variant of the widely studied vehicle routing problem (VRP) is the PDPTW. In the PDPTW, a set of vehicle routes are determined so that a set of transportation requests are served at a minimum cost with the given vehicle fleet [10]. The PDPTW uses a heterogeneous vehicle fleet stationed at multiple depots to satisfy a set of predetermined transportation requests. Each request consists of a pick-up location, a delivery location, and a certain number of items that should be transported. The pick-up time and delivery time of a request should fall in the time windows of both locations. The objective function may include the fixed cost of vehicles and various variable operational costs (see [11]).

The dial-a-ride problem (DARP) is a variation of the PDPTW in which the transported items are people and thus focuses more on the quality of service and the convenience of passengers [11, 12]. For more comprehensive reviews, we refer the interested reader to articles about PDPTW [13] and DARP [12, 14].

2.2. Combining Different Types of Transportation. In urban transportation, different types of items move in heterogeneous carriers. Most studies on combined transportation discuss the combination of two types of items in the same vehicles, which can be mainly classified into two categories: single-tiered and two-tiered models.

In a single-tiered model, vehicles can simultaneously transport goods and passengers to their delivery points. Li et al. [15] introduced a new kind of model called the share-a-ride problem (SARP) based on the dial-a-ride problem. In the SARP, goods and passengers are transported by taxis in the city.

As for the two-tiered model, the first tier typically describes a scheduled transportation mode such as bus and train, whose spare space can be used to carry goods. In the second tier, smaller vehicles transport goods or passengers to their final destinations. Thus, strict synchronization at the transfer points of each tier is necessary [8].

2.3. Fleet Size and Mixed-Fleet Problem. The fleet size and mixed-fleet (FSM) problem was first proposed by Golden et al. [16] in a VRP study, in which multiple types of vehicles (varying in capacity, range, etc.) are available and the size of the fleet is unlimited. Hiermann et al. [17] introduced the electric fleet size and mix VRP with time windows (E-FSMFTWs) and recharging stations, which considers the choice of recharging times and locations.

2.4. Partial Recharging. Electric vehicles (EVs) have rapidly developed in the last decade. However, EVs suffer problems like long recharging times and insufficient charging station locations. Thus, the optimization of EVs has become an important and timely research topic.

Conrad and Figliozzi [18] considered a recharging vehicle routing problem (GVRP), which allows EVs to be recharged at selected customer locations. Erdoğan and Miller-Hooks [19] proposed a green vehicle routing problem and presented a mathematical formulation with recharging stations. EVs can be fully recharged at recharging stations in a predefined time. However, time window and capacity constraints are not considered in their problem. Schneider et al. [20] introduced the electric vehicle routing problem with time window (E-VRPTW) and recharging stations based on the GVRP, in which the recharging time depends on the remaining electricity when an EV arrives at a recharging station. Time window and capacity constraints are also considered in the proposed E-VRPTW. Grandinetti et al. [21] discussed electric pick-up and delivery problem with time windows, which extends the PDPTW with recharging stations.

2.5. Adaptive Large Neighborhood Search. ALNS, first introduced by Ropke and Pisinger [22], is a heuristic algorithm consisting of competing subheuristics whose selection frequencies are adjusted based on their historical performance. ALNS has been widely used and proven effective for solving the general VRP. Ropke and Pisinger [22] applied ALNS to the PDPTW and improved the best-known solution in benchmark instances. Li et al. [23] designed an ALNS algorithm for the SARP proposed in Li et al. [15] and proved its effectiveness in instances generated from real taxi trails and benchmark instances of DARP. Hiermann et al. [17] solved the E-FSMFTW utilizing branch-and-price and an adaptive large neighborhood search with an embedded local search and labeling procedure for intensification. The effectiveness of their proposed algorithms is evidenced by a newly created set of benchmark instances for the E-FSMFTW and the existing single vehicle type benchmark using an exact method. Keskin and Çatay [24] proposed an innovative ALNS for the EVRP-PR, which introduces new removal and insertion mechanisms of recharging stations. The computational results of benchmark instances showed that their proposed method is effective and of high quality.

3. Methodology

3.1. Model Description. In this study, we consider a mixed fleet including MVs and several types of SVs, in which the load modules of MVs can be switched at dedicated change stations to transport any item type served by the SVs. The entire vehicle fleet is assumed to be electric. Each vehicle can be either partially or fully recharged at recharging stations. Each capacity-limited vehicle starts from one of the depots and returns to the depot where it originated. Requests, which contain several types of items with pick-up locations, delivery locations, and time windows, are served by the mixed fleet. As an illustration, Figure 3 displays one route by a vehicle. The vehicle serves two passenger requests and recharges the battery at the recharging station. After visiting a change station, it serves one freight request and returns to the same depot where it originated.

We aim to find the best vehicle configuration of the mixed fleet and routes of these vehicles in the network composed of customer points (pick-up and delivery points), recharging stations, change stations, and depots to achieve an overall minimum cost including the fixed cost of vehicles, the variable cost of the total travel distance, and the change cost at change stations. The problem can be regarded as a variant of the pick-up and delivery problem with time windows (PDPTWs) as defined in the literature review. We refer to it as the multi-depot pick-up and delivery problem with time windows, partial recharging strategies, fleet sizing, and mixed fleet of single-purpose vehicles and multi-purpose vehicles.

3.2. Mathematical Problem Formulation. In this section, we present the mathematical model of the proposed problem. The nomenclature and parameters are displayed in Table 1. In this model, all points except depots can be visited only once. Thus, multiple duplicates of change stations and recharging stations are added to the network to allow multiple visits to a station. Each duplicate can be visited once, and the number of duplicates is large enough to cover the maximum number of possible visits at a change or recharging station. To distinguish the arrival time, load, and electricity when a vehicle starts and ends at a depot, we add one copy of depots as the end depots differed from the start depots. Given n_d depots, depot vertices $\{1, \ldots, n_d\}$ denote the start depots and depot vertices $\{n_d + 1, \ldots, 2 * n_d\}$ denote the end depots. MVs must empty their loads before entering a change station.

There are two classes of vehicles. One is MV, and the other one is SV containing three vehicles types. The request of an item type can be handled by an SV or an MV of the corresponding vehicle type. Although MVs can transport items of multiple types at different moments, the type of the MV is defined as the type of item it can carry at a particular moment.

Let N^* denote all points in the network and p denote the class of a vehicle (i.e., 0 for MV and 1 for SV). We define the binary decision variable x_{ijkd}^{p} as equal to 1 if a vehicle of the class p carries items of type k, travels along the arc (i, j) for $i, j \in N^*$, and originates from the depot d. Otherwise, the variable is 0. Let R denote the request set for all types. We define the binary decision variable z_j^r for tracking the item of request r in the vehicle route. It equals 1 if a vehicle carries the item of request r when arriving at the point j, and 0 otherwise. Given the request $r \in R$, $r^+ \in C$ denotes its pick-up point and $r^- \in C$ denotes its delivery point.

3.3. Objective Formulation. The objective function (1) minimizes the overall costs including the fixed costs of vehicles and variable costs proportional to the distance traveled. The first and second terms denote the fixed costs of SVs and MVs, respectively. The fixed cost of vehicles contains the purchase cost and the labor cost (e.g., driver and packing if any) amortized to each trip. The third and fourth terms represent the variable costs brought by SVs and MVs, respectively. The variable cost contains the cost of energy consumption per distance unit. The fifth term represents change costs, which are proportional to the number of module changes. The meaning of its parameters is displayed in Table 1.

$$\min \Bigg[\sum_{k\in K} f^k \sum_{d\in D} \sum_{i\in D} \sum_{j\in C\cup S'} x_{ijkd}^{1} + f^0 \sum_{k\in K} \sum_{d\in D} \sum_{i\in D} \sum_{j\in C\cup S'} x_{ijkd}^{0} + \sum_{k\in K} \alpha^k \sum_{i\in N} \sum_{j\in N'} \sum_{d\in D} d_{ij} x_{ijkd}^{1} + \alpha^0 \sum_{k\in K} \sum_{i\in N} \sum_{j\in N'} \sum_{d\in D} d_{ij} x_{ijkd}^{0} + f_c \sum_{i\in F'} \sum_{j\in N'} \sum_{k\in K} \sum_{d\in D} x_{ijkd}^{0} \Bigg]. \tag{1}$$

3.4. Constraints. The model characteristics are defined in this subsection. The constraints related to depots and the mixed fleet are based on the formulation proposed by Salhi et al. [25], which considers multiple depots and a mixed fleet

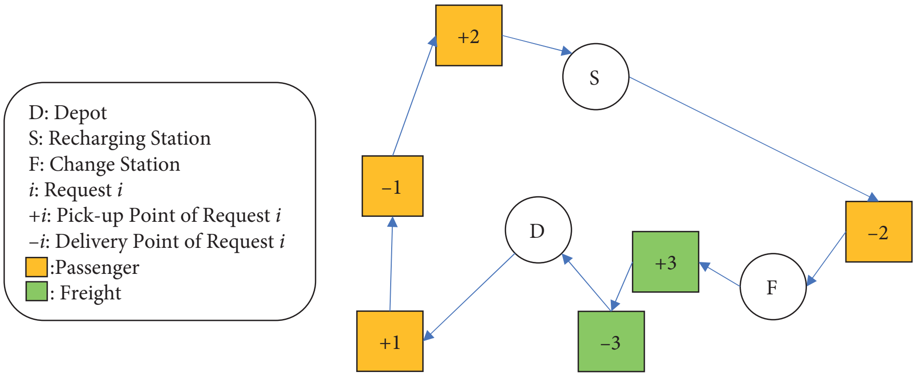


Figure 3: An example of a vehicle route.

Table 1: Sets of points, parameters, and decision variables.

K	Vehicle types (i.e., transported item types)
C_k	Customer points for item type k
C	Customer points for all item types, $C = \cup C_k$
R	Requests, in which a request has r a pick-up point r^+ and a delivery point r^-
C_k^+	Pick-up points for item type k
D	Depots where vehicles start
D'	Depots where vehicles end
F'	Duplicated change stations
S'	Duplicated recharging stations
N	Customer and change points with start depots, $N = C \cup D \cup F' \cup S'$
N'	Customer and change points with end depots, $N' = C \cup D' \cup F' \cup S'$
N^*	All points in the network, $N^* = N \cap N'$
P	Vehicle classes, 1 for SVs, 0 for MVs
f^k	Fixed cost of the SV type k
f^0	Fixed cost of the MV
α^k	Variable cost of the SV type k
α^0	Variable cost of the MV
f_c	Change cost of the MVs at any change station
d_{ij}	The distance/cost for traveling from i to j
$[e_i, l_i]$	Time range for $i \in C \cup D \cup D'$
s_i	Service time for a customer point i.
τ_i	Starting time of a point i when visited by a vehicle at arrival.
E^k	Energy storage capacity for SVs of type k.
E^0	Energy storage capacity for MVs.
g^k	Recharging time per energy unit for vehicle type k.
g^0	Recharging time per energy unit for MVs.
η^k	Energy consumption per unit distance for vehicle type k.
η^0	Energy consumption per unit distance for MVs.
Q^k	Load capacity for SVs of type k.
Q^0	Load capacity for MVs
q_j	Volume of the item at customer point j
M_t	Upper bound of time.
M_y	Upper bound of energy, $M_y = \max\{E^k\}$
M_q	Upper bound of load, $M_q = \max\{Q^k\}$.
x^p_{ijkd}	1, if a vehicle of the class p, carrying items of type k, traveling along arc (i, j) for $i, j \in N^*$, and originating from depot d. 0, otherwise
z^r_j	1, if a vehicle carries the item of request r when arriving at point j. 0, otherwise
y_i	Energy level when a vehicle arrives at point i
Y_i	Energy level when a vehicle leaves the recharging station i
u_i	Remaining capacity at point i when being visited

in a VRP. We extend the formulation by using the variables z_j^c tracking movements of items on vehicle routes and introduce features of MVs, change stations, and recharging stations.

$$\sum_{p\in P}\sum_{i\in N}\sum_{d\in D} x_{ijkd}^p = 1, \quad \forall k \in K, \forall j \in C_k^+, \tag{2}$$

$$\sum_{p\in P}\sum_{i\in N}\sum_{d\in D} x_{ijkd}^p = 0, \quad \forall k \in K, \forall j \notin C_k^+, \tag{3}$$

$$\sum_{p\in P}\sum_{i\in N}\sum_{k\in K}\sum_{d\in D} x_{ijkd}^p \le 1, \quad \forall j \in S', \tag{4}$$

$$\sum_{i\in N}\sum_{k\in K}\sum_{d\in D} x_{ijkd}^0 \le 1, \quad \forall j \in F', \tag{5}$$

$$\sum_{i\in N}\sum_{k\in K}\sum_{d\in D} x_{ijkd}^1 = 0, \quad \forall j \in F', \tag{6}$$

$$\sum_{i\in N'}\sum_{k\in K}\sum_{d\in D} x_{jikd}^1 = 0, \quad \forall j \in F', \tag{7}$$

$$\sum_{i\in D}\sum_{k\in K}\sum_{d\in D} x_{ijkd}^0 = 0, \quad \forall j \in F', \tag{8}$$

$$\sum_{i\in N} x_{ijkd}^p = \sum_{i\in N} x_{jikd}^p, \quad \forall p \in P, \forall k \in K, \forall j \in C \cup S', \forall d \in D, \tag{9}$$

$$\sum_{i\in N}\sum_{k\in K} x_{ijkd}^0 = \sum_{i\in N'}\sum_{k\in K} x_{jikd}^0, \quad \forall j \in F', \forall d \in D, \tag{10}$$

$$\begin{gathered}\sum_{j'\in N'} x_{jj'kd}^0 - M\left(1 - x_{ijkd}^0\right) \le 0, \\ \forall i \in N, \forall j \in F', \forall k \in K, \forall d \in D,\end{gathered} \tag{11}$$

$$\sum_{p\in P}\sum_{k\in N}\sum_{d\in D} x_{iikd}^p = 0 \quad \forall i \in N^*, \tag{12}$$

$$\begin{aligned} x_{d_1ikd_2}^p = 0, \quad & \forall p \in P, \forall i \in C \cup F' \cup S', \forall k \in K, \\ & \forall d_1, d_2 \in D, d_1 \ne d_2, \end{aligned} \tag{13}$$

$$\begin{aligned} x_{id_1kd_2}^p = 0, \quad & \forall p \in P, \forall i \in C \cup F' \cup S', \forall k \in K, \forall d_1 \in D', \\ & \forall d_2 \in D, d_1 \ne d_2 + n_d, \end{aligned} \tag{14}$$

$$\sum_{p\in P}\sum_{k\in K}\sum_{d\in D} x_{ijkd}^p = 0, \quad \forall i, j \in D, \tag{15}$$

$$\sum_{p\in P}\sum_{k\in K}\sum_{d\in D} x_{ijkd}^p = 0, \quad \forall i, j \in D', \tag{16}$$

$$\sum_{p\in P}\sum_{k\in K}\sum_{d\in D} x_{ijkd}^p = 0, \quad \forall i \in D, \forall j \in D', \tag{17}$$

$$\sum_{p\in P}\sum_{k\in K}\sum_{d\in D} x_{ijkd}^p = 0, \quad \forall i \in D', \forall j \in D, \tag{18}$$

$$x_{ijkd}^p \in \{0, 1\}, \quad \forall p \in P, \forall i \in N^*, \forall j \in N^*, \forall k \in K, \forall d \in D. \tag{19}$$

Constraints (2) and (3) ensure that each pick-up point is served by a vehicle of the corresponding type. Constraint (4) guarantees that each recharging station may be visited by all vehicles at most once. Constraints (5)-(7) enforce that module change stations can be exclusively visited by MVs. Constraint (8) ensures MVs cannot drive from a depot to a change station. The flow conservation for customer points and recharging stations is handled by Constraint (9). Constraint (10) ensures the conservation of flow for change stations. Constraint (11) guarantees that a vehicle, which visits a change station, must change its vehicle type. M is a sufficiently large parameter for the conditional constraint. Constraint (12) avoids self-circles for all points. Constraints (13) and (14) ensure that vehicles return to the same depot where they depart. Constraints (15)-(18) prevent vehicles directly traveling from depots to depots. Constraint (19) defines the binary decision variable.

The following set of constraints record the carrying status of requests:

$$\sum_{r\in R} z_j^r = 0, \quad \forall j \in D \cup D' \cup F', \tag{20}$$

$$\sum_{p\in P}\sum_{k\in K}\sum_{d\in D} x_{ijkd}^p = 1 \longrightarrow z_i^r = z_j^r, \quad \forall i \in N, \forall j \in N, \forall r \in R, i \ne r^+, i \ne r^-, i \ne j, \tag{21}$$

$$\sum_{p\in P}\sum_{k\in K}\sum_{d\in D} x_{r^+jkd}^p = 1 \longrightarrow z_j^r = 1, \quad \forall j \in N', \forall r \in R, r^+ \ne j, \tag{22}$$

$$\sum_{p\in P}\sum_{k\in K}\sum_{d\in D} x^{p}_{r^{-} jkd} = 1 \longrightarrow z^{r}_{j} = 0, \quad \forall j \in N', \forall r \in R,\ r^{-} \neq j, \tag{23}$$

$$z^{r}_{r^{-}} = 1, \quad \forall r \in R, \tag{24}$$

$$z^{r}_{j} \in \{0, 1\}, \quad \forall j \in N^{*}, \forall r \in R, \tag{25}$$

Constraint (20) defines the initial situation for all start depots and guarantees that vehicles have emptied their loads before visiting end depots and change stations. Constraint (21) passes the carrying status of requests from i to j if they are not picked up or delivered at i. Constraint (22) ensures that vehicles collect items at pick-up points of requests. Constraint (23) guarantees that vehicles drop items at delivery points of requests. Constraint (24) ensures that the delivery point must be served after its corresponding pick-up point. Constraint (25) defines the binary decision variable.

The last set of constraints record the arriving time, energy level, and load. To track the energy usage, the decision variable y_i is defined as the energy level when a vehicle arrives at the point i. The decision variable Y_i is defined as the energy level when a vehicle leaves the recharging station i. The recharged amount of electricity is $(Y_i - y_i)$, which follows the definition first introduced by Keskin and Çatay [24]. The decision variable τ_i records the time when a vehicle visits the point i.

In the proposed model, each type of SV and the MV has a different energy storage capacity, recharging time per energy unit, and energy consumption per unit distance. When expressing parameters of vehicle types, we use the index of $\{1, \ldots, K\}$ representing SV types and 0 representing MVs. The decision variable u_j describes the current remaining capacity at the point j.

$$e_j \leq \tau_j \leq l_j, \forall j \in C \cup D \cup D', \tag{26}$$

$$\tau_i + \left(t_{ij} + s_i\right) x^{p}_{ijkd} - M_t\left(1 - x^{p}_{ijkd}\right) \leq \tau_j, \quad \forall p \in P, \forall i \in C, \forall j \in N', \forall k \in K, \forall d \in D, i \neq j, \tag{27}$$

$$\tau_i + t_{ij} x^{0}_{ijkd} + t_{swap} - M_t\left(1 - x^{0}_{ijkd}\right) \leq \tau_j, \quad \forall i \in F', \forall j \in N', \forall k \in K, \forall d \in D, i \neq j, \tag{28}$$

$$\tau_i + t_{ij} x^{1}_{ijkd} + g^{k}\left(Y_i - y_i\right) - M_t\left(1 - x^{1}_{ijkd}\right) \leq \tau_j, \quad \forall i \in S', \forall j \in N', \forall k \in K, \forall d \in D, i \neq j, \tag{29}$$

$$\tau_i + t_{ij} x^{0}_{ijkd} + g^{0}\left(Y_i - y_i\right) - M_t\left(1 - x^{0}_{ijkd}\right) \leq \tau_j, \quad \forall i \in S', \forall j \in N', \forall k \in K, \forall d \in D, i \neq j, \tag{30}$$

$$0 \leq y_j \leq Y_j, \quad \forall j \in S', \tag{31}$$

$$0 \leq Y_i \leq E^{0} + M_y\left(1 - x^{0}_{ijkd}\right), \quad \forall i \in S', \forall j \in N', k \in K, d \in D, i \neq j, \tag{32}$$

$$0 \leq Y_i \leq E^{k} + M_y\left(1 - x^{1}_{ijkd}\right), \quad \forall i \in S', \forall j \in N', k \in K, d \in D, i \neq j, \tag{33}$$

$$0 \leq y_j \leq y_i - \left(\eta^{0} d_{ij}\right) x^{0}_{ijkd} + M_y\left(1 - x^{0}_{ijkd}\right), \quad \forall i \in C \cup F', \forall j \in N', k \in K, d \in D, i \neq j, \tag{34}$$

$$0 \leq y_j \leq y_i - \left(\eta^{k} d_{ij}\right) x^{1}_{ijkd} + M_y\left(1 - x^{1}_{ijkd}\right), \quad \forall i \in C \cup F', \forall j \in N', k \in K, d \in D, i \neq j, \tag{35}$$

$$0 \leq y_j \leq E^{0} - \left(\eta^{0} d_{ij}\right) x^{0}_{ijkd} + M_y\left(1 - x^{0}_{ijkd}\right), \quad \forall i \in D, \forall j \in N', k \in K, d \in D, i \neq j, \tag{36}$$

$$0 \leq y_j \leq E^{k} - \left(\eta^{k} d_{ij}\right) x^{1}_{ijkd} + M_y\left(1 - x^{1}_{ijkd}\right), \quad \forall i \in D, \forall j \in N', k \in K, d \in D, i \neq j, \tag{37}$$

$$0 \leq u_j \leq u_i - q_i x^{p}_{ijkd} + M_q\left(1 - x^{p}_{ijkd}\right), \quad \forall p \in P, \forall k \in K, \forall i \in C \cup F' \cup S', \forall j \in N', d \in D, i \neq j, \tag{38}$$

$$0 \le u_j \le Q^0 + M_q\left(1 - x^0_{ijkd}\right), \quad \forall k \in K, \forall i \in D, \forall j \in N', d \in D, i \ne j, \tag{39}$$

$$0 \le u_j \le Q^k + M_q\left(1 - x^1_{ijkd}\right), \quad \forall k \in K, \forall i \in D, \forall j \in N', d \in D, i \ne j. \tag{40}$$

Constraint (26) defines the time windows of depots and customer points for all types. Constraints (27) and (28) describe the time changes at customer points and change stations, respectively. Constraints (29) and (30) consider the time changes at recharging stations for MVs and SVs, respectively. Constraint (31) ensures that the energy level of a vehicle leaving a recharging station is higher than the energy level of the vehicle when arriving. Constraints (32) and (33) guarantee that the energy levels are no more than the maximum energy capacity of the corresponding vehicle types when leaving the recharging station. Constraints (34) and (35) describe the energy change at recharging stations for MVs and SVs, respectively. Constraints (36) and (37) describe the energy consumption from depots to other points for MVs and SVs, respectively. Constraint (38) describes the load changes at customer points, change stations, and recharging stations. Constraints (39) and (40) describe the load changes when leaving start depots for MVs and SVs, respectively.

4. Adaptive Large Neighbourhood Search

This section details the ALNS algorithm for the proposed problem. Section 4.1 gives an overview of the proposed algorithm. Section 4.2 defines the process to generate initial solutions. Section 4.3 discusses the strategy for sequential sharing. Section 4.4 discusses the strategy for partial recharging. Section 4.5 describes the evaluation of solutions. Sections 4.6, 4.7, 4.8, and 4.9 introduce the customer destroy (CD) operators, customer repair (CR) operators, station destroy (SD) operators, and station repair (SR) operators, respectively.

4.1. Solution Generation. The pseudocode of the proposed ALNS is displayed in Algorithm 1. In the first step, an initial solution is created (see Section 4.2). Then, the algorithm iteratively destroys and repairs the current solution. For each iteration, if the new solution is not energy feasible (i.e., nonfulfillment of energy constraints), the station insertion algorithm Greed (see section 4.9) is applied to get an energy feasible solution.

The new solution is accepted or rejected based on a simulated annealing (SA) approach: Let X_{New} and X_{Current} be the new solution and the current solution, respectively. If $f(X_{\text{New}})$ is less than $f(X_{\text{Current}})$, where f is the objective function of the model plus load penalty E_l and time penalty E_t (defined in Algorithm 2), the new solution is accepted as the current solution. Otherwise, the new solution may still be accepted with a probability of $e^{-(f(X_{\text{New}})-f(X_{\text{Current}})/T)}$. After each iteration, the temperature T is multiplied by the cooling rate α and the best solution will be updated if the new solution is best so far. The initial temperature guarantees that the first new solution worse than the current solution will be accepted with a probability of 0.5. If $f(X_{\text{New}})$ is less than $f(X_{\text{Current}})$ and if E_l and E_t are zero, the new solution will be updated as the best solution. Infeasible solutions with better f will be accepted as the current solution, which increases the flexibility of ALNS.

The analysis function f contains the objective of the model given the solution plus a load penalty E_l and a time penalty E_t scaled by parameters β_l and β_t, respectively, as shown in Algorithm 2. $E_l = 0$ guarantees the fulfillment of load constraints. $E_t = 0$ ensures the fulfillment of the time windows. The new solution is always energy feasible according to Algorithm 1. Therefore, a penalty for electricity is not needed. The outputs f, E_t, and E_l are utilized in the SA approach.

The ALNS algorithm's structure is based on the algorithm proposed by Keskin and Çatay [24], which is proven useful for recharging stations in the vehicle routing problem. An iteration of operators for charging stations is followed after every $N_{\text{SO}} - 1$ iterations of operators for destroying and repairing requests.

When choosing the destroy operator and repair operator during each iteration, we use a roulette wheel selection. This mechanism is proposed by Ropke and Pisinger [22]. As shown in Algorithm 1, iterations are divided into segments with the lengths of N_c and N_e for customers and recharging, respectively. The probability P_i of selecting an operator i during a segment s depends on its weight in the previous segment W_l^{s-1} (i.e., $P_i^s = W_i^{s-1} / \sum_{l=1}^{m} W_l^{s-1}$). In particular, the weight of the operator i in the segment s is $W_i^s = W_i^{s-1}(1 - r_p) + r_p \pi_i^{s-1} / \vartheta_i^{s-1}$, where r_p is the roulette wheel parameter, ϑ_i^{s-1} represents the usage of the operator i during the segment $s-1$, and π_i^{s-1} is the score of the operator i during the segment $s-1$. For each iteration, the score of the chosen operator π_i^s is increased by σ_1, σ_2, or σ_3, which are scores if the iteration generates the best solution so far, improves the current solution, or accepts a worse solution according to the SA criterion.

After each customer/recharging segment, the scores and weights of the customer/recharging operators are reset. In the reset of the segment s, if the operator i is not selected within the segment (i.e., $\pi_i^s = 0$), the weight of this operator in the next segment $s+1$ will be $W_i^{s+1} = W_i^s(1 - r_p) + r_p \sigma_3 / 3$. The small value $\sigma_3/3$ guarantees that W_i^s does not approach zero so that every operator can be selected even if they have bad performance for a period.

4.2. Initial Solution. In the generation of the initial solution, requests are handled according to their type. Out of all requests from the same type, one request is randomly selected as the base request to generate a vehicle. The start depot is chosen so that f increases the least for this vehicle carrying only the base request. The request is randomly

```
Generate an initial solution
$j \leftarrow 1$
while $j \le Maximum$ do
    if $j \equiv 0(\bmod N_{SO})$ then
        Select SD operator and remove stations on the current solution
        Select SR operator and repair solution to get a new solution
    else
        Select CD operator and remove customer on the current solution
        Select CR operator and repair solution to get a new solution
        Updating change stations of the new solution
        if energy infeasible solution then
            Perform the station insertion algorithm Greed on the new solution
        end
    end
    Apply the sequential sharing strategy
    Apply the partial recharging strategy
    Algorithm 2: Assignment of vehicles' class and solution evaluation
    Using the SA criterion to accept/reject the new solution as the current solution and updating the best solution if it is feasible and best so far
    $T \leftarrow \alpha * T$
    if $j \equiv 0(\bmod N_c)$ then
        reset scores and update weights of CD and CR operators
    end
    if $j \equiv 0(\bmod N_e)$ then
        reset scores and update weigths of SD and SR operators
    end
    $j \leftarrow j + 1$
end
```

ALGORITHM 1: ALNS.

```
Input new solution $X$
Output $f(s)$, $E_t$ and $E_t$
for each vehicle in the solution do
    if requests of different types in the vehicle then
        Assign the vehicle as a MV
    else
        calculate the objective value $\mathrm{Obj}_0$ contributed by the vehicle if it is multi-purpose
        calculate the objective value $\mathrm{Obj}_1$ contributed by the vehicle if it is single-purpose
        Assign the vehicle with the class $p$ of lower $\mathrm{Obj}_p$
    end
    If the arriving time of point $i$: $\tau_i$ is earlier than its time-window, the vehicle will wait
    If the vehicle load at point $i$ is beyond its load capacity, record the excess load as $v_i$
end
Calculate the load penalty $E_l = \sum_{i \in N^*} v_i$
Calculate the time penalty $E_t = \sum_{i \in C \cup D \cup D'} \max(\tau_i - l_i, 0)$
Sum up the objective value contributed by each vehicle as $Objective$
$f(s) \leftarrow \mathrm{Objective} + \beta_l * E_l + \beta_t * E_t$
```

ALGORITHM 2: Assignment of vehicles' class and solution evaluation.

assigned to an MV or SV. Then, the remaining requests will be inserted into the best position while considering the feasibility of time and load. Requests with earlier possible visiting times have priorities in the insertion order. If it is impossible to insert any remaining request into this vehicle, a new vehicle is generated based on one request randomly chosen from the remaining requests. The request insertion and vehicle generation processes are repeated until all remaining requests of this type are inserted. The process is then for all types. Finally, the station insertion algorithm (see Section 4.9) is performed to make the initial solution feasible in energy constraints.

4.3. Strategy for Sequential Sharing. Sequential sharing does not allow vehicles to simultaneously carry different types of items. New change stations may be in need when inserting a request into a vehicle with at least one different type of request. Conversely, existing change stations may be removed when deleting a request whose neighboring points on both sides belong to the same type. After applying destroy and repair operators, the proposed algorithm checks every two successive customer points. Then, if the two customer points on both sides are not of the same type, the change station adding the lowest distance increment is inserted. If the two customer points are of the same type, the change station is removed from the solution.

However, the insertion of new change stations into a vehicle route is performed after the destroy and repair operators have been applied. The insertion of a change station adds additional travel distance to the route, which is initially not considered when applying the destroy and repair operators. Therefore, the distance matrix equivalent as the direct distance plus the possible additional travel distance (see Equation (41)) is used in order to integrate the additional travel distance into the route, hence correctly updating the cost values.

$$d_{ij}' = \begin{cases} \min_{f \in F'} \left[d_{if} + d_{fj} + f_c \right] & \forall i \in C_{k_1}, \forall j \in C_{k_2}, k_1 \neq k_2 \\ d_{ij} & \text{Otherwise} \end{cases} \tag{41}$$

If two points are not of the same type, their equivalent distance is the distance same as the two points as endpoints to pass the nearest change station. Otherwise, their equivalent distance is the same as the direct distance.

4.4. Partial Recharging Strategy. Keskin and Çatay [24] showed that: "if an optimal solution exists such that an EV leaves the depot with its battery partially charged, then the same EV departing from the depot fully charged is also optimal since fully recharging the battery at the depot does not delay the departure time of the EV." Therefore, vehicles start with a fully recharged battery in our proposed algorithm.

We propose a simple and optimal partial recharging strategy that avoids any time window violations. Since the recharging time increases with the energy that is recharged, after obtaining an energy feasible solution, our proposed algorithm sets the recharged amount of energy at each station to a minimum value, which ensures that the EV arrives at the next station/depot with an empty battery.

4.5. Assignment of Vehicles' Class and Solution Evaluation. In the proposed problem, there are K kinds of SVs and one kind of MV. The purpose of a vehicle is important since each kind of vehicle has a different capacity, maximum energy capacity, recharging speed, and energy-consuming speed. In Algorithm 2, the objective function is computed and the purpose of vehicles is changed based on the evaluation result.

4.6. Customer Destroy Operators. The customer destroy operators delete or move customer requests.

(i) Random location: in the current solution, a number γ_c of points are selected including customer points, change stations, and recharging stations. Then, requests containing the selected customer points are removed.

(ii) Path: inspired by Demir et al. [26], a request is randomly selected. Then, stations and requests with at least one point within the sequence between the pick-up point and the delivery point of the preselected request are removed.

(iii) Worst distance: inspired by Keskin and Çatay [24], the request with the $\lceil |R| \mu^{\kappa} \rceil^{\text{th}}$ largest distance increment, calculated as the total distance of the vehicle minus the total distance after the removal, is removed. Here, $|R|$ means the total number of requests, μ is a random number from 0 to 1, and κ is a parameter influencing the randomness.

(iv) Shaw: inspired by Ropke and Pisinger [22], the idea of this operator is to remove similar requests. For two requests $r_1, r_2 \in R$, the similarity is defined as follows:

$$R(r_1, r_2) = \frac{1}{d_{r_1^+ r_2^+} + d_{r_1^- r_2^-} + \lambda \left(\left| \tau_{r_1^+} - \tau_{r_2^+} \right| + \left| \tau_{r_1^-} - \tau_{r_2^-} \right| \right)}, \tag{42}$$

where τ_i is the arriving time at the point i. Equation (42) considers the differences in distances and arriving time between the two pick-up points and the two delivery points. The two requests with maximum similarity are removed.

(v) Random vehicle: a random vehicle is selected and deleted.

(vi) Pair: two requests are selected and removed. The information of their original vehicles (i.e., the vehicle from which each request is removed) is swapped between those vehicles.

(vii) Cross-swap: two vehicles are randomly selected. All points with empty loads, when being visited, are recorded. For each vehicle, the route is broken into two parts based on their one random recorded point. The back half of each vehicle route is connected to the front half of the other vehicle route. Finally, a time feasibility check is conducted from the breaking point to the end depots for the two vehicles, which deletes the requests containing infeasible points until all remaining requests are time feasible.

Table 2: Results of parameter tuning (Obj $*10^4$).

	Default	Tested							
β_t				Penalty for the time infeasibility					
	6	1	5	5.5	6.5	7	**8**	10	20
Obj	2.240 9	2.417 4	2.395 2	2.221 9	2.213 5	2.238 5	**2.195 4**	2.336 5	2.362 4
β_l				Penalty for the load infeasibility					
	100	10	50	**80**	90	120	150	200	300
Obj	2.227	2.217 7	2.204 5	**2.195 2**	2.216 3	2.230 6	2.223	2.211 5	2.199 4
N_{so}				Period to destroy and repair recharging stations					
	10	3	4	5	6	7	**8**	9	11
Obj	2.208 6	2.361 7	2.222 2	2.252 6	2.201 5	2.235 7	**2.192 3**	2.195 1	2.199 8
N_c				Length of customer segment for resetting scores					
	25	10	20	22	27	**30**	32	35	40
Obj	2.243 3	2.301	2.192 2	2.206 4	2.188 3	**2.187 3**	2.208 1	2.224 5	2.204 2
N_e				Length of station segment for resetting scores					
	200	100	150	175	190	210	**220**	250	300
Obj	2.191	2.239 7	2.208 6	2.209 5	2.252 2	2.217 1	**2.175 5**	2.180 6	2.197 7
r_p				Roulette wheel parameter					
	0.25	0.15	**0.2**	0.3					
Obj	2.185 1	2.267 2	**2.181 6**	2.191 1					
κ				Randomness in the *Worst Distance*					
	5	2	3	**4**	6	7	8		
Obj	2.198 0	2.181 8	2.276 8	**2.180 0**	2.182 3	2.245 1	2.206 2		
λ				Parameter for time differences in the Shaw removal					
	2	0.1	1	1.25	**1.5**	2.5	5	10	100
Obj	2.204 8	2.283 8	2.197 4	2.195 7	**2.179 0**	2.189 2	2.211 2	2.201 6	2.248 4
γ_c				Proportion for deleting customers in destroy operators					
	0.2	0.1	0.125	0.15	0.175	0.225	0.25	0.3	
Obj	**2.193 8**	2.23	2.205	2.242 4	2.197 6	2.243 5	2.230 6	2.237 4	
γ_s				Proportion for deleting stations in destroy operators					
	0.2	0.1	**0.15**	0.25					
Obj	2.201 8	2.197 7	**2.193 9**	2.213 4					
$\sigma_{1,2,3}$				Scores in the simulated annealing					
	15,10,5	15,5,10	15,10,13	15,10,8					
Obj	**2.180 7**	2.210 3	2.235 4	2.193 4					

(viii) Swap: two requests are randomly selected, and their pick-up points and delivery points are swapped. Therefore, no requests are deleted.

4.7. Customer Repair Operators. The customer repair operators insert requests deleted in destroy operators. The best position for inserting a request i is defined as the position leading to the lowest distance increment compared to the ones brought by other positions. The distance increment brought by the insertion of request i is defined as follows:

$$\Delta d_i = \begin{cases} d_{\text{prev}(i^+),i^+} + d_{i^+,\text{next}(i^+)} - d_{\text{prev}(i^+),\text{next}(i^+)} + d_{\text{prev}(i^-),i^-} + d_{i^-,\text{next}(i^-)} - d_{\text{prev}(i^-),\text{next}(i^-)}, & i^+ \text{and } i^- \text{are adjacent,} \\ d_{\text{prev}(i^+),i^+} + d_{i^+,i^-} + d_{i^-,\text{next}(i^-)} - d_{\text{prev}(i^+),\text{next}(i^-)}, & \text{Otherwise,} \end{cases} \tag{43}$$

where prev and next mean the previous point and the next point, respectively.

(i) Original vehicle: the insertion order of requests is first randomly shuffled. Then, each request is inserted into its original vehicle where it is removed at the best position. If the request's original vehicle does not exist after destroy operators, it will be inserted into a random vehicle.

(ii) Inter vehicle: the process is similar to the first repair operator. The only difference is that the insertion considers all positions among all vehicles.

(iii) New vehicle: the operator generates vehicles and inserts requests following the same method in the generation of the initial solution.

(iv) Random insertion: the order is randomly disrupted in which vehicles are inserted. Requests are sorted by their earliest possible visiting time for insertion. Then, each request in the list is inserted by order into the best position within the vehicle while considering the time and load feasibility. If all remaining requests cannot be inserted into the vehicle, the next vehicle will be considered. If there are still remaining requests and all vehicles have been considered, operator *New Vehicle* will be executed for these requests.

4.8. Station Destroy Operators. The station destroy operators are used to eliminate or move recharging stations in the solution, which may improve the overall performance by removing inefficient ones or adjusting their positions. Four operators are implemented. A fixed number γ_s of recharging stations are selected for destroying operators, except for the inefficient visit operator.

(i) Worst usage: following the operator proposed by Keskin and Çatay [24], this operator aims to make use of the battery as much as possible. Thus, recharging stations are sorted by the energy levels when they are visited. γ_s recharging stations are removed with the highest energy level.

(ii) Inefficient visit: we propose a new destroy operator for recharging stations. To be recharged, vehicles need to visit recharging stations, which increases the total travel distance. Thus, avoiding a large increment with a short recharging stop is reasonable. The proposed operator sorts the recharging stations by the ratio of the recharged amount to distance increment by station and counts the number of stations with zero recharging amount as γ_0. Then, the first $\max(\gamma_0, \gamma_s)$ stations are removed.

(iii) Random station: γ_s recharging stations are randomly removed in order to diversify the solution.

(iv) Random walk: γ_s recharging stations are selected and then moved one position forward or backward in their vehicles' routes.

4.9. Station Repair Operators. The station repair operators are utilized to make the solution electric feasible by inserting recharging stations. There are three operators implemented, and all of them are based on Keskin and Çatay [24].

(i) Greed: this operator checks the route from the start depot. Once a point is visited with a negative battery state, a recharging station is inserted into the arc between this point and its previous point, which brings the least distance increment. If the insertion at this arc is not feasible, the previous arc is considered.

(ii) Comparison: this operator checks the first point visited with a negative battery state and looks at arcs before this point. If the nearest feasible arc for insertion has an infeasible previous arc or a feasible previous arc for insertion of more distance increments, a recharging station with the least distance increment is inserted into the arc. Otherwise, the previous arc is considered in the same way.

(iii) Best arc: once a point is visited with a negative battery state, all arcs between the point and the previous station or depot are considered. The least distance increment brought by the insertions of a recharging station at all feasible arcs is calculated. The station at the arc generating the shortest distance increment is inserted.

To avoid the special case that there is no feasible arc between the last station/depot to the point visited with a negative battery state, we assume a relatively large amount of vehicle energy capacity in the experiments.

5. Numerical Experiments and Analysis

In order to validate our proposed ALNS algorithm, 15 small instances are solved using ALNS and the solver Gurobi 7.5.2. The results of both methods for each instance are presented. The problems are solved using an 8-core i7-8700 CPU of a 3.20 GHz computer. In addition to the numerical results, we record and analyze the selection probabilities of operators and the objective values of one optimization run. Finally, 15 large-size cases are tested to compare the results by pure SV, pure MV, or mixed fleets.

5.1. Parameter Tuning. Inspired by the tuning approach of Ropke and Pisinger [22], Demir et al. [26], and Keskin and Çatay [24] for testing each setting of a parameter, the proposed ALNS is executed 20 times for only 1,500 iterations. Then, the average objective value of 20 trials is recorded. Finally, the setting with the best average objective value is chosen as the parameter for the numerical experiments in the remainder of this study. Only one parameter is varied at a time while keeping the others as default values if they are not tuned or tuned values otherwise. We tuned parameters according to the order in Table 2. An artificial instance of nine requests with three types is used in tuning. Table 2 displays the details of the parameter tuning results. The best parameter values are indicated in bold. We set a high cooling rate α of 0.995 to make the first 1,500 iterations converge fast while manually tuning the cooling rate as 0.999 5 for experiments to make the program result in a good performance in the entire iterations.

5.2. Small-Size Instances. We have manually created 15 instances. Table 3 displays general information about the number of requests, the number of recharging stations, depots, and change stations for each instance. As for the request vector, the number in position i of the vector represents how many requests are there for each type i. The instances have different numbers of recharging stations, change stations, depots, requests, and request types to ensure that the instances represent a varied set of possible scenarios. The fixed costs of SVs and the MV are randomly chosen from the range [2000, 4000], which enables our program to tradeoff between shorten routes and fewer vehicles. The fixed

Table 3: Results for small instances.

Instance	Requests	$[n_S, n_d, n_F]$	Avg.$t_l\,(s)$	Avg.$t_a\,(s)$	$t_e\,(s)$	Avg.dev	Optimum
1	[3,1]	[2,1,1]	0.316 807	2.112 883	105.22	0.00%	5
2	[3,1]	[2,1,1]	0.194 065	2.003 296	269.6	0.00%	5
3	[3,1]	[1,2,1]	0.322 63	1.768 072	5.16	0.00%	5
4	[3,2]	[1,2,1]	0.178 373	2.017 891	30.42	0.00%	5
5	[3,2]	[1,1,2]	0.513 943	1.901 372	4.09	0.00%	5
6	[3,1]	[2,2,1]	0.254 505	2.347 111	2787.43	0.00%	5
7	[3,1]	[3,1,1]	0.517 603	2.162 484	2887.55	0.71%	3
8	[2,2]	[1,2,1]	0.104 138	1.809 524	2.94	0.00%	5
9	[3,3]	[1,1,2]	0.572 434	2.186 973	21.37	0.00%	5
10	[3,1]	[1,3,1]	0.968 397	1.630 676	180.1	0.57%	0
11	[3,2]	[1,2,2]	0.708 596	1.738 11	341.12	0.00%	5
12	[2,2,2]	[1,2,1]	0.549 16	1.611 34	19.06	0.00%	5
13	[2,1,3]	[1,2,2]	0.659 06	1.597 949	267.50	0.00%	5
14	[2,2,3]	[1,2,2]	1.496 54	2.440 7	7200+[1]	0.01%	4
15	[2,2,3]	[1,2,2]	2.158 38	3.102 671	2481.43	0.34%	2

[1] The objective bound limit was not reached within 2 h. The objective value reported is the last value computed by the MIP solver.

Table 4: Time window settings of base instances (pick-up ⟶ delivery, hour).

Type 1	Type 2	Type 3
[0,10] ⟶ [1,12]	Half requests: [5,14] ⟶ [6,15] Half requests: [12,24] ⟶ [12,24]	[6,18] ⟶ [6,18]

cost of MV can be higher than all, some, or none of the modeled SV types. The positions, time windows, loads of customer points, and capacity of vehicles are also varied from instance to instance to increase their diversity.

Each instance is solved by the proposed ALNS algorithm five times and the MIP solver one time. The maximum number of iterations of ALNS is set to 10,000 in Case 1-13, 15,000 in Case 14, and 20,000 in Case 15. In Table 3, n_S, n_d, n_F are the numbers of recharging stations, depots, and change stations, respectively. Avg.t_l means the time to find the best solution. Avg.t_a means the time to finish all iterations. t_e denotes the time obtaining the optimal solution by the exact algorithm. Avg.Dev denotes the average percentage of the result obtained by the ALNS algorithm that deviates from the optimal solution obtained by the MIP solver. Optimum records how many times the ALNS reaches the optimal solution within 5 trials.

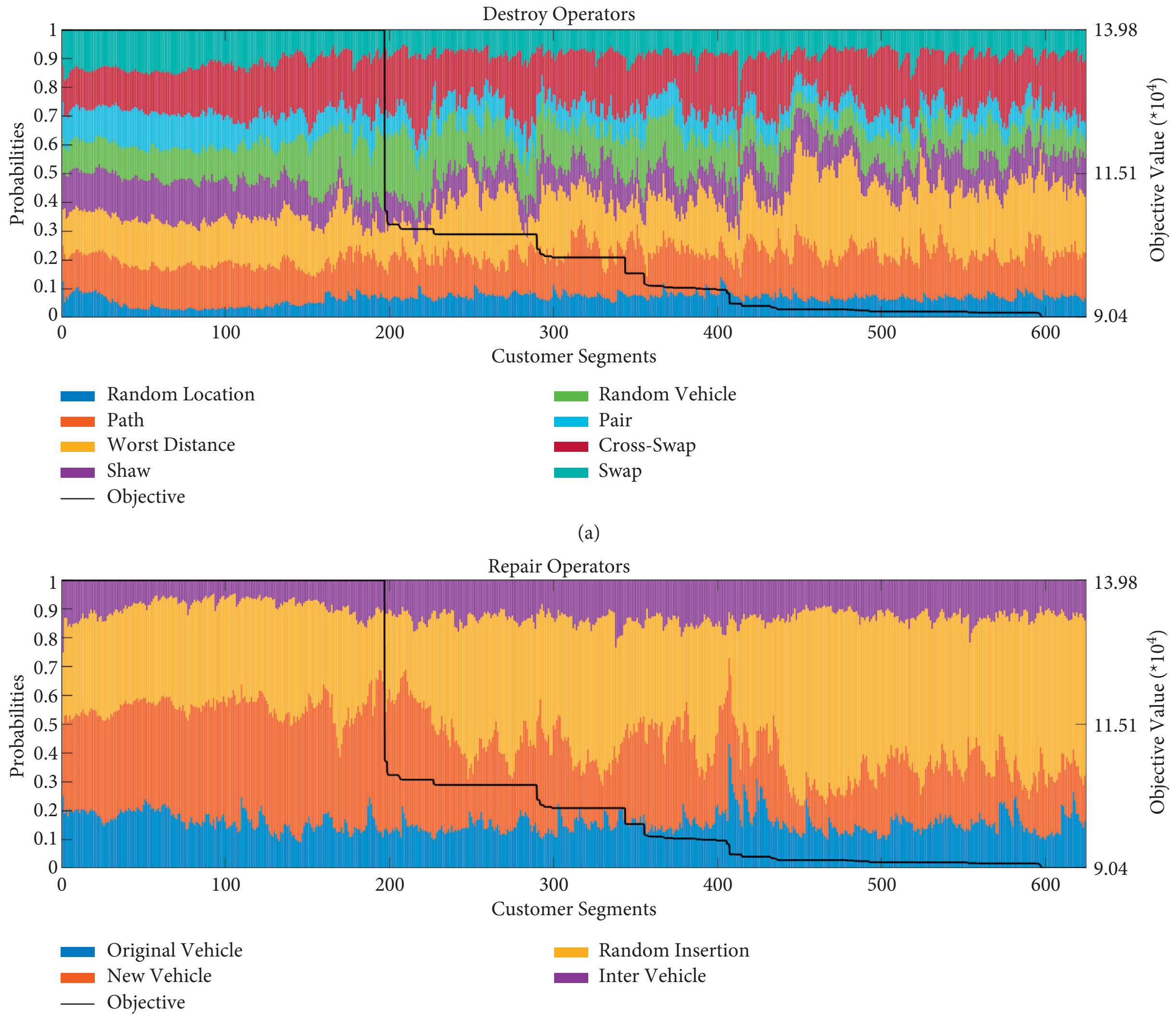
As Table 3 shows, the Avg.t_l of cases 1-13 is less than 1 second, which indicates that the proposed algorithm can reach a good solution fast. The Avg.t_a of cases 1-15 except the last case is less than 2.5 seconds, which suggests that the proposed ALNS normally has stable and fast computation time under the same stopping conditions. Comparing the Avg.t_l and the Avg.t_a with the t_e for all cases, hence, we could show that the proposed ALNS is efficient to solve the problems.

At the same time, there is at most a 0.71% average deviation for all the 15 cases. The ALNS reaches the optimum solution for all 5 trials in 11 out of 15 cases. Although the ALNS is stuck in a locally optimal solution in Case 10, the deviation is small. Therefore, the quality of solutions generated by the ALNS is high.

5.3. Operator Analysis. To analyze the operators implemented in this study, we run the proposed algorithm on instance L90-3 (see detailed instance description in 5.4). As mentioned in Section 4, iterations are divided into fixed-length segments and the probability for choosing one operator at one segment depends on its performance on the previous segment. Figures 4 and 5 display the objective curves and selection probabilities at all segments for customer and station operators, respectively. Components of the stacked bar at a segment represent the probabilities of operators at this segment.

Looking at the customer destroy operators in Figure 4(a), before the objective starts to decrease, the probabilities of the operators Cross-swap and Random Vehicle increase, while the probabilities of operators Swap, Pair, and Shaw decrease. At the same time, the operators Worst Distance and Path keep their probabilities. The probability of the operator Random Location increases first, then decreases, and later increases again. With the decrease in objective value, the probabilities of the operators Worst Distance, Path, and Cross-swap dominate while the probability of the operator Random Vehicle continuously decreases. Additionally, the operators Random Location, Shaw, Pair, and Swap remain at low probabilities throughout the entire optimization, indicating poor optimization performance.

Considering the customer repair operators in Figure 4(b), Original Vehicle and Inter Vehicle have low probabilities throughout the entire optimization, indicating poor possibilities to improve the objective value. Before the drop of the objective, the algorithm tries to generate lots of new vehicles proven by the increasing probability of the operator New Vehicle while the operator Random Insertion keeps a high probability at the same time. After the objective value reduces, the operator Random Insertion dominates and fewer new vehicles are generated.



(a)

(b)

Figure 4: The probabilities of customer operators at each segment. (a) Customer destroy operators. (b) Customer repair operators.

Before the decrease in the objective in Figure 5(a), the probability of the station destroy operator Random Walk continuously decreases while the other three operators share approximately equal probabilities. After the objective starts to decline, the operator Inefficient Visit dominates while the other three operators fluctuate at relatively lower probabilities. Toward the end of the optimization, all four operators share roughly equal probabilities.

In Figure 5(b), the three implemented station repair operators have approximately equal probabilities before the objective starts to decrease. Subsequently, the operators oscillate for the majority of the optimization until returning to an equal share.

In ALNS, we apply both conventional operators and problem-specific operators. Among the conventional operators, Worst Distance and Path are frequently chosen during the optimization, suggesting superior performance. The problem-specific operators Cross-swap, Random Insertion, and Inefficient Visit show good performance, while Pair performs worst according to the reported selection probabilities.

5.4. Large-Size Cases. To evaluate the effectiveness of the consolidated urban transport by MVs given requests with uneven time distribution, we create three base instances. For requests of each type in all base instances, their time windows are concentrated in specific time periods during the day to simulate demands of different types, which are unevenly distributed in time. These three base instances have a total number of requests of 30, 50, and 90 each. The total number of requests is divided into three item types. Base instance L30 has 9, 14, and 7 requests, respectively, for three item types, while base instance L50 has 17, 16, and 17 requests, respectively. Base instance L90 contains 30 requests for each type. Three base instances have the same distribution of time windows as shown in Table 4.

In Table 4, the left side of an arrow denotes a time window of the pick-up points. The right side of an arrow denotes a time window of the delivery points. The time windows for type 1 are concentrated around the first half of a day. Time windows for type 2 have two kinds representing

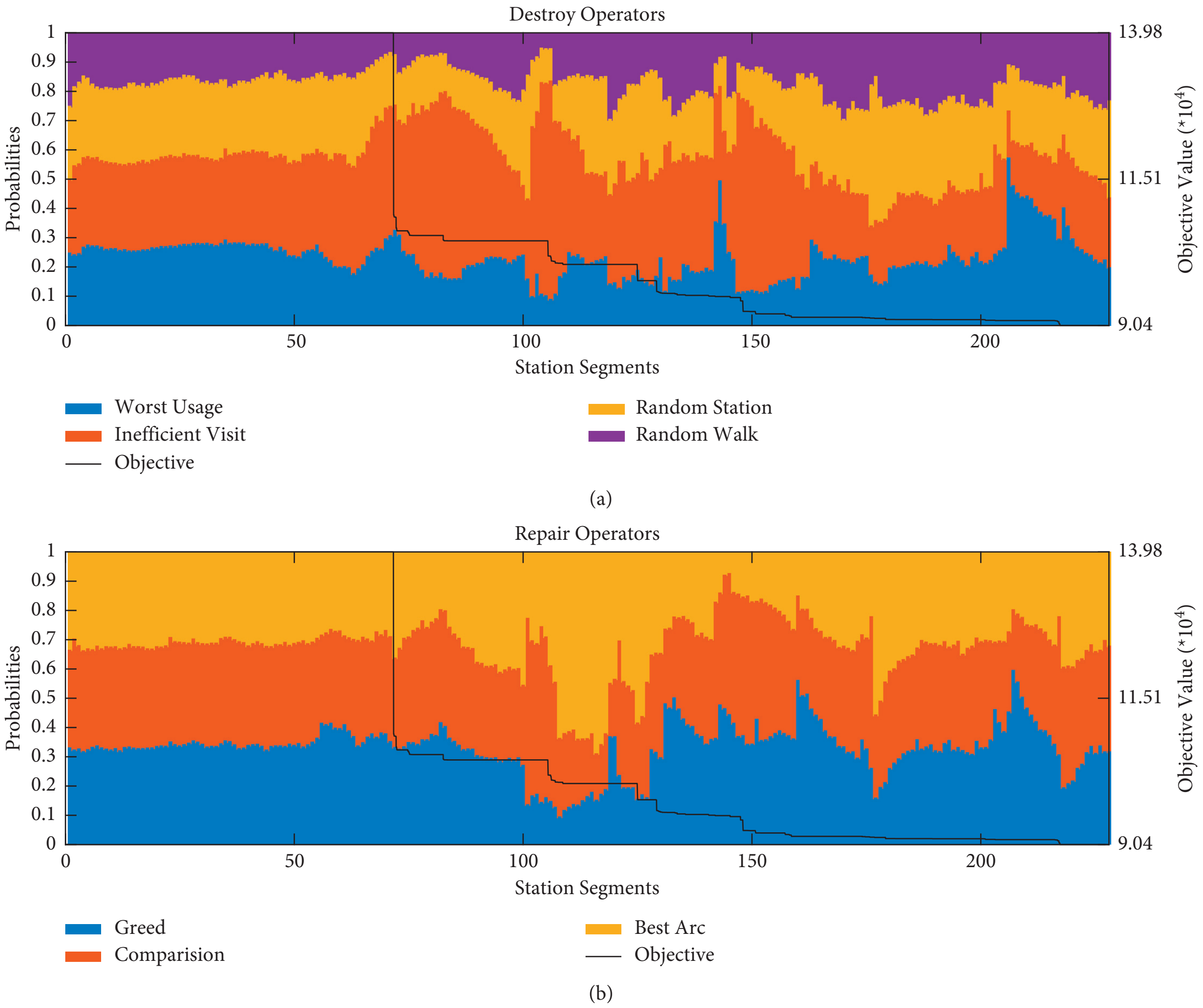


Figure 5: The probabilities of station operators at each segment. (a) Station destroy operators. (b) Station repair operators.

morning and evening peaks of passengers. The time windows for type 3 cover the whole daytime. Requests of different types need to be served in different periods (i.e., unevenly distributed time windows for different demand types).

To study the performance of pure SVs, pure MVs, and the mixed vehicle fleet, the proposed program can be slightly modified to obtain solutions for pure MVs and pure SVs besides the mixed vehicle fleet. To consider pure SV fleets, a large enough value may be assigned as the fixed cost of MVs so that no MV will be utilized. For pure MV fleets, all vehicles may be assigned as MVs in Algorithm 2.

At the same time, the MV's fixed cost f^0 is an important factor for decision-makers to consider the fleet configuration. For simplicity, we assume that all SV types have the same fixed cost $f^* = 2000$. Furthermore, we create two different situations for MVs based on their fixed costs. One situation assumes that MVs have equal fixed cost f^* as SVs and the other situation assumes that MVs have a slightly higher fixed cost of 1.2 $f^* = 2400$. To control other factors, all SVs and MVs have the same parameters of recharging performance, energy capacity, and volume capacity.

As Table 5 and Figure 6 show, the three base instances are computed with different vehicle fleet characteristics and MV cost structures. Each instance is computed 3 times using 50,000 iterations, and the solution with the overall best objective is displayed. The program uses the same parameter settings as shown in Table 2 except larger N_c of 80, which extends the length of resetting customer operator scores to accommodate longer maximum iteration and more requests. We record the total travel distance, objective value, and vehicle configuration for each instance.

In instances with the same fixed cost structure (i.e., L30-1 to L30-3, L50-1 to L50-3, and L90-1 to L90-3), MVs and mixed vehicle fleets can improve the objective value by 5-15% compared to SVs. At the same time, fewer vehicles are utilized in pure MV and mixed vehicle fleet solutions. The introduction of MVs (i.e., both mixed fleet and pure MVs) implies fewer vehicles and lower objectives if the same fixed costs are assumed.

In instances L30-4 to L30-5, L50-4 to L50-5, and L90-4 to L90-5, the improvement of the objective value in L30-4 is much smaller than those in L30-2, L30-3, and L30-5. The pure MV fleet even has a higher objective value than the SV

Table 5: Results of large-scale instances.

Instance	Fleet	f^0	Distance		Objective	
			Value	Percentage	Value	Percentage
L30-1	SV	Inf	11 498.47	100.00%	27 498.47	100.00%
L30-2	MV	f^*	11 405.70	99.19%	23 705.70	86.21%
L30-3	Mixed	f^*	11 247.01	97.81%	23 547.01	85.63%
L30-4	MV	1.2 f^*	11 247.01	97.81%	25 947.01	94.36%
L30-5	Mixed	1.2 f^*	11 070.83	96.28%	24 070.83	87.54%
L50-1	SV	Inf	21 975.75	100.00%	51 975.75	100.00%
L50-2	MV	f^*	21 640.93	98.48%	47 840.93	92.04%
L50-3	Mixed	f^*	21 606.89	98.32%	47 906.89	92.17%
L50-4	MV	1.2 f^*	22 985.32	104.59%	52 285.32	100.60%
L50-5	Mixed	1.2 f^*	21 872.38	99.53%	48 872.38	94.03%
L90-1	SV	Inf	41 421.75	100.00%	95 421.75	100.00%
L90-2	MV	f^*	42 029.33	101.47%	90 629.33	94.98%
L90-3	Mixed	f^*	41 057.86	99.12%	89 557.86	93.85%
L90-4	MV	1.2 f^*	42 169.33	101.80%	100 669.33	105.50%
L90-5	Mixed	1.2 f^*	42 598.40	102.84%	93 598.40	98.09%

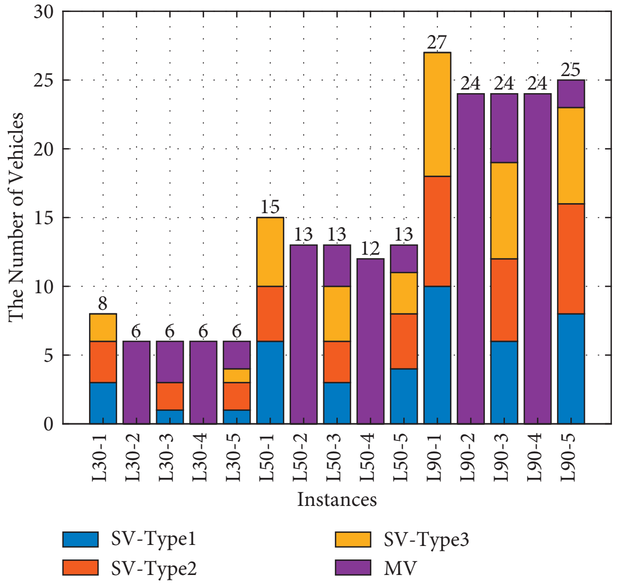


Figure 6: Vehicle configurations of large-scale instances.

fleet in L50 and L90. However, the mixed vehicle fleet can still improve the objective value compared to the SV fleet when the MV has a slightly higher fixed cost than SVs. This is surprising, but a potential explanation for the increased objective values for large-scale instances with pure MV fleets and increased fixed costs (i.e., L50-4 and L90-4) can be the underlying demand pattern. The more the customer requests are in a certain area, the less the time and space gaps of the customer points are. Hence, the more evenly distributed the requests are within that area. Since the operational benefit of MVs mainly stems from the efficient consolidation of unevenly distributed demand in time and/or space, the higher demand density potentially neutralizes this operational advantage of MVs. However, the total number of vehicles for both fleets, mixed and MV, can be reduced compared to SVs.

As for the total travel distance, there are no significant trends regarding the number of requests, fleet characteristics, and MV cost structures given the results. As a potential explanation, fewer vehicles require more travel distance although MVs enable more flexible routes. Therefore, fewer or more total distance depends on the adversarial relationship between more flexible routes and fewer vehicles. However, compared with pure SVs (i.e., L30-1, L50-1, and L90-1), the total distances of other instances have not changed more than 5%.

In the analyzed instances, it can be seen that MV and mixed vehicle fleets utilize fewer vehicles to serve the requests. This is true for scenarios with an equal or a higher fixed cost for MVs than for SVs. The introduction of MVs (i.e., both mixed fleet and pure MVs) brings improvements in terms of the objective value and fleet size, if MVs have the equivalent fixed cost as SVs. When increasing the fixed costs of MV by 20% compared to SV costs, the objective value of pure MVs rises or decreases less than the situation of equal fixed costs. However, the mixed vehicle fleet can still reduce the objective value in such a situation. At the same time, MVs do not significantly change the total distance. Namely, the MVs improve the objective mainly by fewer vehicles.

6. Conclusions and Future Works

This study investigates MVs in the pick-up and delivery problem with features of multiple depots, mixed fleet, and partial recharging strategies. The mathematical formulations of the problem and an efficient ALNS algorithm are proposed. Additionally, we propose new heuristic operators and strategies in order to accommodate the problem-specific characteristics induced by change stations and partial recharging.

In several numerical experiments, we show that the proposed ALNS can find solutions for small-scale cases with high qualities in an efficient time. We display probabilities' variation of proposed operators over entire iterations in one run and analyze it combining the objective curve. Through large-scale instances, the mixed fleet can reduce the total cost compared to SVs in scenarios where MVs have the same or higher fixed cost compared to SVs. MVs can lead to smaller fleet sizes of pure MVs or mixed fleets but have little positive or negative influence on the total travel distance. The improvement of the objective brought by MVs mainly derives from fewer vehicles.

Solution algorithms to the problem are not limited to our proposed ALNS. Some of the most representative computational intelligence algorithms are promising, such as ant colony optimization [27, 28], artificial bee colony algorithm [29, 30], monarch butterfly optimization [31, 32], and Harris Hawks optimization [33].

The MV is a promising direction for a more efficient urban transport system by enabling fewer fleet size while approximately keeping the total travel distance. When transitioning from the current fleet of SVs to a fleet of MVs, there will be a long period that SVs and MVs coexist. At the same time, SVs may have comparative advantages in some aspects. For example, some SVs are efficient in transporting items of a certain type. Therefore, the coexistence of SVs and MVs is still possible from a long-term perspective. The proposed model can help decision-makers in the transition phase and/or the coexistence of SVs and MVs. The mixed vehicle fleet analysis has shown that the transition toward MV is beneficial since the coexistence of SVs and MVs brings a reduction in the objective value and fleet size. Hence, the transportation system could be already improved in the transition phase. Pure MVs may also be the future trend if they have an even lower fixed cost compared to SVs.

To increase the practical applicability and generality of the model, a future research direction is to further investigate extra large-scale situations using real demand datasets. Another future research direction is the addition of dynamic demand considerations to the proposed problem so that decision-makers can dynamically accept or delete requests.

As for the aspect of MVs, different options of MV operations can be considered in future studies. Namely, there may be MVs with heterogeneous capacities, speed, or other vehicle-related parameters. Besides, MVs do not have to be able to transport items of all types. For example, some MVs may only be switched between passengers and freight, while some vehicles may only be switched between freights and recycling. Flexible changes between two types may be more feasible in a shorter time period and may already lead to benefits.

Data Availability

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

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] J. Eliasson, *The stockholm Congestion Charges: An Overview*, KTH Royal Institute of Technology location, Stockholm, Sweden, 2014.

[2] D. Firth, *City of stockholm, the City of stockholm Traffic Administration*, KTH Royal Institute of Technology location, Stockholm, Sweden, 2012.

[3] R. Sims, R. Schaeffer, F. Creutzig et al., “Climate Change 2014: Mitigation of Climate Change. Contribution of Working Group III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change,” Cambridge University Press, Cambridge, UK, IPCC 2014, 2014.

[4] M. Savelsbergh and T. Van Woensel, “Fiftyth anniversary invited article-city logistics: challenges and opportunities,” *Transportation Science*, vol. 50, no. 2, pp. 579–590, 2016.

[5] J. Los, F. Schulte, M. T. J. Spaan, and R. R. Negenborn, “The value of information sharing for platform-based collaborative vehicle routing,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 141, Article ID 102011, 2020.

[6] Z. Chen and X. Li, “Designing corridor systems with modular autonomous vehicles enabling station-wise docking: discrete modeling method,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 152, Article ID 102388, 2021.

[7] T. Liu, A. Ceder, and A. Rau, “Using deficit function to determine the minimum fleet size of an autonomous modular public transit system,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2674, no. 11, pp. 532–541, 2020.

[8] A. Mourad, J. Puchinger, and C. Chu, “A survey of models and algorithms for optimizing shared mobility,” *Transportation Research Part B: Methodological*, vol. 123, pp. 323–346, 2019.

[9] J. Hatzenbühler, E. Jenelius, G. Gidófalvi, and O. Cats, “Multi-purpose vehicle assignment for combined passenger and freight transport,” Accepted for Presentation: Transportation Research Board (TRB) 101, Washington, DC, USA, 2022.

[10] P. Toth and D. Vigo, “MOS-SIAM Series on Optimization,” *Vehicle Routing: Problems, Methods, and Applications*, Society for Industrial and Applied Mathematics : Mathematical Optimization Society, Philadelphia, PA, USA, 2nd edition, 2014.

[11] J. Desrosiers, Y. Dumas, M. M. Solomon, and F. Soumis, "Chapter 2 Time constrained routing and scheduling," in *Handbooks in Operations Research and Management Science*vol. 8, , pp. 35–139, Elsevier, 1995.

[12] K. F. Doerner and J.-J. Salazar-González, "Chapter 7: pickup-and-delivery problems for people transportation," *Vehicle Routing*, Society for Industrial and Applied Mathematics : Mathematical Optimization Society, Philadelphia, PA, USA, pp. 193–212, 2014.

[13] M. Battarra, J.-F. Cordeau, and M. Iori, "Chapter 6: pickup-and-delivery problems for goods transportation," *Vehicle Routing*, Society for Industrial and Applied Mathematics : Mathematical Optimization Society, Philadelphia, Pennsylvania, PA, USA, pp. 161–191, 2014.

[14] J.-F. Cordeau and G. Laporte, "The dial-a-ride problem: models and algorithms," *Annals of Operations Research*, vol. 153, no. 1, pp. 29–46, 2007.

[15] B. Li, D. Krushinsky, H. A. Reijers, and T. van Woensel, "The Share-a-Ride Problem: people and parcels sharing taxis," *European Journal of Operational Research*, vol. 238, no. 1, pp. 31–40, 2014.

[16] B. Golden, A. Assad, L. Levy, and F. Gheysens, "The fleet size and mix vehicle routing problem," *Computers & Operations Research*, vol. 11, no. 1, pp. 49–66, 1984.

[17] G. Hiermann, J. Puchinger, S. Ropke, and R. F. Hartl, "The electric fleet size and mix vehicle routing problem with time windows and recharging stations," *European Journal of Operational Research*, vol. 252, no. 3, pp. 995–1018, 2016.

[18] R. G. Conrad and M. A. Figliozzi, "The recharging vehicle routing problem," in *Proceedings of the Sixty first Annual IIE Conference*, vol. 8, Nevada, NV, USA, May 2011.

[19] S. Erdoğan and E. Miller-Hooks, "A green vehicle routing problem," *Transportation Research Part E: Logistics and Transportation Review*, vol. 48, no. 1, pp. 100–114, 2012.

[20] M. Schneider, A. Stenger, and D. Goeke, "The electric vehicle-routing problem with time windows and recharging stations," *Transportation Science*, vol. 48, no. 4, pp. 500–520, 2014.

[21] L. Grandinetti, F. Guerriero, F. Pezzella, and O. Pisacane, "A pick-up and delivery problem with time windows by electric vehicles," *International Journal of Productivity and Quality Management*, vol. 18, no. 2/3, p. 403, 2016.

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

[23] B. Li, D. Krushinsky, T. Van Woensel, and H. A. Reijers, "An adaptive large neighborhood search heuristic for the share-a-ride problem," *Computers & Operations Research*, vol. 66, pp. 170–180, 2016.

[24] M. Keskin and B. Çatay, "Partial recharge strategies for the electric vehicle routing problem with time windows," *Transportation Research Part C: Emerging Technologies*, vol. 65, pp. 111–127, 2016.

[25] S. Salhi, A. Imran, and N. A. Wassan, "The multi-depot vehicle routing problem with heterogeneous vehicle fleet: formulation and a variable neighborhood search implementation," *Computers & Operations Research*, vol. 52, pp. 315–325, 2014.

[26] E. Demir, T. Bektaş, and G. Laporte, "An adaptive large neighborhood search heuristic for the Pollution-Routing Problem," *European Journal of Operational Research*, vol. 223, no. 2, pp. 346–359, 2012.

[27] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 26, no. 1, pp. 29–41, 1996.

[28] B. Yu and Z. Z. Yang, "An ant colony optimization model: the period vehicle routing problem with time windows," *Transportation Research Part E: Logistics and Transportation Review*, vol. 47, no. 2, pp. 166–181, 2011.

[29] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Technical report-tr06, Erciyes university, engineering faculty, computer engineering department, Kayseri, Turkey, 2005.

[30] D. Karaboga and B. Basturk, "On the performance of artificial bee colony (abc) algorithm," *Applied Soft Computing*, vol. 8, no. 1, pp. 687–697, 2008.

[31] G.-G. Wang, S. Deb, and Z. Cui, "Monarch butterfly optimization," *Neural Computing & Applications*, vol. 31, no. 7, pp. 1995–2014, 2015.

[32] Y. Feng, S. Deb, G.-G. Wang, and A. H. Alavi, "Monarch butterfly optimization: a comprehensive review," *Expert Systems with Applications*, vol. 168, Article ID 114418, 2021.

[33] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: algorithm and applications," *Future Generation Computer Systems*, vol. 97, pp. 849–872, 2019.