Routing Electric Vehicle Fleet for Ride-sharing

Jie Shi, Yuanqi Gao, and Nanpeng Yu
Department of Electrical Engineering and Computer Science
University of California, Riverside
Riverside, CA 92521 USA
Email: jshi005@ucr.edu, ygao024@ucr.edu, and nyu@ece.ucr.edu

***Abstract*—Providing ride-sharing services with an electric vehicle (EV) fleet can significantly enhance urban mobility, reduce transportation sector energy consumption, and improve air quality. This paper develops an EV fleet routing algorithm for ride-sharing services. The EV fleet routing problem is rigorously formulated on a complete directed graph as a mixed integer nonlinear programming problem. The intrinsic characteristics of the EV routing problem allow us to transform the optimization formulation into an equivalent mixed integer linear programming problem. Numerical testing results show that the proposed method can find globally optimal EV routes to provide ride-sharing services. The simulation results also reveal a trade-off between customer waiting time and total distance traveled by the EV fleet.**

***Index Terms*—Electric vehicle, vehicle routing problem, ride-sharing, MILP.**

I. Introduction

Stricter environmental regulations, higher emission standards, and generous government incentives accelerate the global adoption of electric vehicles (EVs). It is estimated that the global EV stock will range between 9 million and 20 million by 2020 and between 40 million and 70 million by 2025 [1]. In the meantime, the rise of innovative ride-sharing platforms such as Uber and Didi is transforming urban mobility by providing timely and convenient transportation services with great efficiency. Recently, the ride-sharing service providers started to add EVs to their vehicle fleet to further reduce greenhouse gas (GHG) emissions and improve air quality. The electrification of ride-sharing services is especially beneficial to residents in disadvantaged communities who face steeper barriers to clean vehicle adoption. Although the introduction of EVs into the ride-sharing services has great potential in reducing GHG emissions, it also complicates the fleet routing and dispatch problem. Hence, it is critical to develop an EV fleet routing algorithm which minimizes customer waiting time, vehicle operating costs, and electricity consumption. In this paper, we provide a rigorous formulation and efficient solution to the EV fleet routing problem for ride-sharing services.

The vehicle routing problem (VRP) has been extensively studied since the seminal work on truck dispatch was published in 1959 [2]. Many researchers have worked on solving the original VRP and its variants. A comprehensive review of the VRP and its solutions can be found in [3]. Recently, the electric vehicle routing problem (EVRP) received a lot of attention from the research community. In the EVRP, the EV charging activities need to be considered. The EVRP is more complicated than the VRP because EVs' potential repetitive visits to the same charging station need to be modeled. As a pioneer, [4] first studied this type of problem with alternative fuel vehicles. Inspired by [5], the dummy vertices were leveraged to deal with vehicles' multiple visits to the same location. Using a similar technique, the EVRP is formulated in [6] with time window constraints. The recharging processes and calculation of EV arriving times at different locations introduced bilinear terms in the problem formulation. The EVRP is simplified as a mixed integer linear programming problem (MILP) by forcing EVs to be fully charged in all charging sessions [6]. Various extensions of the EVRP have been studied by considering partial recharging of EVs [7], the effects of EV load on energy consumption, a heterogeneous EV fleet [8], the road conditions [9], and nonlinear charging functions [10].

Very few studies directly addressed the problem of determining the optimal routes of an EV fleet for ride-sharing service (EVRP-RS). The EVRP-RS can be considered as an extension of the vehicle routing problem with multiple depots (MDVRP), which requires different vehicles leaving from and returning to different locations. A comprehensive review of the MDVRP can be found in [11]. The nonlinearity induced by charging time and customer waiting time makes the EVRP-RS more difficult to solve than the MDVRP. Several papers studied the EVRP-RS. However, they either assumed a well established routing algorithm already exists [12] or used greedy algorithms [13], [14] to determine the charging schedules and routes of EVs. These papers did not provide a rigorous problem formulation to determine the optimal routes and charging schedule for an EV fleet. A recent work [15] did develop an optimization framework to solve the EVRP-RS. However, it did not allow an EV to visit the same charging station more than once and used a cumbersome three-index formulation.

Building on top of existing literature, this work makes two unique contributions. First, we define the EV fleet routing problem on a complete directed graph and formulate the EVRP-RS as a mixed integer nonlinear programming (MINLP) problem. The formulation not only has a compact two-index form but also takes EVs' multiple visits to the same charging station into account. Second, we successfully transformed the MINLP problem into an equivalent MILP problem where global optimum can be found in many instances.

The rest of the paper is organized as follows. Section II formulates the EVRP-RS problem as a MINLP. Section III converts the MINLP into an equivalent MILP. Numerical studies are carried out in Section IV to validate the effectiveness of the proposed method and analyze the trade-off between customer waiting time and total distance traveled by the EV fleet. Section V concludes this paper.

II. Problem Formulation

In this section, the problem of determining the optimal routes for an EV fleet for ride-sharing services is formulated as a mixed integer nonlinear programming problem. The optimization problem is equivalent to finding the set of active edges in a complete directed graph.

A. Assumptions and Notations

The assumptions and notations used in the problem formulation are introduced in this subsection. To derive an analytically tractable formulation, three simplifying assumptions are made. First, it is assumed that the desired pickup time, pickup location, and drop-off location of all the customers are known prior to the operating time window. Second, it is assumed that EVs will start charging immediately after arriving at the charging stations. Third, it is assumed that a homogeneous EV fleet is being managed.

A complete directed graph can be used to describe all possible routing and charging plans of an EV fleet for ride-sharing. Let $G = (V, E)$ denote a complete directed graph, where $V = \{v_i | i = 1, \cdots, N\}$ is the set of vertices and $E = \{x_{ij} | i, j = 1, \cdots, N\}$ is the set of edges. Let S_C, S_I, S_T, and S_{CS} be a partition of V:

- $S_C = \{v_i | i = 1, 2, \cdots, N_C\}$
- $S_I = \{v_i | i = N_C + 1, N_C + 2, \cdots, N_C + N_{EV}\}$
- $S_T = \{v_i | i = N_C + N_{EV} + 1, N_C + N_{EV} + 2, \cdots, N_C + 2N_{EV}\}$
- $S_{CS} = \{v_i | i = N_C + 2N_{EV} + 1, N_C + 2N_{EV} + 2, \cdots, N\}$

where N_C is the number of customer requests. N_{EV} is the number of EVs in the fleet. Each vertex in S_C represents a customer request. Each vertex in S_I denotes an EV parked at its initial location. Each vertex in S_T represents an EV parked at its terminal location. Vertices in S_{CS} denote copies of charging stations that enable the modeling of multiple visits to the charging stations. Let N_{CS} be the number of charging stations. Then the cardinality of S_{CS} should not exceed $(N_C + N_{EV}) \times N_{CS}$. The size of S_{CS} is a design variable which determines the trade-off between graph complexity and the potential benefits received from enabling multiple visits to the charging stations.

Each vertex $v_i \in V$ contains a tuple $B_i = \{t_i^S, P_i^{ST}, P_i^{ED}\}$ that stores the starting time t_i^S, the starting position P_i^{ST}, and the end position P_i^{ED} of v_i. The definitions of the elements in a tuple depend on the type of v_i:

- If $v_i \in S_C$, then t_i^S is the desired pick-up time, P_i^{ST} is the pickup location, and P_i^{ED} is the drop-off location of the corresponding customer request.

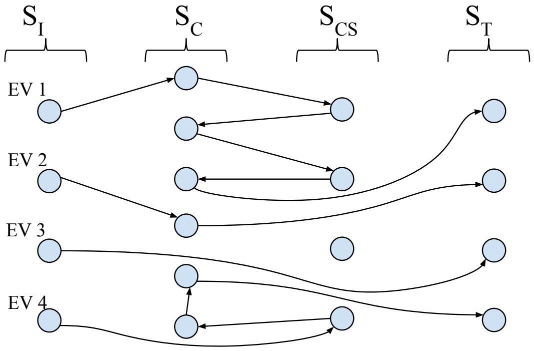

Fig. 1. A sample routing schedule.

- If $v_i \in S_I$, then t_i^S is the time when the EV departs from its initial location. $P_i^{ST} = P_i^{ED}$ is the initial location of the corresponding EV.
- If $v_i \in S_T$, then t_i^S is the time when the EV reaches its terminal location. $P_i^{ST} = P_i^{ED}$ is the terminal location of the corresponding EV.
- If $v_i \in S_{CS}$, then t_i^S is the time when the EV reaches the corresponding charging station. $P_i^{ST} = P_i^{ED}$ is the location of the corresponding charging station.

B. Control Variables

There are two sets of control variables. The first set, $\{T_i^{CS} | v_i \in S_{CS}\}$, defines the charging time spent on each visit to one of the charging stations. The second set, $\{x_{ij} | i, j = 1, \cdots, N\}$, is related to the edges in the directed graph. x_{ij} is a binary variable which takes the value of 1 if the edge from v_i to v_j is active and 0 otherwise. Here, an active edge is defined as follows:

- If $v_i \in S_C$, then an active edge from v_i to v_j means there exists one and only one EV that serves customer request of v_i and moves to the starting position of v_j immediately after the service is completed.
- If $v_i \in S_I$, then an active edge from v_i to v_j means the corresponding EV leaves from its initial location at initial operating time and moves to the starting position of v_j.
- If $v_i \in S_{CS}$, then an active edge from v_i to v_j means an EV gets charged at the corresponding charging station and moves to the starting position of v_j immediately after the charging session is completed.

The following edges are always set as inactive. We denote this set of inactive edges as E_{IA}.

- All the edges pointing toward vertices in S_I are set as inactive since they are the initial vertices.
- All the edges leaving from vertices in S_T are set as inactive since they are the terminal vertices.
- The edges between any two vertices in S_{CS} are set as inactive to avoid consecutive visits to the charging stations.

Fig. 1 shows a sample routing schedule with 4 EVs, 6 customer requests, and 1 charging station. The size of S_{CS} is set to be 4. At the start of the operating window, 4 EVs are located at their initial locations. The active edges represent the routing schedule of EV fleet. For example, the routing schedule of EV 1 is as follows. In the beginning, EV 1 leaves its initial location to serve the first customer. Then it goes to the charging station to charge the battery. After the charging session, it continues to serve the second customer and then get back to the charging station. Finally, it serves the third customer and heads to its assigned terminal location.

C. Objective Function

The objectives of the EV fleet routing algorithm are to minimize the operating cost of the EV fleet and the total customer waiting time. The operating cost is composed of two parts: EV maintenance cost and charging cost. The maintenance cost and the charging cost are assumed to be proportional to the total distance traveled by the EV fleet and the total charging time, respectively. The objective function can be formulated as

$$\min_{\{x_{ij}|i,j=1,\cdots,N\},\{T_m^{CS}|v_m\in S_{CS}\}} \quad Cost^{op} + Cost^{wt} \tag{1}$$

where

$$Cost^{op} = \sum_{\substack{i\neq j\\ i,j=1,\cdots,N}} s_{ij}x_{ij}C^M + \sum_{\{m|v_m\in S_{CS}\}} T_m^{CS}P^{CS}C^E \tag{2}$$

$$Cost^{wt} = \beta\sum_{k=1}^{N_C} w_k \tag{3}$$

where s_{ij} is the distance between the end position of v_i and the starting position of v_j. T_m^{CS} is the charging time (h) of the corresponding EV at vertex v_m. P^{CS} is the charging rate (kW) of the EV. w_k is the waiting time (h) of kth customer. C^M is the maintenance cost per mile ($/mile) of an EV. C^E is the cost of electricity ($/kWh). β is the cost of customer waiting time ($/h).

D. Constraints

Three groups of constraints need to be enforced when routing EV fleet for ride-sharing. These groups of constraints are: path constraints, energy constraints, and time constraints. We formulate the three groups of constraints below.

1) Path Constraints: The path constraints ensure that the EV routing schedule is feasible and efficient. They are listed as follows.

$$\sum_{v_j\in V, v_j\neq v_i} x_{ij} = 1, \quad \forall v_i \in S_C \tag{4}$$

$$\sum_{v_j\in V, v_j\neq v_i} x_{ij} \leq 1, \quad \forall v_i \in S_{CS} \tag{5}$$

$$\sum_{v_j\in V, v_j\neq v_i} x_{ji} - \sum_{v_k\in V, v_k\neq v_i} x_{ik} = 0, \quad \forall v_i \in S_C \cup S_{CS} \tag{6}$$

$$\sum_{v_j\in V, v_j\neq v_i} x_{ij} = 1, \quad \forall v_i \in S_I \tag{7}$$

$$\sum_{v_i\in V, v_i\neq v_j} x_{ij} = 1, \quad \forall v_j \in S_T \tag{8}$$

$$x_{ij} = 0, \quad \forall x_{ij} \in E_{IA} \tag{9}$$

(4) ensures each customer request is only served once. (5) ensures that the charging stations do not have to be visited an EV. (6) enforces the flow conservation. (7) and (8) ensure each EV leaves from its initial location and reaches its terminal location. (9) represents the inactive edges defined in Section II.B.

2) Energy Constraints: The energy constraints describe the electricity consumptions of EVs in the routing process and enforce the battery capacity limits.

$$\begin{aligned} s_{ij}E^{PM}\cdot x_{ij} - (1-x_{ij})E_{max} \leq E_i - E_j \leq \\ s_{ij}E^{PM}\cdot x_{ij} + (1-x_{ij})E_{max}, \\ \forall v_i \in S_I, v_j\in V, v_j\neq v_i \end{aligned} \tag{10}$$

$$\begin{aligned} (s_{ij}E^{PM} + E_i^C)x_{ij} - (1-x_{ij})E_{max} \leq E_i - E_j \leq \\ (s_{ij}E^{PM} + E_i^C)x_{ij} + (1-x_{ij})E_{max}, \\ \forall v_i \in S_C, v_j\in V, v_j\neq v_i \end{aligned} \tag{11}$$

$$\begin{aligned} (s_{ij}E^{PM} - T_i^{CS}P^{CS})x_{ij} - (1-x_{ij})E_{max} \leq E_i - E_j \leq \\ (s_{ij}E^{PM} - T_i^{CS}P^{CS})x_{ij} + (1-x_{ij})E_{max}, \\ \forall v_i \in S_{CS}, v_j\in V, v_j\neq v_i \end{aligned} \tag{12}$$

$$E_i + T_i^{CS}P^{CS} \leq E_{max}, \quad \forall v_i \in S_{CS} \tag{13}$$

$$E_i \geq 0, \quad \forall v_i \in S_{CS} \tag{14}$$

$$E_i = E_i^{ini}, \quad \forall v_i \in S_I \tag{15}$$

$$\begin{aligned} E_i \geq \min_{c\in\{1,2,\cdots,N_{CS}\}} (\left\|P_i^{ED} - P_c^{CS}\right\| \cdot E^{PM}), \\ \forall v_i \in S_T \end{aligned} \tag{16}$$

where s_{ij} is the distance between the end position of v_i and the starting position of v_j. E^{PM} is the EV electricity consumption per mile. E_i^C is the electricity required for serving customer request $v_i \in S_C$. $\left\|P_i^{ED} - P_i^{ST}\right\|$ is the distance between P_i^{ED} and P_i^{ST}. Hence $\forall v_i \in S_C$, $E_i^C = \left\|P_i^{ED} - P_i^{ST}\right\| \cdot E^{PM}$. E_i is the remaining energy in the battery of the corresponding EV when it reaches the starting position of vertex v_i. E_{max} is the battery capacity of EVs. P_c^{CS} is the position of the cth charging station.

(10) defines the difference in battery energy between E_i and E_j when $v_i \in S_I$. If the edge from v_i to v_j is active, then the difference in battery energy should be equal to the energy consumed for trip s_{ij}. (11) defines the difference in battery energy between E_i and E_j when $v_i \in S_C$. If the edge from v_i to v_j is active, then the difference in battery energy should be equal to the summation of energy consumed for the customer service and energy consumed for making the trip corresponding to s_{ij}. (12) defines the difference in battery energy between E_i and E_j when $v_i \in S_{CS}$. If the edge from v_i to v_j is active, then the difference in battery energy should be equal to the energy received from the charging station subtracted from the energy consumed for making trip corresponding to s_{ij}. Note that if the edge from v_i to v_j is inactive, then the above constraints are not enforced. (13) ensures that the remaining battery energy never exceeds the

battery capacity when the corresponding EV leaves a charging station. (14) ensures the remaining battery energy is greater than or equal to zero when EV arrives at a charging station. This constraint, along with the energy transition constraints, enforce that $E_i \geq 0$ when $v_i \in S_C$. (15) defines the initial battery energy of EVs. (16) ensures each EV has enough energy to reach the nearest charging station from its terminal location.

3) Time Constraints: The time constraints describe the temporal relationships among different vertices and specify the waiting times of customers.

$$\begin{aligned} -T_{max}(1-x_{ij}) + \max(t_j^R - t_i, T_i + T_{ij})x_{ij} \leq t_j - t_i \\ \leq T_{max}(1-x_{ij}) + \max(t_j^R - t_i, T_i + T_{ij})x_{ij}, \\ \forall v_i \in V, v_j \in S_C, v_i \neq v_j \end{aligned} \tag{17}$$

$$\begin{aligned} -T_{max}(1-x_{ij}) + (T_i + T_{ij})x_{ij} \leq t_j - t_i \\ \leq T_{max}(1-x_{ij}) + (T_i + T_{ij})x_{ij}, \\ \forall v_i \in V, v_j \in S_{CS} \cup S_T, v_i \neq v_j \end{aligned} \tag{18}$$

$$T_i = 0, \quad v_i \in S_I \cup S_T \tag{19}$$

$$T_i = \frac{\left\| P_i^{ED} - P_i^{ST} \right\|}{v_{EV}}, \quad v_i \in S_C \tag{20}$$

$$T_i = T_i^{CS}, \quad v_i \in S_{CS} \tag{21}$$

$$T_{ij} = \frac{s_{ij}}{v_{EV}}, \quad \forall v_i \in V, v_j \in V, v_i \neq v_j \tag{22}$$

$$t_i = t_i^{ini}, \quad v_i \in S_I \tag{23}$$

$$w_i = \max(0, \sum_{j \in V, j \neq i} (T_j + T_{ji} + t_j - t_i^R)x_{ji}), \quad \forall v_i \in S_C \tag{24}$$

where T_{max} denotes the maximum length of system operating time. t_j^R is the desired pickup time of customer request $v_j \in S_C$. t_i is defined as follows:

- If vertex $v_i \in S_C$, then t_i is the time when corresponding EV starts its customer service of v_i.
- If vertex $v_i \in S_{CS} \cup S_T$, then t_i is the time when the corresponding EV reaches the starting position of v_i.
- If vertex $v_i \in S_I$, then t_i is the initial operating time of the corresponding EV.

T_i denotes the time spent on vertex v_i. T_{ij} is the time spent on moving from the end position of v_i to the starting position of v_j. v_{EV} is the average speed of an EV.

(17) defines the time difference between t_j and t_i where $v_j \in S_C$. If the edge from v_i to v_j is active, then the time difference should be $\max(t_j^R - t_i, T_i + T_{ij})$. As shown in Fig. 2, if the corresponding EV reaches the start position before the desired pickup time, it has to wait for the customer. Thus $t_j = t_j^R$. If the corresponding EV reaches the starting position later than the desired pickup time, then the customer has to wait for the EV. Hence $t_j = t_i + T_i + T_{ij}$. (18) defines the time difference between t_j and t_i where $v_j \in S_{CS} \cup S_T$. If the edge from v_i to v_j is active, then the time difference should be $T_i + T_{ij}$. Note that if the edge from v_i to v_j is inactive, then these two constraints are not enforced. (19) defines the time spent on any vertex in $S_I \cup S_T$ being zero. (20) calculates the time spent on serving customer request of vertex $v_i \in S_C$. (21) defines the time spent on charging at vertex $v_i \in S_{CS}$. (22) calculates the time spent on traveling between the end position of vertex v_i and the starting position of vertex v_j. (23) specifies the initial operating time of corresponding EV. (24) calculates the waiting time of each customer.

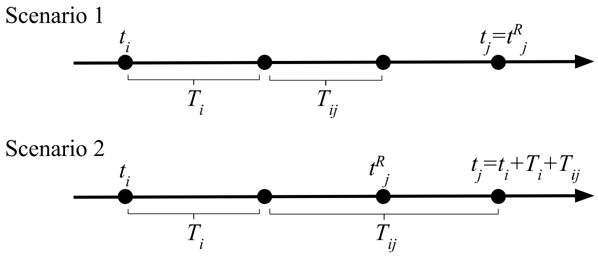

Fig. 2. Time constraints under two scenarios.

E. MINLP Problem

The EV ride-sharing routing problem formulated above is a MINLP due to the bilinear terms and $\max(\cdot)$ functions in the energy and time constraints. By regrouping the constraints into two sets according to their linearity, the problem can be summarized as follows:

$$\min_{\{x_{ij}|i,j=1,\cdots,N\},\{T_m^{CS}|v_m \in S_{CS}\}} \qquad Cost^{op} + Cost^{wt}$$

subject to:

Linear constraints: (4)-(11), (13)-(16), (19)-(23)

Nonlinear constraints: (12), (17), (18), (24)

In general, it is difficult to final globally optimal solutions for a MINLP problem. However, we can convert the nonlinear constraints of this particular problem into linear ones. Hence, the MINLP problem can be transformed into a MILP problem.

III. Reformulation of Nonlinear Constraints

In this section, the nonlinear constraints of the original optimization problem are converted into linear ones. The resulting MILP problem can then be solved with existing commercial solvers such as Gurobi and Mosek.

A. How to Deal with Nonlinear Conditional Constraints?

Note that the nonlinear constraints in Section II have a special structure. They can be described by if-else statements in an algorithm. Here we demonstrate how to convert these conditional nonlinear constraints into linear ones using (17) and (24) as an example.

Constraints (17) and (24) can be expressed by Algorithm 1. Note that the nested if-else statements shown in Algorithm 1 can be represented by a set of linear constraints. Let's start with the inner if-else statements. A binary variable α_j is introduced to construct the following linear inequality constraints to represent the inner if-else statements.

$$t_j^R - t_i - T_i - T_{ij} \geq -M(1-\alpha_j) \tag{25}$$

Algorithm 1: Representation of constraints (17) and (24).

```
if $x_{ij} = 0$ then
    do nothing;
else
    $t_j - t_i = \max(t_j^R - t_i, T_i + T_{ij})$
    if $t_j^R - t_i > T_i + T_{ij}$ then
        $t_j = t_j^R$;
        $w_j = 0$;
    else
        $t_j = t_i + T_i + T_{ij}$;
        $w_j = t_i + T_i + T_{ij} - t_j^R$;
```

$$t_j^R - t_i - T_i - T_{ij} \leq M\alpha_j \tag{26}$$

$$-M(1-\alpha_j) \leq t_j - t_j^R \leq M(1-\alpha_j) \tag{27}$$

$$-M(1-\alpha_j) \leq w_j \leq M(1-\alpha_j) \tag{28}$$

$$-M\alpha_j \leq t_j - t_i - T_i - T_{ij} \leq M\alpha_j \tag{29}$$

$$-M\alpha_j \leq w_j - t_i - T_i - T_{ij} + t_j^R \leq M\alpha_j \tag{30}$$

where M is a real number that is sufficiently large.

When $\alpha_j = 0$, (26), (29), and (30) are binding. This corresponds to the else part of the inner if-else statements. Similarly when $\alpha_j = 1$, (25), (27), and (28) are bindings. This corresponds to the if part of the inner if-else statements.

Now, denote this set of linear inequalities (25)-(30) as $\boldsymbol{A}_j \leq \mathbf{0}$. Then we can represent the outer if-else statement as

$$\boldsymbol{A}_j \leq M(1 - x_{ij}) \cdot \mathbf{1} \tag{31}$$

When $x_{ij} = 0$, none of constraints are enforced. When $x_{ij} = 1$, $\boldsymbol{A}_j \leq \mathbf{0}$ is enforced and the constraints represented by the inner if-else statements are enforced. Now, the nonlinear constraints (17) and (24) are successfully transformed into a set of equivalent linear inequality constraints.

B. Other Nonlinear Constraints

We apply the same procedures to the other nonlinear constraints in the MINLP formulation and derive the equivalent linear constraints below.

(12) can be transformed to

$$E_i - E_j + T_i^{CS} P^{CS} - s_{ij} E^{PM} \leq M(1 - x_{ij}) \tag{32}$$

$$E_i - E_j + T_i^{CS} P^{CS} - s_{ij} E^{PM} \geq -M(1 - x_{ij}) \tag{33}$$

$$\forall v_i \in S_{CS}, v_j \in V, v_j \neq v_i$$

(18) can be transformed to

$$t_j - t_i - T_i^{CS} - T_{ij} \leq M(1 - x_{ij}) \tag{34}$$

$$t_j - t_i - T_i^{CS} - T_{ij} \geq -M(1 - x_{ij}) \tag{35}$$

$$\forall v_i \in V, v_j \in S_{CS} \cup S_T, v_i \neq v_j$$

Now, all the nonlinear constraints have been converted into linear ones. The original MINLP problem is transformed into an equivalent MILP problem. In this study, we use the Gurobi's MILP engine to solve the EV ride-sharing scheduling problem.

IV. Numerical Study

To demonstrate the effectiveness of the proposed EV fleet routing method, various case studies are carried out. First, the optimality of the EV routing solutions derived from the proposed algorithm is validated through 50 small-scale test cases with randomly generated customer requests. Globally optimal solutions are found in all test cases. Second, the trade-off between customer waiting time and total distance traveled by the EV fleet is explored by gradually changing the cost associated with customer waiting time β. The simulation results show that as β increases, the total customer waiting time decreases and the total distance traveled by the EV fleet increases.

A. Simulation Settings

In the case studies, it is assumed that the EV fleet has four vehicles with a battery configuration similar to that of the 2017 Nissan LEAF. The per mile electricity consumption of an EV, E^{PM}, is 0.25 kWh/mile. The battery capacity, E_{max}, is assumed to be 30 kWh. The charging rate, P^{CS}, is 6 kW. The average travel speed of an EV, v_{EV}, is assumed to be 20 mph. The fleet of EV provides ride-sharing service within a square region. The length of each side is 40 miles. The square region can be represented by a 2D coordinate system with its centroid located at (20, 20) and one corner located at (0, 0). All EVs can only move either horizontally or vertically. There is one charging station located at the center of square, which can provide charging services to all EVs simultaneously.

Suppose we have 6 customer requests to be served. Their desired pick-up times are generated through a Poisson process with an expectation of 2 customer requests per hour. The coordinates of pick-up and drop-off locations are sampled independently from a uniform distribution defined on the square. The cardinality of copies of charing station, S_{CS}, is set to be 4. The maintenance cost of an EV is set as C^M =0.04 \$/mile. The electricity cost, C^E, is set to be 0.15 \$/kWh. This is selected based on the average retail price across California in 2016 according to U.S. Energy Information Administration.

B. Optimality of EV Fleet Routing Solution

50 simulation cases were conducted to show that the optimality of EV fleet routing solution can be achieved for a small-scale EV ride-sharing program. In each of the test cases, the EVs are assumed to be parked at the charging station in the beginning of each operation window. At the end of the operation window, all EVs are required to return to the charging station. The customer requests are generated through a Poisson process as described in the simulation settings. The initial battery levels of EVs are sampled from a uniform distribution of $[0, 30]$ kWh. The customer waiting time, β, is set as 5 \$/h. The EVRP-RS problems of the 50 simulation cases are solved by the proposed optimization algorithm. The duality gap of all the solutions are zero, which means the global optimum is obtained for every simulation case. The average computation time of the small-scale EV fleet routing problem is 23.3 seconds on an entry level Dell workstation.

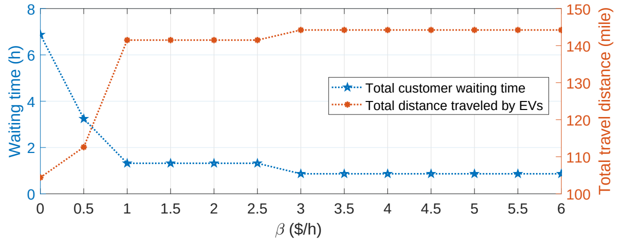

Fig. 3. Total customer waiting time and total distance traveled by EVs under different values of β.

TABLE I
CUSTOMER REQUESTS INFORMATION

Customer	Pick-up time (h)	Pick-up location	Drop-off location
1	0.450	(11.586, 18.294)	(5.007, 5.643)
2	1.629	(9.696, 23.088)	(7.024, 22.211)
3	1.813	(24.722, 24.839)	(8.801, 9.281)
4	2.137	(9.759, 24.955)	(24.309, 16.710)
5	2.805	(20.417, 7.010)	(13.693, 11.537)
6	3.114	(29.751, 22.657)	(29.414, 7.043)

TABLE II
INITIAL STATES OF EVS

	Initial battery	Initial location
EV 1	30 kWh	(30, 30)
EV 2	20 kWh	(10, 10)
EV 3	10 kWh	(20, 20)
EV 4	0 kWh	(20, 20)

C. Trade-off between Wait Time and Total Distance Traveled

In this subsection, we explore the trade-off between customer waiting time and total distance traveled by the EV fleet. We first select one of the 50 customer requests samples generated in Section IV.B. The detailed customer requests information can be found in Table I. The initial locations and battery levels of the four EVs are reported in Table II. This EVRP-RS problem is repetitively solved by varying the customer waiting cost β from 0 $/h to 6 $/h.

Fig. 3 shows the total customer waiting time and total distance traveled by EVs under different values of β. The total waiting time of the customers reaches the highest level when $\beta = 0$. Since no penalty is assigned for the customer waiting time, the EV fleet routing algorithm yields a dispatch solution which minimizes the total travel distance. As β increases, the total customer waiting time decreases while the total distance traveled by the EV fleet increases. The saturation effect takes place when β becomes greater than 3 $/h.

V. CONCLUSIONS

This paper develops an algorithm to determine the optimal routes for an EV fleet to provide ride-sharing services. The EV fleet routing problem is formulated on a complete directed graph as a MINLP problem. The MINLP problem is then converted into an equivalent MILP problem. Numerical studies show that the proposed method can easily find globally optimal EV routes for small-scale problems. The simulation results also show that as we increase the costs associated with customer waiting time, the total customer waiting time decreases while the total distance traveled by the EV fleet increases. Hence, the EV ride-sharing service providers need to strike a balance between fleet operating costs and user waiting time.

Although the proposed MILP-based algorithm does provide globally optimal EV routes for small-scale problems, the MILP problem is NP-hard and does not scale well with the number of customers and EVs. In the future, we plan to develop deep reinforcement learning based approach to solve the EV fleet routing problem for ride-sharing services.

ACKNOWLEDGMENT

We gratefully acknowledge the support from National Science Foundation (NSF) under award #1637258 and Department of Energy (DOE) under award DE-EE0007328.

REFERENCES

[1] K. Naceur and J. Gagné, “Global EV outlook 2017,” *International Energy Agency (IEA): Paris, France*, 2016.

[2] G. B. Dantzig and J. H. Ramser, “The truck dispatching problem,” *Management Science*, vol. 6, no. 1, pp. 80–91, Oct. 1959.

[3] P. Toth and D. Vigo, *The vehicle routing problem*. SIAM, 2002.

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

[5] J. F. Bard, L. Huang, P. Jaillet, and M. Dror, “A decomposition approach to the inventory routing problem with satellite facilities,” *Transportation Science*, vol. 32, no. 2, pp. 189–203, May 1998.

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

[7] 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,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 71, pp. 111–128, Nov. 2014.

[8] J. Lin, W. Zhou, and O. Wolfson, “Electric vehicle routing problem,” *Transportation Research Procedia*, vol. 12, pp. 508–521, 2016.

[9] R. Basso, P. Lindroth, B. Kulcsr, and B. Egardt, “Traffic aware electric vehicle routing,” in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Nov. 2016, pp. 416–421.

[10] A. Montoya, C. Guéret, J. E. Mendoza, and J. G. Villegas, “The electric vehicle routing problem with nonlinear charging function,” *Transportation Research Part B: Methodological*, vol. 103, pp. 87–110, Sep. 2017.

[11] J. R. Montoya-Torres, J. L. Franco, S. N. Isaza, H. F. Jiménez, and N. Herazo-Padilla, “A literature review on the vehicle routing problem with multiple depots,” *Computers & Industrial Engineering*, vol. 79, pp. 115–129, Jan. 2015.

[12] G. S. Bauer, J. B. Greenblatt, and B. F. Gerke, “Cost, energy, and environmental impact of automated electric taxi fleets in Manhattan,” *Environmental Science & Technology*, vol. 52, no. 8, pp. 4920–4928, Mar. 2018.

[13] T. D. Chen, K. M. Kockelman, and J. P. Hanna, “Operations of a shared, autonomous, electric vehicle fleet: Implications of vehicle & charging infrastructure decisions,” *Transportation Research Part A: Policy and Practice*, vol. 94, pp. 243–254, Dec. 2016.

[14] N. Kang, F. M. Feinberg, and P. Y. Papalambros, “Autonomous electric vehicle sharing system design,” *Journal of Mechanical Design*, vol. 139, no. 1, p. 011402, Jan. 2017.

[15] T. Chen, B. Zhang, H. Pourbabak, A. Kavousi-Fard, and W. Su, “Optimal routing and charging of an electric vehicle fleet for high-efficiency dynamic transit systems,” *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 3563–3572, Jul. 2018.