



# A branch-and-price algorithm for two-echelon electric vehicle routing problem

Zhiguo Wu<sup>1</sup> · Juliang Zhang<sup>1</sup>

Received: 5 April 2021 / Accepted: 17 May 2021 / Published online: 28 May 2021  
© The Author(s) 2021

## Abstract

Motivated by express and e-commerce companies' distribution practices, we study a two-echelon electric vehicle routing problem. In this problem, fuel-powered vehicles are used to transport goods from a depot to intermediate facilities (satellites) in the first echelon, whereas electric vehicles, which have limited driving ranges and need to be recharged at recharging stations, are used to transfer goods from the satellites to customers in the second echelon. We model the problem as an arc flow model and decompose the model into a master problem and pricing subproblem. We propose a branch-and-price algorithm to solve it. We use column generation to solve the restricted master problem to provide lower bounds. By enumerating all the subsets of the satellites, we generate feasible columns by solving the elementary shortest path problem with resource constraints in the first echelon. Then, we design a bidirectional labeling algorithm to generate feasible routes in the second echelon. Comparing the performance of our proposed algorithm with that of CPLEX in solving a set of small-sized instances, we demonstrate the former's effectiveness. We further assess our algorithm in solving two sets of larger scale instances. We also examine the impacts of some model parameters on the solution.

**Keywords** Two-echelon · Electric vehicles · Branch-and-price algorithm · Column generation · Labeling algorithm

## Introduction

With the depletion of fossil energy and the urgent need for environmental protection, increasing nations have used electric vehicles (EVs) to replace fuel-powered vehicles for urban transport and logistics. For example, the European Commission's Urban Mobility Package aims to achieve zero emissions of urban logistics by 2030 [20], and its partners agree on a green deal about zero-emission urban logistics in the Netherlands [24]. This means that the use of gasoline- or diesel-powered freight vehicles will be limited in many cities. Consequently, the sales of new energy vehicles have registered phenomenal growth in key markets such as Europe, the Middle East, China, and Australia [31]. The cumulative sales of new energy vehicles in China reached 2.7 million in 2018 [41]. In this environment, many logistics (e.g., SF express) and e-commerce companies (e.g., JD.com, Amazon) have used EVs for logistics and distribution in cities

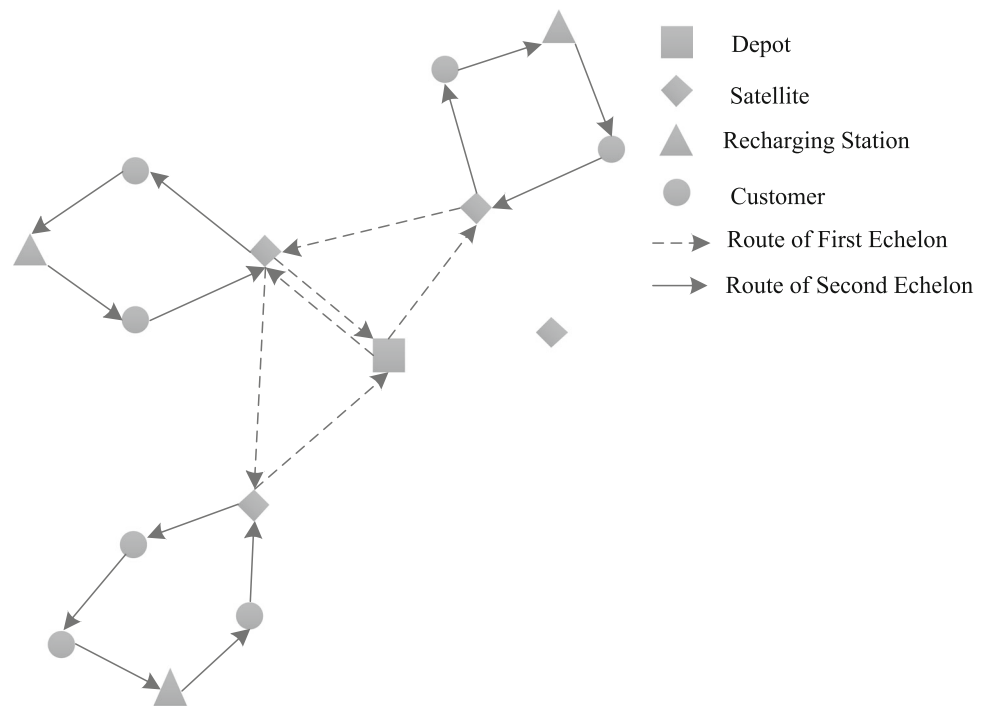
[51]. Different from gasoline- and diesel-powered vehicles, EVs have limited driving ranges and need en-route recharging, which rely on the volume of batteries and the limited recharging infrastructure because of the immature battery technology. The adoption of EVs gives rise to new challenges for the vehicle scheduling and routing problem, which motivates research on tailoring models to capture the limitations of EVs. Specifically, EVs need to decide when and which recharging stations to recharge when they are running out of battery power. As a result, many studies on the electric vehicle routing problem (EVRP) have recently appeared in the literature [21,26,36,39]. The majority of such works on the EVRP only consider a single echelon where the distribution center delivers goods to the customers directly.

In contrast, the two-echelon (or multi-echelon) distribution system is very efficient for city logistics and e-commerce logistics [13,15]. For a B2C e-commerce firm, customers place orders online on its platform, and then, the orders are picked and collected in the depot, which are generally located on the outskirts of cities. The packages are transported to satellites that are located close to the city centers. After that, the packages are assigned to distributors to deliver to customers. Such a two-echelon distribution system can keep

✉ Zhiguo Wu  
17113144@bjtu.edu.cn

<sup>1</sup> School of Economics and Management, Beijing Jiaotong University, Beijing, China

**Fig. 1** An instance of the 2E-EVRP



large fossil fuel-powered vehicles out of the city centers and use EVs with lower pollution to transfer the packages from the satellites to the customers. Motivated by this real practice of transport logistics, we study the two-echelon electric vehicle routing problem (2E-EVRP) to help express and e-commerce companies address their key transport planning and scheduling problem. In the first echelon, fossil fuel-powered trucks transport goods from the depot to a subset of satellites. The second echelon consists of transferring goods from the satellites to the final customers using EVs.

Figure 1 shows an instance with one depot, four satellites, seven customers, and three recharging stations. Our model for the 2E-EVRP is to craft an optimal transport plan to minimize the total cost. We use the arc flow formulation to model the problem and design an exact algorithm to solve it.

We make the following contributions in this study:

(1) Motivated by the distribution systems adopted by B2C firms and express firms, we propose the 2E-EVRP and build an arc flow model.

(2) We develop an efficient branch-and-price (BP) algorithm to solve the 2E-EVRP. Specifically, we decompose the arc flow model as an integer master problem and apply relaxations of the integer master problem to derive lower bounds for the optimal solution. We design a bidirectional labeling algorithm to generate feasible routes with recharging stations in the second echelon. We adopt the state-space relaxation and the *ng*-route relaxation as acceleration strategies to reduce the time required to solve the subproblem.

(3) Comparing the performance of the BP algorithm with that of CPLEX in solving a set of small-scale instances, we

demonstrate the former's effectiveness. We further assess the performance of the BP algorithm in solving two sets of larger scale instances. We also examine the impacts of some model parameters on the solution and cost.

The rest of the paper is organized as follows: We give a review to identify the research gap in Sect. 2. We formulate the 2E-EVRP as an arc flow model and decompose the model as an integer master problem and apply relaxation to derive the lower bounds in Sect. 3. We propose a BP algorithm with accelerate strategies in Sect. 4. We present the results of numerical studies in Sect. 5. Finally, we conclude the paper and give suggestions for future research in Sect. 6.

## Literature review

There are many variants of VRP [6,19,25]. However, there are three streams of literature related to our work, namely the two-echelon vehicle routing problem (2E-VRP), the electric vehicle routing problem (EVRP), and the two-echelon electric vehicle routing problem (2E-EVRP).

## Two-echelon vehicle routing problem

Crainic et al. [12] considered the 2E-VRP in early time, which is formally introduced by Perboli et al. and Crainic et al. [11,45]. Cuda et al. [14] gave an extensive review of the problem. In the following, we only review the works that are most closely related to our study.

There are some studies that provide exact algorithms for the problem. Perboli and Tadei [43] proposed two meta-heuristics and some new families of valid inequalities derived from the traveling salesman problem (TSP) and the vehicle routing problem (VRP). Jepsen et al. [33] presented an edge flow model and applied a relaxation, which is similar to the formulation introduced by Perboli et al. [45]. Based on this, they proposed a branch-and-cut algorithm to solve the problem. Baldacci et al. [5] built a new model of the 2E-VRP and derived some new lower bounds. Based on this, they proposed an exact method that transforms the 2E-VRP into multi-depot capacitated vehicle routing problems (MDCVRPs) with a limited set. Dellaert [15] proposed an arc-based formulation and developed a branch-and-price algorithm that can solve instances of up to five satellites and 100 customers. Perboli et al. [44] further gave two groups of valid inequalities: one was an extension of the existing VRP valid inequalities and the other consisted of explicit valid inequalities for the 2E-VRP.

Because of the hard-solving of the problem, researchers have developed heuristics algorithms for it. Hemmelmayr et al. [29] proposed an adaptive large neighborhood search (ALNS) heuristic for the 2E-VRP. Breunig et al. [8] presented a meta-heuristic, in which some tailored operators were designed to optimize the selection of satellites. Grangier et al. [28] considered integrated constraints (e.g., time windows, synchronization, multiple trips) in the second echelon and proposed an ALNS to solve the problem. Li et al. [37] adopted Clarke and Wright's savings heuristic algorithm, and further improved through a local search phase.

The above works assume that the transport in both echelons performed by gasoline- or diesel- powered vehicles, and do not consider the vehicle en-route recharging. In contrast, we consider the 2E-EVRP, where EVs are used in the second echelon, so we need to consider when and where to recharge the EVs.

## Electric vehicle routing problem

Since 2000, green supply chain [1,17,18,52] and city logistics [27,38,48] have attracted extensive interest, and the use of EVs is thought of as an effective means for realizing green distribution. There has been a growing interest in the EVRP, since Gonçalves et al. [26] first studied it. Pelletier et al. [42] gave a comprehensive review and suggested some research perspectives for the EVRP. In the following, we only review the works that are most closely related to our study.

Liao et al. [39] considered the battery charging or swapping in the EVRP, proposed a dynamic programming to solve the EVRP with the shortest travel time, and developed an algorithm for the fixed tour EV touring problem. Hiermann et al. [30] studied a mixed vehicle routing

problem that involves various types of EVs with different acquisition costs, driving ranges, and payloads. The results showed that the type and number of EVs to be used depend on the typical customer distribution in the urban area. Desaulniers et al. [16] considered four variants of the EVRP and designed customized branch-and-price algorithms for each variant. Montoya et al. [40] studied the EVRP with piecewise linear charging functions and proposed a hybrid meta-heuristic to solve it. Zhang et al. [53] introduced a new model considering the energy consumption caused by coal-fired generation. Keskin and Çatay [35] compared the total recharging costs under three recharging configurations, i.e., normal, fast, and super-fast, where partial recharging was permitted. Froger et al. [22] further studied the EVRP with a general nonlinear charging function, assuming that the EVs can be fully or partially recharged. Cortés-Murcia et al. [10] proposed a new variant that allows at most one customer to visit when an EV is recharging en-route. Hua and Xu [23] studied the location of charging stations to serve all EVs in a given area. Tahami et al. [49] provided a compact formulation of the EVRP, and three exact methods were introduced to solve the formulation and its augmented variants. The above works focus on the single-echelon EVRP. In contrast, we consider the 2E-EVRP in which EVs are used in the second echelon, while fossil fuel-powered trucks are used in the first echelon.

## Two-echelon electric vehicle routing problem

To our best knowledge, there are only three studies on the 2E-EVRP up to now [7,34,50]. Jie et al. [34] studied the 2E-EVRP in which the battery of EV can be replaced at the stations. They proposed a hybrid algorithm combining the column generation (CG) with an ALNS to deal with the problem. Wang et al. [50] studied the 2E-EVPR with time windows of satellites and customers. The randomly generated instances were solved by GUROBI. In our paper, we focus on the exact algorithm to find the optimal solution of 2E-EVRP.

Breunig et al. [7] studied the 2E-EVRP in which EVs were only used in the second echelon, and proposed an exact algorithm for it based on the algorithm proposed by Baldacci et al. [5] for 2EVRP. The algorithm consists of two main steps: (i) generate the set of all first echelon routes and calculate an initial lower bound; (ii) fix the selected set of first echelon routes and transform the resulting problem into MDCVRP, which is further solved by the procedure proposed in Baldacci and Mingozzi [2]. Different from them, we propose an arc flow model and decompose the model into an integer master problem and a pricing subproblem. We then design a branch-and-price algorithm to solve the problem.

### Problem formulation

In this section, we formulate the 2E-EVRP as an arc flow model and decompose the model as an integer master problem. The relaxation of the integer master problem is used to derive lower bounds for the optimal solution.

### Arc flow model

We model the 2E-EVRP on graph  $G = \{V, A\}$ . The vertex set  $V = \{0\} \cup S \cup C \cup F$ , where 0 represents the depot,  $S$  and  $C$  represents the sets of satellites and customers, respectively, and  $F$  is the set of recharging stations at second echelon. The arc set  $A = A_1 \cup A_2$  consists of two subsets, namely the set of arcs at first echelon  $A_1 = \{(i, j)|i, j \in \{0\} \cup S, i \neq j\}$  and the set of arcs at second echelon  $A_2 = \{(i, j)|i, j \in S \cup C \cup F, i \neq j\} \setminus A'_2$ , where  $A'_2 = \{(i, j)|i, j \in S\} \cup \{(i, j)|i, j \in F\}$ .

Suppose the trucks and the EVs are located at the depot and satellites, respectively. The trucks are homogeneous and their load capacity is  $Q_1$ . The EVs are also homogeneous, and their load capacity and battery capacity are  $Q_2$  and  $Q_3$ , respectively. When an EV travels on arc  $(i, j)$ , it incurs transport cost  $c_{ij}$  and electricity consumption  $e_{ij}$ . We assume that  $c_{ij}$  and  $e_{ij}$  are linear functions of the distance  $d_{ij}$ , i.e.,  $c_{ij} = \alpha d_{ij}$  and  $e_{ij} = \beta d_{ij}$ ,  $\alpha, \beta > 0$ . Let  $E$  be the maximum recharging time that is required to recharge  $Q_3$  units of electricity, we have  $E = \gamma Q_3$ , where  $\gamma > 0$  is the charging rate. Here, we assume that  $\alpha = \beta = \gamma = 1$  to simplify the presentation and analysis. In fact, the values of these parameters do not affect the effectiveness of our proposed algorithm.

Let the fixed costs of a truck and an EV be  $u_1$  and  $u_2$ , respectively. Suppose the battery is fully recharged on a visit to a recharging station and an EV can visit a recharging station many times. However, consecutive visits to two recharging stations are forbidden.

Each customer  $i \in C$  has a demand  $q_i$ . We assume that the depot and satellites are large enough to handle the demand of all customers. The satellites are intermediate stations that are used to transfer goods from the depot to the customers. Each route of the first echelon starts from and ends with the depot after visiting some satellites. The goods delivered by a truck cannot exceed the load capacity  $Q_1$ . Each satellite can be visited by more than one truck. However, a satellite can be visited by a truck no more than once. Any route of the second echelon starts from a satellite, visits a subset of customers and recharging stations, and then returns to the same satellite. The total quantity of goods delivered by an EV cannot exceed the load capacity  $Q_2$  and the remaining battery power of EVs cannot be negative. Each customer can only be visited once. Table 1 summarizes the notation used throughout the paper. Then, the 2E-EVRP problem can be modeled as the following mixed integer programming (MIP):

**Table 1** The notation used in the paper

Variable	Definition
0	Depot;
$S, C, F$	Sets of satellites, customers and recharging stations, respectively;
$A_1, A_2$	Arcs of the first and second echelon, respectively;
$Q_1, Q_2$	The capacity of trucks and EVs, respectively;
$Q_3$	Maximum battery capacity of EVs;
$K_1, K_2$	Set of trucks and EVs;
$k_1, k_2$	Number of trucks and EVs, respectively;
$c_{ij}$	Transport cost of arc $(i, j)$ ;
$e_{ij}$	Energy consumption of arc $(i, j)$ ;
$q_i$	Demand of customer $i \in C$ ;
$d_{ij}$	Distance between vertex $i$ and vertex $j$ ;
$h_{isk}^+, h_{isk}^-$	The remaining battery power when EV $k$ arrives and leaves vertex $i$ from satellite $s$ , respectively;
$b_{ik}$	The amount of goods delivered to satellite $i$ by truck $k$ ;
$x_{ijk}$	Binary decision variable demonstrating whether truck $k$ passes through arc $(i, j)$ ;
$y_{ijsk}$	Binary decision variable demonstrating whether EV $k$ from satellite $s$ passes through arc $(i, j)$ .

$$\text{Min } \sum_{k \in K_1} \sum_{(i,j) \in A_1} c_{ij} x_{ijk} + \sum_{k \in K_1} \sum_{j \in S} u_1 x_{0jk} + \sum_{s \in S} \sum_{k \in K_2} \sum_{(i,j) \in A_2} c_{ij} y_{ijsk} + \sum_{s \in S} \sum_{k \in K_2} \sum_{(s,j) \in A_2} u_2 y_{sjk} \tag{1}$$

$$\sum_{(0,j) \in A_1} x_{0jk} \leq 1 \quad \forall k \in K_1 \tag{2}$$

$$\sum_{(i,j) \in A_1} x_{ijk} = \sum_{(j,i) \in A_1} x_{jik} \quad \forall i \in S, k \in K_1 \tag{3}$$

$$\sum_{i \in S} b_{ik} \leq Q_1 \sum_{(0,j) \in A_1} x_{0jk} \quad \forall k \in K_1 \tag{4}$$

$$\sum_{k \in K_1} b_{sk} = \sum_{k \in K_2} \sum_{i \in C} \sum_{(i,j) \in A_2} q_i y_{ijsk} \quad \forall s \in S \tag{5}$$

$$\sum_{\bar{s} \in S \setminus \{s\}} \sum_{k \in K_2} ( \sum_{(s,j) \in A_2} y_{sj\bar{s}k} + \sum_{(i,s) \in A_2} y_{is\bar{s}k} ) = 0 \tag{6}$$

$$\sum_{s \in S} \sum_{k \in K_2} \sum_{(i,j) \in A_2} y_{ijsk} = 1 \quad \forall j \in C \tag{7}$$

$$\sum_{(i,j) \in A_2} y_{ijsk} = \sum_{(j,i) \in A_2} y_{jisk} \quad \forall j \in C \cup F, s \in S, k \in K_2 \tag{8}$$

$$\sum_{i \in C} \sum_{(i,j) \in A_2} q_i y_{ijsk} \leq Q_2 \quad \forall s \in S, k \in K_2 \tag{9}$$

$$h_{jsk}^+ + e_{ij} - h_{isk}^- \leq Q_3(1 - y_{ijsk}) \quad \forall (i, j) \in A_2, s \in S, k \in K_2 \tag{10}$$

$$h_{isk}^- = Q_3 \quad \forall i \in S \cup F, s \in S, k \in K_2 \tag{11}$$

$$h_{isk}^+ = h_{isk}^- \quad \forall i \in C, s \in S, k \in K_2 \tag{12}$$

$$x_{ijk} \in \{0, 1\} \tag{13}$$

$$\forall (i, j) \in A_1, k \in K_1$$

$$y_{ijsk} \in \{0, 1\} \quad \forall (i, j) \in A_2, s \in S, k \in K_2 \tag{14}$$

$$h_{isk}^+, h_{isk}^- \geq 0 \tag{15}$$

$$\forall i \in V \setminus \{0\}, s \in S, k \in K_2$$

$$b_{ik} \geq 0 \quad \forall i \in S, k \in K_1. \tag{16}$$

The objective function (1) is to minimize the total cost including the fixed cost and the transport cost in both echelons. Constraints (2) guarantee that a truck visits a satellite no more than once. Constraints (3) and (8) show the flow balance of vehicles. Constraints (4) and (9) ensure the capacity of vehicles cannot be exceeded. Constraints (5) indicate the balance of goods receiving and delivery, which means that all goods received by a satellite from the depot must be delivered to customers. Constraint (6) prevents the situation that an EV returns to a different satellite from the starting satellite. Constraints (7) specify that each customer can only be visited once. Constraints (10) reveals the relationship of battery power when an EV transfers from vertex  $i$  to vertex  $j$ , which also eliminates the sub-tours in the route. Constraints (11) impose that the battery power is full when an EV departs from a satellite or a recharging station. Constraints (12) keep the battery power of an EV the same when it arrives at or leaves a customer.

### Dantzig–Wolfe decomposition

We shall use the idea of Dantzig–Wolfe decomposition to propose an algorithm for the problem. The MIP model (1)–(16) will be decomposed into an integer master problem and a pricing subproblem by the Dantzig–Wolfe decomposition. Let  $R_1$  and  $R_2$  be the feasible routes sets of the first and second echelon, respectively. Let  $R_2^s$  be the set of feasible routes in the second echelon starting from and ending at the satellite  $s$ , where  $R_2 = \bigcup_{s \in S} R_2^s$ .

Let  $\lambda_r$  be an integer variable that indicates the times route  $r \in R_1$  is implemented. For each route  $r \in R_1 \cup R_2$ ,  $c_r$  denotes its transport cost and  $a_r^i (i \in V \setminus \{0\})$  equals to 1 if vertex  $i$  is visited by route  $r$  and 0 otherwise. Let  $p_r^s$  be the quantity of the goods delivered to satellite  $s$  using route  $r \in R_1$ . For a route  $r \in R_2^s$ ,  $\theta_r^s$  is a decision variable that is equal to 1 if route  $r$  is a component of the optimal distribution strategy and 0 otherwise. Table 2 summarizes the notation of the integer master problem used throughout the paper. The integer master problem (IMP) is shown as follows:

**Table 2** The notation used in the integer master problem

Variable	Definition
$R_1, R_2$	Sets of feasible routes of the first and second echelons, respectively;
$R_2^s$	Set of feasible routes in the second echelon starting from and ending at satellite $s \in S$ ;
$c_r$	Transport cost of route $r \in R_1 \cup R_2$ ;
$a_r^i$	Binary variable indicating whether vertex $i$ is visited by route $r \in R_1 \cup R_2$ ;
$\lambda_r$	Decision variable indicating the times route $r \in R_1$ is implemented;
$p_r^s$	Decision variable indicating how many goods does route $r \in R_1$ transport to satellite $s$ ;
$\theta_r^s$	Binary decision variable indicating whether a route $r \in R_2^s$ is implemented.

$$\text{Min} \sum_{r \in R_1} (c_r + u_1)\lambda_r + \sum_{s \in S} \sum_{r \in R_2^s} (c_r + u_2)\theta_r^s \tag{17}$$

$$\sum_{r \in R_2^s} \theta_r^s \sum_{i \in C} a_r^i q_i = \sum_{r \in R_1} p_r^s \quad \forall s \in S \tag{18}$$

$$\sum_{s \in S} \sum_{r \in R_2^s} a_r^i \theta_r^s = 1 \quad \forall i \in C \tag{19}$$

$$\sum_{s \in S} a_r^s p_r^s \leq Q_1 \lambda_r \quad \forall r \in R_1 \tag{20}$$

$$\sum_{r \in R_1} \sum_{s \in S} (1 - a_r^s) p_r^s = 0 \tag{21}$$

$$\lambda_r \in \mathbb{N}_+ \quad \forall r \in R_1 \tag{22}$$

$$\theta_r^s \in \{0, 1\} \forall r \in R_2^s \quad \forall s \in S \tag{23}$$

$$p_r^s \in \mathbb{N}_+ \quad \forall r \in R_1, s \in S. \tag{24}$$

The objection function (17) is the sum of the fixed cost and transport cost in both echelons. Constraints (18) ensure the flow balance at each satellite, which means the total amount of goods received by a satellite from the depot equals that shipped from the satellite to customers. Constraints (19) ensure that each customer is visited only once in the second echelon. Constraints (20) guarantee that the capacity of the trucks is not exceeded. Constraint (21) imposes the relationship between  $a_r^s$  and  $p_r^s$  that a route of the first echelon cannot deliver goods to a satellite if the route does not visit it. Constraints (22)–(24) restrict the domains of the associated variables.

We relax constraints (22) and (23) as  $\lambda_r \geq 0$  and  $\theta_r^s \geq 0$  to generate lower bounds for the model. The linear relaxation of the IMP (17)–(24) is called the master problem. The lower bound of the master problem can be strengthened by imposing the capacity constraint



$$\sum_{r \in R_1} \lambda_r \geq \left\lceil \frac{\sum_{i \in C} q_i}{Q_1} \right\rceil, \quad (25)$$

which imposes a minimum number of vehicles used at the first echelon to transport the goods from the depot to satellites.

## Branch-and-price algorithm

For model (17)–(25), there are too many variables to solve by a standard MIP solver or the branch-and-bound algorithm.

### Algorithm 1 The BP algorithm

**Input:** Data on the depot, satellites, customers, and vehicles, and the maximum running time  $T_{\max}$ ;

**Output:** The optimal solution  $r^*$ ;

- 1: Initialize the columns set  $\zeta = \emptyset$  and the search tree  $\mathcal{T} = \emptyset$ ;
- 2: Call **IH** to generate a feasible solution  $\psi$  of the second echelon and get the initial columns set  $\zeta$ . Calculate  $UB_0$ ,  $UB \leftarrow UB_0$  (see Sect. 4.2);
- 3: Call **CG** to solve  $P^0$ , and get the solution  $S^0$  and its cost  $C(P^0)$  (see Sect. 4.3);
- 4:  $LB \leftarrow C(P^0)$ ,  $\mathcal{T} \leftarrow \mathcal{T} \cup P^0$ ;
- 5: **while**  $T \leq T_{\max}$  and  $UB \neq LB$  and  $\mathcal{T} \neq \emptyset$  **do**
- 6:   Choose a branch node  $P$  such that  $C(P) = \min_{P' \in \mathcal{T}} \{C(P')\}$ . Obtain its solution  $S$ ;
- 7:    $P^0 \leftarrow P$ ,  $S^0 \leftarrow S$ ,  $\mathcal{T} \leftarrow \mathcal{T} \setminus P^0$ ;
- 8:   **if**  $S^0$  is an integer solution and  $C(P^0) < UB$  **then**
- 9:      $UB \leftarrow C(P^0)$  and prune the branch nodes  $\bar{P}$  such that  $C(\bar{P}) \geq UB$ ;
- 10:   **else if**  $S^0$  is a non-integer solution **then**
- 11:     Branch  $P^0$  into  $P^1$  and  $P^2$ , call **CG** to get the optimal solutions  $S^1$  and  $S^2$  and their costs  $C(P^1)$  and  $C(P^2)$ ;
- 12:      $\mathcal{T} \leftarrow \mathcal{T} \cup \{P^1, P^2\}$ ,  $LB \leftarrow \min_{P' \in \mathcal{T}} \{C(P')\}$ ;
- 13:   **end if**
- 14: **end while**
- 15: **return**  $r^*$

Therefore, we propose a BP algorithm for it based on its structure. Noting that each route  $r \in R_1 \cup R_2$  corresponds to a variable ( $\lambda_r$  or  $\theta_r^s$ ) in the master problem, we keep the set of routes  $R_1$  unchanged and replace the sets  $R_2^s (s \in S)$  by the initial subsets of routes  $\widehat{R}_2^s \subset R_2^s (s \in S)$ , where  $|\widehat{R}_2^s| \ll |R_2^s|$ . We call the master problem with the variables (columns)  $\lambda_r (r \in R_1)$  and  $\theta_r^s (s \in S, r \in \widehat{R}_2^s)$  the *restricted master problem (RMP)*. Then, we adopt the column generation (CG) algorithm to solve the RMP. We branch the RMP into two subproblems based on a branching strategy if the solution of the RMP is fractional. Therefore, our algorithm comprises three main components: (i) an insertion heuristic (IH) to generate the initial subsets of routes  $\widehat{R}_2^s \subset R_2^s (s \in S)$ , (ii) a CG algorithm that evaluates the lower bounds of IMP

(17)–(25) by solving the RMP, and (iii) a branching strategy that forces the solution of the RMP to be integers.

## The branch-and-price algorithm framework

In this subsection, we discuss the BP algorithm, which is called Algorithm 1. At first, the initial set of variables (columns)  $\zeta$  is generated. Based on the initial set of columns  $\zeta$ , we call CG to calculate the new lower bound. Then, we explore the branch nodes of search tree  $\mathcal{T}$  and update the lower bound and upper bound. The optimal solution  $r^*$  is obtained when the lower bound  $LB$  equals the upper bound  $UB$ .

Since the number of satellites is generally relatively small, we enumerate all the subsets of the satellites with the depot in the first echelon. Then, we find the shortest route of each subset that starts from and ends at the depot by solving the ESPPRC. These shortest routes constitute the set of the first echelon routes  $R_1$ . In the second echelon, we apply an insertion heuristic (Algorithm 2) to generate a feasible solution  $\psi$  of the second echelon, where  $\psi = \bigcup_{s \in S} \widehat{R}_2^s$ . Then, we get the initial set of variables as  $\zeta = \{\lambda_r | r \in R_1\} \cup \{\theta_r^s | s \in S, r \in \widehat{R}_2^s\}$ .

Once the feasible solution  $\psi$  of the second echelon is fixed, the 2E-EVRP can be transformed into solving the split delivery vehicle routing problem at the first echelon. We use the CPLEX to solve this problem and derive the initial upper bound  $UB_0$ .

We denote  $\mathcal{T}$  as the search tree to represent the original problem and its subproblems. The RMP with the initial set of columns is regarded as the original problem and is thought of as the root node of  $\mathcal{T}$ . In each iteration, we select an unfathomed node from  $\mathcal{T}$  which is denoted as  $P^0$ . The optimal solution and its corresponding value of  $P^0$  are denoted as  $S^0$  and  $C(P^0)$ , respectively. The upper bound in the current iteration is  $UB$ . If it is an integer solution, the  $UB$  will be updated and the unnecessary branch nodes will be pruned. If  $S^0$  is not an integer solution and  $C(P^0) < UB$ , we branch

## Insertion heuristic algorithm

In this subsection, we design an insertion heuristic (IH) algorithm to obtain a feasible solution  $\psi$  of the second echelon, which consists of the routes that satisfy the resource constraints without en-route recharging. The algorithm consists of two steps: (i) assign customers to their nearest satellites; (ii) route the assigned customers of each satellite. We summarize IH algorithm as Algorithm 2.

### Algorithm 2 Insertion heuristic

**Input:** Distance  $d_{ij}$ , cost  $c_{ij}$ , and electricity consumption  $e_{ij}$  for each arc  $(i, j) \in A_2$ , demand  $q_i$  for each customer  $i \in C$ , EVs' capacity  $Q_2$ , and the maximum recharging time  $E$ ;

**Output:** A feasible solution  $\psi$  of the second echelon;

---

```

1: For each  $s \in S$ , initialize set  $\Gamma_s = \emptyset$ ,  $D_s = \emptyset$ ;
   ***** assign customers to its nearest satellite *****
2: for  $i \in C$  do
3:    $\varphi_i = \emptyset$ ;
4:   for  $s \in S$  do
5:     Calculate the distance  $d_{is}$  between customer  $i$  and satellite  $s$ ;
6:      $\varphi_i \leftarrow \varphi_i \cup \{d_{is}\}$ ;
7:   end for
8:    $d_{ik} = \min_{s \in S} \{d_{is} | d_{is} \in \varphi_i\}$ ;
9:    $\Gamma_k \leftarrow \Gamma_k \cup \{i\}$ ,  $D_k \leftarrow D_k \cup \{d_{ik}\}$ ;
10: end for
   ***** route the assigned customers of each satellite *****
11: for  $s \in S$  do
12:    $n = 1$ ;
13:   while  $\Gamma_s \neq \emptyset$  do
14:      $P_s^n = \{s, \bar{i}, s\}$ ,  $T^{load} = q_{\bar{i}}$ ,  $T^{rt} = e_{s\bar{i}} + e_{\bar{i}s}$ ;
15:     repeat
16:       Insert customer  $u \in \Gamma_s$  with the minimum insertion cost into route  $P_s^n$ , such
       that the load capacity and battery capacity constraints are satisfied, and
       remove  $u$  from  $\Gamma_s$ ;
17:     until no such customer in  $\Gamma_s$  exists
18:      $n \leftarrow n + 1$ ;
19:   end while
20: end for
21: return  $\psi = \bigcup_{s \in S} P_s^n$ 

```

---

$P^0$  into two subproblems  $P^1$  and  $P^2$  (child nodes) and add  $P^1$  and  $P^2$  to the search tree  $\mathcal{T}$ . The optimal solutions of  $P^1$  and  $P^2$  can be obtained by invoking the CG algorithm. Since  $P^0$  is fathomed, we remove  $P^0$  from the search tree  $\mathcal{T}$ . Then, the lower bound will be updated.

## Column generation algorithm

In this subsection, we describe the column generation (CG) algorithm that generates lower bounds for the IMP (17)–(25). To state the algorithm, we first introduce the pricing subproblem, which aims at finding a column with the minimum reduced cost.

For a given satellite  $s \in S$ , let  $\alpha_s$  and  $\pi_i$  ( $i \in C$ ) be the dual variables associated with constraints (18) and (19), respectively. Suppose there is a route  $r \in R_2$  starting from satellite  $s$ . Let  $y_{ij}$  be a binary variable to indicate whether this route passes through arc  $(i, j)$ .  $h_{ij}$  means the remaining battery power when EV leaves vertex  $i$ .  $a_i$  denotes whether customer  $i$  is visited by this route. Then, the pricing subproblem is shown as follows:

$$\min \bar{c} = \sum_{(i,j) \in A_2} c_{ij}y_{ij} + u_2 - \alpha_s \sum_{i \in C} a_i q_i - \sum_{i \in C} a_i \pi_i \quad (26)$$

$$\sum_{j \in C \cup F} y_{ij} = a_i \quad \forall i \in C \quad (27)$$

$$\sum_{i \in C \cup F} y_{ij} = a_j \quad \forall j \in C \quad (28)$$

$$\sum_{j \in C \cup F} y_{sj} = 1 \quad (29)$$

$$\sum_{i \in C \cup F} y_{is} = 1 \quad (30)$$

$$\sum_{(i,j) \in A_2} y_{ij} = \sum_{(j,i) \in A_2} y_{ji} \quad \forall i \in C \cup F \quad (31)$$

$$\sum_{i \in C} q_i a_i \leq Q_2 \quad (32)$$

$$h_j^+ + e_{ij} - h_i^- \leq Q_3(1 - y_{ij}) \quad \forall (i, j) \in A_2 \quad (33)$$

$$h_i^- = Q_3 \quad \forall i \in S \cup F \quad (34)$$

$$h_i^+ = h_i^- \quad \forall i \in C \quad (35)$$

$$y_{ij} \in \{0, 1\} \quad \forall (i, j) \in A_2 \quad (36)$$

The objective function (26) seeks for the column with minimum reduced cost. Constraints (27)–(28) imply the relationship between  $y_{ij}$  and  $a_i$ . Constraints (29)–(30) restrict the route starting from and ending at the same satellite. Constraints (31)–(35) are the same as constraints (8)–(12). Constraints (36)–(38) are the variable domains.

The pricing subproblem is subject to cost, capacity, and electricity constraints, which are called resource constraints. We can re-formulate the subproblem as an ESPPRC in  $G'=(V', A_2)$  where  $V' = S \cup F \cup C$ . Since the ESPPRC is strongly NP-hard, we adopt dynamic programming to solve it.

To compute the reduced costs of the feasible routes in  $G'$ , we replace the arc cost  $c_{ij}$  by a modified cost  $\bar{c}_{ij}$  as follows:

$$\bar{c}_{ij} = \begin{cases} c_{ij} + 0.5u_2 - 0.5\alpha_s q_j - 0.5(\pi_i + \pi_j) & \text{if } i \in S, j \in C \cup F \\ c_{ij} - 0.5\alpha_s(q_i + q_j) - 0.5(\pi_i + \pi_j) & \text{if } i, j \in C \cup F \end{cases} \quad (39)$$

The variables  $\pi_i$  and  $q_i$  are set to 0 if  $i \in S \cup F$ . For any feasible route  $r$ , the reduced cost  $\bar{c}_r$  equals the sum of the modified costs of the arcs in the route.

The CG algorithm is an iterative algorithm that alternates between solving the RMP and the pricing subproblem. We denote the set of feasible routes as  $R'_2$ , which is generated by a labeling algorithm.

**Algorithm 3** CG Algorithm

**Input:**The RMP of a branch node and its column set  $\zeta$ ;

**Output:**The optimal solution  $S^*$ ;

```

1: do {
2:   Solve the RMP to get a solution  $S$ , and dual variables  $\alpha_s$  and  $\pi_i$ ;
3:   Use the dual variables  $\alpha_s$  and  $\pi_i$  to construct the pricing subproblem;
4:   Call the labeling algorithm to generate a feasible route set  $R'_2$  (see Sect. 4.3.1);
5:   Calculate the reduced cost  $\bar{c}_r$  for each  $r \in R'_2$  and  $\bar{c}_k = \min_{r \in R'_2} \{\bar{c}_r\}$ ;
6:   if  $\bar{c}_k \geq 0$  then
7:      $S^* \leftarrow S$ ;
8:     break;
9:   else
10:     $\zeta \leftarrow \zeta \cup \{\theta_k^s\}$ ;
11:  end if
12: } while ( $\bar{c}_k \leq 0$ );
13: return  $S^*$ 

```

$$a_i \in \{0, 1\} \quad \forall i \in C \quad (37)$$

$$h_i^+, h_i^- \geq 0 \quad \forall i \in V \setminus \{0\}. \quad (38)$$

Let  $S$  be a feasible solution and  $S^*$  be the optimal solution of the RMP, respectively. We present the CG algorithm in detail in Algorithm 3.



### Labeling algorithm

In this subsection, we discuss the labeling algorithm that generates a set of feasible routes  $R'_2$  of the second echelon. Each feasible route in  $R'_2$  starts from a satellite and ends at the same satellite, and satisfies a set of resource constraints. To present conveniently, we call the starting satellite  $s$  and the ending satellite  $t$ . Therefore, each feasible route corresponds to an  $s - t$  path. In the labeling algorithm, each label of vertex  $i$  represents a partial path from a satellite  $s$  to vertex  $i$ . Labels extend forwardly using the resource extension functions (REFs) and those that violate the resource constraints will be discarded [32]. Furthermore, a dominance rule is introduced to eliminate partial paths from which the  $s - t$  path with the minimum reduced cost cannot be yielded by extending them.

Because of the symmetry of our model, we use a bidirectional search to generate the set of feasible routes of the second echelon. The bidirectional search consists of two mono-directional searches, namely the forward search from  $s$  to  $t$  and the backward search from  $t$  to  $s$ . To present clearly the bidirectional search, we introduce the mono-directional search first.

**Mono-directional search** A mono-directional forward search extends the labels from  $s$  to  $t$ . In the search, a partial path  $r$  that starts from  $s$  to vertex  $i$  can be represented by a label  $(X_i, Y_i, Z_i, (H_i^{custk})_{k \in C}, s)$ , where  $X_i$  is the reduced cost along path  $r$ ,  $Y_i$  is the cumulative load transferred along path  $r$ ,  $Z_i$  is the cumulative recharging time required, since last recharge along path  $r$ , and  $H_i^{custk}$  is the number of visiting times for customer  $k \in C$  along path  $r$ .

The components of the initial label at satellite  $s$  are set to 0. Suppose that  $(X_i, Y_i, Z_i, (H_i^{custk})_{k \in C}, s)$  is a label of vertex  $i \in s \cup C \cup F$  and  $j \in C \cup F$  is a successor of  $i$ , i.e., arc  $(i, j) \in A_2$ . When the partial path  $r$  extends from  $i$  to  $j$ , a new label  $(X_j, Y_j, Z_j, (H_j^{custk})_{k \in C}, s)$  of vertex  $j \in C \cup F$  is generated according to the following REFs:

$$X_j = X_i + \bar{c}_{ij} \tag{40}$$

$$Y_j = Y_i + q_j \tag{41}$$

$$Z_j = \begin{cases} e_{ij} & \text{if } i \in F \\ Z_i + e_{ij} & \text{otherwise} \end{cases} \tag{42}$$

$$H_j^{custk} = \begin{cases} H_i^{custk} + 1 & \text{if } k = j \\ H_i^{custk} & \text{otherwise} \end{cases} \tag{43}$$

If one of the conditions  $Y_j > Q_2, Z_j > E$ , and  $H_j^{custk} > 1$  for a customer  $k \in C$  holds, label  $(X_j, Y_j, Z_j, (H_j^{custk})_{k \in C}, s)$  is infeasible and discarded.

To avoid enumerating all the feasible partial paths, each new generated label is examined by the dominance rule. Given that REFs (40)–(43) are nondecreasing, we define the dominance rule as follows:

**Definition 1** Let  $L^n = (X_n, Y_n, Z_n, (H_n^{custk})_{k \in C}, s)$ ,  $n \in \{1, 2\}$ , be two labels associated with different paths starting from the same satellite  $s$  and ending at the same vertex  $i$ . Label  $L^2$  is dominated by  $L^1$  if

$$X_1 \leq X_2 \tag{44}$$

$$Y_1 \leq Y_2 \tag{45}$$

$$Z_1 \leq Z_2 \tag{46}$$

$$H_1^{custk} \leq H_2^{custk} \quad \forall k \in C \tag{47}$$

and at least one of the inequalities is strict.

If  $L^1$  dominates  $L^2$ ,  $L^2$  is discarded.

**Bidirectional search** In the bidirectional search, labels are extended both forward from satellite  $s$  to its successors and backward from  $t$  to its predecessors. Labels, REFs, and the dominance rules of the backward search can be defined analogously as those of the forward search.

In the bidirectional search, we select a resource as the critical resource for which its value is nondecreasing as the path extends. We choose the capacity of the vehicles as the critical resource and set the midpoint as  $Q_2/2$ . The restriction condition for the forward search is

$$Y \leq Q_2/2. \tag{48}$$

A forward label is no allowed to be extended if its component  $Y$  exceeds  $Q_2/2$ . Simultaneously, backward labels are extended backward from  $t$  except  $Y > Q_2/2$ .

Let  $L_i^{fw}$  and  $L_i^{bw}$  be the sets of forward and backward labels associated with vertex  $i$ , respectively. The forward and backward labels should be merged to form an  $s - t$  path. Let  $(X_i, Y_i, Z_i, (H_i^{custk})_{k \in C}, s) \in L_i^{fw}$  and  $(X'_i, Y'_i, Z'_i, (H_i^{custk})_{k \in C}, s) \in L_i^{bw}$  be the labels of a forward path and a backward path starting from the same satellite  $s$  and ending at the same vertex  $i$ , respectively. An  $s - t$  path can be yielded by merging these two labels only if the labels meet the following conditions:

$$Y_i + Y'_i - q_i \leq Q_2 \tag{49}$$

$$H_i^{custk} + H_i^{custk'} \begin{cases} \leq 1 & \text{if } k \neq i \\ = 2 & \text{otherwise} \end{cases} \tag{50}$$

and the additional conditions, which correspond to two cases of the merged vertex.

**Case 1.** The ending vertex is a recharging station, i.e.,  $i \in F$ , if

$$\max\{Z_i, Z'_i\} \leq E. \tag{51}$$

**Case 2.** The ending vertex is a customer, i.e.,  $i \in C$ , if

$$Z_i + Z'_i \leq E. \tag{52}$$

**Acceleration strategies**

We adopt two strategies to reduce the computing time to solve the pricing subproblem. One strategy is the state-space relaxation, which is proposed by Christofides et al. [9]. This strategy is achieved by removing the label component  $(H^{custk})_{k \in C}$  that is used to record the number of visiting times. The other strategy is the *ng*-route relaxation introduced by Baldacci et al. [3].

The state-space relaxation has been used to solve the TSP [4,46]. In their paper, the component  $(H^{custk})_{k \in C}$  is replaced by the sum of all customers' visits  $\delta = \sum_{k \in C} H^{custk}$ . Here, we remove the component  $(H^{custk})_{k \in C}$  from the label instead of replacing it by  $\delta$  to obtain the label with relaxation. Let  $\overline{L}^{fw}$  and  $\overline{L}^{bw}$  be the sets of forward and backward labels with relaxation, respectively. The labeling algorithm is performed to extend the labels of  $\overline{L}^{fw}$  and  $\overline{L}^{bw}$  in each round of the CG algorithm. If the labeling algorithm fails to generate columns with negative reduced costs, it is implemented with the set of labels  $L^{fw}$  and  $L^{bw}$ .

For each vertex  $i \in C \cup F$ , let  $N_i$  be its neighborhood that contains the  $|N_i| - 1$  closest vertices (customers and recharging stations) and  $i$  itself. Let  $P = \{s, i_1, i_2, \dots, i_p\}$  be a feasible path associated with the subset  $\prod(P) \subseteq V(P)$ , where  $\prod(P)$  is

$$\prod(P) = \begin{cases} \{i_k | i_k \in \bigcap_{s=k+1}^p N_{i_s}\} \cup \{i_p\} \cap C & \text{if } i_p \in C \\ \{i_k | i_k \in \bigcap_{s=k+1}^p N_{i_s}\} \cup \{i_p\} \cap C \cup F & \text{if } i_p \in F \end{cases} \tag{53}$$

A *ng*-route is a non-necessarily elementary path  $P = \{s, i_1, i_2, \dots, i_p=i\}$  starting from a satellite  $s$ , visiting some vertices, and ending at vertex  $i$ , such that  $i \notin \prod(P')$ , where  $P' = \{s, i_1, i_2, \dots, i_{p-1}\}$  is a *ng*-path. The label  $L_{i_p}$  is allowed to extend to a vertex  $i_{p+1}$ , such that  $i_{p+1} \notin \prod(P)$  and all of the label components at vertex  $i_{p+1}$  satisfy the conditions. When  $N_i$  contains all of the vertices, such that  $N_i \supseteq C$ , it is obvious that label  $L_{i_p}$  generates the elementary paths. If  $|N_i|$  is small enough, it is difficult to produce a near-elementary

route, which means that the subproblem can be solved easily. To trade off the computing time and solution quality, we set  $|N_i|$  to be 10 in the numerical studies.

**Branching strategy**

To derive an integer solution, we resort to branching. The branching strategy consists of three types of branching decisions [47]. Let  $\{\lambda_r | r \in R_1\}$  and  $\{\hat{\theta}_r^s | s \in S, r \in \widehat{R}_2^s\}$  be the solutions of the latest RMP. The variables  $\{\lambda_r | r \in R_1\}$  are preferred to branch first. Suppose  $\lambda_w$  is the most fractional variable, i.e., its decimal part is closest to 0.5. Two child nodes are created, namely one adds the constraint  $\lambda_r \geq \lceil \lambda_w \rceil$  and the other adds the constraint  $\lambda_r \leq \lfloor \lambda_w \rfloor$ .

If all of the variables  $\{\lambda_r | r \in R_1\}$  are integers, we evaluate the number of vehicles  $\sum_{r \in \widehat{R}_2^s} \hat{\theta}_r^s$  used by each satellite  $s \in S$ .

Assume that satellite  $k$  uses the most fractional number of vehicles. Similarly, two child nodes are produced, namely

one adds the constraint  $\sum_{r \in \widehat{R}_2^k} \theta_r^k \leq \left\lfloor \sum_{r \in \widehat{R}_2^k} \hat{\theta}_r^k \right\rfloor$  and the other

adds the constraint  $\sum_{r \in \widehat{R}_2^k} \theta_r^k \geq \left\lceil \sum_{r \in \widehat{R}_2^k} \hat{\theta}_r^k \right\rceil$ .

The third branching decision is used if  $\{\lambda_r | r \in R_1\}$  and  $\{\sum_{r \in \widehat{R}_2^s} \hat{\theta}_r^s | s \in S\}$  are integers. Let  $t_{ijr}$  be the number of times that a vehicle travels the arc  $(i, j) \in A_2$ . The total number of times all vehicles passed through arc  $(i, j)$  is  $\sum_{r \in \widehat{R}_2^s} t_{ijr} \hat{\theta}_r^s$ . Assuming that  $(u, v) \in A_2$  is the most fractional arc, we create two child nodes, namely one adds the constraint  $\sum_{r \in \widehat{R}_2^s} t_{uvr} \theta_r^s \leq 0$  and the other adds the constraint  $\sum_{r \in \widehat{R}_2^s} t_{uvr} \theta_r^s \geq 1$ .

The second and third branching decisions are imposed by adding inequalities to the RMP. Each inequality has a corresponding dual variable. For a given satellite  $s \in S$ , let  $I_{BR}^s$  be the set of the inequalities that have been added to the RMP at the current branch node and  $\beta^s$  be the set of the corresponding dual variables of the second decision. Let  $I_{BA}^s$  be the set of inequalities and  $\gamma^s$  be the set of dual variables of the third decision. Then, the subproblem is

$$\begin{aligned} \min_{s \in S, r \in \widehat{R}_2^s} \bar{c}_r = & c_r + u_2 - \alpha_s \sum_{i \in C} a_r^i q_i \\ & - \sum_{i \in C} a_r^i \pi_i - \sum_{k=1}^{|I_{BR}^s|} \beta_k^s - \sum_{k=1}^{|I_{BA}^s|} t_{ijr} \gamma_k^s. \end{aligned} \tag{54}$$

If  $I_{BR}^s$  or  $I_{BA}^s$  is empty, the corresponding part in (54) is 0.

To reduce the memory requirement, we adopt the depth-first search strategy to explore the search tree.

## Numerical studies

In this section, we present numerical studies to evaluate the effectiveness of the BP algorithm. We present the test instances used in the studies in Sect. 5.1 and the performance of the algorithm in Sect. 5.2. We present the sensitivity analysis of recharging station density, battery capacity, and fixed costs of EVs in Sect. 5.3.

We coded the algorithm in Python 3.6.3 and used CPLEX 12.8.0 to solve the RMP. We conducted the numerical studies on a computer with an Inter(R) Core(TM) i7-9700K CPU (3.6GHz) and 32GB of RAM memory.

### The test instances

As the 2E-EVRP problem in our paper is different from those considered in existing studies, no benchmarks' instances are available to test the performance of our proposed algorithm. Therefore, we generate new instances for the numerical studies.

First, we generated a set of small-sized instances, labeled as Set 1, on the basis of instances in Breunig et al. [7]. We generated second and third sets of instances, labeled as Set 2 and Set 3, respectively, which are based on the instances proposed by Breunig et al. [7] and Desaulniers et al. [16], respectively. Set 1 has 12 instances, each of which has ten customers and no more than two recharging stations. Set 2 has 12 instances, each of which has 20 customers and two recharging stations. Set 3 also has 12 instances, each of which has 20 customers and two recharging stations. To apply the instances to our problem, we need to preprocess these instances. First of all, we randomly removed some customers if the number of customers is great than our instances. Afterward, for the instances from Breunig et al. [7], we removed the recharging function as well as the capacity limit of the satellites. For the instances from Desaulniers et al. [16], we randomly inserted the satellites into the graph. All the instances can be downloaded by accessing URL: <https://github.com/Novigram/Insatances-for-2E-EVRP.git>.

### Computational results

In this subsection, we report the results of applying our BP algorithm to solve the generated instances. First, we compare the performance of the BP algorithm with that of CPLEX (arc flow model) on Set 1 (the small-sized instances) and report the results in Table 3. We give the names of the instances in the first column, e.g., “E-c10-s3-r1-1” means that the first instance, which has ten customers, three satellites, and one

recharging station. This naming rule also applies to Set 2 and Set 3. The column “Opt” reports the optimal solution of CPLEX. If CPLEX cannot find the optimal solution within 3,600 s, the best upper bound is considered as the best approximate solution. The columns “UB” and “LB” denote the upper bound and lower bound founded on terminating the algorithm. The column “T(s)” shows the computation time (seconds) of CPLEX and the BP algorithm, respectively. The column “Gap” reports the percentage gap between the upper bound and lower bound.

defined as  $100(\text{UB}-\text{LB})/\text{UB}$ .

From Table 3, we see that the BP algorithm can solve all of the 12 instances and find the optimal solutions. In addition, the BP algorithm solves the instances (except E-c10-s3-r1-5) faster than CPLEX. On average, the BP algorithm takes 13.9727 s, whereas CPLEX takes 47.2367 s to solve an instance in Set 1. The BP algorithm is 70.42% faster than CPLEX. Table 3 shows the correctness of the model and the effectiveness of our algorithm.

Next, we evaluate the performance of the BP algorithm in solving some larger scale problems (Set 2 and Set 3). We set the maximum CPU running time as 100,000 s. We report the computational results in Table 4.

In Table 4, the upper bound in column “UB” marked with an asterisk means that the BP algorithm finds the optimal solution. The column “RLB” denotes the root lower bound, which is the optimal value of the RMP at the root node.

From Table 4, we see that the BP algorithm performs well in solving the 2E-EVRP in moderate computing time. Overall, the BP algorithm solves 15 out of the 24 instances within the maximum running time. The BP algorithm finds the optimal solutions for seven instances in Set 2. For the other instances, the average percentage gap between the upper bound and lower bound is small, which is approximately 1.1 %. For the instances in Set 3, the BP algorithm solves eight instances within the maximum running time. For the other instances, the percentage gaps between the upper and lower bounds are very small and acceptable.

We can also observe that some instances are very hard to solve (E-c20-s2-r2-1, E-c20-s2-r2-2, etc.), while other are very easy to solve (E-c20-s2-r2-5, E-c20-s2-r2-9, etc.), despite they have the same dimensions. The computing time is mainly affected by the number of branch nodes. Each branch process means that the CG has to be executed many times, which is very time-consuming. If an integer solution of a branch node is produced, this node is no longer branched. Therefore, the earlier the integer solution is generated, the fewer of branch nodes are produced. If the search tree has fewer branches, the running time will be smaller.

**Table 3** Performance of the BP algorithm on the instances of set 1

Instance	CPLEX		BP			
	Opt	T	LB	UB	Gap(%)	T(s)
E-c10-s3-r1-1	315	70.45	315	315	0	4.4677
E-c10-s3-r1-2	287	7.83	287	287	0	2.8900
E-c10-s3-r1-3	315	114.78	315	315	0	21.5105
E-c10-s3-r1-4	330	79.69	330	330	0	4.7020
E-c10-s3-r1-5	282	3.06	282	282	0	3.7960
E-c10-s3-r1-6	258	3600	258	258	0	55.9486
E-c10-s3-r1-7	279	4.73	279	279	0	3.7178
E-c10-s3-r1-8	334	48.52	334	334	0	4.7801
E-c10-s3-r1-9	288	52.78	288	288	0	20.3702
E-c10-s3-r1-10	286	51.53	286	286	0	14.7309
E-c10-s3-r1-11	230	27.22	230	230	0	5.3112
E-c10-s3-r1-12	270	106.25	270	270	0	25.4471
Avg.	289.5	47.2367	289.5	289.5	0	13.9727

**Table 4** Performance of the BP algorithm on the instances of Sets 2 and 3

Instance	LB	UB	RLB	Gap(%)	T(s)
Set2					
E-c20-s2-r2-1	4854	4854*	4791.1	0	14.76
E-c20-s2-r2-2	4913	4913*	4792.16	0	29.88
E-c20-s2-r2-3	4986	4986*	4853.19	0	19391.02
E-c20-s2-r2-4	5400	5400*	5363.28	0	72.02
E-c20-s2-r2-5	5661.41	5669	5477.07	0.13	100000
E-c20-s2-r2-6	4079	4079*	3998.11	0	43731.56
E-c20-s2-r2-7	6396	6396*	6185.29	0	37145.77
E-c20-s2-r2-8	6690	6690*	6645.28	0	3752.18
E-c20-s2-r2-9	6187.56	6306	5918.11	1.88	100000
E-c20-s2-r2-10	9567.8	9586	9510.13	0.19	100000
E-c20-s2-r2-11	10384.81	10449	10351.36	0.61	100000
E-c20-s2-r2-12	6350.74	6527	6311.52	2.70	100000
Avg.	6289.19	6321.25	6183.05	1.10	
Set3					
E-c20-s2-r2-13	517	517*	460.6	0	158.2
E-c20-s2-r2-14	457	457*	403.4	0	1569.1
E-c20-s2-r2-15	336	336*	291.04	0	90421.07
E-c20-s2-r2-16	566	566*	494.6	0	49228.99
E-c20-s2-r2-17	402.97	430	362.81	6.29	516334.3
E-c20-s2-r2-18	507.63	545	481.23	6.86	271460.1
E-c20-s2-r2-19	497	497*	438.72	0	49498.76
E-c20-s2-r2-20	394.67	435	373.63	9.27	100000
E-c20-s2-r2-21	593.3	637	571.26	6.86	100000
E-c20-s2-r2-23	542	542*	483.98	0	4903.78
E-c20-s2-r2-24	481	481*	432.2	0	507.15
Avg.	480.63	493	432.51	7.32	

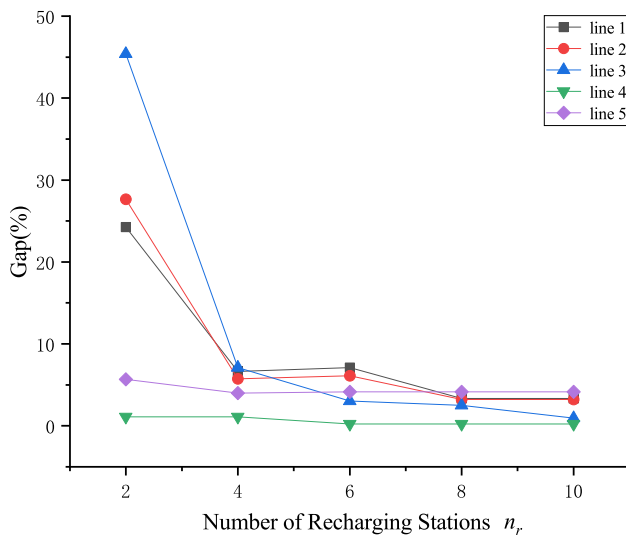


Fig. 2 Impact of recharging stations density on the total cost

### Sensitivity analysis

In this subsection, we investigate the impacts of the recharging stations density, battery capacity, and fixed cost of EVs on the optimal solution. We selected five instances, namely E-c20-s2-r2-1 (instance 1), E-c20-s2-r2-2 (instance 2), E-c20-s2-r2-4 (instance 3), E-c20-s2-r2-14 (instance 4), and E-c20-s2-r2-24 (instance 5) from Sets 2 and 3 to generate three additional sets of instances, namely Sets 4, 5, and 6, to conduct the sensitivity analysis. Set 4 consists of 25 instances, which are divided into five groups, with  $n_r \in \{2, 4, 6, 8, 10\}$  recharging stations. Set 5 also has 25 instances, which are divided into five groups with different battery capacities. The battery capacity of the five groups is  $n$  times the original battery capacity  $Q_3$ , where  $n \in \{1, 1.3, 1.5, 1.7, 2\}$ . Set 6 contains 25 instances, which are divided into five groups with different fixed costs of the EVs. The fixed costs of the five groups are  $u_2 \in \{100, 200, 300, 400, 500\}$ . The other parameters were the same as before.

In Sets 5 and 6, we fixed the numbers of the depot, satellites, and recharging stations for each instance. However, we varied the number of recharging stations  $n_r$  in the different groups of Set 4. For each instance of Set 4, we added the additional recharging stations in the graph when  $n_r$  changes. Suppose the x-coordinates of all the vertices (depot, satellites, and recharging stations) are in the interval  $[x_{min}, x_{max}]$  and the y-coordinates are in the interval  $[y_{min}, y_{max}]$ . We randomly located the additional recharging stations in the rectangular area  $[x_{min}, x_{max}] \times [y_{min}, y_{max}]$ . Figure 2 depicts the relationship between the number of recharging stations and the total cost. Figure 3 shows the changes in the total cost concerning the battery capacity. Figure 4 shows the impacts of the fixed cost of the EVs on the transport cost (transport dis-

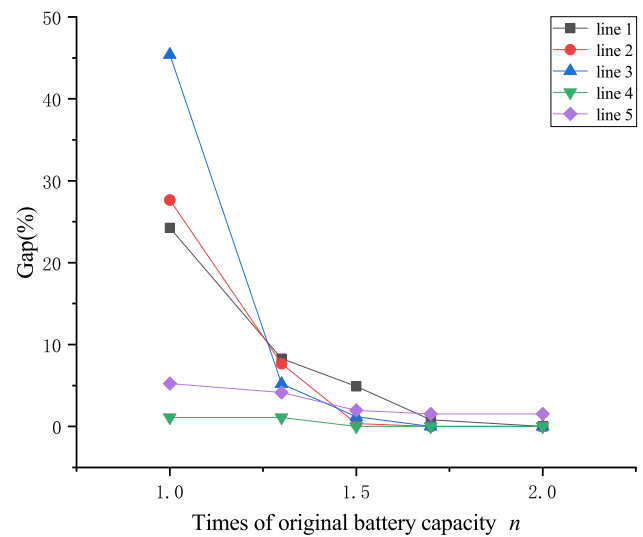


Fig. 3 Impact of battery capacity on the total cost

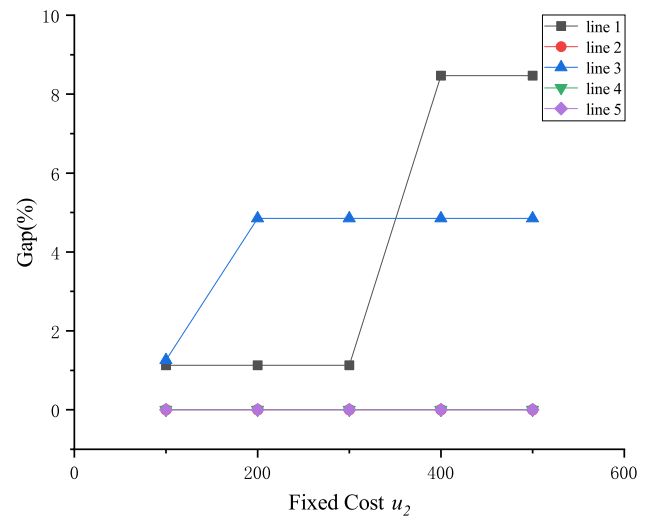


Fig. 4 Impact of fixed cost on the transport cost

tance). The “Gap” in Figs. 2 and 3 means the percentage gaps between the total costs with and without the battery capacity restriction. Since the total cost consists of the transport and fixed costs, changing the fixed cost directly affects the total cost, which cannot show the impact on the transport cost. Therefore, the “Gap” in Fig. 4 is the percentage gap between the transport costs with and without the vehicle fixed cost. In Fig. 2, the points on each line  $n \in \{1, 2, 3, 4, 5\}$  represent the five instances with  $n_r \in \{2, 4, 6, 8, 10\}$ , which are derived from instance  $n$ . Overall, the percentage gap decreases with the recharging station density, but the decline is slower as the recharging station density becomes larger. In other words, the total cost will decrease with increasing recharging station density. However, when the recharging station density reaches a certain level, adding new recharging stations will result in a slower decline in the total cost.



In addition, there is no significant change in the percentage gaps on the line 4 and line 5, i.e., the recharging stations density has a slight impact on the total cost. This is because the optimal values of instance 4 and instance 5 with  $n_r = 2$  are very close to their corresponding optimal values without the battery capacity restriction. When new recharging stations are added, there is no better option that can significantly narrow the gap. Thus, the location of the recharging stations is very important. If the recharging stations are located at appropriate positions, the BP algorithm can generate promising solutions even if the recharging station density is very low.

In Fig. 3, the points on each line  $n \in \{1,2,3,4,5\}$  represent the five instances with different battery capacity, which are derived from instance  $n$ . When the battery capacity is small, a slight increase in the battery capacity can considerably reduce the transport cost. When the battery capacity is large, increasing the battery capacity has a small impact on the total cost. Moreover, when the recharging stations density is high, the effect of increasing the battery capacity to save cost is not obvious. In other words, when the recharging infrastructure is adequate, the battery capacity has a small impact on the transport cost.

In Fig. 4, the points on each line  $n \in \{1,2,3,4,5\}$  represent the five instances with different fixed costs, which are derived from instance  $n$ . Figure 4 shows the impact of the fixed costs of the EVs on the transport cost in the line 1 and line 3. When the fixed cost increases to a certain extent, the vehicle will change from the original distribution route to visiting more customers, so the number of vehicles used is reduced. However, the fixed cost has no impact on the transport cost in the other three lines. This is mainly caused by one or more of the following reasons:

(i) The high loading rates of the EVs make it impossible for the EVs to access more customers.

(ii) The distribution of customers in each region is relatively centralized, so the distances between the different regions are long.

(iii) The battery capacity of the EVs is small and there is no available recharging station nearby, which makes the EVs unable to undertake more distribution tasks.

## Conclusions

In this paper, motivated by the express and e-commerce companies' distribution practices, we study the 2E-EVRP, which is an extension of the 2E-VRP considering battery capacity and en-route recharging. We propose a BP algorithm to solve the problem, which uses enumeration to generate feasible routes for the first echelon and the labeling algorithm for the second echelon. Conducting computational experiments to evaluate the effectiveness of the BP algorithm, we show that the algorithm can find the optimal solutions efficiently. We

demonstrate the effectiveness of the BP algorithm by comparing it with CPLEX for solving small-scale test instances. We further assess the performance of the BP algorithm in solving middle-scale test instances. We also explore the impacts of recharging station density, battery capacity, and fixed cost of the EVs on the optimal solution.

Based on our study, we suggest some topics worthy of future research. In practice, a host of other factors, e.g., time windows, the limited number of EVs, nonlinear recharging, and discharging characteristics, need to be considered when manager vehicle schedules. Extending our model and algorithm to consider other factors is a topic worth studying in the future. In this paper, we propose a BP algorithm to solve the problem. Other solution algorithms, be they exact or approximate, could be devised to solve the problem.

**Author Contributions** Zhiguo Wu designed the proposed algorithm and conducted the numerical experiments. Juliang Zhang adjusted the structure and checked the grammar of this paper. Zhiguo Wu is the major contributor in writing the manuscript. All authors read and approved the final manuscript.

**Funding** This work was supported in part by the National Natural Science Foundation of China under grant numbers 71661167009, 71901027, the NSFC/RGC joint support scheme under grant number N\_PolyU531/16, and Beijing Logistics Informatics Research Base.

**Availability of data and materials** All data generated or analyzed during this study are included in this published article [and its supplementary information files].

## Declarations

**Conflicts of interest** We declare that we have no conflict of interest.

**Ethics approval** Not applicable.

**Consent to participate** Not applicable.

**Consent for publication** Written informed consent for publication was obtained from all participants.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Awaga AL, Xu W, Liu L, Zhang Y (2020) Evolutionary game of green manufacturing mode of enterprises under the influence of government reward and punishment. *Adv Production Eng Manag* 14(4):416–430. <https://doi.org/10.14743/apem2020.4.375>
- Baldacci R, Mingozzi A (2009) A unified exact method for solving different classes of vehicle routing problems. *Math Program* 120(2):347
- Baldacci R, Mingozzi A, Roberti R (2011) New route relaxation and pricing strategies for the vehicle routing problem. *Oper Res* 59(5):1269–1283. <https://doi.org/10.1287/opre.1110.0975> Cited By 84
- Baldacci R, Mingozzi A, Roberti R (2012) New state-space relaxations for solving the traveling salesman problem with time windows. *Inform J Comput* 24(3):356–371
- Baldacci R, Mingozzi A, Roberti R, Calvo RW (2013) An exact algorithm for the two-echelon capacitated vehicle routing problem. *Oper Res* 61(2):298–314
- Barros L, Linfati R, Escobar JW (2020) An exact approach for the consistent vehicle routing problem (convrp). *Adv Prod Eng Manag* 15(3):255–266. <https://doi.org/10.14743/apem2020.3.363>
- Breunig U, Baldacci R, Hartl RF, Vidal T (2019) The electric two-echelon vehicle routing problem. *Comput Oper Res* 103:198–210
- Breunig U, Schmid V, Hartl RF, Vidal T (2016) A large neighbourhood based heuristic for two-echelon routing problems. *Comput Oper Res* 76:208–225
- Christofides N, Mingozzi A, Toth P (1981) State-space relaxation procedures for the computation of bounds to routing problems. *Networks* 11(2):145–164
- Cortés-Murcia DL, Prodhon C, Murat Afsar H (2019) The electric vehicle routing problem with time windows, partial recharges and satellite customers. *Transportation Research Part E: Logistics and Transportation Review* 130:184–206. <https://doi.org/10.1016/j.tre.2019.08.015> <http://www.sciencedirect.com/science/article/pii/S1366554518310597>
- Crainic TG, Perboli G, Mancini S, Tadei R (2010) Two-echelon vehicle routing problem: a satellite location analysis. *Proc-Soc Behav Sci* 2(3):5944–5955
- Crainic TG, Ricciardi N, Storchi G (2004) Advanced freight transportation systems for congested urban areas. *Trans Res Part C* 12(2):119–137
- Crainic TG, Ricciardi N, Storchi G (2009) Models for evaluating and planning city logistics systems. *Trans Sci* 43(4):432–454
- Cuda R, Guastaroba G, Speranza MG (2015) A survey on two-echelon routing problems. *Comput Oper Res* 55:185–199
- Dellaert N, Dashty Saridarq F, Van Woensel T, Crainic TG (2019) Branch-and-price-based algorithms for the two-echelon vehicle routing problem with time windows. *Trans Sci* 53(2):463–479
- Desaulniers G, Errico F, Irnich S, Schneider M (2016) Exact algorithms for electric vehicle-routing problems with time windows. *Oper Res* 64(6):1388–1405
- Du J, Li Q, Qiao F, Yu L (2018) Estimation of vehicle emission on mainline freeway under isolated and integrated ramp metering strategies. *Environ Eng Manag J* 17(5):1237–1248
- Du J, Li Q, Qiao F, Yu L (2019) Temporal characteristics and forecasting of pm2.5 concentration based on historical data in Houston, USA. *Conserv Recycl* 147:145–156
- Du J, Qiao F, Yu L (2020) Improving bus transit services for disabled individuals: demand clustering, bus assignment, and route optimization. *IEEE Access* 8:121564–121571. <https://doi.org/10.1109/ACCESS.2020.3007322>
- EC: Together towards competitive and resource-efficient urban mobility. <https://ec.europa.eu/transparency/regdoc/rep/1/2013/EN/1-2013-913-EN-F1-1.Pdf> (2013)
- Felipe Á, Ortuño MT, Righini G, Tirado G (2014) A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges. *Trans Res Part E* 71:111–128
- Froger A, Mendoza JE, Jabali O, Laporte G (2019) Improved formulations and algorithmic components for the electric vehicle routing problem with nonlinear charging functions. *Comput Oper Res* 104:256–294
- H G, X Y (2019) Optimal deployment of charging stations and movable charging vehicles for electric vehicles. *J Syst Manag Sci* 9(1):105–116
- GDZ: Zero emission stadslogistiek. <http://greendealzes.connekt.nl> (2016)
- Gocken T, Yaktubay M (2019) Comparison of different clustering algorithms via genetic algorithm for vrptw. *Int J Simul Modelling* 18(4):574–585
- Gonçalves F, Cardoso SR, Relvas S, Barbosa-Póvoa A (2011) Optimization of a distribution network using electric vehicles: A vrp problem. In: *Proceedings of the IO2011-15 Congresso da associação Portuguesa de Investigação Operacional*, Coimbra, Portugal, pp. 18–20
- Gonzalez-Feliu J, Semet F, Routhier JL (2014) Sustainable urban logistics: concepts, methods and information systems. Springer, Berlin
- Grangier P, Gendreau M, Lehuédé F, Rousseau LM (2016) An adaptive large neighborhood search for the two-echelon multiple-trip vehicle routing problem with satellite synchronization. *Euro J Oper Res* 254(1):80–91
- Hemmelmayr VC, Cordeau JF, Crainic TG (2012) An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics. *Comput Oper Res* 39(12):3215–3228
- Hiermann G, Puchinger J, Ropke S, Hartl RF (2016) The electric fleet size and mix vehicle routing problem with time windows and recharging stations. *Euro J Oper Res* 252(3):995–1018
- IEA: Global ev outlook 2019. [https://webstore.iea.org/download/direct/2807?fileName=Global\\_EV\\_Outlook\\_2019.pdf](https://webstore.iea.org/download/direct/2807?fileName=Global_EV_Outlook_2019.pdf) (2019)
- Irnich S, Desaulniers G (2005) Shortest path problems with resource constraints. In: *Column generation*, pp. 33–65. Springer
- Jepsen M, Spoorendonk S, Ropke S (2013) A branch-and-cut algorithm for the symmetric two-echelon capacitated vehicle routing problem. *Trans Sci* 47(1):23–37
- Jie W, Yang J, Zhang M, Huang Y (2019) The two-echelon capacitated electric vehicle routing problem with battery swapping stations: formulation and efficient methodology. *Euro J Oper Res* 272(3):879–904
- Keskin M, Çatay B (2018) A matheuristic method for the electric vehicle routing problem with time windows and fast chargers. *Comput Oper Res* 100:172–188
- Keskin M, Laporte G, Çatay B (2019) Electric vehicle routing problem with time-dependent waiting times at recharging stations. *Comput Oper Res* 107:77–94
- Li H, Zhang L, Lv T, Chang X (2016) The two-echelon time-constrained vehicle routing problem in linehaul-delivery systems. *Trans Res Part B* 94:169–188
- Li HY, Xu W, Cui Y, Wang Z, Xiao M, Sun ZX (2020) Preventive maintenance decision model of urban transportation system equipment based on multi-control units. *IEEE Access* 8:15851–15869. <https://doi.org/10.1109/ACCESS.2019.2961433>
- Liao CS, Lu SH, Shen ZJM (2016) The electric vehicle touring problem. *Transportation Research Part B: Methodological* 86:163–180
- Montoya A, Guéret C, Mendoza JE, Villegas JG (2017) The electric vehicle routing problem with nonlinear charging function. *Trans Res Part B* 103:87–110

41. NBSPRC: Cumulative value of sales of new energy vehicles. <http://data.stats.gov.cn/easyquery.htm?cn=B01&zb=A03011M&sj=2019A> (2019)
42. Pelletier S, Jabali O, Laporte G (2016) 50th anniversary invited article-goods distribution with electric vehicles: review and research perspectives. *Trans Sci* 50(1):3–22
43. Perboli G, Tadei R (2010) New families of valid inequalities for the two-echelon vehicle routing problem. *Electron Notes Discrete Math* 36:639–646
44. Perboli G, Tadei R, Fadda E (2018) New valid inequalities for the two-echelon capacitated vehicle routing problem. *Electron Notes Discrete Math* 64:75–84
45. Perboli G, Tadei R, Vigo D (2011) The two-echelon capacitated vehicle routing problem: models and math-based heuristics. *Trans Sci* 45(3):364–380
46. Righini G, Salani M (2009) Decremental state space relaxation strategies and initialization heuristics for solving the orienteering problem with time windows with dynamic programming. *Comput Oper Res* 36(4):1191–1203
47. Santos FA, Mateus GR, da Cunha AS (2015) A branch-and-cut-and-price algorithm for the two-echelon capacitated vehicle routing problem. *Trans Sci* 49(2):355–368
48. Schneider M, Stenger A, Goeke D (2014) The electric vehicle-routing problem with time windows and recharging stations. *Trans Sci* 48(4):500–520
49. Tahami H, Rabadi G, Haouari M (2020) Exact approaches for routing capacitated electric vehicles. *Transportation Research Part E: Logistics and Transportation Review* 144:102126. <https://doi.org/10.1016/j.tre.2020.102126> <http://www.sciencedirect.com/science/article/pii/S1366554520307742>
50. Wang D, Zhou H, Feng R (2019) A two-echelon vehicle routing problem involving electric vehicles with time windows. In: *Journal of Physics: Conference Series*, vol. 1324, p. 012071. IOP Publishing
51. xianjichina.com: New energy logistics vehicle replacing fuel logistics vehicle has become a trend. [https://www.xianjichina.com/news/details\\_83369.html](https://www.xianjichina.com/news/details_83369.html) (2018)
52. Zhang H, Cui Y (2019) A model combining a bayesian network with a modified genetic algorithm for green supplier selection. *SIMULATION: Transactions of The Society for Modeling and Simulation International* 95(12):1165–1183
53. Zhang S, Gajpal Y, Appadoo S, Abdulkader M (2018) Electric vehicle routing problem with recharging stations for minimizing energy consumption. *Int J Prod Econ* 203:404–413

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.