1-2023

A diversity-enhanced memetic algorithm for solving electric vehicle routing problems with time windows and mixed backhauls

Jianhua XIAO

Jingguo DU

Zhiguang CAO
Singapore Management University, zgcao@smu.edu.sg

Xingyi ZHANG

Yunyun NIU

Citation

XIAO, Jianhua; DU, Jingguo; CAO, Zhiguang; ZHANG, Xingyi; and NIU, Yunyun. A diversity-enhanced memetic algorithm for solving electric vehicle routing problems with time windows and mixed backhauls. (2023). *Applied Soft Computing*. 134,.
Available at: https://ink.library.smu.edu.sg/sis_research/8193

Applied Soft Computing 134 (2023) 110025

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A diversity-enhanced memetic algorithm for solving electric vehicle routing problems with time windows and mixed backhauls

Jianhua Xiao [a,e], Jingguo Du [a], Zhiguang Cao [b,f], Xingyi Zhang [c], Yunyun Niu [d,*]

[a] Research Center of Logistics, Nankai University, Tianjin 300071, China
[b] Institute for Infocomm Research (I2R), A*STAR, 138632, Singapore
[c] Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, School of Computer Science and Technology, Anhui University, Hefei 230601, China
[d] School of Information Engineering, China University of Geosciences in Beijing, Beijing 100083, China
[e] The Laboratory for Economic Behaviors and Policy Simulation, Nankai University, Tianjin 300071, China
[f] Singapore Institute of Manufacturing Technology (SIMTech), A*STAR, 138634, Singapore

A R T I C L E I N F O

Article history:
Received 4 June 2020
Received in revised form 24 November 2022
Accepted 6 January 2023
Available online 13 January 2023

Keywords:
Electric vehicles
Vehicle routing problem
Memetic algorithm
Time windows
Mixed backhauls

A B S T R A C T

The electric vehicle routing problem (EVRP) has been studied increasingly because of environmental concerns. However, existing studies on the EVRP mainly focus on time windows and sole linehaul customers, which might not be practical as backhaul customers are also ubiquitous in reality. In this study, we investigate an EVRP with time windows and mixed backhauls (EVRPTWMB), where both linehaul and backhaul customers exist and can be served in any order. To address this challenging problem, we propose a diversity-enhanced memetic algorithm (DEMA) that integrates three types of novel operators, including genetic operators based on adaptive selection mechanism, a selection operator based on similarity degree, and modification operators for tabu search. Experimental results on 54 new instances and two classical benchmarks show that the proposed DEMA can effectively solve the EVRPTWMB as well as other related problems. Furthermore, a case study on a realistic instance with up to 200 customers and 40 charging stations in China also confirms the desirable performance of the DEMA.

1. Introduction

With increasing concerns about environmental deterioration and resource shortages, electric vehicles (EVs) are gaining more popularity in logistics [1,2]. As the primary power source of EVs, electricity would significantly reduce environmental pollution and carbon emissions compared with conventional energy. Therefore, EVs are deemed as an important alternative to internal combustion engine vehicles (ICEVs), especially in urban areas where vehicles need to brake and start frequently [3,4]. Studies related to the electric vehicle routing problem (EVRP) are also emerging in the research community, most of which are conducted based on the traditional vehicle routing problem (VRP). On the one hand, similar to the VRP, a fleet of vehicles in the EVRP departs from a depot to sequentially serve a set of customers at different known locations under a capacity constraint [5]. On the other hand, unlike the VRP, the vehicles in the EVRP need to occasionally recharge at some charging stations to maintain autonomy and extend the travel range, which makes route planning more complicated and challenging [6].

Theoretically, the EVRP is a variant of the VRP, and some key properties can be inherited from the VRP when studying the EVRP. As a representative one, time windows that prescribe the time constraints for the service have been widely investigated in the EVRP [7–14]. Time windows for serving customers can be treated as either hard or soft constraints, with certain affordable penalties being imposed on violations of the latter. Regardless of the difference between them, existing studies have indicated that time windows in the EVRP strongly impact the routes of vehicles, the selection of vehicle types, recharging activity, and even the determination of charging station locations. Although some success has been achieved for the EVRP with time windows (EVRPTW), these studies have concentrated on route planning solely with delivery demands from linehaul customers while ignoring the pickup demands from backhaul customers along the same route. In realistic logistics, apart from linehaul customers with delivery demands, backhaul customers with pickup demands are also often considered, such as in barreled water delivery, where it is necessary to pick up empty barrels from customers. Thus, the assumption of only linehaul customers makes the obtained solutions of the EVRPTW less effective. Contrastingly, the traditional VRP that considers both pickup and delivery

* Corresponding author.
E-mail address: yniu@cugb.edu.cn (Y. Niu).

https://doi.org/10.1016/j.asoc.2023.110025
1568-4946/

demands is defined as the VRP with backhauls (VRPB), where backhaul customers are assumed to be served only after linehauls [15,16]. In real-world applications, however, the demands of linehauls and backhauls should be met with equal priority rather than dominating each other. Therefore, the VRPB is extended further to the VRP with mixed backhauls (VRPMB), where linehaul and backhaul customers are served in an arbitrary order [17,18]. In this study, we further extend the EVRPTW with mixed backhauls (EVRPTWMB), a more comprehensive and practical EVRP model. For the sake of more realistic concerns, we also stipulate that electricity consumption is related to the weight of cargo carried on the vehicle and the travel distance. Compared to the VRP, EVRP, and their existing variants, it is evident that the settings and assumptions in the EVRPTWMB make it more challenging to solve.

Various solution methods regarding the general EVRP and its variants have been proposed. Most of these are based on metaheuristic algorithms because the EVRP is computationally NP-hard [12,19,20]. Among them, evolutionary algorithms [21–23], adaptive large neighborhood search [11,14,24], and variable neighborhood search [7,9,25] have been developed and have exhibited satisfactory performance in solving these problems. Nevertheless, the presence of mixed backhauls in the EVRPTWMB poses a major challenge to the search ability of these algorithms because the interrelated customer visiting sequence, vehicle loading, and maximum traveling range may lead to unstable performance. More importantly, the inferior population diversity may make them easily trapped in a local optimum, thereby deteriorating solution quality. To efficiently and effectively solve this problem, we propose a diversity-enhanced memetic algorithm (DEMA), which integrates three types of novel operators, i.e., genetic operators based on adaptive selection mechanism, a selection operator based on similarity degree, and modification operators for tabu search. The contributions of this study are summarized as follows:

(1) This is the first attempt to study the electric vehicle routing problem with time windows and mixed backhauls (EVRPTWMB), where customers are characterized as delivery (linehaul) and pickup (backhaul) entities with time window constraints but no service priority restrictions. Apart from the service sequence, the optimal route with the minimum total travel distance also considers the effect of load and travel distance on electricity consumption.

(2) A diversity-enhanced memetic algorithm (DEMA) is proposed by leveraging a genetic algorithm framework and a tabu search framework to solve the EVRPTWMB. The specifically designed genetic and selection operators can facilitate the search of the entire solution space more efficiently within the algorithm. Moreover, modification operators for improving the solution quality are exploited to enhance the local search ability.

(3) We synthesize two benchmarks with a total of 54 instances for the EVRPTWMB, based on which the effectiveness of the DEMA and its components is verified. Additionally, we further evaluate the performance of the DEMA based on 36 VRPMB instances, 27 VRPTW instances, and one real case study. The experimental results demonstrate that the DEMA performs robustly in terms of solution quality.

The remainder of this study is organized as follows. Section 2 briefly reviews the existing work. Next, Section 3 establishes the mathematical model of the studied problem. Section 4 elaborates on the proposed algorithm. Section 5 presents the experimental results. Finally, Section 6 concludes the study and discusses future work.

2. Related work

A growing number of studies on the EVRP have been carried out in recent years owing to social concerns over low carbon consumption and environmental preservation [26–29]. In this section, we review the EVRP related models and solution approaches. Specifically, the EVRP was first proposed by Conrad and Figliozzi [30], where EVs were allowed to recharge at certain customer locations to renew their trips. Erdoğan and Miller-Hooks [31] introduced EVs into the green VRP, where they needed to additionally visit the scattered charging stations to extend their travel ranges due to the limited battery capacity. However, neither of these studies considered the time window constraint for serving customers, which may substantially impact the operations in logistics planning. Therefore, Schneider et al. [7] made the first attempt at the EVRPTW and introduced that the batteries of EVs were fully recharged based on a linear charging function. Keskin et al. [12] studied the EVRPTW model with soft time windows and time-dependent queueing durations at the stations in light of the restricted capacity of charging stations.

For the electricity volume of EVs for a single recharge, the strategy can be classified as full and partial recharging. Full recharging for a long duration to replenish electricity may affect the timeliness of the logistics service and increase customer dissatisfaction. Therefore, some studies have instead focused on the EVRPTW with a partial recharging strategy. Bruglieri et al. [9] designed partial recharging strategies by setting the recharging level for an EV at each station as a decision variable, and then presented an EVRPTW model to minimize the total duration of travel time, waiting and recharging time, and the number of employed EVs. Keskin and Çatay [10] investigated an EVRPTW model with partial recharging by relaxing the full recharging constraint. Desaulniers et al. [32] presented four variants of the EVRPTW with single or multiple chargers per route and full or partial recharging strategies, respectively.

In other studies, hybrid fleets containing conventional vehicles and EVs were considered, given that companies may not be able to afford a pure fleet of EVs. Goeke and Schneider [33] proposed an EVRPTW model with a hybrid fleet where, unlike conventional vehicles, EVs adopt an energy consumption function that incorporates speed, the gradient of the terrain, and cargo load distribution. Hiermann et al. [34] studied the EVRPTW with various types of vehicles that differed in capacities and costs, where each type of vehicle also had a respective cluster of charging stations. Additionally, numerous efforts have been devoted to optimizing the EV routes and the charging station locations simultaneously. Wang and Song [8] investigated an electric vehicle location-routing problem with time windows and multiple charging stations, which aimed to optimize not only the route plan of capacitated EVs but also the charging strategy. Schiffer and Walther [35] designed a location routing framework for EVs to optimize the placement of charging stations and vehicle routing simultaneously.

The literature reviewed above focused on optimizing route planning solely for linehaul customers without considering backhaul customers. Granada-Echeverri et al. [36] realized the importance of backhaul customers in logistics distribution and formulated a model for the EVRP with backhauls. Nevertheless, it is less practical to prioritize linehaul customers and fail to consider the time windows for serving customers. Therefore, in this study, we propose the EVRPTWMB which integrates EVRPTW and mixed backhauls while taking into account the impact of both travel distance and load on electricity consumption. In Table 1, we summarize the related literature by classifying the works based on the main features of the studied problem.

Being classical metaheuristic methods successfully applied to various NP-hard problems, evolutionary algorithms are widely

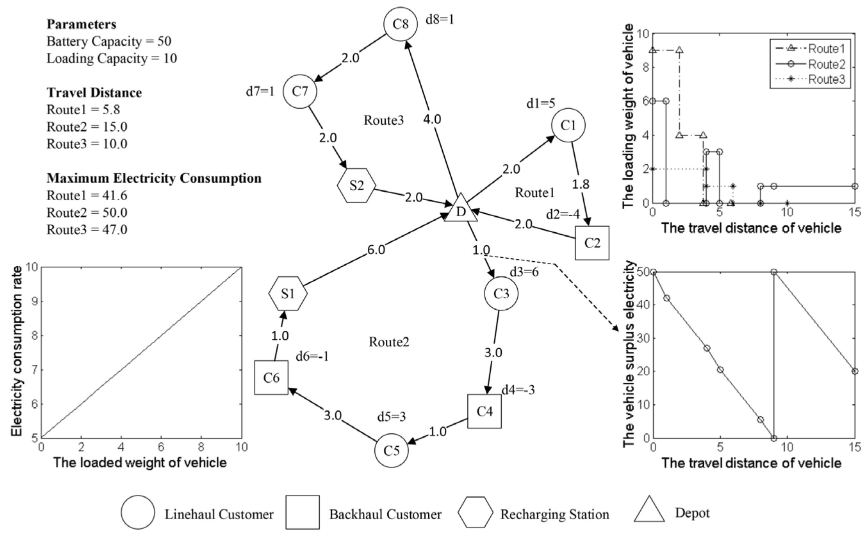

Fig. 1. Illustration of a feasible solution to EVRPTWMB. It is worth noting that route 2 has the longest travel distance. However, since the duration of high loading in route 2 is short, the vehicle maintains a low power consumption rate on average, which leads to a satisfactory total electricity consumption.

Table 1
Overview of the Related Literature.

Reference	Objective Function	Energy Cost Function	Capacity	Fleet composition		Recharge Strategy		Customer			Time window	Load	Location
				Homogeneous	Heterogeneous	Full	Partial	Linehaul	Backhaul	Mixed backhaul			
Conrad & Figliozzi(2011)	1,2,3,4	Linear	√	EVs		√	√	√					
Erdoğan & Miller-Hooks(2012)	2	Linear		AFs		√		√					
Schneider et al.(2014)	1,2	Linear	√	EVs		√		√			√		
Bruglieri et al.(2015)	1,2,3,4	Linear	√	EVs			√	√			√		
Goeke & Schneider(2015)	2,3,6,7,8	Nonlinear	√		EVs and ICEVs		√	√			√	√	
Wang & Song(2015)	2,3,6,7	Linear	√	EVs		√	√	√				√	
Desaulniers et al.(2016)	2	Linear	√	EVs		√	√	√			√		
Keskin & Çatay(2016)	1,2	Linear	√	EVs		√	√			√			
Hiermann et al.(2016)	1,2	Linear	√		EVs	√		√			√		
Schiffer & Walther(2017)	1,2,5,11,12	Linear	√	ICEVs			√	√			√		√
Keskin et al.(2019)	1,2,9,10	Linear	√	EVs			√	√			√		
Granada-Echeverri et al.(2020)	2	Linear	√	EVs		√			√				
Ours	1,2	Linear	√	EVs		√				√	√	√	

Note: The numbers in the second column represent the index of the objective functions as follows. 1: the number of vehicles, 2: the travel distance, 3: recharging cost, 4: waiting cost, 5: station installation cost, 6: total time cost, 7: fuel cost, 8: battery cost, 9: unit time penalty for violated time windows, 10: driver wage, 11: fixed costs of building a freight replenishment facility, and 12: fixed costs of building a combined facility.

used to solve the EVRP as well. Among the representative works, Lin et al. [21] used a genetic algorithm optimization model to achieve an economically and environmentally cost-efficient green transportation scheme for forward and reverse logistics. Masmoudi et al. [22] proposed a hybrid evolutionary algorithm that combined a few diversification mechanisms and a variable neighborhood search algorithm to tackle a dial-a-ride problem with EVs. Roberto et al. [37] presented an improved genetic algorithm by replacing the mutation operation with the λ-opt local search to handle a green vehicle routing problem, which aimed to minimize CO_2 emissions per route. Hiermann et al. [23] designed a sophisticated metaheuristic that integrated a genetic algorithm with local and large neighborhood search to solve a complex EVRP in conjunction with conventional and plug-in hybrid vehicles. Zhang et al. [38] integrated an adaptive large neighborhood search algorithm with several removal operators and a fuzzy simulation method to solve the EVRP with time windows and different charging station locations. In this study, we exploit a memetic algorithm that combines global and local searches to deal with the EVRPTWMB and ameliorate the performance by leveraging the characteristics of the studied problem.

3. Problem formulation

In the EVRPTWMB, commodities are delivered from a central depot to linehaul customers, and collections are picked up from backhaul customers and transported to the central depot. Other assumptions and settings are as follows: For each customer, the geographical location, demand, and time window for service are known beforehand. A fleet of electric vehicles with identical load and battery capacities is employed to carry out the service. During transportation, the electricity consumed by a vehicle is affected by both the travel distance and loaded weight. This implies that a single tour that can serve customers without recharging in the load-independent case may not be able to serve all the assigned customers in the load-dependent case because of the runout of the battery. Meanwhile, it may employ additional vehicles to fulfill the time window constraints that might otherwise be violated, which could increase the acquisition cost of EVs. As such, it would be ideal to properly adjust the service sequence for linehaul and backhaul customers on a longer route to achieve a low loaded weight along this route. Fig. 1 illustrates an example with eight customers, two charging stations, and three routes, where route 2 exhibits a satisfactory total electricity consumption. In addition, when the electricity is insufficient, the vehicle can visit a charging station for replenishment. The time spent recharging is proportional to the amount of electricity consumed. Fig. 2 shows that even if recharging activities affect the arrival times of vehicles to customers, directly minimizing the number of visits to charging stations would not necessarily lead to a better solution. Those hard assumptions and relationships aforementioned make it challenging to attain an optimal solution to the EVRPTWMB.

Let V be the set of customer nodes and F be the set of unique charging stations. Meanwhile, node 0 represents the depot, and we have $V = L \cup B$, where $L = \{1, \ldots, n\}$ and $B = \{n+1, \ldots, n+m\}$ represent linehaul and backhaul customers, respectively. Each

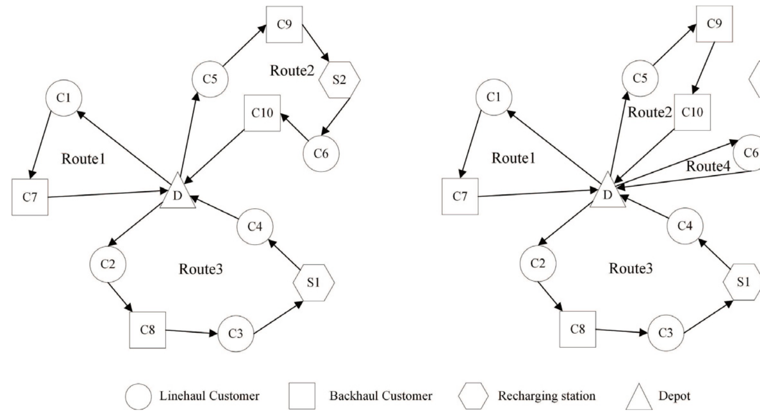

Fig. 2. An exemplary instance for reducing the visits to charging stations. It can be seen from the routes on the left that the second vehicle cannot reach customer 6 without recharging. Thus, the routes on the right show that an additional vehicle must be assigned to serve this customer, which will increase the total travel distance.

Table 2
Parameter and Variable Definitions of the EVRPTWMB.

Factor	Definition
V	Set of customer nodes
V_0	Set of all the customer nodes and the depot
V'	Set of all the customer nodes and the visited charging stations, $V' = V \cup F'$
V_0'	Set of all nodes and the depot, $V' \cup \{0\}$
0	Depot node
L	Set of linehaul customers
B	Set of backhaul customers
F'	Set of visits to charging stations, dummy vertices of set of charging stations F
F_0'	Set of all the visited charging stations and the depot, $F' \cup \{0\}$
A'	All the possible arcs among the customers, the visited charging stations, and the depot
d_{ij}	Travel distance of the vehicle from node i to node j
C	Capacity of the vehicle
t_{ij}	Travel time from node i to node j
τ_i	The arrival time at node i
s_i	Service time for node i
M	A sufficiently large value
g	Charge rate
Q	Maximum storage capacity of the battery
$[e_i, l_i]$	Time window constraint of node i
q_i	Demand of node i
h	Electricity consumption rate of the electric vehicle per unit of distance
w	Electricity consumption rate of the electric vehicle per weight per unit of distance
y_i	Remaining electricity when the vehicle arrives at node i
u_i	Remaining capacity when the vehicle leaves node i
K	Number of all available electric vehicles
x_{ij}	Binary decision variable indicating if the vehicle travels from node i to node j

EV starts and ends its route at the depot, and we use V_0 to represent the set of customers, including the depot. As a charging station might be visited more than once, we use a set of dummy nodes F' to represent and distinguish the multiple visits to the nodes in F. As such, V' represents all the customers and the visited charging stations with $V' = V \cup F'$, F_0' denotes all the visited charging stations and the depot with $F_0' = F' \cup \{0\}$, and V_0' denotes all the nodes including the depot with $V_0' = V' \cup \{0\}$. Consequently, the EVRPTWMB can be defined on a complete directed graph $G = (V_0', A')$, where $A' = \{(i, j) : i, j \in V_0', i \neq j\}$ represents all the possible arcs among the customers, the visited charging stations, and the depot. The distance traveled by an EV from nodes i to j is represented by d_{ij}. The objective of the EVRPTWMB is to minimize the total cost (i.e., travel distance) for serving all linehaul and backhaul customers by optimizing the routes while taking the recharging activity of the EVs into account. The related notation is summarized in Table 2.

In this study, the mathematical model of EVRPTWMB is formulated as a mixed integer program (MIP) as follows. It is also worth noting that, as is commonly accepted for studying VRPTW, a solution with fewer vehicles is always preferable due to the high acquisition cost of EVs, although it was not explicitly expressed in the MIP.

$$\min \sum_{(i,j)\in A'} x_{ij} d_{ij}, \tag{1}$$

$$\sum_{j\in V_0', j\neq i} x_{ij} = 1, \forall i \in V, \tag{2}$$

$$\sum_{j\in V_0', j\neq i} x_{ij} \leq 1, \forall i \in F', \tag{3}$$

$$\sum_{i\in V'_0, j\neq i} x_{ij} - \sum_{i\in V'_0, j\neq i} x_{ji} = 0, \forall j \in V', \tag{4}$$

$$\tau_i + s_i + t_{ij}x_{ij} - M(1 - x_{ij}) \le \tau_j, \forall i \in V_0, \forall j \in V'_0, i \neq j, \tag{5}$$

$$\tau_i + t_{ij}x_{ij} + g(Q - y_i) - (M + gQ)(1 - x_{ij}) \le \tau_j, \quad \forall i \in F', \forall j \in V'_0, i \neq j, \tag{6}$$

$$e_j \le \tau_j \le l_j, \forall j \in V'_0, \tag{7}$$

$$0 \le u_i \le C, \forall i \in V'_0, \tag{8}$$

$$-q_i \le u_i \le C, \qquad \forall i \in B, \tag{9}$$

$$0 \le u_i \le C - q_i, \qquad \forall i \in L, \tag{10}$$

$$0 \le u_i \le u_j + q_i x_{ij} + C(1 - x_{ij}), \quad \forall (i, j) \in A', i \neq j, \tag{11}$$

$$0 \le y_j \le y_i - ((h + w \cdot u_i) \cdot d_{ij})x_{ij} + Q(1 - x_{ij}), \quad \forall i \in V, j \in V'_0, i \neq j, \tag{12}$$

$$0 \le y_j \le Q - ((h + w \cdot u_i) \cdot d_{ij})x_{ij} + Q(1 - x_{ij}), \quad \forall i \in F'_0, j \in V'_0, i \neq j, \tag{13}$$

$$\sum_{i\in V'} x_{i,0} \le K, \tag{14}$$

$$\sum_{j\in V'} x_{0,j} \le K, \tag{15}$$

$$\sum_{i\in V'} x_{i,0} = \sum_{j\in V'} x_{0,j}, \tag{16}$$

$$x_{ij} \in \{0, 1\}, \forall (i, j) \in A', i \neq j. \tag{17}$$

Specifically, constraint (2) ensures that there is exactly only one arc to enter and one arc to leave for each customer. Constraint (3) ensures that the dummy nodes of each charging station cannot be visited more than once. Constraint (4) ensures flow conservation by guaranteeing that the number of incoming arcs for each node equals the number of outgoing arcs. Constraints (5)–(7) ensure that the vehicles must arrive at the customer locations within the given time windows. Constraint (8) ensures that the loaded weight of a vehicle does not exceed the maximum loading capacity. Constraints (9)–(11) describe the loaded weight relationship between nodes i and j. For each node, there is a given demand $q_i > 0$ for the linehaul, $q_i < 0$ for the backhaul, and $q_i = 0$ for the charging stations and the depot. Constraints (12) and (13) refer to changes in the battery state. Constraints (14) and (15) refer to the given maximum number of EVs. In this study, we only consider the case wherein the number of routes is not greater than the EV quantity, and each route corresponds to a vehicle. Constraint (16) ensures that the vehicles must return to the depot.

4. The proposed algorithm

In this section, we first present the framework of the proposed DEMA. Then, we elaborate on the specifically designed genetic operators, selection operator and modification operators that distinguish the DEMA from others.

Algorithm 1 General framework of the proposed DEMA

Input: G (the EVRPTWMB to be solved).
Output: s^* (the best obtained solution of the EVRPTWMB).
1: $P \leftarrow$ Generate an initial population that satisfies all the constraints
2: $\phi, \varphi, \psi \leftarrow$ initialize the penalty coefficients for the violation of the constraints of capacity, time windows, and battery capacity
3: **while** termination criterion not fulfilled **do**
4: $O \leftarrow$ GO-ASM (P) /* Generate an offspring population by operating the parent population through Genetic Operators based on Adaptive Selection Mechanism */
5: $O' \leftarrow$ Repair (O) /* Repair the infeasible individuals in the O */
6: $P' \leftarrow$ ISSD ($P \cup O'$) /* Select the elite individuals from the combined population through objective values and Improved Selection Operators based on Similarity Degree */
7: $P \leftarrow$ TS (P') /* Modify the route information of each individual in the obtained new population through Tabu Search */
8: $s^* \leftarrow$ Update s^* if the best solution of P has a lower objective value
9: **return** s^*

4.1. Framework of the diversity-enhanced memetic algorithm

The overall framework of the proposed DEMA is described in Algorithm 1 and encompasses four main steps. Firstly, an initial population is randomly generated while ensuring that all individuals are feasible. Secondly, genetic operators based on an adaptive selection mechanism (GO-ASM) are proposed to generate an offspring population from the parent population. Specifically, the GO-ASM includes large-step crossover and mutation operators and can adaptively select the appropriate one according to the optimization status, thereby enhancing the robustness of the algorithm. To maintain a high quality offspring population, the infeasible individuals yielded by the crossover or mutation operations will undergo a repair procedure to make it as feasible as possible. Then, we select a portion of the elite feasible individuals from the combination of the parent and offspring populations and take them as part of the new population. The remaining part of the new population is composed of the infeasible individuals selected by the improved selection operators based on their similarity degrees (ISSD) to the combined population. The selection mechanism for infeasible individuals intends to obtain new populations with high diversity and prevent premature convergence. Finally, modification operators are designed for tabu search to further improve the quality of each individual in the new population by adjusting the customer service sequence and recharging scheme. Subsequently, the four operations are repeated for the new population obtained in each iteration until the termination criterion is satisfied, and the best solution is considered as the final solution. In this study, the maximum number of iterations was set as the termination condition.

4.2. Initial population

The individuals that constitute the initial population are generated by progressively constructing feasible routes that satisfy the restrictions on capacity, time windows, and electricity limits. Route construction begins with a random selection of a customer. The remaining customers are then sorted in ascending order of the angle between themselves, the chosen customer, and the depot. Following this order, customers are iteratively inserted into the active route at the position causing the minimal increment in travel distance while respecting the time window constraints. If the insertion of a customer triggers the battery capacity violation, we will insert a customer along with a charging station, where the best position with minimal increment is determined according to the difference between the total travel distance of the route after and before the insertion of the customer and the charging station. For the insertion of a charging station, we primarily consider the

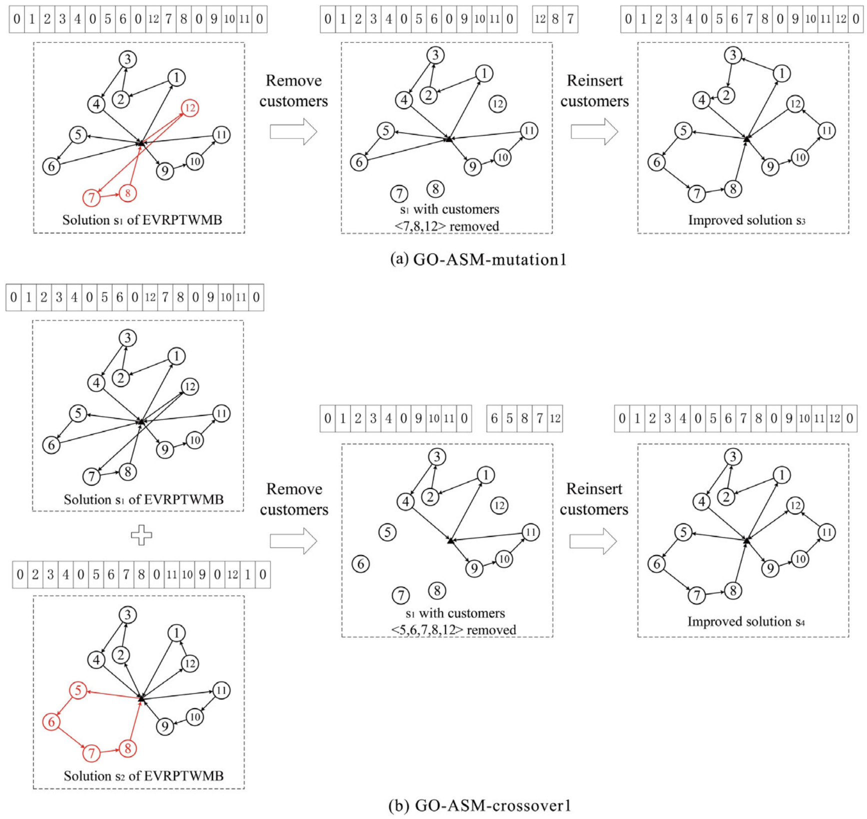

Fig. 3. An example of the proposed genetic operators based on adaptive selection mechanism, where we assume that an individual $\langle 0, 1, 2, 3, 4, 0, 5, 6, 0, 12, 7, 8, 0, 9, 10, 11, 0\rangle$ is obtained for the EVRPTWMB by the proposed DEMA.

first customer at which the vehicle will arrive with potentially negative electricity, and then the best station which increases the travel distance by the smallest possible amount is inserted on the arc between that customer and the preceding one. However, if this insertion fails to meet the time window requirement, then the preceding arcs will be considered in the same way. Finally, when no customer can be inserted owing to a violation of the capacity or time windows, we activate a new route until all customers have been visited.

4.3. Genetic operators based on adaptive selection mechanism

As mentioned in Section 3, the EVRPTWMB has a set of complex and strict constraints. Given the impact of charging decisions and time windows, attaining the global optimal solution is challenging. To alleviate this issue, we propose the GO-ASM to operate the customer nodes, which consists of a crossover operator and a mutation operator, and switches between them based on an adaptive selection mechanism.

To better illustrate the GO-ASM, an example is depicted in Fig. 3, where we assume that the customers are denoted from 1 to 12 and the depot is denoted as 0. Fig. 3(a) shows the process of the mutation operation, that is, GO-ASM-mutation1, designed to remove an inferior part of the individual and then rebuild it. As shown, we assume that a feasible individual consists of four routes, i.e., $\langle 0, 1, 2, 3, 4, 0\rangle$, $\langle 0, 5, 6, 0\rangle$, $\langle 0, 12, 7, 8, 0\rangle$, and $\langle 0, 9, 10, 11, 0\rangle$. The average travel distance of the route $\overline{RD}$ is defined as follows to evaluate the quality of each route for an individual:

$$\overline{RD} = \frac{\sum_{i,j\in\mathcal{R}} x_{ij} d_{ij}}{N_{\mathcal{R}}}, \tag{18}$$

where $\mathcal{R} = \langle 0, c_1, \dots, c_n, 0\rangle$ is one route of the individual and $N_{\mathcal{R}}$ is the total number of customers in $\mathcal{R}$. The inferior route will have a higher $\overline{RD}$ value instead. Therefore, the GO-ASM-mutation1 first selects a route randomly out of the four through a roulette wheel evaluated by $\overline{RD}$, and all the customers in this route are removed. These customers are then sequentially reinserted into the best position one by one after being randomly shuffled. The best insertion position for each customer is determined by inserting the customer at all other possible positions and selecting the one with the lowest cost. In particular, the cost function is defined as Eq. (19), which is adapted from Schneider et al. [7] and applicable to both feasible and infeasible routes. Thus, we also evaluate the quality of a solution by summing the cost of multiple routes. Specifically, $f(R)$ denotes the total travel distance of the route; $P_{cap}(R)$, $P_{tw}(R)$, and $P_{batt}(R)$ correspond to the capacity, time windows, and battery capacity violations, respectively; φ, ϕ, and ψ are the weights for the three violations, where the violations are computed in the same way as Schneider et al. [7] did.

$$f_{cost}(R) = f(R) + \varphi P_{cap}(R) + \phi P_{tw}(R) + \psi P_{batt}(R). \tag{19}$$

Fig. 3(b) presents the process of the crossover operation, i.e., GO-ASM-crossover1. It aims to exploit the superior part of an individual to guide the evolution of another individual. As shown, we assume that s_1 and s_2 are two feasible individuals for the EVRPTWMB instance, and s_2 is better than s_1. Firstly, the GO-ASM-crossover1 selects a route from s_2 using a roulette wheel. In contrast to the GO-ASM-mutation1, the superior route should have a higher probability of being selected in the GO-ASM-crossover1, so we adopt the reciprocal of $\overline{RD}$ as the measurement. Then, all customers of the selected route, i.e., $\langle 5\rangle$, $\langle 6\rangle$, $\langle 7\rangle$, and $\langle 8\rangle$ in the example, are removed from s_1. Simultaneously, the most affected route (i.e., $\langle 0, 12, 7, 8, 0\rangle$) is also removed, where if multiple routes have the same number of removed customers, a random one will be chosen. Finally, the selected route from s_2 is directly inserted into s_1, and other unvisited customers are reinserted individually into the best position after being randomly shuffled. The principle for determining the best insertion position for each customer is the same as that of the GO-ASM-mutation1. In particular, during the customer removal process via the aforementioned genetic operators, any unnecessary charging stations in the sub-routes are removed from the individual as well.

Since the performance of these two genetic operators may vary during the iterative search process, an adaptive selection mechanism is proposed to switch between them automatically. Inspired by Hiermann et al. [23], we assign a weight to each genetic operator, and employ a roulette-wheel scheme measured by the weights to select which one to use. To achieve this, we first assign a score π to each genetic operator to record how well they have performed during each generation. At the beginning of each iteration, the scores for the operators are all set to zero, and they are updated with increments of σ_1, σ_2, and σ_3 (i.e., $\sigma_1 > \sigma_2 > \sigma_3 > 0$) depending on the following conditions:

- σ_1, if the offspring individual is the best in the current population.
- σ_2, if the offspring individual is better than its parent.
- σ_3, if the offspring individual is worse than its parent.

For each genetic operator, at the beginning of the algorithm, we initialize each operator with equal weights. Subsequently, the weight ω_{i+1} is updated after the completion of each iteration as follows,

$$\omega_{i+1} = \begin{cases} \theta\dfrac{\pi}{\tau} + (1-\theta)\omega_i, & \tau \neq 0, \\ \omega_i, & \tau = 0, \end{cases} \tag{20}$$

where τ is the number of times we have attempted to use the operator to generate the offspring individuals in one generation; $\theta \in [0, 1]$ is a factor that reflects the sensitivity of the selection probability to the recent score. If θ is equal to 0, the weight depends only on the past performance. If θ is equal to 1, the weight is determined solely by the recent performance. Algorithm 2 describes the detailed procedure of the GO-ASM.

Algorithm 2 GO-ASM(P)

Input: P (the current population of EVRPTWMB).
Output: O (the offspring population).
while $size(O) \leq PopSize$ **do**
  Select an operator via roulette-wheel according to the weight ω_i
  if the first method is chosen **then**
    $s \leftarrow$ Randomly select one individual from P
    $\overline{RD} \leftarrow$ Calculate the average travel distance for each route in s by Eq. (18)
    $R \leftarrow$ Choose an inferior route from s via roulette-wheel
    $s' \leftarrow$ Remove all customers of R from s and reinsert them to the best insertion position
  else if the second method is chosen **then**
    $s_1, s_2 \leftarrow$ Randomly select two individuals from P
    $\overline{RD} \leftarrow$ Calculate the average travel distance for each route in s_2 by Eq. (18)
    $R \leftarrow$ Choose an superior route from s_2 via roulette-wheel
    $s' \leftarrow$ Remove customers of R and the ones affected by R from s_1, insert R to s_1 and reinsert remaining unvisited customers into the best insertion position
  $O \leftarrow$ Append the new individual s'
  Update the weight ω_i
return O

4.4. Repair operators

The main purpose of the GO-ASM is to remove some customers and reinsert them into other unaffected routes, so as to reduce the number of vehicles in a solution. However, this may cause the generated offspring individuals to violate the capacity and time window constraints. Furthermore, although some unnecessarily visited charging stations are removed concurrently with customer removal, the impact of changing the customer service sequence on electricity consumption is ignored. Therefore, some infeasible individuals may exist in the offspring population. We apply a neighborhood structure composed of several operators to repair the infeasible individuals to improve the offspring population quality further.

We firstly judge which of the three constraints this infeasible route violates, and then perform the appropriate intra-route operation (i.e., move within the route). If the time window constraint is violated, the service sequence of the infeasible customer will be moved forward. For a capacity restriction violation, we move the infeasible customer according to its type. In particular, the linehaul customers near the back end of the route will be exchanged with the backhaul customers near the front end of the route, and vice versa. Additionally, if the electricity is insufficient, we consider the first customer (or depot) at which the vehicle arrives with negative electricity and insert the charging station closest to this customer.

Following the intra-route operations, we apply three inter-route operators (i.e., move between the routes), including *2-opt**, *relocation*, and *exchange* [7]. The *exchange* operator is only for customer nodes, and the *2-opt** and *relocation* operators are both for customers and charging stations. To circumvent an extremely large neighborhood size, we only move geographically close customers, at least one of which must be among the top 50 nodes that are closest to the relocated one [39]. To reduce the computing time required by the repair phase, these operators are performed in turn to search for no more than a predefined number of neighborhoods. Each repair move is evaluated, and the procedure continues until the infeasible individuals become feasible, or no further repair moves are available. When the latter occurs, the best neighborhood solution with the lowest cost according to Eq. (19) is compared with the original individual. If it has a lower cost, then it replaces the original individual. The repair procedure for the offspring population is presented in Algorithm 3.

Algorithm 3 Repair(O)

Input: O (the obtained offspring population through GO-ASM).
Output: O' (the repaired offspring population).
Append feasible individuals in O into O'
for each infeasible individual s in O **do**
  $s^* \leftarrow$ Record the best neighborhood solution of s and initialize it as s itself
  for each operator in the neighborhood structure **do**
    $\mathcal{N}(s) \leftarrow$ Initialize the neighborhood set of s as $\varnothing$
    while $size(\mathcal{N}(s)) \leq MaxNebSize$ **do**
      $s' \leftarrow$ Perform the neighborhood operator on s
      Append s' into $\mathcal{N}(s)$
      if s' becomes feasible **then**
        Append s' into O'
        break
    if s' becomes feasible **then**
      break
    if the best individual of $\mathcal{N}(s)$ is better than s^* **then**
      Update s^* with the best individual
  if s^* is better than s **then**
    Append s^* into O'
  else if s^* is worse than s **then**
    Append s into O'
return O'

4.5. Improved selection operators based on similarity degree

Following the aforementioned repair procedure, infeasible individuals may still exist in the offspring population. To illustrate

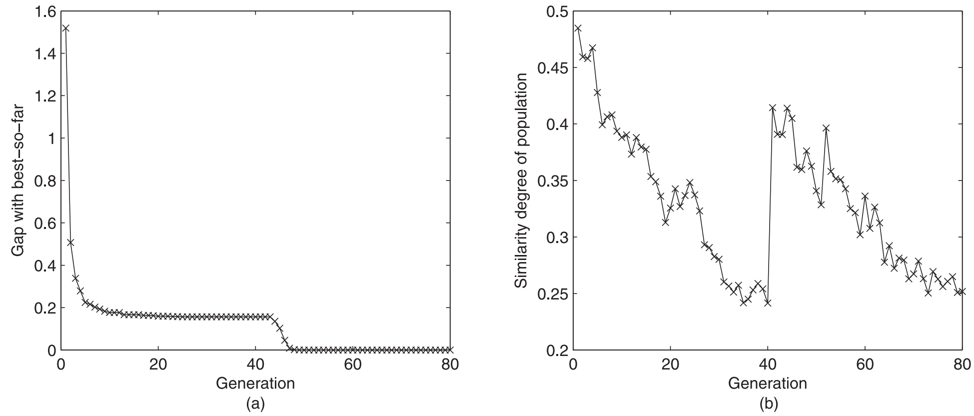

Fig. 4. A case study on the $C201$ instance, where 10 infeasible individuals are added to the population at the 41st generation. (a) the gap between the best solution of each generation and the optimal solution. (b) the similarity degree for the population of each generation.

the effectiveness of the infeasible individuals in terms of population improvement, Fig. 4 depicts the results for a given instance. Within the first 40 generations, the algorithm allows only feasible individuals to survive. In the 41st generation, the algorithm adds 10 infeasible individuals into the population, and maintains a certain proportion of infeasible individuals in later generations. It has been revealed that it is rewarding to enhance population diversity by allowing the survival of infeasible individuals, which may accelerate the algorithm in escaping local optima. Here, we design an independent selection mechanism to preserve favorable infeasible individuals.

Algorithm 4 ISSD($P \cup O'$)

Input: $P \cup O'$ (the combined population).
Output: P' (the new population).
1: $SP_1, SP_2 \leftarrow$ Join all feasible individuals into subpopulation SP_1, and all infeasible individuals into SP_2
2: $SP_1' \leftarrow$ Select the elite individuals from SP_1 based on the objective values
3: $Simi \leftarrow$ Calculate the similarity degree between the infeasible individuals of SP_2 and SP_1' according to Eq. (21).
4: $[s_1, s_2, \ldots] \leftarrow$ Do nondominated sorting on SP_2 based on their cost and $Simi$
5: $SP_2' \leftarrow$ Select a certain proportion of infeasible individuals according to the results of nondominated sorting
6: $P' \leftarrow SP_1' \cup SP_2'$
7: **return** P'

We firstly combine the parent population with the offspring population and then divide this combined population into feasible and infeasible subpopulations. For the subpopulation of feasible individuals, the key idea is to select and preserve individuals with shorter travel distance and fewer vehicles to ensure that the entire population can continuously evolve toward the optimal solution. For the subpopulation of infeasible individuals, it is intuitive that an infeasible individual is useless if it shares many of the same characteristics as feasible individuals. Therefore, the nondominated sorting method is adopted to select individuals that not only have good quality but are also dissimilar to the feasible subpopulation. To evaluate the degree of similarity between an individual and a population, we define the relevant metric as follows:

$$Simi(s_k) = \frac{\sum_{i,j \in V} x_{ij} \times n_{ij}}{\sum_{i,j \in V} x_{ij}}, \tag{21}$$

where x_{ij} refers to whether $arc(i, j)$ is selected in the infeasible individual, and n_{ij} is a counter that counts the number of times $arc(i, j)$ appears in all individuals of the feasible subpopulation.

As described in Algorithm 4, the detailed ISSD consists of four major steps. First, all candidate individuals are divided into SP_1 and SP_2 according to their feasibility. Second, some elite individuals from SP_1 are selected for SP_1' based on their objective values. Third, for all infeasible individuals in SP_2, we calculate the similarity degree $Simi$ using Eq. (21) with respect to SP_1'. Finally, all infeasible individuals are divided into different ranks by nondominated sorting according to their cost and $Simi$ values, and then selected into SP_2'. Specifically, individuals in the first rank are not dominated by any individual in other ranks, whereas individuals in the higher ranks are dominated by the ones in the lower ranks. The selection of infeasible individuals for SP_2' begins at the first rank and proceeds towards the higher ones. If the number of infeasible individuals required by SP_2' is less than those contained in the rank, a roulette-wheel scheme based on the reciprocal of $Simi$ is used to ensure that the individual with a lower similarity degree will be selected with high probability. Afterward, by merging SP_1' and SP_2', a new population is constructed. The size of SP_2' adaptively decreases with the number of iterations, as determined as follows:

$$SP_2' = \vartheta_{\min} - \frac{\vartheta_{\max} - \vartheta_{\min}}{MaxGen} \times iter, \tag{22}$$

where $\vartheta_{\max}$ and $\vartheta_{\min}$ denote the maximal and minimal self-adaption probabilities, and $MaxGen$ and $iter$ represent the predefined maximum generation and current iteration, respectively.

4.6. Modification operators for tabu search

The tabu search process aims to modify the route information of all individuals in the new population obtained through the aforementioned genetic, repair, and selection procedures to enhance their quality further. We apply several neighborhood operators in each iteration to generate a composite neighborhood $\mathcal{N}(s)$ of the individual s. The recharging scheme modification operation adopts the stationInRe operator proposed by Schneider et al. [7] to insert and remove the charging stations. As for customer sequence modification, we apply the *2-opt** operator for inter-route moves, and propose two other operators described below. It is worth noting that a move in tabu search that can reduce the number of employed vehicles or induce a lower cost is superior, which is always adopted for execution.

(i) Time-based relocation: it randomly removes one customer from a route and inserts the removed customer into another route or a different position on the same route,

Table 3
The Parameter Settings of the Proposed DEMA.

Factor	Description	Value
φ, ϕ, ψ	the penalty coefficients corresponding to capacity, time and electric power	[15,10,10]
MaxGen	the maximum generation number of DEMA	100
PopSize	the size of population	10
$\vartheta_{\max}$	the maximum allowable proportion of infeasible individuals in population	0.25
$\vartheta_{\min}$	the minimum allowable proportion of infeasible individuals in population	0.1
$\sigma_1, \sigma_2, \sigma_3$	the increased score of crossover operator selection	[10,5,1]
θ	the sensitive factor of the adaptive selection mechanism	0.1
MaxIter	the maximum iteration number of tabu search for route modification	4*[V]
MaxNebSize	the maximum size of neighborhood	3*[V]
ε_1, ε_2	bounds on the number of iterations for which a move is declared tabu	[5,10]

which must be a superior move. It is important that the position is selected from candidate position sets composed of customers with similar time windows.

(ii) Distance-based exchange: it randomly selects a pair of geographically close customers and swaps them, which is not restricted to the same route but must be a superior move.

To prevent cycling moves, the superior moves that were recently examined are forbidden and inserted into a constantly updated tabu list. Considering that charging station visits have a strong impact on electricity and time windows, we define the attributes of the tabu list in the form (ξ, r, μ, ζ), which indicates that the arc ξ is prohibited to be inserted between μ and ζ in route R_r, where μ and ζ can denote either a charging station or the depot. Additionally, the duration for each attribute remained in the tabu list is randomly sampled from the interval $[\varepsilon_1, \varepsilon_2]$. A particular case is that the tabu status of a move is lifted in advance if a so-called aspiration criterion is satisfied, i.e., a new best yet feasible individual is generated. After each iteration, the best individual with nontabu move is selected from the generated candidate individuals, and the current individual is superseded by it. The tabu search procedure shown in Algorithm 5 for each individual terminates after a certain number of iterations.

Algorithm 5 TS(P)

Input: P' (the new population of EVRPTWMB).
Output: P (the improved population of EVRPTWMB).

```
for each individual s in P' do
    s' ← Record the best neighborhood solution of s and initialize it as s itself
    TaList ← Initialize the tabu list as ∅
    while termination criterion not fulfilled do
        N(s) ← Generate the composite neighborhood of s based on the several neighborhood operators
        s ← Select the best individual with nontabu move from N(s)
        Update TaList
        if s is feasible and superior to s' then
            Update s' and TaList
    Append s' into P
return P
```

5. Experimental results and analysis

In this section, we describe a series of numerical tests conducted to evaluate the performance of the proposed DEMA. The DEMA is compared with the commercial solver CPLEX, and several state-of-the-art heuristic algorithms on 54 instances, two classical benchmarks, and one real case. The DEMA is implemented in Python, and the code is publicly available on GitHub (https://github.com/ci-ke/EVRPTWMB). Due to the randomness of the algorithm, each experiment is independently run 10 times. All experiments are implemented on a PC with a 3.4 GHz CPU, an Intel Core i5-7500, and 16 GB of memory.

In addition, for the parameters of the DEMA and its components, we first randomly select five large-scale instances from the new EVRPTWMB benchmark. Extensive trial experiments are performed to determine the best settings for our experiments, as summarized in Table 3.

5.1. Performance of the DEMA on small-scale instances

In this subsection, we use the commercial solver, CPLEX, to solve the proposed EVRPTWMB problem. The results of DEMA are compared with the optimal (or near-optimal) solutions obtained by CPLEX to evaluate its performance. Considering that no available benchmarks exactly comply with the EVRPTWMB problem, we firstly design a new benchmark by adapting existing ones. The new EVRPTWMB benchmark with 27 large-scale instances is formed by merging the instances adopted in Schneider et al. [7] for the EVRPTW, and the setting rules utilized in Salhi and Nagy [40] for mixed backhauls. Each instance consists of 100 customers, 21 charging stations, and at most 15 vehicles. Meanwhile, every second customer is set as a backhaul customer with the same supply which equals the original demand. Then, we select only the first 10 customers and retain all charging stations and vehicles for each instance to generate 27 small-scale EVRPTWMB instances.

In the MIP formulation presented in Section 3, a set of charging station copies is used to represent multiple visits to the same charging station. Theoretically, there is no restriction on the number of copies for each charging station. However, CPLEX is characterized by searching for the optimal solution directly through the constraints and objective functions. If the number of copies is large, it is computationally intractable for CPLEX to solve the problem in a limited amount of time given the massive search space [19]. Thus, we perform the following trial experiments to determine the number of charging station copies for CPLEX before conducting comparison with the DEMA. For simplicity, we firstly set ten copies for each charging station, and CPLEX is employed to solve the MIP formulation with a time limit of 72,000 s. Then, the number of copies is decreased to the maximum number of copies in each instance based on the results. It is worth noting that even if vehicles can serve all customers without recharging, we need to allow them to visit each charging station at least once. Finally, with the above settings, we use CPLEX with a time limit of 3600s and the DEMA to solve the EVRPTWMB problem and record the corresponding optimal solutions and computation time.

Table 4 provides an overview of the results obtained by the DEMA and CPLEX for solving the studied EVRPTWMB. It can be observed that the results obtained by the DEMA are identical to the optimal solutions of CPLEX for all instances. Furthermore, the runtimes of the DEMA are much shorter than those of CPLEX in most instances, which verifies the fact that the search strategies embedded in the DEMA significantly reduce the search space and improve the search efficiency. Therefore, the results demonstrate that the DEMA can determine the optimal vehicle routes with high efficiency for the small-scale EVRPTWMB.

Table 4
The Objective Values and Runtime of the DEMA and CPLEX Solver on Small-scale EVRPTWMB Instances.

Problem	CPLEX			DEMA				
	m	f	$t(s)$	m	f	Δm	$\Delta f(\%)$	$t(s)$
EC201S	2	152.29	**1.48**	2	152.29	0	0.00	2.04
EC202S	2	152.29	**1.20**	2	152.29	0	0.00	1.65
EC203S	1	143.96	3.69	1	143.96	0	0.00	**2.53**
EC204S	1	141.03	4.95	1	141.03	0	0.00	**3.02**
EC205S	2	152.29	6.67	2	152.29	0	0.00	**5.18**
EC206S	1	152.09	**4.81**	1	152.09	0	0.00	5.09
EC207S	1	144.63	22.28	1	144.63	0	0.00	**6.10**
EC208S	1	152.09	6.94	1	152.09	0	0.00	**3.25**
ER201S	1	228.52	12.34	1	228.52	0	0.00	**4.73**
ER202S	1	223.06	10.72	1	223.06	0	0.00	**5.26**
ER203S	1	199.73	17.16	1	199.73	0	0.00	**6.12**
ER204S	1	181.69	24.58	1	181.69	0	0.00	**7.37**
ER205S	1	234.54	35.48	1	234.54	0	0.00	**5.49**
ER206S	1	192.30	18.75	1	192.30	0	0.00	**6.48**
ER207S	1	173.04	11.42	1	173.04	0	0.00	**8.53**
ER208S	1	173.04	12.73	1	173.04	0	0.00	**7.07**
ER209S	1	212.37	21.89	1	212.37	0	0.00	**8.35**
ER210S	1	212.37	29.72	1	212.37	0	0.00	**6.58**
ER211S	1	199.51	31.44	1	199.51	0	0.00	**7.45**
ERC201S	1	201.34	59.22	1	201.34	0	0.00	**6.15**
ERC202S	1	142.36	48.92	1	142.36	0	0.00	**8.76**
ERC203S	1	142.36	128.55	1	142.36	0	0.00	**10.92**
ERC204S	1	137.78	323.56	1	137.78	0	0.00	**11.06**
ERC205S	1	197.40	188.49	1	197.40	0	0.00	**7.72**
ERC206S	1	198.25	100.64	1	198.25	0	0.00	**6.82**
ERC207S	1	186.97	1820.72	1	186.97	0	0.00	**13.27**
ERC208S	1	141.37	521.09	1	141.37	0	0.00	**11.87**
Avg.	–	–	128.50	–	–	0.00	0.00	6.62

Note: m denotes the number of vehicles, f denotes travel distance, and $t(s)$ denotes runtime. Δm and Δf represent the gaps to the CPLEX solutions in terms of the number of vehicles and travel distance, respectively. Additionally, Avg. is the average value of all the results obtained for the corresponding column.

5.2. Effectiveness of the DEMA components

In this subsection, we conduct experiments on the new EVRPTWMB benchmark described in Section 5.1 to verify the effectiveness of the DEMA and its components for solving large-scale instances. DEGA only accepts the solutions obtained by the designed genetic operators, selection operator, and repair operators, whereas DETS refers solely to the tabu search strategy. To this end, we consider DEGA and DETS as two baselines for comparison with the DEMA.

Table 5 presents the results obtained using the DEMA, DETS, and DEGA. It can be seen that the DEMA achieves the best objective values in all instances. Additionally, comparing the results of DETS and DEGA with those of the DEMA indicates that the designed genetic operators can effectively reduce the number of vehicles, and the modification operators for the tabu search are conducive to significantly improving the quality of the obtained solutions. Furthermore, in each instance, the DEMA obtains the same vehicle number but a different travel distance. Consequently, we use the average objective values as the metric in the third column. The results in the third and fourth columns show that the DEMA exhibits desirable robustness in most instances. Therefore, the DEMA based on the hybrid strategy can effectively search for high-quality solutions, and its components have positive impacts on the search process.

5.3. The effectiveness of the proposed DEMA on benchmarks

In this subsection, we demonstrate the robustness and generalizability of the proposed DEMA for different but relevant problems. Since the VRPMB and VRPTW have both been extensively studied and are close to the EVRPTWMB, we apply the DEMA to solve these two problems and compare them with the respective state-of-the-art algorithms.

The time window is set to [0,M], where M is an infinite value, to adapt the DEMA to solving the VRPMB. Two parameters related to electricity consumption, h and w, are set to zero, and there are no charging stations in the graph. In doing so, the constraints of the time window and electricity will not affect the solution-seeking process of the DEMA. Similarly, when dealing with the VRPTW, we eliminate the electricity constraint by setting h and w to 0. Since all customers only have delivery demands in the VRPTW, the backhaul assumption also needs to be removed.

In terms of the VRPMB, we select 36 instances from the benchmark proposed by Salhi and Nagy [40]. For each instance, the number of customers ranges between 50 and 150. All instances can be divided roughly into three types (T(10%), Q(25%), and H(50%)) according to the proportions of pickup demands among all customers. Furthermore, the proposed DEMA is compared with two state-of-the-art algorithms, i.e., the NSP [41] and SSMOEA [42]. Specifically, the NSP is a heuristic algorithm based on neighborhood search that integrates four neighborhood structures and a precise perturbation algorithm. The SSMOEA is developed based on an evolutionary algorithm that exploits similarity to maintain population diversity. Both algorithms have demonstrated promising capabilities in terms of finding high-quality solutions for solving the VRPMB.

Table 6 records the results of the three algorithms when solving the VRPMB, and the results of the two baselines are obtained from their original work. It can be seen that the gap between the average results of the DEMA and the best-so-far solutions is no more than 3% in each instance, and the gaps between the best results obtained are even smaller. More notably, the DEMA can find the best-so-far solution in a number of instances, and the average gap of average results attained by the DEMA is better than that of SSMOEA. Therefore, the proposed DEMA has a favorable capability for solving the VRPMB.

Regarding the VRPTW, we select the benchmark proposed in Solomon [43]. The instances in the original benchmark can

Table 5
The Objective Values Obtained by the DEMA and its Components on Large-scale EVRPTWMB Instances.

Problem	DEMA				DETS				DEGA			
	m	f	f_{avg}	sd.	m	f	Δm	$\Delta f(\%)$	m	f	Δm	$\Delta f(\%)$
EC201L	**4**	**645.16**	648.50	3.43	5	762.00	1	18.11	9	1309.86	5	103.03
EC202L	**4**	**646.52**	655.25	5.08	4	779.57	0	20.58	8	1285.59	4	98.85
EC203L	**4**	**645.36**	649.12	4.22	4	718.10	0	11.27	9	1357.23	5	110.31
EC204L	**4**	**640.52**	643.23	3.53	4	713.87	0	11.45	8	1367.77	4	113.54
EC205L	**4**	**641.13**	642.44	1.07	4	709.54	0	10.67	9	1186.71	5	85.10
EC206L	**4**	**643.45**	643.21	2.68	4	718.17	0	11.61	7	1002.00	3	55.72
EC207L	**4**	**638.17**	641.00	2.79	4	710.37	0	11.31	9	1145.20	3	79.45
EC208L	**4**	**638.17**	640.54	2.16	4	712.31	0	11.62	7	1105.50	3	73.23
ER201L	**3**	**1322.99**	1327.17	3.75	5	1197.60	2	−9.48	7	2313.27	4	74.85
ER202L	**3**	**1068.51**	1078.44	6.73	5	1015.26	2	−4.98	7	2308.21	4	116.02
ER203L	**3**	**918.36**	922.80	4.53	4	938.62	1	2.21	8	1912.13	5	108.12
ER204L	**2**	**810.53**	813.31	3.72	3	741.25	1	−8.55	6	1438.30	4	77.45
ER205L	**3**	**1031.35**	1039.49	5.15	4	986.81	1	−4.32	7	2183.26	4	111.69
ER206L	**3**	**965.80**	975.53	8.20	4	913.86	1	−5.38	7	2006.87	4	107.79
ER207L	**3**	**829.65**	844.30	14.91	4	826.13	1	−0.42	5	1624.14	2	95.76
ER208L	**2**	**745.71**	748.14	1.42	3	739.47	1	−0.84	5	1516.50	3	103.36
ER209L	**3**	**886.59**	895.65	6.44	4	890.29	1	0.42	7	1787.00	4	101.56
ER210L	**3**	**875.78**	880.63	3.60	4	871.06	1	−0.54	7	1858.48	4	112.21
ER211L	**3**	**794.22**	801.00	4.80	3	854.99	0	7.65	5	1677.61	2	111.23
ERC201L	**4**	**1488.07**	1495.55	5.33	7	1324.17	3	−11.01	8	3058.45	4	105.53
ERC202L	**4**	**1269.16**	1272.99	3.25	6	1163.84	2	−8.30	6	2379.44	2	87.48
ERC203L	**3**	**1119.07**	1035.88	12.07	5	1002.83	2	−10.39	7	2096.32	4	87.33
ERC204L	**3**	**897.31**	906.28	8.34	4	876.48	1	−2.32	8	1647.23	5	83.57
ERC205L	**4**	**1194.03**	1200.45	5.49	5	1156.63	1	−3.13	8	2305.30	4	93.07
ERC206L	**3**	**1230.38**	1245.05	12.35	4	1184.55	1	−3.72	7	2246.71	4	82.60
ERC207L	**3**	**1051.35**	1065.24	11.92	4	1047.24	1	−0.39	7	2197.53	4	109.02
ERC208L	**3**	**871.86**	884.14	10.84	4	857.28	1	−1.67	9	1767.30	6	102.71
Avg.	–	–	–	5.84	–	–	0.93	1.53%	–	–	3.89	95.95%

Note: m denotes the number of vehicles and f denotes travel distance. We provide the gap between the best results obtained by the baselines and those of the DEMA in columns Δm and Δf, respectively. In addition, f_{avg} represents the average value of the DEMA, sd. denotes the standard deviation, and Avg. is the average value of all the results obtained for the corresponding column.

be divided roughly into three types according to the geographical information of customers (i.e., random, clustered, and mixed random and clustered). Given the number of instances of each type, C2, R2, and RC2 are selected for testing, with a total of 27 instances. In this way, we can ensure that the proposed DEMA is tested using different customer distributions. Additionally, the proposed DEMA is compared with two strong algorithms for solving the VRPTW, i.e., HACO [44] and HSFLA [45]. In particular, HACO is developed based on ant colony optimization and three mutation operators, which also incorporate the Pareto optimality for multiobjective optimization. HSFLA is a swarm intelligence heuristic algorithm developed based on memetic evolution, which also integrates neighborhood search.

Table 7 presents the results of the three algorithms for solving the VRPTW, and those for the two baselines are obtained from their original work. It indicates that the DEMA achieves the best-known results for six instances of the C2 type, and it also finds the optimal number of vehicles for each instance with the exception of R207. Moreover, the results in the last two columns show that the gaps between the best and average values obtained by the DEMA and the best-so-far solutions are within an acceptable range for most instances. Although the average gaps of the best and average results obtained by the DEMA (i.e., 1.32% and 2.02%) are slightly inferior to those of HACO and HSFLA, the two baselines are state-of-the-art algorithms designed for this benchmark specifically. Thus, we can conclude that the DEMA can effectively solve the VRPTW problem to a desirable extent.

5.4. Real case

In this subsection, we verify the effectiveness of the proposed DEMA on a real case from JD.com, Inc., which is one of the largest e-commerce companies in China. The case can be found at https://jdata.jd.com/html/detail.html?id=5. For the experiment, 200 customers and 40 charging stations are randomly selected from the original case. The DEMA, DEGA, and DETS are employed to solve this real case. The best results in terms of travel distance are depicted in Fig. 5, where the number of employed vehicles is 15, 20, and 18 respectively. It implies that the DEMA achieves the shortest travel distance with fewer vehicles. In addition, Fig. 6 depicts the optimal solution obtained by the DEMA, where 15 vehicles are dispatched to serve customers, and only 12 stations are used for recharging during the service. We also observe that most of the routes are interlaced. This is because the algorithm tends to reduce the number of vehicles by alternately serving the linehaul and backhaul customers for a vehicle, so that a single route can potentially be extended.

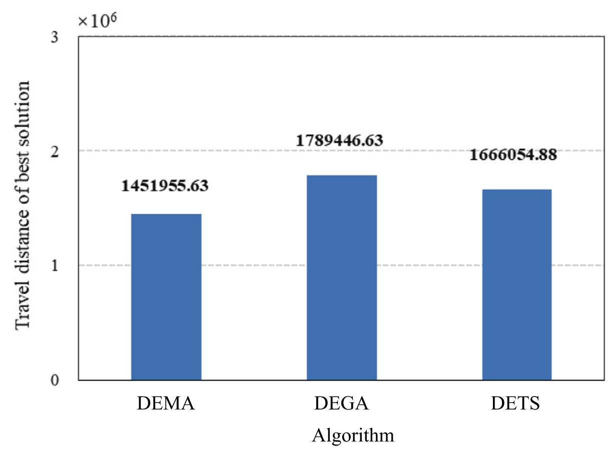

Fig. 5. The average best results of DEMA, DEGA and DETS for JD case.

Table 6
The Objective Values Obtained by the Proposed DEMA and Two Baselines on the Benchmark from Salhi and Nagy [40].

Problem	Best-so-far	NSP		SSMOEA		DEMA			
		f_{best}	f_{avg}	f_{best}	f_{avg}	f_{best}	f_{avg}	Δf_{best}(%)	Δf_{avg}(%)
CMT1H	462	**462**	–	468	481.42	465.07	469.07	0.66	1.53
CMT1Q	490	**490**	–	**490**	504.64	494.24	500.21	0.86	2.08
CMT1T	520	521	–	**520**	525.53	**520.06**	520.25	0.00	0.05
CMT2H	661	**661**	–	668	684.64	664.69	668.40	0.56	1.12
CMT2Q	734	735	–	**734**	751.52	741.36	748.81	1.00	2.02
CMT2T	783	788	–	793	806.62	799.39	804.42	2.09	2.74
CMT3H	701	719	–	717	734.14	706.62	708.59	0.80	1.08
CMT3Q	747	755	–	751	777.84	**747.46**	750.81	0.00	0.50
CMT3T	798	801	–	806	827.82	805.61	807.89	0.95	1.24
CMT4H	829	833	–	838	865	850.24	853.88	2.56	3.00
CMT4Q	915	**915**	–	921	945.52	923.24	926.73	0.90	1.28
CMT4T	993	996	–	**993**	1018.81	1003.19	1005.78	1.03	1.29
CMT6H	555	**555**	–	**555**	568.4	558.68	559.16	0.66	0.75
CMT6Q	555	**555**	–	**555**	565.87	559.32	559.71	0.78	0.85
CMT6T	555	**555**	–	**555**	571.17	**555.43**	559.73	0.00	0.85
CMT7H	900	**900**	–	902	924.42	907.60	929.54	0.84	0.98
CMT7Q	901	**901**	–	905	925.02	902.95	904.96	0.22	0.44
CMT7T	903	**903**	–	907	928.33	**903.05**	910.69	0.00	0.85
CMT8H	865	866	–	**865**	883.08	871.72	877.86	0.78	1.49
CMT8Q	865	866	–	**865**	880.32	873.09	876.88	0.94	1.37
CMT8T	866	**866**	–	**866**	889.72	875.60	878.56	1.11	1.45
CMT9H	1159	**1159**	–	1167	1195.97	1182.31	1185.23	2.01	2.46
CMT9Q	1162	1163	–	1168	1192.46	1187.91	1190.26	2.23	2.43
CMT9T	1164	1165	–	**1164**	1196.64	1193.26	1194.62	2.51	2.63
CMT11H	818	819	–	**818**	836.87	823.92	824.80	0.72	0.83
CMT11Q	939	940	–	941	950.13	943.67	947.18	0.50	0.87
CMT11T	999	**999**	–	1001	1015.51	1006.46	1009.62	0.75	1.06
CMT12H	629	630	–	630	664.6	636.15	641.97	1.14	2.06
CMT12Q	729	**729**	–	740	764.13	731.85	745.06	0.39	2.20
CMT12T	788	**788**	–	790	807.77	**788.72**	790.68	0.00	0.34
CMT13H	1540	**1540**	–	1546	1564.16	1554.14	1557.10	0.92	1.11
CMT13Q	1543	1544	–	1546	1561.22	1552.73	1555.33	0.63	0.80
CMT13T	1544	**1544**	–	1545	1564.59	1562.03	1571.31	1.17	1.77
CMT14H	822	**822**	–	**822**	840.81	823.11	823.96	0.13	0.24
CMT14Q	822	**822**	–	**822**	843.91	823.82	826.24	0.22	0.52
CMT14T	827	**827**	–	828	851.94	839.70	845.31	1.54	2.21
$Gap_{avg.}$	–	0.19%	–	0.41%	2.81%	–	–	0.88%	1.34%

Note: f_{best} denotes the best solutions found, and f_{avg} denotes the average obtained results. We present the gaps between the obtained best solutions and the average results of the DEMA and the best-so-far solutions in columns Δf_{best} and Δf_{avg}, respectively. In addition, $Gap_{avg.}$ is the average gap of all the results obtained for the corresponding column.

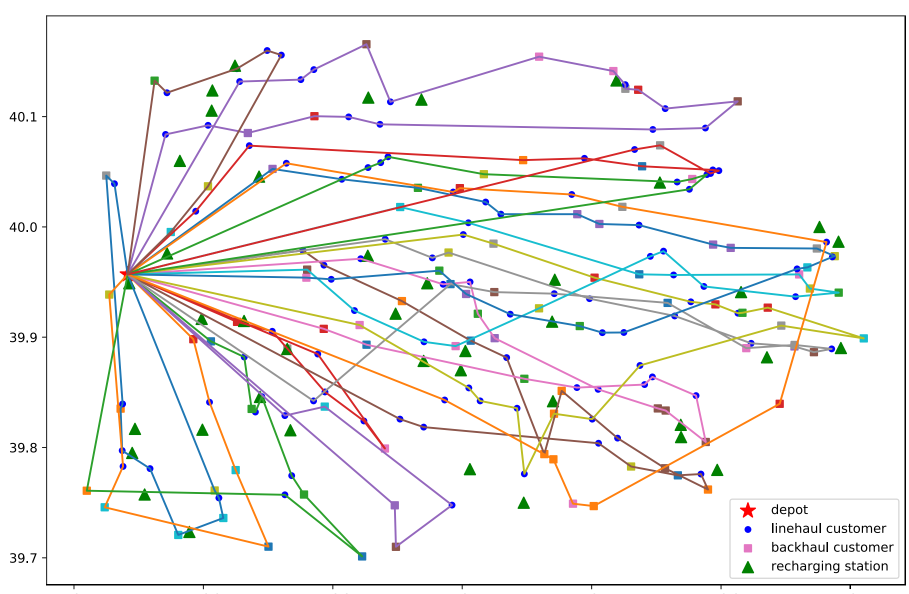

Fig. 6. The optimal routes obtained by DEMA on JD case.

Table 7
The Objective Values Obtained by the Proposed DEMA and Two Baselines on the Benchmark from Solomon [43].

Problem	Best-so-far		HACO		HSFLA		DEMA					
	m	f	m	f	m	f	m	f	f_{avg}	Δ_m	Δf_{best}(%)	Δf_{avg}(%)
C201	3	591.56	**3**	**591.56**	**3**	**591.56**	**3**	**591.56**	**591.56**	0	0.00	0.00
C202	3	591.56	**3**	**591.56**	**3**	**591.56**	**3**	**591.56**	**591.56**	0	0.00	0.00
C203	3	591.17	**3**	**591.17**	**3**	**591.17**	**3**	**591.17**	**591.17**	0	0.00	0.00
C204	3	590.60	**3**	598.72	**3**	**590.60**	**3**	591.45	595.14	0	0.14	0.77
C205	3	588.16	**3**	**588.16**	**3**	588.88	**3**	588.88	588.88	0	0.12	0.12
C206	3	588.49	**3**	588.9	**3**	**588.49**	**3**	**588.49**	**588.49**	0	0.00	0.00
C207	3	588.29	**3**	592.46	**3**	**588.29**	**3**	**588.29**	**588.29**	0	0.00	0.00
C208	3	588.32	**3**	**588.32**	**3**	**588.32**	**3**	**588.32**	**588.32**	0	0.00	0.00
R201	4	1252.37	**4**	**1252.37**	**4**	**1252.37**	**4**	1258.24	1262.77	0	0.47	0.83
R202	3	1191.7	4	1091.21	**3**	**1191.7**	**3**	1203.96	1215.31	0	1.03	1.98
R203	3	939.5	4	937.56	**3**	**939.5**	**3**	945.78	957.91	0	0.67	1.96
R204	2	825.52	3	794.56	**2**	**825.52**	**2**	850.98	856.34	0	3.08	3.73
R205	3	994.43	**3**	**994.43**	**3**	**994.43**	**3**	1026.47	1028.10	0	3.22	3.39
R206	3	906.14	**3**	**906.14**	**3**	**906.14**	**3**	913.68	920.97	0	0.83	1.64
R207	2	890.61	3	825.48	**2**	**890.61**	3	811.51	817.06	1	−8.88	−8.26
R208	2	726.50	**2**	**726.50**	**2**	726.82	**2**	733.54	738.73	0	0.97	1.68
R209	3	909.16	**3**	**855**	**3**	909.16	**3**	912.78	918.27	0	6.76	7.40
R210	3	939.37	**3**	**939.37**	**3**	**939.37**	**3**	954.54	956.57	0	1.61	1.83
R211	2	885.71	4	761.5	**2**	**885.71**	**2**	930.83	935.57	0	5.09	5.63
RC201	4	1406.94	**4**	1423.7	**4**	**1406.94**	**4**	1429.12	1445.48	0	1.58	2.74
RC202	3	1365.64	4	1263.53	**3**	**1365.64**	**3**	1460.79	1473.32	0	6.97	7.88
RC203	3	1049.62	4	1048	**3**	**1049.62**	**3**	1084.49	1089.57	0	3.32	3.81
RC204	3	798.46	**3**	799.52	**3**	**798.46**	**3**	807.11	815.94	0	1.08	2.19
RC205	4	1297.65	**4**	1410.3	**4**	**1297.65**	**4**	1328.10	1342.65	0	2.35	3.47
RC206	3	1146.32	**3**	**1146.32**	**3**	**1146.32**	**3**	1164.67	1190.70	0	1.60	3.87
RC207	3	1061.14	4	1140.67	**3**	**1061.14**	**3**	1062.05	1105.67	0	0.09	4.20
RC208	3	828.14	4	828.71	**3**	**828.14**	**3**	856.95	859.58	0	3.48	3.80
$Gap_{avg.}$	–	–	0.37	−0.80%	0.00	0.24%	–	–	–	0.04	1.32%	2.02%

Note: m denotes the number of vehicles and f denotes travel distance. The DEMA obtains the same vehicle number but different travel distance in each instance, and the average value is presented by f_{avg}. Δm represents the gap to the best-so-far solutions in terms of the number of vehicles. We present the gaps between the obtained best solutions and the average results of the DEMA and the best-so-far solutions in columns Δf_{best} and Δf_{avg}, respectively. In addition, $Gap_{avg.}$ is the average gap of all the results obtained for the corresponding column.

In summary, the proposed DEMA demonstrates competitive performance in terms of its solution quality and robustness when it is used to address the EVRPTWMB and other relevant variants.

6. Conclusion

In this study, we investigate a challenging variant of the electric vehicle routing problem with time windows and mixed backhauls, called the EVRPTWMB. To address this problem, we proposed a diversity-enhanced memetic algorithm (DEMA) that integrates three novel operators: (1) genetic operation; (2) selection of both feasible and infeasible individuals; and (3) modification of route information. The experimental results obtained from 27 small-scale instances, 27 large-scale instances, and two classical benchmarks show that the DEMA achieves competitive performance in solving the proposed EVRPTWMB problem and its variants, including VRPTW and VRPMB. Furthermore, a study on a real JD case demonstrates that the proposed DEMA achieves a satisfactory solution quality in solving a large-scale EVRPTWMB with up to 200 customers and 40 charging stations. In the future, we plan to study customers with simultaneous linehaul and backhaul properties. In addition, we will leverage deep (reinforcement) learning to solve the routing problems in this study [46–48].

CRediT authorship contribution statement

Jianhua Xiao: Conceptualization, Methodology. **Jingguo Du:** Software, Writing – original draft. **Zhiguang Cao:** Writing – review & editing. **Xingyi Zhang:** Methodology. **Yunyun Niu:** Conceptualization, Supervision.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the National Natural Science Foundation of China [grant numbers 62072258, 61772290, 61872325, 62172373]; the Fundamental Research Funds for the Central Universities, China [grant number 63192616, 2652019028]; the China Scholarship Council [grant number 201806405004]; and the Special Foundation for Philosophy and Social Science Laboratories of Ministry of Education of China [grant number ZX20220103].

References

[1] R. Dekker, J. Bloemhof, I. Mallidis, Operations research for green logistics can overview of aspects, issues, contributions and challenges, European J. Oper. Res. 219 (3) (2012) 671–679.

[2] P. Kleindorfer, K. Singhal, L. Wassenhove, Sustainable operations management, Prod. Oper. Manage. 14 (4) (2005) 482–492.

[3] W. Feng, M. Figliozzi, An economic and technological analysis of the key factors affecting the competitiveness of electric commercial vehicles: a case study from the USA market, Transp. Res. C 26 (2013) 135–145.

[4] A.G.H. Kek, R.L. Cheu, Q. Meng, Distance-constrained capacitated vehicle routing problems with flexible assignment of start and end depots, Math. Comput. Modelling 47 (1–2) (2008) 140–152.

[5] L. Zhen, Z.H. Xu, C.L. Ma, L.Y. Xiao, Hybrid electric vehicle routing problem with mode selection, Int. J. Prod. Res. 58 (2) (2020) 562–576.

[6] O. Sassi, A. Oulamara, Electric vehicle scheduling and optimal charging problem: complexity, exact and heuristic approaches, Int. J. Prod. Res. 55 (2) (2017) 519–535.

[7] M. Schneider, A. Stenger, D. Goeke, The electric vehicle-routing problem with time windows and recharging stations, Transp. Sci. 48 (4) (2014) 500–520.

[8] L.Y. Wang, Y.B. Song, Multiple charging station location-routing problem with time window of electric vehicle, J. Eng. Technol. Rev. 8 (5) (2015) 190–201.
[9] M. Bruglieri, F. Pezzella, O. Pisacane, S. Suraci, A variable neighborhood search branching for the electric vehicle routing problem with time windows, Electron. Notes Discrete Math. 47 (2015) 221–228.
[10] M. Keskin, B. Çatay, Partial recharge strategies for the electric vehicle routing problem with time windows, Transp. Res. C 65 (2016) 111–127.
[11] M. Schiffer, G. Walther, Strategic planning of electric logistics fleet networks: a robust location-routing approach, Omega 80 (2018) 31–42.
[12] M. Keskin, G. Laporte, B. Çatay, Electric vehicle routing problem with time-dependent waiting times at recharging stations, Comput. Oper. Res. 107 (2019) 77–94.
[13] G. Macrina, L.D.P. Pugliese, F. Guerriero, G. Laporte, The green mixed fleet vehicle routing problem with partial battery recharging and time windows, Comput. Oper. Res. 101 (2019) 183–199.
[14] W.C. Jie, J. Yang, M. Zhang, Y.X. Huang, The two-echelon capacitated electric vehicle routing problem with battery swapping stations: formulation and efficient methodology, European J. Oper. Res. 272 (3) (2019) 879–904.
[15] M. Goetschalckx, C. Jachobs-Blecha, The vehicle routing problem with backhauls, European J. Oper. Res. 42 (1) (1989) 39–51.
[16] P. Toth, D. Vigo, A heuristic algorithm for the symmetric and asymmetric vehicle routing problems with backhauls, European J. Oper. Res. 113 (3) (1999) 528–543.
[17] A.C. Wade, S. Salhi, An investigation into a new class of vehicle routing problem with backhauls, Omega 30 (6) (2002) 479–487.
[18] F. Belmecheri, C. Prins, F. Yalaoui, L. Amodeo, Particle swarm optimization algorithm for a vehicle routing problem with heterogeneous fleet, mixed backhauls, and time windows, J. Intell. Manuf. 24 (2013) 775–789.
[19] A. Montoya, C. Guéret, J.E. Mendoza, J.G. Villegas, The electric vehicle routing problem with nonlinear charging function, Transp. Res. B 103 (2017) 87–110.
[20] X.R. Zuo, Y.Y. Xiao, M. You, I. Kaku, Y. Xu, EA new formulation of the electric vehicle routing problem with time windows considering concave nonlinear charging function, J. Clean. Prod. 236 (2019) 117687.
[21] C.H. Lin, K.L. Choy, G.T.S. Ho, T.W. Ng, A genetic algorithm-based optimization model for supporting green transportation operations, Expert Syst. Appl. 41 (7) (2014) 3284–3296.
[22] M.A. Masmoudi, M. Hosny, E. Demir, K.N. Genikomsakis, N. Cheikhrouhou, The dial-a-ride problem with electric vehicles and battery swapping stations, Transp. Res. E 118 (2018) 392–420.
[23] G. Hiermann, R.F. Hartl, J. Puchinger, T. Vidal, Routing a mix of conventional, plug-in hybrid, and electric vehicles, European J. Oper. Res. 272 (1) (2019) 235–248.
[24] M. Keskin, B. Çatay, A matheuristic method for the electric vehicle routing problem with time windows and fast chargers, Comput. Oper. Res. 100 (2018) 172–188.
[25] J. Hof, M. Schneider, D. Goeke, Solving the battery swap station location-routing problem with capacitated electric vehicles using an AVNS algorithm for vehicle-routing problems with intermediate stops, Transp. Res. B 97 (2017) 102–112.
[26] Z.G. Cao, H.L. Guo, J. Zhang, D. Niyato, U. Fastenrath, Finding the shortest path in stochastic vehicle routing: a cardinality minimization approach, IEEE Trans. Intell. Transp. Syst. 17 (6) (2016a) 1688–1702.
[27] Z.G. Cao, H. Guo, J. Zhang, D. Niyato, U. Fastenrath, Improving the efficiency of stochastic vehicle routing: a partial lagrange multiplier method, IEEE Trans. Veh. Technol. 65 (6) (2016b) 3993–4005.
[28] Z.G. Cao, H.L. Guo, W. Song, K.Z. Gao, Z.H. Chen, L. Zhang, et al., Using reinforcement learning to minimize the probability of delay occurrence in transportation, IEEE Trans. Veh. Technol. 69 (3) (2020a) 2424–2436.
[29] Z.G. Cao, H.L. Guo, W. Song, K.Z. Gao, L.J. Kang, X.X. Zhang, et al., Improving the performance of transportation networks: a semi-centralized pricing approach, IEEE Trans. Intell. Transp. Syst. 22 (10) (2020b) 6353–6364.
[30] R.G. Conrad, M.A. Figliozzi, The recharging vehicle routing problem, in: Proceedings of the 2011 Industrial Engineering Research Conference, Nevada, Reno, 2011, pp. 1–8.
[31] S. Erdoğan, E. Miller-Hooks, A green vehicle routing problem, Transp. Res. E 48 (1) (2012) 100–114.
[32] G. Desaulniers, F. Errico, S. Irnich, M. Schneider, Exact algorithms for electric vehicle-routing problems with time windows, Oper. Res. 64 (6) (2016) 1388–1405.
[33] D. Goeke, M. Schneider, Routing a mixed fleet of electric and conventional vehicles, European J. Oper. Res. 245 (1) (2015) 81–99.
[34] G. Hiermann, J. Puchinger, S. Ropke, R.F. Hartl, The electric fleet size and mix vehicle routing problem with time windows and recharging stations, European J. Oper. Res. 252 (3) (2016) 995–1018.
[35] M. Schiffer, G. Walther, The electric location routing problem with time windows and partial recharging, European J. Oper. Res. 260 (3) (2017) 995–1013.
[36] M. Granada-Echeverri, L.C. Cubides, J.O. Bustamante, The electric vehicle routing problem with backuals, Int. J. Ind. Eng. 10 (1) (2020) 135–152.
[37] D. Roberto, S. Mauceri, P. Carroll, F. Pallonetto, A genetic algorithm for a green vehicle routing problem, Comput. Oper. Res. 64 (2018) 65–74.
[38] S. Zhang, M.Z. Chen, W.Y. Zhang, X.Y. Zhuang, Fuzzy optimization model for electric vehicle routing problem with time windows and recharging stations, Expert Syst. Appl. 145 (2021) 113123.
[39] Y. Nagata, O. Bräysy, W. Dullaert, A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows, Comput. Oper. Res. 37 (4) (2010) 724–737.
[40] S. Salhi, G. Nagy, A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling, J. Oper. Res. Soc. 50 (1999) 1034–1042.
[41] Y. Jun, B.I. Kim, New best solutions to VRPSPD benchmark problems by a perturbation based algorithm, Expert Syst. Appl. 39 (5) (2012) 5641–5648.
[42] A. García-Nájera, J.A. Bullinaria, M.A. Gutiérrez-Andrade, An evolutionary approach for multi-objective vehicle routing problems with backhauls, Comput. Ind. Eng. 81 (2015) 90–108.
[43] M.M. Solomon, Algorithms for the vehicle routing and scheduling problems with time window constraints, Oper. Res. 35 (2) (1987) 254–265.
[44] H.Z. Zhang, Q.W. Zhang, L. Ma, Z.Y. Zhang, Y. Liu, A hybrid ant colony optimization algorithm for a multi-objective vehicle routing problem with flexible time windows, Inform. Sci. 490 (2019) 166–190.
[45] J.P. Luo, X. Li, M.R. Chen, H.W. Liu, A novel hybrid shuffled frog leaping algorithm for vehicle routing problem with time windows, Inform. Sci. 316 (2015) 266–292.
[46] L. Xin, W. Song, Z.G. Cao, J. Zhang, Step-wise deep learning models for solving routing problems, IEEE Trans. Ind. Inform. 17 (7) (2020) 4861–4871.
[47] Y.X. Wu, W. Song, Z.G. Cao, J. Zhang, A. Lim, Learning improvement heuristics for solving routing problems, IEEE Trans. Neural Netw. Learn. Syst. 33 (9) (2022) 5057–5069.
[48] L. Xin, W. Song, Z.G. Cao, J. Zhang, Multi-decoder attention model with embedding glimpse for solving vehicle routing problems, in: Proceedings of 35th AAAI Conference on Artificial Intelligence (AAAI), 2021, pp. 12042-12049.