4-2019

Route planning for a fleet of electric vehicles with waiting times at charging stations

Baoxiang LI
Singapore Management University, bxli@smu.edu.sg

Shashi Shekhar JHA
Singapore Management University, shashij@smu.edu.sg

Hoong Chuin LAU
Singapore Management University, hclau@smu.edu.sg

Citation

LI, Baoxiang; JHA, Shashi Shekhar; and LAU, Hoong Chuin. Route planning for a fleet of electric vehicles with waiting times at charging stations. (2019). *Evolutionary computation in combinatorial optimization: 19th European Conference, EvoCOP 2019, Leipzig, Germany, April 24-26, 2019: Proceedings*. 11452, 66-82.
Available at: https://ink.library.smu.edu.sg/sis_research/4372

Published in Evolutionary computation in combinatorial optimization: 19th European Conference, EvoCOP 2019, Leipzig, Germany, April 24-26, 2019: Proceedings. pp. 66-82. Springer. LNCS, vol 11452.
https://doi.org/10.1007/978-3-030-16711-0_5

Route Planning for a Fleet of Electric Vehicles with Waiting Times at Charging Stations

Baoxiang Li, Shashi Shekhar Jha, and Hoong Chuin Lau

Fujitsu-SMU Urban Computing and Engineering Corporate Lab,
School of Information Systems,
Singapore Management University

Abstract. Electric Vehicles (EVs) are the next wave of technology in the transportation industry. EVs are increasingly becoming common for personal transport and pushing the boundaries to become the mainstream mode of transportation. Use of such EVs in logistic fleets for delivering customer goods is not far from becoming reality. However, managing such fleet of EVs bring new challenges in terms of battery capacities and charging infrastructure for efficient route planning. Researchers have addressed such issues considering different aspects of the EVs such as linear battery charging/discharging rate, fixed travel times, etc. In this paper, we address the issue of waiting times due to limited charging capacity at the charging stations while planning the routes of EVs for providing pickup/delivery services. We provide an exact mathematical model of the problem considering waiting times of vehicle based on their arrival at the charging stations. We further develop a genetic algorithm approach that embeds Constraint Programming to solve the problem. We test our approach on a set of benchmark Solomon instances.

Keywords: Electric vehicle routing problem, mixed integer linear programming, constraint programming, genetic algorithm

1 Introduction

With rising pollution levels globally, cities are adopting means and technologies that are social and environmental friendly for various activities. The movement of vehicles on the roads is one of the major contributors of the overall rise in the air pollution over the globe. Recent advancements in the electric vehicle technology with high power and compact batteries have opened up new modalities to shape the transportation for people and goods. The prominent advantage of using electric vehicles is that they do not emit the green house gases that act as the major contributor to the global pollution. For large scale logistics operations such Electric Vehicles (EVs) now provide more viable green technology to serve customers (mostly referred as green logistics). However, EVs have their own share of drawbacks which limits their usage. At the very basic, it is the limited driving range with one complete charge that EVs can scale up to. The maximum driving range of the EVs is around 100-150 miles which gets further reduced by the operating and environmental conditions such as low temperatures, battery age, weight of the vehicle with payload, etc. Further EVs can take more than half-an-hour to completely charge their batteries which is also dependent on the type of batteries and the capacity of charging outlets.

To make way for the EVs to join the mainstream commercial logistics services, one needs to consider all such constraints for route planning and scheduling of these vehicles. Although researchers have considered the limited battery capacity of the EVs in planning the routes, the limited capacity of charging stations have scarcely been addressed in the literature. The scarcity of charging station resources is particularly pertinent in cities where EVs are still in the infant stage of adoption. The station locations, number of charging outlets and charge delivery capacities put limitations to the number of EVs that can be charged simultaneously at a charging station. In addition, due to the long charging time of EVs with limited charging infrastructure, one needs to consider the significant queuing time at the charging stations while planning routes for logistics fleet operations.

In this paper, our focus is to plan the routes for a fleet of EVs for logistics operations by allowing the EVs to queue and wait at the fixed capacity charging stations. We present an exact mathematical formulation of the classical VRP problem with the considerations of limited battery charge, cargo capacity constraints and waiting times at the charging stations. One major difficulty in modeling this problem is to handle the queuing sequence at each charging station, which we overcome by using a subtour elimination technique commonly used for solving the Traveling Salesman Problem (TSP). Such considerations can make the route planning of EVs more viable for real world operations. Further, the last mile delivery operations can further be streamlined with such EVs taking the central stage with carefully planned routes and visits to the charging stations. Hence, our work extends the current literature by addressing the challenging aspect of route planning for a logistics fleet with an explicit modeling of waiting times at the charging stations. Further in our proposed solution, we handle the constraints (such as charging sequence of EVs) by integrating a Constraint Programming (CP) model in our heuristic approach.

The rest of the paper is organized as follows. In the next section, we discuss the related literature for solving EV routing. The succeeding sections present the problem formulation as an MIP considering the limited battery capacities of EVs and waiting times at the charging stations. Next, we present a GA method to solve large problem instances and the results gathered from the numerical experiments. The last sections concludes the paper with avenues of future research.

2 Related Literature

The Electric Vehicle Routing Problem (EVRP) is a special case of the traditional Vehicle Routing Problem (VRP) with the additional set constraints due to the electric engine technology.

Schneider et al. [1] describe the EVRP with Time Windows (EVRPTW) as an MIP formulation and propose a heuristic combining variable neighborhood search and tabu search. In their problem formulation, the authors considered charging stations having no capacity constraint on number of chargers. In addition, they have not considered any waiting times at the charging stations. Goeke and Schneider [2] extended the EVRPTW model for a mixed fleet of conventional and electric vehicles with time windows and capacity constraint. The authors emphasize on the energy consumption models for the conventional and electric vehicles which is non-linear by nature. Sassi and Oulamara [3] considered an electric vehicle scheduling and optimal charging problem in order to assign vehicles to tours and minimize the charging cost. In their model, they considered constraints related to chargers, electricity grid, and EVs driving range.

In [4], the authors discuss the formulation of a mixed fleet with a case study at the city of Amsterdam. In their objective, the authors factors in the fixed cost of vehicles and the variable cost which depends on the en-route time and distance. Although the authors consider the battery capacity constraints of the electric vehicles, there is no consideration given to the amount of time a vehicle may need to spend at a charging station. The authors in [5] considers partial charging of electric vehicles with non-linear charging functions. The non-linearity of the battery charging function is handled using a piece-wise linear approximation. The authors model the problem to minimize the total time which includes the travel time and time spent at the charging stations.

Qin and Zhang [6] discuss the scheduling of charging activities with minimal waiting time of EVs in a network of EVs and charging stations. The authors consider EVs in general not involving a logistic fleet operation. Although, the authors address the problem of waiting time at the charging stations which translates to the availability of the charging outlets, the routing problem of the vehicle is not considered. Stein et al. [7] propose a market driven approach to create charging schedules. The authors provide two restricted interfaces for collecting the preferences of EV owners. While one of the interfaces reduce the dimensionality of the reporting space, the second interface reduces the reporting space to a fixed set of options. Additionally, Froger et al. [8] address the problem of limited capacity charging stations for vehicle routing and provide matheuristic based approach to solve the problem. EVRP can be considered a specialized problem of the larger set of the Green Vehicle Routing Problem (G-VRP) that focuses on the use of alternative fuel sources for the vehicles. Hence, G-VRP also focuses on similar issues such as the limits to driving range and limited refueling stations. In [9] for example, the authors discuss the G-VRP and provide an MIP formulation for this problem.

As can be noted from the discussions above, the literature on EVs is segregated in two broad groups - those who consider it as a routing problem for fleet of EVs and those who consider the problem of scheduling and congestion at the charging stations. There is also a third variant which is related to the placement of charging stations infrastructure such as the one discussed in [10]. However, the problem of restricted number of charging outlets at the charging stations for the EV fleet operations does not seem to be addressed in the literature. The waiting times due to unavailability of charging outlets or scheduling the vehicle at the charging stations while planning their routes in a traditional VRP setting warrants further considerations to come up with real-world solutions.

Table 1. Variables and parameters of our proposed EVRP model

Notations	Description
0, N+1	Start and end depot
V	Set of customer nodes $\{1 \ldots N\}$
V_0	Set of customer nodes with start depot
V_{N+1}	Set of customer nodes with end depot
$V_{0,N+1}$	Set of customer nodes with start and end depots
V'	Set of nodes - the customer nodes and charging station outlets associated dummy nodes
V'_0	Set of nodes - the customer nodes, charging stations outlets associated dummy nodes, and depot $\{0\}$
V'_{N+1}	Set of nodes - the customer nodes, charging stations outlets associated dummy nodes, and depot $\{N+1\}$
$\mathit{\Omega}$	Set of vehicles $\{1 \ldots W\}$
R	Set of charging station outlets $\{1 \ldots K\}$
R^k	Set of dummy nodes (indicate potential vehicles visit for a station) for charging station outlet k, set size equals to W, and $W \leq N$
Parameters	
T	Maximum time duration of a route
C	Cargo capacity of each vehicle
B	Battery capacity of each vehicle
δ	Rate of battery discharge
β	Rate of battery charging
s_i	Service time at node i
d_{ij}	Distance between two nodes i and j
t_{ij}	Travel time between two nodes i and j
D_i	Demand at node i
Variables	
x_{ij}	Binary decision variable for visit from node i to j; 1 if (i,j) is traveled, 0 otherwise
τ_i	Decision variable specifying the time of arrival at node i
$\tau^{\hbar}_{N+1}$	Decision variable specifying the time of arrival at the end depot for vehicle $\hbar$
w_i	Decision variable specifying the time of start of charging at the charging station outlet i
f_i	Decision variable specifying the remaining cargo on arrival at node i
y_i	Decision variable specifying the remaining charge available on arrival at node i
b^k_{uv}	Binary decision variable for a vehicle visiting the u^{th} dummy node of charging station outlet k in sequence (order) index v; 1 if u takes the sequence (order) index v and 0 otherwise
z^k_u	Decision variable specifying the sequence (order) of charging service at station outlet k for each $u \in R^k$
$a^1_{m,u,v}, a^2_{m,u,v}, a_{m,u,v}$	Intermediate binary decision variables for deriving the service sequence of visits at a charging station outlet
a_{uv}	Binary decision variable to derive the sequence of visit at the charging station; 1 if dummy node v is visited after u and 0 otherwise

3 Mathematical Model

The primary objective of our work is to address the limited capacities of the charging stations which may result in the formation of queues in order to service a logistics fleet of EVs. A charging station consists of a set of charging outlets wherein a vehicle is plugged in to recharge its battery. Such charging outlets could be limited in number at the charging stations depending upon the electric grid capacity and charging stations' space constraints. In this section, we provide the exact mathematical model for our EVRP considering waiting times of the vehicles at the charging stations.

Let V be the set of customer nodes and R be the set of charging stations. Let the start and end depot node instances be denoted by 0 and $N+1$. We will use 0 and $N+1$ in the subscript to denote the set of nodes with start or end depot instance or both as in [1]. Table 1 lists the variables and parameters used in our model. To allow multiple visits to the charging station outlets, we create dummy nodes for every charging stations outlet in R. We further consider a linear discharge of the vehicle batteries and the distance among the nodes is Euclidean. We assume that the vehicles leave the depot with full charge and will not visit charging station outlet before completing their first delivery. Moreover, each visit to the charging station outlet results in full battery charge of the vehicles.

The objective function in our mixed integer program is to minimize the sum of route durations, which includes the travel time, the service time at the customer nodes, the charging and waiting times at the charging station outlets.

$$\min \sum_{\hbar \in \mathit{\Omega}} \tau^{\hbar}_{N+1} \tag{1}$$

Subject to:

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

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

$$\sum_{i \in V'_{N+1}} x_{ji} - \sum_{i \in V'_{0}} x_{ij} = 0, \forall j \in V' \tag{4}$$

$$\tau_i + (t_{ij} + s_i)x_{ij} - T(1 - x_{ij}) \leq \tau_j, \forall i \in V_0, \forall j \in V^{'} \tag{5}$$

$$w_i + \beta(B - y_i) + t_{ij}x_{ij} - (T + \beta B)(1 - x_{ij}) \leq \tau_j, \forall i \in V_0, \forall j \in V^{'}_{N+1} \tag{6}$$

$$\tau_i + (t_{i,N+1} + s_i)x_{i,N+1} - T(1 - x_{i,N+1}) \leq \tau^{\hbar}_{N+1}, \forall i \in V, \hbar \in \mathit{\Omega} \tag{7}$$

$$w_i + \beta(B - y_i) + t_{i,N+1}x_{i,N+1} - (T + \beta B)(1 - x_{i,N+1}) \leq \tau^{\hbar}_{N+1}, \forall i \in \bigcup_{k \in R} R^k, \hbar \in \mathit{\Omega} \tag{8}$$

$$\tau_i \leq w_i, \forall i \in R^k \tag{9}$$

$$u_i - D_i x_{ij} + C(1 - x_{ij}) \geq f_j \geq 0, \forall i \in V_0^{'}, \forall j \in V_{N+1}^{'} \tag{10}$$

$$C \geq f_0 \geq 0 \tag{11}$$

$$y_i - \delta d_{ij} x_{ij} + B(1 - x_{ij}) \geq y_j \geq 0, \forall j \in V_{N+1}^{'}, \forall i \in V \tag{12}$$

$$B - \delta d_{ij} x_{ij} \geq y_j \geq 0, \forall j \in V_{N+1}^{'}, \forall i \in R^k, k \in R \tag{13}$$

$$\sum_{u \in R^k} b_{uv}^k \leq 1, \forall v \in \mathit{\Omega}, \forall k \in R \tag{14}$$

$$\sum_{v \in V} b_{uv}^k - \sum_{v \in V} x_{uv} = 0, \forall u \in R^k, \forall k \in R \tag{15}$$

$$\sum_{v \in V} v b_{uv}^k = z_u^k, \forall u \in R^k, \forall k \in R \tag{16}$$

$$z_v^k - z_u^k - (m - 1) - M a_{m,u,v}^1 \leq 0, \forall m \in \mathit{\Omega} \backslash N, u, v \in R^k \tag{17}$$

$$(m + 1) - (z_v^k - z_u^k) - M a_{m,u,v}^2 \leq 0, \forall m \in \mathit{\Omega} \backslash N, u, v \in R^k \tag{18}$$

$$a_{m,u,v}^1 + a_{m,u,v}^2 - a_{m,u,v} \leq 1, \forall m \in \mathit{\Omega} \backslash N, u, v \in R^k \tag{19}$$

$$\sum_{m \in \mathit{\Omega} \backslash N} a_{m,u,v} = a_{uv}, \forall u, v \in R^k \tag{20}$$

$$(a_{uv} - 1)M \leq \tau_v - \tau_u, \ \forall u, v \in R^k \tag{21}$$

$$w_v - w_u + (1 - a_{uv})M \geq (B - y_u)\beta, \forall u, v \in R^k \tag{22}$$

$$x_{ij} \in \{0, 1\}, \forall i \in V_0^{'}, j \in V_{N+1}^{'} \qquad b_{uv}^k \in \{0, 1\}, \forall u \in R^k, v \in \mathit{\Omega}, k \in R \tag{23}$$

Equation (1) minimizes the sum of routeduration of each individual route. Constraint (2) specifies that all the customer nodes must be visited. Constraint (3) defines the visit to the charging station outlets. The constraint in (4) ensures flow conservation, which basically means all the outflows from a node and inflows to a node must be equal i.e. all the vehicles that leave a node must be equal to the number of vehicles that arrived at that node. Constraints (5) and (6) provide the arrival time at a node from a customer node and charging station outlet respectively. Constraints (7) and (8) provide the arrival time at the depot from a customer node and charging station outlet. Constraint (9) forces that the start of charging at the charging station outlet must be after the arrival and possible waiting. Constraints (10) and (11) is for the fulfillment of customer demand while constraints (12) and (13) are to ensure that the vehicles batteries always have positive charge.

Constraints (14)-(22) are meant to derive the waiting times of the vehicles that queue at a charging station outlet. The basic idea here is to sequence the vehicle based on their arrivals at the charging station outlet and use this sequence to calculate the start of charging of the vehicles at the charging station outlet. Constraint (14) forces that whenever a vehicle visits the charging station outlet, it gets assigned a sequence index. Constraint (15) ensures that all the vehicles that has arrived, also leave the charging station outlet, while (16) assigns the sequence index to the vehicles that arrive at the charging station outlet after serving a given customer node. Constraints (17)-(20) ensure the vehicle sequence of charging at the charging station outlet. Finally, constraints (21)-(22) calculate the time to start the charging based on the sequence index of the vehicles.

The binary variables $a^1_{m,u,v}$, $a^2_{m,u,v}$ and $a_{m,u,v}$ are introduced to get the visiting order of the EVs at a charging station outlet based on m. The associated constraints (17)-(20) ensures that, if $a^1_{m,u,v}$ equals to 1, $Z^k_v - Z^k_u > m - 1$, similarly, if $a^2_{m,u,v}$ equals to 1, it means $Z^k_v - Z^k_u < m + 1$, and if $a_{m,u,v}$ equals to 1, it means $Z^k_v - Z^k_u = m + 1$. Constraint (17-20) is to define the sequence of vehicles visit at charge station and deduct the waiting time. Note that m belongs to the set of {1,..., N-1}, hence, the big M in (17) and (18) can be defined as N.

4 Genetic Algorithm (GA)

In this Section, we present our Genetic Algorithm (GA) approach to solve our EVRP problem defined above. The initial population size is set to Λ. We use a greedy algorithm to generate the initial population of solutions, which vary the order of selection of customer nodes to generate different individuals of the population. The fitness function used to evaluate an individual solution is the total route duration of all the routes to serve all the customers in the problem instance. The children solutions are improved by applying a local search heuristics and battery charge station visiting algorithm.

To generate an initial solution, we first assign an empty route that start and end at depot to each vehicle. We then randomly insert a customer node at the best position of the route for the vehicle with the smallest additional cost. If the battery charge of the vehicle cannot support visiting to the next customer, we add a charging station outlet visit. Once the route has been updated, we block the corresponding charging station outlet time interval occupied by this vehicle. When no more customer nodes could be added to the first vehicle, we proceed to the second vehicle, and so on. We repeat this process until all customers are fully served.

We represent each individual solution (chromosome) as a sequence of node labels. Hence, each chromosome starts with a depot label and ends with a depot label. All the customers and stations labels are inserted in between based on the routes taken. The end/start of each route is marked by the depot label. Hence, a single chromosome can be divided into a set of segments with starting and ending depot labels. Each such segment represents a vehicle route. Figure 1 depicts the representation of a chromosome with three routes, two charging station outlets and eight customer nodes wherein **D** denotes depot, **C** denotes customer nodes, and **S** denotes the charging station outlets. Hence, when we evaluate the fitness function, we will translate the chromosome to multi-routes, which is split by **D**, then, we calculate the sum of route durations.

D	C1	C2	S1	C3	S2	D	C5	S2	C4	C6	D	C7	S1	C8	D

Fig. 1. An individual solution representation in the GA

Table 2. Variables and parameters of the CP model

Parameters	
$succ_i$	Successive node in a path after serving customer i
$t_{i,succ_i}$	Travel time from node i to the successive node
λ_i	Battery charging time plus service time associated with node i
Variables	
arr_i	Earliest arrival time to start service at a node i
$serv_i$	Service start time at node i
s_k	Start depot node of vehicle k
e_k	End depot nodes of vehicle k
R	Set of charging station
R^o	Set of nodes (dummy nodes) of o^{th} charging location
C^o	Charging station outlet capacity of o^{th} charging location
V	Set of nodes in the vehicles route
E	Set of end depot nodes in the vehicles route

4.1 Constraint Programming (CP) for Fitness Evaluation

We vary the order of selection of customer nodes to generate different individuals of the population. The fitness function used to evaluate an individual solution is the total route duration to serve all the customers in the problem instance.

Let $c(s)$ be the routing cost (that is, sum of the travel time and service time) of route s. The fitness function f is defined to be the sum of routing costs and waiting times of all routes, each route has a fitness value $f(s) = c(s)+w(s)$, where $w(s)$ is the total waiting time incurred for route s. Interestingly, for a route s, $c(s)$ is easy to calculate, but the calculation of waiting time is complicated, as vehicles can visit charging station outlet many times, and wait either at charging station or customer locations. This makes the calculation of the consequent waiting times (if any) non-trivial. For this purpose, we apply a Constraint Programming (CP) model for calculating the optimal fitness value.

For the CP model, we first find that the solution satisfies the capacity and battery charge constraints for the vehicle. If the constraints are satisfied, then, we apply the CP model to calculate the fitness value of the total route duration.

$$\min \sum_{k \in \Omega} serv_{e_k} - arr_{s_k} \tag{24}$$

Subject to:

$$arr_i \leq serv_i, \forall\ i \in V \tag{25}$$

$$serv_i + \lambda_i + t_{i,succ_i} = arr_{succ_i}, \forall\ i \in V \backslash E \tag{26}$$

$$CUMULATIVE(\{serv_i, \lambda_i : i \in R^o\}, 1, C^o), \forall\ R^o \in R \tag{27}$$

The variables and parameters used in the CP model are listed in Table 2. The objective function (24) minimizes the total route duration. Constraints (25)–(26) define the earliest arrival time, and start time of charging at node i. The charging resource constraint (27) is modeled using the global constraint, where $serv_i$, λ_i, 1 and C^o represent the start time of charging, duration, resource requirement, and the capacity of the resource respectively. The cumulative constraint (27) specifies the requirements on tasks which need to be scheduled on a number of resources. It expresses the fact that at any time instant the total use of these resources for the tasks does not exceed a given limit. For our problem, we have a list of charging tasks, each task requires only one resource, and the upper limit of the amount of charging resources limit equals to C^o.

The average running time for our CP model is around 2 seconds for 100-nodes problem instances. Hence, we provide the following model simplification procedure before running the CP model: 1) If a vehicle never visits any charging station outlet, we can remove the whole path of vehicle k from the model; 2) If there are fewer than C^o visits for o^{th} charging station, it is unnecessary to check the cumulative constraint (27) as the constraint is automatically satisfied; 3) Suppose a charging station has only been visited by the same vehicle k, then we can also remove the whole path of vehicle k since there is no dependency on other vehicles.

Unfortunately, it is still quite time-consuming to apply CP to compute the exact waiting time in every iteration, hence, we only trigger CP under two cases: 1) when we manage to find a solution that satisfies vehicle capacity and time window constraints; and 2) after every p (a parameter that we set default value to 20) iterations. In other cases, we apply a heuristic to estimate the waiting times. The basic idea of this heuristic is to simulate a virtual booking system that will compute an approximated waiting time and station capacity violation whenever a vehicle visits a charging station as follows.

Heuristic for Charging Station Waiting Time Calculation Let us define a charging station visit as a special task. Since waiting time is a global variable that depends on the arrival times at *all* charging station visited by all vehicles, we are not able to compute the exact waiting time by simply considering each trip independently. Therefore, for a given charging task i in the solution, we apply Algorithm 1 to estimate the waiting time. Let τ_i be the earliest service start time for charging task i, and λ_i be the charge time at charge station m.

- Case 1: charging station m has an available (i.e. unoccupied) time interval at $[\tau_i, \tau_i + \lambda_i]$.
  This is very straightforward - we assign $[\tau_i, \tau_i + \lambda_i]$ to charging task i (shown in lines 1-5 in Algorithm 1).
- Case 2: charging station m has an available time interval with length at least λ_i but later than τ_i. Let Θ be the set of booked charging outlets intersect with time τ_i, and τ_i' denote the earliest completion time of charging tasks in Θ. We assign $[\tau_i', \tau_i' + \lambda_i]$ to charging task i (shown in lines 6-8).

Algorithm 1: Virtual charging station outlet booking algorithm

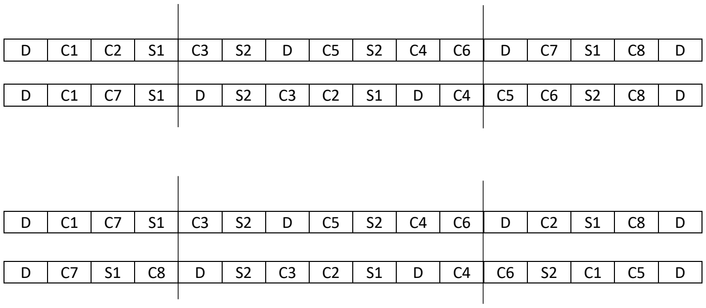
```
   Input: battery charge task $i$, earliest service start time $\tau_i$, service duration $\lambda_i$,
          initial $\varpi_i = 0$, $\overline{v_i} = 0$
 1   if charge station $\epsilon_j$ is available at time interval $[\tau_i, \tau_i + \lambda_i]$
 2      $\varpi_i = 0$;
 3      assign battery charge task $i$ to time interval $[\tau_i, \tau_i + \lambda_i]$ on $\epsilon_j$;
 4     stop
 5    end if
 6 Let $\Theta$ be the set of booked battery charge tasks that intersect with time $\tau_i$,
   and $\tau_i'$ denotes the earliest completion time of tasks in $\Theta$
 7 if $[\tau_i', \tau_i' + \lambda_i]$ is within the planning horizon (no penalty)
 8   assign battery charge task $i$ to $[\tau_i', \tau_i' + \lambda_i]$
 9   record waiting time $\varpi_i$, and update charging outlet booking
10 else
11   assign battery charge task $i$ after $\tau_i$
12 record waiting time $\varpi_i$ and penalty $\overline{v_i}$, and update charging outlet booking
13 end if
   Output: $\varpi_i$ and $\overline{v_i}$
```

- Case 3: there is no available time interval that can fit charging task i. We assign task i after time τ_i (line 11). Note that by doing so, the booked charging outlets time interval will be overlapping (i.e. the solution is infeasible). Therefore, penalties need to be imposed (line 12).

Overall, the total waiting time $w(s)$ is evaluated by sum of all tasks ϖ_i and $\overline{v_i}$.

4.2 Selection and Crossover

The top K elite individuals are kept for the next generation based on their fitness. All the individuals will participate to produce new individuals generated by applying the crossover operator. Noted that the crossover rate is 100%, as we will apply our local search operator in Section 4.4, which is more powerful than general mutation operator in the GA. The pair of parents for crossover are chosen using tournament selection method [11]. This means, we randomly sample two individuals from the individuals. The partially matched crossover operator is used to create two new individuals to replace the parents. For given parents P_1 and P_2, two cutting points are chosen. With the middle sub-path kept intact, we swap the arc and nodes between two parents. Figure 2 shows an example. The top two rows represent parents individuals, and bottom two rows stand for the children individuals. The crossover operation results in the creation of new children. All the customers must be served in our problem instance, hence if left arcs or nodes cannot be inserted to the child because of vehicle load or vehicle available time infeasibility, we add empty vehicles and insert the leftover arcs/nodes to the empty vehicles. The two new individuals along with the current population form the new population for the next generation of GA. We terminate GA when there is no improvement in the best solution for a fixed number of generations or the number of iterations reaches the upper limit.



D	C1	C2	S1	C3	S2	D	C5	S2	C4	C6	D	C7	S1	C8	D
D	C1	C7	S1	D	S2	C3	C2	S1	D	C4	C5	C6	S2	C8	D

D	C1	C7	S1	C3	S2	D	C5	S2	C4	C6	D	C2	S1	C8	D
D	C7	S1	C8	D	S2	C3	C2	S1	D	C4	C6	S2	C1	C5	D

Fig. 2. Crossover operator

4.3 Columns Based Chromosome Generation

In every 1000 iterations of GA generations, we generate new chromosomes by recording the last 1000 iterations of feasible vehicle paths in the population as columns. We find the best combination of those columns, generate new chromosomes and add them to the population. Let R be the set of all columns, V be the set of customers included in the columns, and P stands for presetting price of a customer. Parameters include c_r and a_r^i, where c_r equals to the travel time of column and r, a_r^i is a parameter that equals 0 or 1, where 1 indicates node i is served by column r. Binary decision variables include y_i and z_r, y_i which equals to 1 if customer i is selected. Similarly, z_r equals to 1 if column (trip) r is selected. We aim to generate better solutions from a population of historical solutions, with the Set Covering formulation shown as follows:

$$\max \sum_{i \in V} P y_i - \sum_{r \in R} c_r z_r \tag{28}$$

Subject to:

$$\sum_{r \in R} a_r^i z_r = y_i, \quad \forall \ i \in V \tag{29}$$

We assume serving a customer leads to revenue P, hence the objective function (28) maximize the revenue minus the total travel time. We use y_i to check customer i be served or not in Constraints (29). If not all the customers can be served, we will insert the leftover customers to the solution one by one to the position with the smallest insertion cost added.

4.4 Local Search on Solutions

Once a new chromosome been generated, we improve the solution by local search. Customer nodes are selected and added to a perturbation set. To select the customer nodes, we use an objective value based operator. By checking the vehicle routes, we calculate the objective value after removing one customer node, then, descending sort the values, we choose the top 10% customer nodes and added to the perturbation set.

After the customer nodes selection, three perturbation operators (I1-I3) are used. The probability of using first operator is set to 0.5 while the probability of using second and third operators is set to 0.25. We accept the new solution if the fitness value is better after applying local search.

1-by-1 (I1): The selected customer nodes are sequentially removed one by one and reinserted into the best position (the highest improvement for the current objective value).

1-by-1 different vehicle (I2): Suppose, customer node i is removed from some route k, this operator tries to insert the customer node i into different route (which is not k) in a better position.

1-by-1 same vehicle (I3): Suppose, customer node i is removed from some route k, this operator tries to insert the node i into the same route k again but in a better position.

4.5 Management of Charging Station Visits

If no feasible solution can be obtained because of the violation of battery limit constraint, we add charging station outlet visits (described in Algorithm 2) to fix the infeasibility. Basically, we include charging station visits if only the vehicles cannot reach to the customer nodes ρ_i with the available charge in their batteries. The position to insert the charging station visit can be any location before ρ_i, hence, we find the best position to insert the closest charging station (line 4-9 of Algorithm 2). In case, a charging station cannot be inserted (due to long distance) then we split the visits to customers to two or more vehicles.

During the GA iterations, once a charging station has been removed, we reinsert it only if the same would lead to a better fitness value. Besides that, every 20 iterations, we double check the current iteration solution, and remove useless charging stations and insert new charging station. Once a new best solution or new best feasible solution has been found, the charging station visit gets updated as well.

5 Numerical Experiments

To evaluate our approach, we performed numerical experiments using two sets of problem instances on an Intel Xeon E5-2667v4 8C/16T (3.2GHz) 16 core CPU 32 GB RAM machine. We used a batch file to utilize all the 16 cores while running the numerical experiments wherein each instance used one single core. The first set is the self-generated small instances (having 5 and 10 number of customers), aimed to compare the solutions between the heuristic approach and math model (using CPLEX solver version 12.8.0) along with the verification of the solution details such as travel time, waiting time and charging time. The second set is based on instances provided by Schneider et al. [1] that are adapted from Solomon [12] instances to include charging stations. These instances are provided in three different categories with 100 customer nodes. With our specific problem setup that focuses on limited charging capacity at the charging stations,

Algorithm 2: Insert charge station visits in a given vehicle path

```
Input: path $(\rho_1, \rho_2, \dots, \rho_N)$, fitness value $f(s)$ and $f(s)_{old}$ (both equals to
       fitness value of path $s$)
do
  for all $\rho_1, \rho_2, \dots, \rho_N$ do
    check the battery when vehicle arrive at location $\rho_i$
    if battery is lower than 0
      find the best charge station in $1, \dots, K$ and best insertion point between
      $\rho_1, \dots, \rho_{i-1}$ with smallest fitness value
      if lowest fitness value is smaller than $f(s)_{old}$
          insert charge station, update path $s$, and $f(s)_{old} := f(s)$, $f(s)$:=
          smallest fitness value
      else break;
    end if
  end for
while $f(s) < f(s)_{old}$
Output: $f(s)$ and updated path $s$
```

we realized that generating instances suitable this problem is in itself a complex problem. As the literature do not provide Solomon instances with more than 100 customer nodes, we only focus on the variety of 100 nodes instances for our experimentation. According to our problem definition, we ignore the customer time windows available in the Solomon instances. All test instances can be found at https://unicen.smu.edu.sg/research/urban-logistics/electric-vehicles-routing-problem-waiting-times-charging-stations.

For the the first set of instances, the battery capacity was fixed at 30, charging station outlet capacity equals to 1, and the vehicle capacity was set to 100. The service time at all the customer nodes was set to 10 time units. The second set of instances used the parameters as defined in the type "C", "R" and "RC" of Solomon instances. The vehicle battery capacity was set to 62.14 or 79.69, number of charging station locations was set to a random number between 5-18, charging station outlet capacity equals to 1, the location is randomly chosen within the spatial distribution, and charging rate was set to full charge battery divided by average customer service time. For GA, we use the following parameter values: initial population size $\Lambda = 100$, number of top elite individual $K = 20$, number of iterations was set to 5000, penalty parameter equals to 20 for waiting time approximation, and customer price in column based operator was set to twice of average travel time between nodes. All the experiments were performed for 10 times and the best result from the multiple runs are reported.

Before discussing the results, we depict the evolution of GA in Figure 3 for one typical instance. As can be seen, the fitness value decreases in the number of generations, and improves from 5979 to 2218 in about 3000 generations and stop improving afterwards.

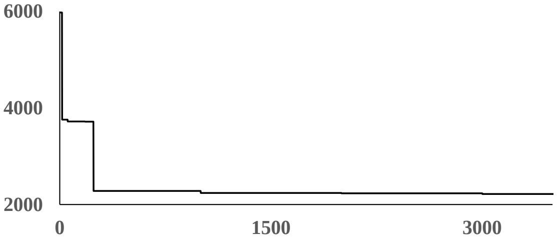


Fig. 3. GA converge curve

5.1 Results

Table 3 compares the results from the exact math model solved using Cplex and the GA based heuristics for the first set of small instances. The table shows the fitness value which is the sum of route durations, travel time (TT), service time (ST), the combined waiting and charging time (WCT), and the running time in seconds for all different instances. As can be observed the gap between the travel time for Cplex and GA is quite comparable for the instances with 5 customer nodes. A similar trend can be noted for the waiting and charging time. However, for instances with 10 customer nodes, GA based solutions seems to spend less time in travel as well as waiting and charging. The results shows that considering waiting time at the charging stations while planning the routes for EV based fleet is crucial as the amount of time spend in waiting and charging is almost equal to the travel time.

One can see, for the 5 customer node instances, Cplex can solve it quickly, while for the instances with 10 customer nodes, we have to stop Cplex after 3 hrs (10800 secs) and get the best solution discovered so far. In comparison to the best solution found by Cplex, the GA heuristics fairs well with very less execution time. For problem instances with 5 customer nodes, there are very low gap between the solutions while in case of instances with 10 customer nodes, GA always finds a better quality solution in very less time as compared to the Cplex.

Table 4 lists the best and average results for the set of Solomon instances using the GA based heuristics approach. The best objective value (from 10 runs) is the sum of the route durations of all the vehicles after they return to the depot. On average, the CPU time equals to 10 minutes overall instances. Since our objective value involve charging and waiting times at the charging stations, the same is not directly comparable with the literature [12]. We observed that the "R" group Solomon instances have longest travel time, while "C" and "RC" group instances have similar travel times. This happens as the distribution of "R" group Solomon instances are more disperse than other two groups. Further, the waiting times of "C" group instances are much longer than other two groups, mainly because of the difference in the rate of charging.

Table 3. Cplex and GA heuristic based solution on 5 and 10 nodes instances

Instance	Cplex					GA				
	Vehicles	Fitness	TT	WCT	Time (sec)	Vehicles	Fitness	TT	WCT	Time (sec)
5-1	3	405	196	159	165	4	427	228	149	10
5-2	3	400	195	155	315	3	400	180	170	11
5-3	4	522	236	236	46	4	554	229	275	16
5-4	4	493	245	198	44	4	497	227	220	11
5-5	4	489	246	193	22	4	489	223	216	19
10-1	4	747	358	289	10800	5	696	316	330	24
10-2	4	783	365	328	10800	4	702	332	270	18
10-3	4	1160	546	514	10800	4	571	280	191	32
10-4	4	984	437	450	10800	4	943	354	489	29
10-5	3	665	318	247	10800	3	622	286	246	35

Table 4. Results of GA based heuristics solution on the 100-node Solomon instances.

Instance	Best Obj.	No. Veh.	TT	WCT	Ave. Obj.	Ave. No. Veh.	Ave.TT	Ave.WCT
c101	10753	12	1133	620	11234	15	1373	862
c102	10475	13	1200	275	10974	16	1540	434
c103	10740	12	1140	600	11712	22	1895	817
c104	10732	15	1255	477	11709	22	1906	803
c105	10698	13	1328	470	11104	13	1390	714
c106	10548	11	1085	463	11064	13	1477	588
c107	10883	17	1497	386	11684	22	1757	927
c108	10654	13	1123	531	11317	25	1902	416
c109	10843	19	1244	599	11319	18	1657	662
r101	2329	14	1246	83	2941	18	1765	176
r102	2385	16	1320	65	2700	18	1587	113
r103	2467	13	1353	114	2885	20	1542	343
r104	2315	14	1216	99	2431	15	1345	86
r105	2392	17	1322	70	2931	22	1789	142
r106	2284	12	1192	92	2605	17	1509	96
r107	2270	14	1193	77	2589	17	1475	115
r108	2250	14	1185	65	2559	15	1457	102
r109	2375	15	1263	112	2545	15	1442	103
r110	2312	14	1209	103	2572	18	1493	79
r111	2343	14	1273	70	2683	17	1561	122
r112	2158	12	1097	61	2324	14	1239	86
rc101	2667	15	1608	59	2883	16	1787	96
rc102	2748	17	1655	93	3038	18	1897	141
rc103	2729	18	1657	72	3088	17	1968	120
rc104	2730	15	1645	85	2902	16	1805	98
rc105	2847	17	1746	101	3175	19	2015	160
rc106	2997	18	1843	154	3199	19	2041	158
rc107	2820	15	1726	94	2960	16	1838	122
rc108	2768	15	1690	78	3207	19	2083	124

Ave. : Average, Obj.: Objective value, TT: travel time
WCT: combined waiting and charging time, No. Veh.: Number of Vehicles

6 Conclusions

The advancement of electric vehicles technology has paved the way forward for EV based logistic fleets for customer delivery. However, the route planning with EVs brings forward new challenges to the already complex problem of vehicle routing. In this paper, we consider the problem of formation of vehicle queues at the charging stations en-route to customers. We consider the extra waiting time due to the queue at the charging stations to formulate an exact mathematical model for route planning of EV based logistics fleet. The evaluations are performed on a a set of problem instances catering to specific requirement of our problem setup. Although it is difficult to get a solution from the exact mathematical model beyond 10 customer nodes, the GA heuristic based solution has shown comparable performance and can be quickly executed for large size instances.In the future, we plan to extend our work by adding more practical constraints in our model such as customer time-windows, partial charging of vehicle at the charging stations, etc. Further, using the current results as benchmark, more comparative evaluations with variety of instances can be carried out in order to validate the scalability and efficiency of our GA based heuristics approach.

References

1. Schneider, M., Stenger, A., Goeke, D.: The electric vehicle-routing problem with time windows and recharging stations. Transportation Science 48(4), 500–520 (2014)
2. Goeke, D., Schneider, M.: Routing a mixed fleet of electric and conventional vehicles. European Journal of Operational Research 245(1), 81–99 (2015)
3. Sassi, O., Oulamara, A.: Electric vehicle scheduling and optimal charging problem: complexity, exact and heuristic approaches. International Journal of Production Research 55(2), 519–535 (2017)
4. Van Duin, J., Tavasszy, L.A., Quak, H.: Towards electric-urban freight: first promising steps in the electric vehicle revolution (2013)
5. Montoya, A., Guéret, C., Mendoza, J., Villegas, J.: The electric vehicle routing problem with partial charging and nonlinear charging function. Ph.D. thesis, LARIS (2015)
6. Qin, H., Zhang, W.: Charging scheduling with minimal waiting in a network of electric vehicles and charging stations. In: Proceedings of the Eighth ACM international workshop on Vehicular inter-networking. pp. 51–60. ACM (2011)
7. Stein, S., Gerding, E.H., Nedea, A., Rosenfeld, A., Jennings, N.R.: Bid2charge: Market user interface design for electric vehicle charging. In: Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems. pp. 882–890. International Foundation for Autonomous Agents and Multiagent Systems (2016)
8. Froger, A., Mendoza, J.E., Jabali, O., Laporte, G.: A Matheuristic for the Electric Vehicle Routing Problem with Capacitated Charging Stations. Research report, Centre interuniversitaire de recherche sur les reseaux d'entreprise, la logistique et le transport (CIRRELT) (Jun 2017), https://hal.archives-ouvertes.fr/hal-01559524

9. Erdoğan, S., Miller-Hooks, E.: A green vehicle routing problem. Transportation Research Part E: Logistics and Transportation Review 48(1), 100–114 (2012)
10. Xiong, Y., Gan, J., An, B., Miao, C., Bazzan, A.L.: Optimal electric vehicle charging station placement. In: IJCAI. pp. 2662–2668 (2015)
11. Miller, B.L., Goldberg, D.E., et al.: Genetic algorithms, tournament selection, and the effects of noise. Complex systems 9(3), 193–212 (1995)
12. Solomon benchmark 100 nodes. https://www.sintef.no/projectweb/top/vrptw/solomon-benchmark/100-customers/, accessed: 2018-06-15