IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received May 30, 2020, accepted June 12, 2020, date of publication June 17, 2020, date of current version July 1, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3003000

The Electric Vehicle Routing Problem With Time Windows and Multiple Recharging Options

HUITING MAO[1], JIANMAI SHI[1], YUZHEN ZHOU[1], AND GUOQING ZHANG[2]
[1]Science and Technology on Information Systems Engineering Laboratory, College of System Engineering, National University of Defense Technology, Changsha 410073, China
[2]Faculty of Engineering, University of Windsor, Windsor, ON N9B 3P4, Canada

Corresponding author: Jianmai Shi (jianmaishi@gmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 71771215, and in part by the Natural Science Fund of Distinguished Young Scholars in Hunan under Grant 2018JJ1035.

ABSTRACT Driven by environmental concerns and new regulations, electric vehicles (EVs) are increasingly becoming popular for package delivery. However, due to their limited driving range, the EV has to be recharged during the route in many situations. A new variant of the electric vehicle routing problem with time windows is investigated through integrating decisions on multiple recharging options, which are partial recharging and battery swapping. A mixed integer programming model is developed to formulate the problem. An improved ant colony optimization (ACO) algorithm hybridized with insertion heuristic and enhanced local search is designed to solve the problem. Also, a new probabilistic selection model in ACO is proposed by integrating the impact of both distances and time windows. Computational experiments based on open data source is utilized to validate the performance of the algorithm, and the results indicate that the newly designed insertion heuristic and local search strategies improve the efficiency for solving the problem. The results for all the instances under the strategy of multiple recharging options are compared with those under strategies of partial recharging and battery swapping, which shows that the former strategy can help saving costs for most of the situations.

INDEX TERMS Electric vehicle, vehicle routing problem, ant colony optimization, partial recharge, battery swapping.

I. INTRODUCTION

Current research in transportation and mobility operations is strongly motivated by the concern about rising greenhouse gas (GHG) emission and climate change. According to statistics, about 20-25% of global energy consumption and CO2 emission are due to the transportation system, in which road transport is a major contributor with 75% share (White Paper on Transport, 2011).

In order to reduce the pollution caused by the usage of petroleum-based fuel, governments and companies began to use alternative fuel vehicles (AFV) in their fleets. AFVs use alternative, greener fuel sources, such as bio-diesel, ethanol, electricity, methanol, hydrogen, natural gas (liquid (LNG) or compressed (CNG)) and propane. Among these alternative fuel vehicles, electric vehicles (EVs) are significantly attractive. Since electricity can be obtained from nuclear, water and other clean energy sources, it can reduce the consumption of non-renewable resources such as coal and oil. Apart from this, according to the report by NASA [1], the usage of EVs have many other advantages. EVs have lower carbon emissions, and pure EVs have zero emissions, which greatly improve air quality. EVs have lower operational costs, and they require less maintenance compared to traditional vehicles. And due to regenerative braking [1], the brakes last longer and the cost per mile is much lower. EVs can accelerate quickly and produce minimal noise. Therefore, they can be used in the densely populated area to meet the noise limits.

Although all the benefits associated with EVs are sustainable for the environment, EVs have some inconveniences for users in practical operations. The driving range of the EV is shorter than that of the combustion-engine vehicles. The number and distribution of public charging stations are limited. The recharging time is very long, while the refueling time of traditional vehicles can be neglected. Especially when the delivery time windows are predetermined by customers, the traditional recharging strategy that the battery must be recharged into the full state consumes longer recharging time,

The associate editor coordinating the review of this manuscript and approving it for publication was Keli Xiao.

which has a significant impact on the routing of EVs. One solution to this issue is partial recharging strategy allowing the battery to be recharged into the required level, which takes less recharging time [2]. Another solution is to utilize the Battery Swapping Stations (BSSs). At a BSS, the battery of the EV can be replaced with a new fully recharged battery in only about 5 minutes [3]. The savings in time is crucial for efficiently routing of EVs in urban package distribution where the time windows are usually narrow. Besides, BSSs enable a longer battery life compared to fast recharging. However, BSSs raise additional issues in building the infrastructure, battery design and compatibility, battery degradation and ownership that lead to a higher cost for logistic firms [2]. Although the BSSs are expensive, the convenience and time conservation are attractive for many commercial companies. And in recent years, for protecting the environment, the government provide subsidy for the electric vehicles and recharging infrastructures. Thus, many cities in China have built sufficient recharging networks which can supply the multiple recharging options. Most of these recharging stations are operated by the third parties, which can provide recharging service for EVs. Wang and Lin [4] considered the establishment of a convenient recharging network which supplies multiple types of recharging stations to meet the varied demands of EV users. Wang *et al.* [5] investigated the joint optimal policy for a subsidy on electric vehicles and infrastructure construction in a highway network.

The Electric Vehicle Routing Problem with Time Windows (EVRPTW) is an important extension to classical VRP, where the EVs in the fleet can be recharged. In this paper, we investigate the situation that all the stations are equipped with two recharging technologies, battery swapping and fast recharging, and also EVs can be partially recharged according to their requirement. The two optional recharging technologies can improve the flexibility of route planning, while the optimization of this new variant becomes more complex. To our best knowledge, the EVRP with time windows and multiple recharging modes (EVRP-TW&MC) has not been solved efficiently. To include the new decision variables and constraints, a new mixed integer programming model is developed. The ant colony algorithm is an optimization method inspired by the search process of ants, which is in accordance with the routing process of EVs. Thus, we designed an improved ant colony algorithm through combining with the insertion heuristic and local search strategies. To improve the search efficiency for selecting customers to visit, we proposed a probabilistic selection model integrating the impact of both distances and time windows. Computational experiments are conducted based on the data set reported by Schneider *et al.* [6], and the results confirm the efficiency of the improved ant colony algorithm.

The reminder of the paper is organized as follows: Section 2 summarizes the related literature. Section 3 presents the mathematical model. The algorithm is shown in Section 4. Section 5 reports the computational experiments. Finally, conclusions are discussed in Section 6.

II. LITERATURE REVIEW

The Vehicle Routing Problem (VRP) introduced by Dantzig and Ramser [7] is a classical NP-hard combinatorial optimization problem. Over the years, many valuable extensions to VRP have been proposed through including different real-world constraints [8]. Electric VRP (EVRP) is one of the critical extensions and grain increasing attention in recent years, due to the benefits of EV bringing to the environment.

Schneider *et al.* [6] studied the Electric Vehicle Routing Problem with Time Windows, where the traditional recharging option is applied to enable the EV completing a longer route. They presented a hybrid heuristic combining the variable neighborhood search with Tabu search, and designed 56 instances based on the classical Solomen data set to test the algorithm. Ding *et al.* [9] investigated the EVRP with the possibility of strategic partial recharging, in which it assumes that only one EV could be recharged at a recharging station at the same time. A heuristic method, consisted of variable neighborhood search and Tabu search, with simple charging time adjustment processes is proposed to solve large instances of the problem. Keskin and Çatay [2] developed an adaptive large neighborhood search (ALNS) algorithm for the EVRPTW where the partial recharging strategy is suggested. The results of experiments confirmed the superiority of partial recharging to full recharging at minimizing the total distance.

Considering the fleet size and composition, Hiermann *et al.* [10] introduced the Electric Fleet Size and Mix Vehicle Routing Problem with Time Windows and Recharging Stations and proposed a hybrid heuristic based on an ALNS with an embedded local search and labeling procedure. Catay and Keskin [11] studied the EVRPTW with single charge and quick charge cases respectively, and developed two mathematical models for these two situations. The experiments on small size instances indicated that the quick charging strategy can reduce the fleet size. Montoya *et al.* [12] investigated the EVRP with nonlinear charging function, and designed a hybrid metaheuristic integrating the iterated local search and the heuristic concentration to solve the problem. Hiermann *et al.* [13] proposed an algorithm combining genetic algorithm with LNS to solve the same problem. Macrina *et al.* [14] introduced a meta-heuristic based on iterated local search (ILS) to solve the problem with partial recharge and time windows. Macrina *et al.* [14] considered the mixed use of electrical and conventional vehicles and partial recharging, and solved it by LNS algorithm. Koç *et al.* [15] extended EVRP with nonlinear charging function by considering shared charging stations, and a multi start heuristic is designed to minimize the total cost. Cortés-Murcia *et al.* [16] investigated the EVRPTW with partial recharge and satellite customers which aimed to minimize the charging time at the charging station.

Most of the above works investigated the EVRP with one charging strategy. The problem becomes more complex when multiple charging technologies are considered. Felipe *et al.* [17] introduced GVRP with partial recharge and three different traditional charging technologies which varied in charging speed. Keskin and Çatay [18] studied the EVRPTW with partial recharge and considered three recharging configurations, and solved it by ALNS combined with an exact method. These two works focused on the combine of charging technologies with different charging speed and costs. Recently, Verma [3] presented the EVRPTW with both traditional recharging and battery swapping options. However, it is restricted that the battery must be recharged fully, that is, partial recharging is not allowed. Until now, the EVRP with partial recharging and battery swapping options is still not addressed.

Ant colony optimization (ACO) algorithm proposed by Dorigo *et al.* [19] is inspired by the food-seeking behaviors of ant colonies in nature, which is an important algorithm for solving the VRP and its extensions [20], [21]. Yu *et al.* [22] showed the competitiveness of the hybrid ACO algorithm embed with Tabu Search for VRPTW through the test on the Solomon's data set. Yu *et al.* [23] proposed an improved ACO algorithm to solve VRP. They suggested a new strategy to update the pheromone called ant-weight strategy which thinking of both the local and global feature and applied a mutation operation in a random fashion. Zhang and Tang [24] presented a hybrid ACO combining the ACO with Scatter Search for solving VRP. Balseiro *et al.* [25] employed two ant colonies to minimize the number of vehicles and total travelling time respectively for the Time-Dependent VRPTW. To prevent the search process from getting trapped in the local optima, Ding *et al.* [26] introduced a disaster operator and λ-interchange mechanism into ACO to solve the VRPTW. Yin and Chuang [27] proposed an adaptive memory artificial bee colony algorithm for the green vehicle routing problem, which shows higher efficiency compared to the Tabu search. Cao *et al.* [28] developed an artificial immune system based hybrid algorithm for the VRP with interval demands. Recently, the ACO algorithm is used as an alternative solution tool to solve EVRP in the work of Zhang *et al.* [29]. They established a model of EVRP seeking to minimize the energy consumption and evaluated the effectiveness of the proposed ACO algorithm. Li *et al.* [30] proposed an ACO algorithm to solve multi-depot green VRP by improving the process of pheromone update.

In the literature, the long recharging time of the traditional recharge solution is still a significant problem for the EVRPTW. Recently, partial recharging and battery swapping strategies are thought as effective ways because more customers with time windows can be visited due to the savings on recharging time, which may result in a cost saving. Also, the partial recharge strategy can reduce the recharging cost. Thus, it becomes very important for both transportation firms and academic researchers to investigate the EVRP under multiple recharging options.

III. MATHEMATICAL MODEL

In this paper, EVRP-TW&MC concerns a set of customers with known demands, delivery time windows and service durations. A homogeneous fleet of EVs with fixed loading capacities and limited driving ranges is used for the package delivery. During a travelling route, the EV consumes the battery charge proportionally with the distance traversed and may visit a station in order to continue its route. At each station, two recharging options are provided. The first one is to recharge the battery partially so that the EV can visit the following customers. The advantage of this option is that the driver can recharge the battery according to the energy requirement in the following travel, which is cost efficiency. But the recharging process is time consuming and the EV should stay at the station for a longer time. The second one is battery swapping whose operation time is very short compared to the travel time of the route. In this way, regardless of the charge level, the current battery would be replaced by a fully charged one. Thus, the cost is higher than the former option.

In EVRP-TW&MC, all EVs are assumed to depart from the depot with a fully charged battery and return to the depot. Let $V = \{1, \ldots, \mathrm{N}\}$ denote the set of customers, and F denote the set of recharging stations and their copies since a recharging station may be visited more than once. Let V' be a set of vertices with $V' = V \cup F$. In order to differentiate the respective instance of the depot, the set is subscripted with 0 or $N+1$. Let vertex 0 and $N+1$ denote the starting depot and ending depot respectively. Thus, $V'_0 = V' \cup \{0\}$ and $V'_{N+1} = V' \cup \{N+1\}$. Then the problem can be defined on a complete directed graph $G = (V'_{0,N+1}, A)$ with the set of arcs $A = \{(i,j)|i,j \in V'_{0,N+1}, i \neq j\}$. Each arc is associated with a distance d_{ij} and a travel time t_{ij}. The battery is consumed at a constant rate of r and the traveled arc consumes a quantity of $r \cdot d_{ij}$ battery power. Let q_i be the recharge quantity if the partial recharging technology is selected in the recharging station and the battery is recharged at a recharging rate of g. Otherwise the battery will be swapped with a fully charged battery with the cost c_s per battery swap. Since the swapping of battery can be completed in very short time, we assume the consuming time is zero compared to the travel time of the route. Each vertex $i \in V$ has a positive demand D_i, a service time s_i and a time window $[e_i, l_i]$. All EVs have a load capacity of C and a battery capacity of Q. The decision variables, τ_i, u_i, and v_i denote the service starting time, remaining cargo level and remaining charge level at customer $i \in V'_{0,N+1}$, respectively. The battery state of charge on departure from recharging station $i \in F$ is defined as Y_i. δ_i is the waiting time of the EV at vertex $i \in N$. The binary decision variable x_{ij} takes 1 if arc (i,j) is traversed and 0 otherwise. Let variable y_i take 1 if the BSS is used at the station $i \in F$ and 0 otherwise. Let variable z_i take 1 if the

TABLE 1. Notations in the model formulation.

Sets:	
V	set of customers;
F	set of recharging stations and their copies;
$\{0\}$	the starting depot;
$\{N+1\}$	the ending depot;
V'	set of customers and recharging stations;
V'_0	set of customers, starting depot and recharging stations;
V'_{N+1}	set of customers, ending depot and recharging stations;
$V'_{0,N+1}$	set of all the vertices;
Parameters:	
d_{ij}	the distance between vertex i and j;
t_{ij}	the travel time between vertex i and j;
r	the battery consume rate;
g	the battery charge rate;
D_i	the demand of customer i;
s_i	the service time of customer i;
e_i	the earliest start of service at vertex i;
l_i	the latest start of service at vertex i;
C	the EV's load capacity;
Q	the capacity of the EV's battery;
c_f	the fixed cost of the EV;
c_t	per unit travel cost;
c_r	per unit recharging cost;
c_w	per unit waiting cost;
c_s	per battery swapping cost;
Decision variables:	
τ_i	service start time of vertex i;
u_i	remaining cargo level of the EV when it departures vertex i;
q_i	the recharging quantity if the partial recharging technology is selected in recharging station i;
v_i	remaining charge level of the EV when it departures vertex i;
Y_i	battery state of the EV's charge when it departures from station $\forall i \in F$;
δ_i	waiting time of the EV at the vertex $i \in V$;
x_{ij}	is equal to 1, if arc (i, j) is traversed; otherwise 0;
y_i	is equal to 1, if the BSS is chosen at the station $i \in F$; otherwise 0;
z_i	is equal to 1, if the partial recharging is chosen at the station $i \in F$; otherwise 0.

partial recharging option is chosen at the station $i \in F$ and 0 otherwise.

The above notations and the other related notations are summarized in TABLE 1.

The objective is to minimize the total cost which consists of the fixed cost of employed vehicles, the travel cost, the waiting cost, the recharging cost and the battery swapping cost. The mixed integer non-linear programming model can be formulated as below.

$$\mathrm{Min}\Pi = c_f \sum_{j\in V'} x_{0j} + c_t \sum_{i,j\in V'} x_{ij}t_{ij} + c_r \sum_{i\in F} z_i q_i + c_w \sum_{i\in V} \delta_i + c_s \sum_{i\in F} y_i \tag{1}$$

$$\text{subject to} \quad \sum_{j\in V'_{N+1}, i\neq j} x_{ij} = 1, \forall i \in V, \tag{2}$$

$$\sum_{j\in V'_{N+1}, i\neq j} x_{ij} \leq 1, \quad \forall i \in F, \tag{3}$$

$$\sum_{i\in V'_0, i\neq j} x_{ij} = \sum_{i\in V'_{N+1}, i\neq j} x_{ji}, \quad \forall j \in V', \tag{4}$$

$$\tau_i + (t_{ij} + s_i)x_{ij} - l_0(1 - x_{ij}) \leq \tau_j, \forall i \in V'_0, \forall j \in V'_{N+1}, i \neq j, \tag{5}$$

$$\tau_i + t_{ij}x_{ij} + gq_iz_i - (l_0 + gQ)(1 - x_{ij}) \leq \tau_j, \forall i \in F, \forall j \in V'_{N+1}, i \neq j, \tag{6}$$

$$e_j \leq \tau_j \leq l_j, \forall j \in V'_{N+1}, \tag{7}$$

$$0 \leq u_j \leq u_i - D_ix_{ij} + C(1 - x_{ij}), \quad \forall i \in V'_0, \forall j \in V'_{N+1}, i \neq j, \tag{8}$$

$$0 \leq u_0 \leq C, \tag{9}$$

$$0 \leq v_j \leq v_i - (r \cdot d_{ij})x_{ij} + Q(1 - x_{ij}), \forall i \in V, \forall j \in V'_{N+1}, i \neq j, \tag{10}$$

$$0 \leq v_j \leq Y_i - (r \cdot d_{ij})x_{ij} + Q(1 - x_{ij}), \forall i \in F, \forall j \in V'_{N+1}, i \neq j, \tag{11}$$

$$Y_i = z_i(v_i + q_i) + y_iQ, \quad \forall i \in F, \tag{12}$$

$$y_j + z_j \leq 1, \quad \forall j \in F, \tag{13}$$

$$Y_i \leq Q, \quad \forall i \in F, \tag{14}$$

$$x_{ij} \in \{0, 1\}, \quad \forall i \in V'_0, \forall j \in V'_{N+1}, i \neq j, \tag{15}$$

$$y_j, z_j \in \{0, 1\}, \forall j \in F, \tag{16}$$

$$q_i \geq 0, \quad \forall i \in F, \tag{17}$$

Constraints (2) and (3) ensure that each customer is visited exactly once and handle the connectivity of visits to recharging stations. The flow conservations constraints (4) guarantee that the number of outgoing arcs equals to the number of incoming arcs at each vertex. Constraints (5) and (6) ensure the time feasibility of arcs leaving the depot/customers and the stations, respectively. We can see that, if there is a sub-tour contains vertex $i, j, \ldots, k$, the arrival time of vertex i would lead to a contradiction, $\tau_i \leq \tau_j \leq \ldots \leq \tau_k \leq \tau_i$, which violates the constraints(5)-(6). Besides, Constraints (7) assures every vertex visited within the time windows. Further, Constraints (5)-(7) eliminate the sub-tours. These constraints are also employed in [17], [23] for eliminating sub-tours in routing problems. Constraints (8) and (9) meet the demand of all customers. Constraints (10) and (11) keep track of the battery state of charge and ensure that it is never negative. Constraints (12) determine the battery state of charge after the recharge at a station. Constraints (13) limit the vehicle to use with recharging capability or battery swapping capability at a station. Constraint (14) make sure that battery state of charge does not exceed the battery capacity. Finally, constraints (15)-(17) restrict the decision variables.

Compared to the general EVRPTW model, more binary and continuous variables have to be introduced in the EVRP-TW&MC model, which makes it more difficult to solve. Thus, we develop an improved ACO algorithm enhanced with local search to solve the problem.

IV. HEURISTICS

In order to solve the EVRP-TW&MC problem, an improved ACO algorithm with enhanced local search (ACO-LS) is proposed. Considering the partial recharging strategy, an aggressive insertion heuristic is designed. The main procedure of IACO-LS is shown in **Algorithm1**.

Algorithm 1 The Main Procedure of ACO-LS

1 Parameter Initialization;
2 Construct an initial solution φ_0;
3 Initialize the pheromone;
4 **Iterate** for Max_iteration times
5 For all k∈ ants, construct NR solution ψ'_k in parallel, until all ants are done;
6 Insertion Heuristics;
7 Local Search;
8 Update Pheromone;
9 **End Iterate**
10 Output the best feasible solution ζ_{best}.

A. IMPROVED ANT COLONY OPTIMIZATION

The ACO algorithm is a meta-heuristic inspired by the foraging behavior of real ants in the wild, which is presented by Dorigo *et al.* [19]. It has been successfully applied as a solution tool to the VRP and VRPTW [20], [26], [31], [32] Considering the impact of time windows, we designed a new model integrating the impact of both distances and time windows (DTW) to calculate the probability for selecting the next customer, which is noted as DTW probabilistic model.

1) INITIAL COLONY CONSTRUCTION

Ant colonies are positive feedback systems. When the ants move, they left a trail of a chemical substance called pheromone. An individual ant can follow the trail left by previous companies and strengthen it with its own pheromone. Consequently, as more ants move, more pheromone is deposited, and the corresponding arc is more attractive for other ants in the future [28].

Noticing the importance of pheromone, the nearest neighborhood search is employed to generate a non-recharging (NR) solution. In the NR solution, only the EV's capacity is considered, while the battery limitation is not considered and no recharging stations would be utilized. For the original EVRP-TW&MC problem, the NR solution is infeasible. Thus, an insertion heuristic is designed to fix the NR solution by inserting recharging stations into the routes over the EV's battery limitation, and construct an initial feasible solution, noted as Ψ_0. The insertion heuristic is presented in the following Subsection B.

The trail intensities of all possible visits between customers or between the customer and the depot are initialized with the same value reciprocal of the objective value of Ψ_0 at the beginning of the algorithm.

In the ant colony optimization, the colony scale is set as P, and each artificial ant simulates the employed electric vehicle and successively select customers to visit. If no customer is accessible because of load capacity, or violating the time window constraints, then the ant goes back to the unique depot and repeats again until all customers are visited. Initially, each ant starts at the depot and the initial routes are empty. In traditional ACO for solving VRP, the decision-making process about selecting the next customer j is based on a probabilistic rule considering the pheromone information and the visibility of the distance. Since the time windows are critical constraints for the VRP with narrow time windows compared to common VRP, they should be considered in the decision-making process. Thus, we design a new model integrating the impact of both distances and time windows to calculate the probability for selecting the next customer, which is noted as DTW probabilistic model. For ant k at the current position of customer i, the DTW probabilistic model is presented as follows.

$$P_{i,j}^k = \begin{cases} \dfrac{\tau_{ij}^{\alpha} \cdot h_{ij}^{\beta} \cdot \omega_{ij}^{\delta}}{\sum\limits_{z \in J_k(i)} \tau_{iz}^{\alpha} \cdot h_{iz}^{\beta} \cdot \omega_{iz}^{\delta}}, & if\ j \in J_k(i), \\ 0, & otherwise, \end{cases} \tag{18}$$

where $P_{i,j}^k$ gives the probability of choosing to combine customers i and j on the route of ant k, $J_k(i)$ is the set of accessible customers for choosing. τ_{ij}, h_{ij} and ω_{ij} denotes the pheromone density, visibility of distance and time windows of edge $< i, j >$ respectively. α, β and δ denotes the relative influence of the pheromone trails and the two visibility values. In this paper, the expectation factors h_{ij} and ω_{ij} are set as follows:

$$h_{ij} = \frac{1}{d_{ij}}, \tag{19}$$

$$\omega_{ij} = \frac{1}{l_j}, \tag{20}$$

where d_{ij} is the length of edge $< i, j >$, and l_j is the latest arrival time of time windows of customer j. It shows that the more urgent the customer j is, the more visibility would be selected.

2) UPDATING PHEROMONE

The Updating of pheromone trails plays a key role in the searching process for obtaining better solutions. Firstly, pheromone updating is conducted by reducing the amount of pheromone on the links between customers or between the customer and the depot. This strategy simulates the natural evaporation of the pheromone so that no path becomes too dominant. The pheromone trails are updated globally which is performed at the end of the constructive phase after insertion heuristics. Different from original ACO, the elitist ant strategy is adopted in this paper. Not only the best solution is used to modify the pheromone trails, but also the acceptable solutions of elitist ants in the ranked list are also used. All the pheromone trails are dynamically updated as follows:

$$\tau_{ij}^{new} = (1-\varphi)\tau_{ij}^{old} + \Delta\tau_{ij}^{ib} + \Delta\tau_{ij}^{\delta_{th}} \tag{21}$$

where τ_{ij}^{new} is the pheromone on the edge $< i, j >$ after updating, τ_{ij}^{old} is the pheromone on the edge $< i, j >$ before updating, and φ is the parameter that controls the speed of evaporation.

$$\Delta\tau_{ij}^{ib} = \begin{cases} \dfrac{Q}{Cost^{ib}}, & \textit{if edge} < i,j > \textit{on the routes}, \\ 0, & \textit{otherwise}. \end{cases}$$
$$\Delta\tau_{ij}^{\delta_{th}} = \begin{cases} \dfrac{Q}{Cost^{\delta_{th}}}, & \textit{if edge} < i,j > \textit{on the routes}, \\ 0, & \textit{otherwise}. \end{cases} \tag{22}$$

where Q is a constant, $Cost^{ib}$ is the total cost of the best-so-far solution of current iteration, and $Cost^{\delta_{th}}$ is the total cost of the solution of elitist ant δ_{th}. The elitist ant strategy expands the range of pheromone updating and accelerates the speed of convergence.

B. INSERTION HEURISTIC

Because of the limited battery capacity of the EV, the cruising range of the EV is constrained and the NR solutions constructed by ACO stand a good chance being battery-energy violated. Thus, these routes in φ_0 need recharging during travelling in order to complete the tour. To minimize the cost of recharging, an aggressive insertion heuristic is proposed. The main idea of the insertion heuristic is to find all the infeasible routes and insert the best recharging station into them one by one until all routes are feasible. The procedure of insertion heuristics is presented in **Algorithm 2**.

Given an infeasible NR route γ_0, let the EV travel along this route and find the farthest node that can be reached within the battery capacity. Select the nearest accessible recharging station (RS), and then insert the selected RS after this node. To find the best inserting place, we try to determine the station insertions on all the arcs between the corresponding customer and the previous depot or RS so that all the feasible routes can be compared.

After recharging stations are inserted in the NR route, a recharging strategy should be designed to select the recharging options and determine the recharging level. Since battery swapping is expensive but time-conserving, it is set that if the recharging level exceeds a threshold value T_0, BSSs will be selected. Otherwise a partial recharging is conducted. T_0 is defined as below.

$$T_0 = \sigma \cdot (Q/g), \tag{23}$$

Algorithm 2 The Procedure of the Insertion Heuristics

Input: An infeasible NR solution γ_0;
Output: A best feasible solution γ_{fea};
1 **For** all infeasible routes in γ_0;
2 Find the farthest node the EV can visit along an infeasible route;
3 **For** all nodes between the previous depot/station and the farthest node
4 Insert the nearest feasible station after the node;
5 Choose the recharging option;
6 Determine the recharging level;
7 Check the time windows;
8 **If** the route is time-violated
9 Remove the time-violated customers from the route;
10 Add the removed customers into the set $V^{unvisit}$;
11 **Else**
12 remove the new feasible route from Ψ_{best} and add it into γ_0;
13 **End if**
14 **End for**
15 **End for**
16 **While** $V^{unvisit} \neq \phi$
17 utilize the nearest neighborhood heuristic to form NR routes for $V^{unvisit}$;
18 Repeat line 1-14;
19 **End While**

where σ is a parameter between 0 and 1, and Q/g denotes the full battery recharge time starting from a zero battery level.

When the recharging level at each station is determined, the feasibility of time windows for all customers should be checked again. Remove the customers whose time windows are not satisfied and add them into the list of $V^{unvisit}$. During the process of station insertion, the first acceptance criterion is keeping as many customers as possible and the second is the lower cost. For the nodes in $V^{unvisit}$, we iteratively employ the nearest neighborhood heuristic and insertion heuristic until all customers are visited.

C. LOCAL SEARCH

Since the ant colony algorithm usually gets trapped into local optimal, local search strategy is considered as an efficient way to enlarge the searching space and improve the solution quality. In this study, local search is conducted by combining removal and insertion operators. The solution is destroyed by removing some customers from the exist routes and then repaired by reinserting the removed customers to the solution. During the searching process, since the stations may need adjustment after the transformation of customers, the removal and insertion operators in the local search is operated on the NR routes with no stations. Then the insertion heuristic is utilized to generate new feasible solutions.

1) REMOVAL OPERATORS

We now present seven removal operators used in the local search. They can be divided into two classes: Route Removal (RR) and Customers Removal (CR). In RR a completed route is removed from the current solution. In CR, a number of λ customers are removed and added into a removal list ς. The value of parameter λ depends on the total number of customers n_c and is determined randomly between $\underline{n_c}$ and $\overline{n_c}$ using a uniform distribution. The seven removal operators are illustrated as follows.

a: RANDOM ROUTE REMOVAL

This operator randomly selects a route from all the routes in the current solution and removes all the nodes in the selected route.

b: SHORTEST ROUTE REMOVAL

This operator selects the shortest route from current solution and removes all the nodes in this route. The idea is to maximize the utility of the load capacity and attempt to reduce the number of vehicles.

c: EARLIEST ROUTE REMOVAL

This operator selects the route which finishes the delivery at the earliest time compared to other routes in the current solution and removes all the nodes in this route. The idea is to fully utilize the working hours for the consideration of practical works.

d: RANDOM CUSTOMER REMOVAL

This operator randomly removes a number of customers from the solution. The purpose of randomly selecting nodes is diversifying the search mechanism.

e: WORST-DISTANCE CUSTOMER REMOVAL

This operator iteratively removes the worst distance customers in current solution, where the worst distance is defined as the sum of distance from the preceding and succeeding customer on the tour.

f: PROXIMITY-BASED CUSTOMER REMOVAL

The operator removes a set of nodes that are related in terms of distance. At the beginning, we randomly select a node and afterwards remove the nearest node of the current removed node until λ customers are added into the list ς. It can be viewed as a special case of the Shaw removal operator.

g: TIME-BASED CUSTOMER REMOVAL

The operator removes a set of nodes that are related in terms of time windows, which can also be viewed as a special case of the Shaw removal operator. At the beginning, we randomly select a node and afterwards remove the node whose the latest start of the service is closest to the current removed node until λ customers are added into the list ς.

2) INSERTION OPERATORS

In this section, we present four insertion operators used in the local search. Insertion operators insert the removed customers in the removal list ς back into the destroyed solution at different places to form a new feasible solution. These operators must maintain the feasibility with respect to the load capacity and time windows, or they generate new routes using additional vehicles. We use the greedy and regret-2 insertion operator from Demir *et al.* [33]. In addition, we propose the Greedy insertion with priority and simulated annealing based insertion as follows.

a: GREEDY INSERTION

This operator repeatedly inserts a node in the best feasible position of the current solution which has the least insertion cost.

b: REGRET-2 INSERTION

Let Δf_i denote the change in the objective value by inserting node i into its best and second best position. Let $i^* = \arg\max_{i\in\varsigma}\{\Delta f_{i2} - \Delta f_{i1}\}$, where Δf_{i1} is the best feasible reinsertion and Δf_{i2} is the second best reinsertion of node i.

c: GREEDY INSERTION WITH PRIORITY

Thinking of the nodes that have multiple insertable positions, we give a priority to these nodes with less insertable positions and reinsert them firstly. Let C_k denotes the numbers of insertable position of node k. The nodes are sorted in the non-decreasing order of C_k and reinserted in this order.

d: SIMULATED ANNEALING BASED INSERTION

This operator utilizes the idea of simulated annealing and assumes that the suboptimal reinsertion of a node can be accepted with a certain probability p. The purpose of this operator is also enlarging the searching space.

During the searching process, all the removal and insertion operators are combined and selected by a roulette-wheel mechanism. The procedure of local search is shown in **Algorithm 3**.

V. COMPUTATIONAL STUDY

In this section, computational experiments are performed to test the proposed ACO-LS algorithm. All the programs were coded in Visual C++ and are implemented on an Intel Core i5 processor with 3GHz speed and 8GB RAM.

A. DATA SET ILLUSTRATION

In this paper, the nodes' distribution and the EV performance in all test instances are from the data set in Schneider *et al.* [6], which were generated based on the well-known Solomon datasets. The data set includes 56 instances with three types, which are instances with 100 customers and 21 recharging stations Clustered (C) distributed, Randomly distributed (R) and Randomly-Clustered distributed (RC). The parameter values for battery capacity, vehicle load capacity, fuel

Algorithm 3 The Local Search

```
Input: A NR solution φ0 with no recharge station;
Output: A best feasible solution Ψbest;
Initialize the weight of the operators;
Insertion heuristics, and generate the current feasible
  solution Ψcurrent;
Let Ψbest ← Ψcurrent;
Iterate for maximum number of iterations
   Randomly choose a removal operator and generate the
    removed node list ς;
    Randomly choose an insertion operator and reinsert
    the nodes in ς into the current
    solution φcurrent;
   Conduct insertion heuristic and generate the feasible
    solution Ψcurrent;
   if the cost of Ψcurrent is lower than Ψbest
      Ψbest ← Ψcurrent;
   End if
   Remove all the stations from the routes in the current
     feasible solution Ψcurrent and generate the new NR
     current solution φcurrent;
End Iterate
```

consumption rate, recharging rate and average velocity are all provided in each instance of the datasets.

For deriving the different cost values specified in Sect. 3, we investigated practical logistics companies and figure out the costs related with the EV operations. The fixed cost for employing an EV, c_f, is set as 200 Yuan (Chinese dollars). The per unit travel cost of the EV, c_t, is set as 10 Yuan/min. The per unit recharging cost, c_r, is set at 100 Yuan/h, and per unit waiting cost c_w is 24 Yuan/h. Since the battery swapping employs more infrastructure, the per battery swapping cost, c_s, is expensive and set as 1.5 times that of a full battery recharged from a zero battery level. The above values are summarized in TABLE 2.

TABLE 2. The values of parameters related with EV costs.

Parameters	**The value**
c_f	200 Yuan
c_t	10 Yuan/min
c_r	100 Yuan/h
c_w	24 Yuan/h
c_s	1.5 times that of a full battery recharged from zero level

In this paper, the ACO-LS algorithm related parameters are set as $P = 30, \alpha = 5, \beta = 5, \delta = 10, \varphi = 0.25$ and $Q = 100$.

B. THE IMPACT OF APPLYING DTW PROBABILISTIC MODEL IN ACO

For improving the efficiency of ACO, one of our main improvements is to design a new DTW probabilistic model integrating the visibility of both distances and time windows, which is used to select the next customer in the solution constructing process. In order to test the impact of applying DTW model, all the 56 instances are solved by the ACO-LS algorithm with general probabilistic model and the ACO-LS with DTW probabilistic model (ACO-LS-DTW).

In the C type instances, all the customers nodes are clustered, which includes 17 instances. The computational results are presented in TABLE 3, where 'VC' refers the cost of all used EVs, 'TC' refers to the total cost calculated by the objective function (1), and $\Delta\%$ means the relative gap between TCs obtained by the algorithm without and with DTW. A negative $\Delta\%$ value means improvement obtained by the ACO algorithm with DTW probabilistic model.

TABLE 3. The results of C type instances by the algorithm with/without DTW.

	Without DTW		With DTW		
Inst..	VC	TC	VC	TC	$\Delta\%$
c101	2400	15463.02	2400	14971.39	-3.18
c102	2400	17468.35	2400	16266.70	-6.88
c103	2400	15704.22	2200	15133.67	-3.82
c104	2200	14394.14	2200	14264.24	-0.92
c105	2400	15653.47	2400	14593.90	-6.77
c106	2400	15947.11	2400	15219.17	-4.56
c107	2200	14541.05	2200	14216.34	-2.23
c108	2200	13971.79	2200	13720.62	-1.80
c109	2200	13648.40	2200	13327.49	-2.35
Average					***-3.61***
c201	900	8195.82	800	7734.22	-5.63
c202	1000	10164.33	800	9651.84	-5.04
c203	1000	10700.60	800	10510.30	-1.78
c204	800	9416.35	800	9688.11	2.88
c205	800	7844.05	800	7728.96	-1.47
c206	800	7952.36	800	7720.37	-2.92
c207	800	7862.21	800	7801.98	-0.77
c208	800	7734.54	800	7676.65	-0.75
Average					***-1.94***

The 17 C type instances can be further divided into two classes, which are the instances with narrow time windows, noted as C1 instances, and the instances with wide time windows, C2 instances. The computational results in TABLE 3 show that the algorithm with DTW model can achieve better solutions for 16 C instances. The total costs for all C1 instances are reduced by an average of 3.61%, while the total costs for all C2 instances the instances are reduced by an average of 1.94%.

Similarly, the R type and RC type instances can also be divided into two classes respectively, we denote R1/RC1 as the instances with narrow time windows and R2/RC2 as the instances with wide time windows. The computational results of R instances are presented in TABLE 4, while the results of RC instances are presented in TABLE 5. We can see that the algorithm with DTW obtains better results for all R and RC instances. The performance on R1 and RC1 instances is also better than that of R2 and RC2 instances respectively. Thus, the DTW probabilistic model performs better on type-1 instances with narrow time windows, which indicates that it is an effective way to take the time windows into consideration in the DTW probabilistic model.

TABLE 4. The results of R type instances by the algorithm with/without DTW.

Inst.	ACO		ACO-DTW		Δ%
	VC	TC	VC	TC	
r101	3800	20954.99	3800	20869.32	-0.41
r102	3800	20869.32	3400	19283.25	-7.60
r103	2800	17561.33	2800	16912.05	-3.70
r104	2400	14442.37	2400	14357.91	-0.58
r105	3000	18006.90	2800	17945.41	-0.34
r106	2800	16985.57	2800	16900.29	-0.50
r107	2600	15691.65	2600	15304.17	-2.47
r108	2200	14309.66	2200	14160.07	-1.05
r109	2600	15967.14	2600	15954.20	-0.08
r110	2600	15181.27	2400	14854.93	-2.15
r111	2400	14596.83	2400	14090.74	-3.47
r112	2400	13940.87	2400	13925.33	-0.11
Average					***-2.03***
r201	800	13343.64	800	13309.18	-0.26
r202	800	12457.06	600	11965.32	-3.95
r203	600	10788.70	600	10598.07	-1.77
r204	600	8704.85	600	8681.57	-0.26
r205	600	11245.32	600	11051.35	-1.72
r206	600	10859.58	600	10786.98	-0.69
r207	600	9824.15	600	9821.09	-0.03
r208	400	8565.78	400	8484.81	-0.95
r209	600	10486.46	600	10198.96	-2.74
r210	600	10107.81	600	9992.51	-1.14
r211	600	9481.79	600	8996.28	-5.12
Average					***-1.69***

TABLE 5. The results of RC type instances by the algorithm with/without DTW.

Inst.	ACO		ACO-DTW		Δ%
	VC	TC	VC	TC	
rc101	3200	21671.14	3200	21451.67	-1.01
rc102	3200	20165.93	3000	19827.57	-1.68
rc103	2800	17940.85	2800	17745.75	-1.09
rc104	2400	16470.78	2400	16305.52	-1.00
rc105	3000	19618.57	3000	19125.69	-2.51
rc106	2800	18392.27	2800	18187.79	-1.11
rc107	2400	16107.99	2400	15893.61	-1.33
rc108	2200	15465.03	2200	15338.85	-0.82
Average					***-1.32***
rc201	1100	15737.04	1000	15470.95	-1.69
rc202	1000	14333.85	1200	14198.12	-0.95
rc203	800	12346.71	800	12129.30	-1.76
rc204	800	10808.73	800	10690.88	-1.09
rc205	1000	13302.98	1000	13249.87	-0.40
rc206	800	13023.98	800	12945.04	-0.61
rc207	800	11429.55	800	11424.93	-0.04
rc208	800	10047.65	800	9852.83	-1.94
Average					***-1.06***

The overall results in TABLE 3-5 show that the algorithm with DTW performs better than the algorithm without DTW for 55 instances, and performs worse in only one instance. It indicates that it is a good strategy to utilize the DTW probabilistic model in the ACO algorithm for solving EVRP-TW&MC.

C. THE IMPACT OF THE ENHANCED LOCAL SEARCH

In order to test the efficiency of the improved ACO, all the 56 instances are also solved by the general ACO, the variable neighborhood search (VNS) algorithm and the ACO-LS. VNS is also an effective algorithm to the VRP and its variants verified by many works [6], [34], [14]. We compared

TABLE 6. The results of C type instances by the algorithm with/without LS and VNS.

Inst.	ACO	VNS	ACO-LS	Δ_1%	Δ_2%
c101	19079.20	17484.51	14971.39	-22	-14
c102	19704.49	18391.66	16266.70	-17	-12
c103	20708.03	16701.14	15133.67	-27	-9
c104	18192.95	13775.97	14264.24	-22	3.5
c105	16042.68	17197.07	14593.90	-9	-15
c106	18881.18	18158.62	15219.17	-19	-16
c107	16141.95	17137.87	14216.34	-12	-17
c108	16049.98	16144.07	13720.62	-15	-15
c109	16147.30	15405.87	13327.49	-17	-13
c201	9858.65	11499.57	7734.22	-22	-33
c202	15832.78	12054.08	9651.84	-39	-19
c203	15575.02	12891.58	10510.30	-33	-18
c204	14405.19	9869.81	9838.34	-32	-0.31
c205	8047.88	9266.09	7728.96	-4	-16
c206	7960.23	8643.82	7720.37	-3	-11
c207	8602.23	9713.18	7801.98	-9	-20
c208	8710.02	9978.13	7676.65	-12	-23

TABLE 7. The results of R type instances by the algorithm with/without LS and VNS.

Inst.	ACO	VNS	ACO-LS	Δ_1%	Δ_2%
r101	23918.32	21317.63	20869.32	-13	-2
r102	22215.22	19542.91	19283.25	-13	-1
r103	19444.02	16498.93	16912.05	-13	2
r104	17646.70	14431.42	14357.91	-19	-1
r105	21389.68	18097.02	17945.41	-16	-1
r106	19517.86	16932.05	16900.29	-13	-0.2
r107	19652.71	15087.42	15304.17	-14	1
r108	16448.81	13826.18	14160.07	-14	2.4
r109	19326.65	16638.69	15954.20	-17	-4
r110	17583.62	15084.99	14854.93	-16	-2
r111	17792.54	14574.32	14090.74	-21	-3
r112	15815.10	13743.12	13925.33	-12	1
r201	16244.08	13959.91	13309.18	-18	-18
r202	16368.64	12752.13	11965.32	-26	-6
r203	15143.08	10476.59	10598.07	-30	1
r204	12630.53	9201.80	8681.57	-31	-6
r205	14001.72	11679.54	11051.35	-20	-5
r206	13433.72	11266.33	10786.98	-20	-4
r207	12546.08	10049.13	9821.09	-22	-2
r208	11600.62	9297.40	8484.81	-27	-9
r209	13060.49	10845.98	10198.96	-22	-6
r210	13959.05	10759.07	9992.51	-28	-7
r211	11477.89	9375.72	8996.28	-22	-4

the results between ACO, VNS and ACO-LS. All the computational results are presented in TABLE 6-8, where Δ_1% means the relative gap between TCs obtained by ACO and ACO-LS, and Δ_2% means the gap between TCs obtained by VNS and ACO-LS. It can be seen that, for all 56 instances, the solutions obtained by ACO-LS are much better than that of the general ACO. The minimal improvement obtained by ACO-LS is 3%, while the maximal improvement is 39%. Besides, we can find that the local search strategy performs better in the type-2 instances which have the wide time windows. And the results showed that the overall performance of ACO-LS is also better than VNS, since there are 46 out of 56 instances achieved obvious improvements by ACO-LS. The fixed cost for vehicles is also important for the operators, and the results for this part is presented in APPENDIX A. Similarly, ACO-LS always obtained the lowest fixed vehicle cost

TABLE 8. The results of RC type instances by the algorithm with/without LS and VNS.

Inst.	ACO	VNS	ACO-LS	$\Delta_1\%$	$\Delta_2\%$
rc101	25012.98	21172.22	21451.67	-14	1
rc102	24727.68	19887.90	19827.57	-20	-0.3
rc103	21875.03	17441.86	17745.75	-19	1.7
rc104	19018.81	15415.46	16305.52	-14	5.7
rc105	24301.62	20087.74	19125.69	-21	-5
rc106	21881.31	18416.22	18187.79	-17	-1
rc107	18966.93	16089.08	15893.61	-16	-1
rc108	18607.42	15106.76	15338.85	-18	1.5
rc201	20080.46	17339.92	15470.95	-23	-11
rc202	18237.26	15206.40	14198.12	-22	-7
rc203	15820.40	13340.06	12129.30	-23	-9
rc204	13715.80	10357.06	10690.88	-22	3
rc205	18004.98	13946.88	13249.87	-26	-5
rc206	16962.39	14186.99	12945.04	-24	-9
rc207	15590.99	13106.94	11424.93	-27	-13
rc208	13196.08	10577.58	9852.83	-25	-7

TABLE 9. The results of C type instances under different charging strategies.

	BS		PR		MR	
Inst.	VC	TC	VC	TC	VC	TC
c101	2400	15435.37	2400	14482.22	2400	14458.24
c102	2200	15962.34	2400	15491.87	2400	15491.87
c103	2200	15664.41	2400	15936.16	2200	14920.98
c104	2000	14149.56	2200	14740.67	2200	13663.18
c105	2200	15031.66	2400	14162.24	2400	14162.24
c106	2200	15890.77	2400	14622.90	2400	14212.54
c107	2200	14509.56	2400	14420.35	2200	13967.48
c108	2200	14278.16	2400	13854.94	2200	13576.68
c109	2000	13761.76	2200	12953.55	2200	12706.37
c201	800	8087.98	800	7734.22	800	7734.22
c202	800	10795.77	800	9021.94	800	9021.94
c203	800	11317.46	800	10223.42	800	10070.03
c204	800	10493.48	800	10400.27	800	9688.11
c205	800	8007.71	800	7727.73	800	7727.73
c206	800	8021.79	800	7698.82	800	7698.82
c207	800	8101.18	800	7743.11	800	7721.87
c208	800	7983.71	800	7706.28	800	7670.44

compared to the other two algorithms except for 2 instances rc204 and rc208.

The ACO-LS is also compared with CPLEX for small instances, since CPLEX cannot solve the 56 instances. Bruglieri *et al.* [35] solved 6 small instances by CPLEX and VNS. All the six instances are solved by ACO-LS and the overall costs obtained are as good as the results in [35].

D. ANALYSIS FOR DIFFERENT CHARGING STRATEGIES

In this experiment, we compare the costs of the EVRPTW under different recharging strategies, which are the strategies of battery swapping (BS), partial recharging (PR) and the multiple recharging (MR) situation studied in this paper. All the instances are solved by the ACO-LS algorithm. The computational results for the C, R and RC type instances are presented in TABLE 9-11 respectively.

Compared to the BSSs, the partial recharging significantly reduced the recharging cost. The results of TABLE 9 showed that an average of 6.22% cost is saved for C-type instances, and the most cost saved is 11.62%. The results of TABLE 10 showed that an average of 10.66% cost is saved for R-type

TABLE 10. The results of R type instances under different charging strategies.

	BS		PR		MR	
Inst.	VC	TC	VC	TC	VC	TC
r101	3800	23072.34	3600	20757.59	3600	20635.22
r102	3200	20778.61	3400	19439.21	3200	18718.07
r103	2600	18195.88	3000	16981.81	2600	16287.97
r104	2200	15801.19	2400	14630.93	2400	14038.46
r105	2600	18950.36	3200	18371.35	2800	17788.55
r106	2600	18215.12	2800	16779.88	2800	16779.88
r107	2400	16273.08	2600	15203.46	2400	14916.47
r108	2000	15120.61	2400	14324.45	2200	13532.12
r109	2600	17246.63	2800	15975.55	2400	15568.32
r110	2200	15682.44	2400	14370.79	2400	14111.53
r111	2400	15658.41	2600	14656.59	2400	14404.39
r112	2200	14866.11	2200	13261.01	2200	13261.01
r201	800	15417.65	800	13207.46	800	13048.64
r202	1000	13081.93	600	11853.87	600	11718.25
r203	600	11987.20	600	10596.28	600	10470.88
r204	800	9357.93	600	8768.49	600	8541.91
r205	600	12632.42	600	11051.35	600	11051.35
r206	600	12504.14	600	10538.24	600	10538.24
r207	800	10376.06	400	9762.68	600	9628.74
r208	600	9384.32	400	8337.95	400	8337.95
r209	600	11666.54	600	10164.74	600	10105.05
r210	600	11362.36	600	9868.40	600	9690.40
r211	600	9837.05	600	8999.92	600	8868.32

TABLE 11. The results of RC type instances under different charging strategies.

	BS		PR		MR	
Inst.	VC	TC	VC	TC	VC	TC
rc101	3200	23274.20	3200	20937.98	3200	20937.98
rc102	2800	21419.62	3000	19848.12	3000	19565.56
rc103	2600	19346.52	2600	17745.65	2600	17260.04
rc104	2200	17116.75	2400	16014.46	2400	16014.46
rc105	2800	20448.12	2800	17790.91	3000	17790.91
rc106	2400	19559.78	2600	17505.63	2600	17334.43
rc107	2200	17701.61	2400	15681.54	2400	15681.54
rc108	2200	17007.71	2200	14905.91	2200	14905.91
rc201	1400	16593.96	1200	15470.95	1200	15470.95
rc202	1200	14303.15	1000	13925.71	1000	13925.71
rc203	1000	13077.26	800	12081.67	800	12081.67
rc204	600	12100.56	600	10636.19	800	10590.00
rc205	1200	14575.15	1000	13342.52	1000	13213.85
rc206	1000	13788.69	800	13086.63	800	12815.27
rc207	800	12373.77	800	11115.83	800	11073.07
rc208	600	11475.55	600	9871.41	600	9745.30

instances, and the most cost saved is 15.72%. And the results of TABLE 11 showed that an average of 9.72% cost is saved for RC-type instances, and the most cost saved is 12.30%. We can see that the R dataset is most impacted by the partial recharging. Here, the conclusion related with the partial recharging strategy is also consistent with that in [2] and [9].

Compared to the partial recharging strategy, an average of 2.98% cost is saved for type-1 instances by the multiple recharging strategy, while an average of 1.46% cost is saved for type-2 instances, which indicates the multiple recharging strategy works well for instances with more constrained time windows.

In the objective function, the recharging cost and fixed cost for EVs are most impacted by employing partial recharging, while the fixed cost of EVs are most impacted by the battery swapping strategy. We can see that the total costs of 38 instances are reduced by the multiple charging strategy,

TABLE 12. The fixed vehicle cost of C type instances by the algorithm with/without LS and VNS.

Inst.	ACO	VNS	ACO-LS
c101	2800	2600	2400
c102	3000	2600	2400
c103	3000	2400	2200
c104	2600	2000	2200
c105	2400	2400	2400
c106	2600	2400	2400
c107	2400	2400	2200
c108	2200	2200	2200
c109	2200	2200	2200
c201	1000	1200	800
c202	1000	1000	800
c203	1000	800	800
c204	1000	1000	800
c205	800	800	800
c206	800	800	800
c207	800	800	800
c208	800	800	800

and an average of 2.29% cost is saved, which is very attractive considering daily operation of the EVRP. The savings mainly attributed to the fixed cost of vehicles since the number of EVs are reduced due to the employment of BSSs. For the other 18 instances, the results of MR are as good as the better results of BS and PR, as the BS and PR strategies are the special cases of the MR strategy.

VI. CONCLUSION

In this paper, we study the electric vehicle routing problem with time windows and multiple recharging options. This new extension on VRP enriched the literature and provided more flexible charging strategy for the EV. To solve this NP-hard problem, an improved ACO algorithm is developed.

First, in solution construction process of ACO, a probabilistic selection model integrating the influence of distances and time windows is designed to facilitate the choose of better customers. Second, a station insertion heuristic is proposed to generate feasible solutions. At last, an enhanced local search is utilized to enlarge the search space. The efficiency of these improvements on ACO is confirmed through experiments on the open data set in Schneider *et al.* [6]. We also compared the multiple charging strategy with other charging strategies, e.g. only partial recharging or battery swapping option adopted. The comparison results indicate that the flexible strategy with multiple options can help the transportation firms save more cost.

In future study, more realistic versions of the problem can be investigated. For example, the heterogenous fleet where the EVs have different vehicle capacities as well as the battery capacities and operating costs can be considered in the model. Also, the multi-depot case is a promising research area since the logistics companies are expanding gradually nowadays, where the EVs can start from any one of several depots in a given area. Besides, it is always an important work to develop more efficient algorithms for solving the VRP and their variants. Thus, other metaheuristics may be developed and compared with the improved ACO algorithm.

TABLE 13. The fixed vehicle cost of R type instances by the algorithm with/without LS and VNS.

Inst.	ACO	VNS	ACO-LS
r101	4600	3800	3800
r102	3800	3400	3400
r103	3400	2800	2800
r104	2800	2400	2400
r105	3800	3000	2800
r106	3200	2800	2800
r107	3400	2600	2600
r108	2600	2200	2200
r109	3200	2600	2600
r110	2800	2400	2400
r111	3000	2600	2400
r112	2800	2400	2400
r201	1200	800	800
r202	1200	800	600
r203	1200	600	600
r204	1000	600	600
r205	1000	600	600
r206	800	800	600
r207	800	600	600
r208	800	800	400
r209	800	600	600
r210	800	600	600
r211	600	600	600

TABLE 14. The fixed vehicle cost of C type instances by the algorithm with/without LS and VNS.

Inst.	ACO	VNS	ACO-LS
rc101	3800	3400	3200
rc102	3400	3000	3000
rc103	3600	2800	2800
rc104	3000	2400	2400
rc105	3400	3000	3000
rc106	3000	2800	2800
rc107	2800	2400	2400
rc108	2800	2200	2200
rc201	1600	1000	1000
rc202	1800	1200	1200
rc203	1200	800	800
rc204	1000	600	800
rc205	1400	1000	1000
rc206	1200	800	800
rc207	1000	800	800
rc208	800	600	800

APPENDIX

See Table 12 to 14 here.

REFERENCES

[1] S. Ou, Z. Lin, Z. Wu, J. Zheng, R. Lyu, S. Przesmitzki, and X. He, "A study of China's explosive growth in the plug-in electric vehicle market," US Dept. Energy, Oak Ridge, TN, USA, 2017, pp. 6283–37831.
[2] M. Keskin and B. Çatay, "Partial recharge strategies for the electric vehicle routing problem with time windows," *Transp. Res. C, Emerg. Technol.*, vol. 65, pp. 111–127, Apr. 2016.
[3] A. Verma, "Electric vehicle routing problem with time windows, recharging stations and battery swapping stations," *EURO J. Transp. Logistics*, vol. 7, no. 4, pp. 415–451, Dec. 2018.
[4] Y.-W. Wang and C.-C. Lin, "Locating multiple types of recharging stations for battery-powered electric vehicle transport," *Transp. Res. E, Logistics Transp. Rev.*, vol. 58, pp. 76–87, Nov. 2013.
[5] Y. Wang, Z. Liu, J. Shi, G. Wu, and R. Wang, "Joint optimal policy for subsidy on electric vehicles and infrastructure construction in highway network," *Energies*, vol. 11, no. 9, p. 2479, 2018.
[6] M. Schneider, A. Stenger, and D. Goeke, "The electric vehicle-routing problem with time windows and recharging stations," *Transp. Sci.*, vol. 48, no. 4, pp. 500–520, Nov. 2014.

[7] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," *Manage. Sci.*, vol. 6, no. 1, pp. 80–91, Oct. 1959.
[8] A. Mor and M. G. Speranza, "Vehicle routing problems over time: A survey," *4OR*, vol. 18, pp. 129–149, Mar. 2020, doi: 10.1007/s10288-020-00433-2.
[9] N. Ding, R. Batta, and C. Kwon, "Conflict-free electric vehicle routing problem with capacitated charging stations and partial recharge," Dept. Ind. Syst. Eng., Univ. Buffalo, New York, NY, USA, Dept. Ind. Manage. Syst. Eng., Univ. South Florida, Tampa, FL, USA, Tech. Rep., 2015.
[10] 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," *Eur. J. Oper. Res.*, vol. 252, no. 3, pp. 995–1018, Aug. 2016.
[11] B. Çatay and M. Keskin, "The impact of quick charging stations on the route planning of electric vehicles," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jul. 2017, pp. 152–157.
[12] A. Montoya, C. Guéret, J. E. Mendoza, and J. G. Villegas, "The electric vehicle routing problem with nonlinear charging function," *Transp. Res. B, Methodol.*, vol. 103, pp. 87–110, Sep. 2017.
[13] G. Hiermann, R. F. Hartl, J. Puchinger, and T. Vidal, "Routing a mix of conventional, plug-in hybrid, and electric vehicles," *Eur. J. Oper. Res.*, vol. 272, no. 1, pp. 235–248, Jan. 2019.
[14] G. Macrina, L. Di Puglia Pugliese, F. Guerriero, and G. Laporte, "The green mixed fleet vehicle routing problem with partial battery recharging and time windows," *Comput. Oper. Res.*, vol. 101, pp. 183–199, Jan. 2019.
[15] Ç. Koç, O. Jabali, J. E. Mendoza, and G. Laporte, "The electric vehicle routing problem with shared charging stations," *Int. Trans. Oper. Res.*, vol. 26, no. 4, pp. 1211–1243, Jul. 2019.
[16] D. L. Cortés-Murcia, C. Prodhon, and H. Murat Afsar, "The electric vehicle routing problem with time windows, partial recharges and satellite customers," *Transp. Res. E, Logistics Transp. Rev.*, vol. 130, pp. 184–206, Oct. 2019.
[17] Á. Felipe, M. T. Ortuño, G. Righini, and G. Tirado, "A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges," *Transp. Res. E, Logistics Transp. Rev.*, vol. 71, pp. 111–128, Nov. 2014.
[18] M. Keskin and B. Çatay, "A matheuristic method for the electric vehicle routing problem with time windows and fast chargers," *Comput. Oper. Res.*, vol. 100, pp. 172–188, Dec. 2018.
[19] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: Optimization by a colony of cooperating agents," *IEEE Trans. Syst., Man Cybern., B (Cybern.)*, vol. 26, no. 1, pp. 29–41, Feb. 1996.
[20] B. Bullnheimer, R. F. Hartl, and C. Strauss, "An improved ant system algorithm for the vehicle routing problem," *Ann. Oper. Res.*, vol. 89, pp. 319–328, Jan. 1999.
[21] L. M. Gambardella, E. Taillard, and G. Agazzi, "MACS-VRPTW: A multiple colony system for vehicle routing problems with time windows," in *New Ideas in Optimization*. 1999, pp. 63–76.
[22] B. Yu, Z. Z. Yang, and B. Z. Yao, "A hybrid algorithm for vehicle routing problem with time windows," *Expert Syst. Appl.*, vol. 38, no. 1, pp. 435–441, 2008.
[23] B. Yu, Z. Z. Yang, and B. Yao, "An improved ant colony optimization for vehicle routing problem," *Eur. J. Oper. Res.*, vol. 196, no. 1, pp. 171–176, 2009.
[24] X. Zhang and L. Tang, "A new hybrid ant colony optimization algorithm for the vehicle routing problem," *Pattern Recognit. Lett.*, vol. 30, no. 9, pp. 848–855, Jul. 2009.
[25] S. R. Balseiro, I. Loiseau, and J. Ramonet, "An ant colony algorithm hybridized with insertion heuristics for the time dependent vehicle routing problem with time windows," *Comput. Oper. Res.*, vol. 38, no. 6, pp. 954–966, Jun. 2011.
[26] Q. Ding, X. Hu, L. Sun, and Y. Wang, "An improved ant colony optimization and its application to vehicle routing problem with time windows," *Neurocomputing*, vol. 98, pp. 101–107, Dec. 2012.
[27] P. Y. Yin and Y. L. Chuang, "Adaptive memory artificial bee colony algorithm for green vehicle routing with cross-docking," *Appl. Math. Model.*, vol. 40, nos. 21–22, pp. 9302–9315, 2016.
[28] E. Cao, R. Gao, and M. Lai, "Research on the vehicle routing problem with interval demands," *Appl. Math. Model.*, vol. 54, pp. 332–346, Feb. 2018.
[29] S. Zhang, Y. Gajpal, S. S. Appadoo, and M. M. S. Abdulkader, "Electric vehicle routing problem with recharging stations for minimizing energy consumption," *Int. J. Prod. Econ.*, vol. 203, pp. 404–413, Sep. 2018.
[30] Y. Li, H. Soleimani, and M. Zohal, "An improved ant colony optimization algorithm for the multi-depot green vehicle routing problem with multiple objectives," *J. Cleaner Prod.*, vol. 227, pp. 1161–1172, Aug. 2019.
[31] S. Mazzeo and I. Loiseau, "An ant colony algorithm for the capacitated vehicle routing," *Electron. Notes Discrete Math.*, vol. 18, pp. 181–186, Dec. 2004.
[32] K. Doerner, M. Gronalt, R. F. Hartl, M. Reimann, C. Strauss, and M. Stummer, "SavingsAnts for the vehicle routing problem," in *Applications of Evolutionary Computing* (Lecture Notes in Computer Science), vol. 2279. Berlin, Germany: Springer, 2001, pp. 11–20.
[33] E. Demir, T. BektaŞ, and G. Laporte, "An adaptive large neighborhood search heuristic for the pollution-routing problem," *Eur. J. Oper. Res.*, vol. 223, no. 2, pp. 346–359, Dec. 2012.
[34] D. C. Paraskevopoulos, P. P. Repoussis, C. D. Tarantilis, G. Ioannou, and G. P. Prastacos, "A reactive variable neighborhood tabu search for the heterogeneous fleet vehicle routing problem with time windows," *J. Heuristics*, vol. 14, no. 5, pp. 425–455, Oct. 2008.
[35] M. Bruglieri, F. Pezzell, O. Pisacane, and S. Suraci, "A variable neighborhood search branching for the electric vehicle routing problem with time windows," *Electron. Notes Discrete Math.*, vol. 47, no. 15, pp. 221–228, 2015.

HUITING MAO received the B.S. degree from Hunan Normal University, Changsha, China, in 2018. She is currently pursuing the M.S. degree with the National University of Defense Technology. Her research interests include vehicle routing and heuristic design.

JIANMAI SHI received the Ph.D. degree in management science and engineering from the National University of Defense Technology (NUDT), Changsha, China, in 2011. From 2008 to 2009, he was a Visiting Ph.D. Student with the University of Windsor, Canada. From 2014 to 2015, he was a Visiting Scholar at McGill University. He is currently an Associate Professor with the School of System Engineering, NUDT. He has authored more than 30 refereed articles, including those published in the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, COR, and *Applied Energy*. His current research interests include vehicle routing, supply chain management, and heuristic design.

YUZHEN ZHOU received the B.S. degree from Anyang Normal University, Anyang, China, in 2014, and the M.S. degree in computational mathematics from Zhengzhou University, Zhengzhou, China, in 2017. She is currently pursuing the Ph.D. degree with the National University of Defense Technology. Her current research interests include vehicle routing and heuristic design.

GUOQING ZHANG received the Ph.D. degree in management sciences from the City University of Hong Kong. He is currently a Professor with the University of Windsor. His main research interests include optimization, decision support systems, supply chain management, the modeling of manufacturing, operational research, and operations management.