Routing and Charge Planning Strategies for Ridesharing EV Fleets

Ashutosh Singh and Arobinda Gupta
Dept. of Computer Science & Engineering, Indian Institute of Technology Kharagpur, WB-721302, India

Keywords: Electric Vehicle, Ridesharing Fleet, Request Assignment, Charge Scheduling.

Abstract: Ridesharing systems have become an important part of urban transportation. At the same time, electric vehicle (EV) adoption is also growing at a fast pace as an eco-friendly and sustainable transportation option. To operate a ridesharing system with EV fleets, scheduling an EV fleet to serve passenger requests requires consideration of both the requests, and the available charge and potential future charge requirements of the EVs. In this paper, we address the problem of scheduling EVs by a ridesharing operator, and propose four algorithms that schedule passenger requests while taking into consideration charging requirements of the EVs. Detailed simulation results are presented on a real world data set to show that the algorithms perform well.

1 INTRODUCTION

Ridesharing systems have become an important part of urban transportation, providing on-demand, convenient, and accessible transportation services to passengers, reshaping the way people move within cities. Within this evolving landscape, Electric Vehicle (EV) fleets can play a pivotal role offering a sustainable and eco-friendly solution to meet the demands of ridesharing. An EV fleet is a collection of electric vehicles that are owned, operated, or managed by a single central entity such as a ridesharing service operator.

A ridesharing operator receives passenger requests for rides and schedules vehicles under its control based on different criteria/constraints. In comparison to a fleet of non-EV vehicles, scheduling an EV fleet should consider both passenger requests, and the current available charge and potential future charge requirements of the EVs. As the operator controls all the EVs, it can have complete knowledge about the EVs at any point in time, which includes their location, current charge levels etc. This can allow the operator to plan routes and manage charging schedules of the EVs more efficiently, leading to lower operating costs. The focus of this paper is addressing the problem of scheduling passenger requests by a ridesharing operator with EV fleet while considering the charging needs of the EVs under different constraints.

The problem addressed can be divided into two parts, the assignment of EVs to passenger requests, and scheduling the charging of EVs at appropriate charging stations at appropriate times. The first part of the problem closely resembles the Vehicle Routing Problem (VRP), a well-studied optimization problem in logistics and several works have addressed different variants of the problem. Mor et al. (Mor and Speranza, 2020) provide a comprehensive survey of the existing works on the vehicle routing problem. Similarly, the fleet charging problem has also been investigated, Ma et al. (Ma and Fang, 2022) provide a survey of the recent developments in this area. However, there has been very little work at the intersection of these two problems. Solving the EV fleet scheduling problem with passenger requests needs to consider both – the aspect of vehicle assignment and routing, and the aspect of charge scheduling. The few works that have addressed both these issues simultaneously (Lu et al., 2012; Chen et al., 2018; Zalesak and Samaranayake, 2021; Yu et al., 2021) mostly formulate the problem as a mixed-integer linear programming (MILP) problem and use optimization solvers to solve the problem, or use a reinforcement learning approach. These approaches have an exponential time complexity in terms of the problem size, and only work when the problem size (number of EVs, requests, etc.) is small, thus making these approaches non-scalable. Also, most of these works solve the offline version of the problem, and have explored a limited set of objective functions.

In this paper, we address the problem of assigning EVs to passenger requests while taking into consideration the charge availability and charging needs of the EVs. We propose an entirely algorithmic approach to the two parts of the online version of the

Singh, A. and Gupta, A.
Routing and Charge Planning Strategies for Ridesharing EV Fleets.
DOI: 10.5220/0013282100003941

In *Proceedings of the 11th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2025)*, pages 76-87
ISBN: 978-989-758-745-0; ISSN: 2184-495X

problem, exploring new objective functions. In particular, we propose two strategies each for assigning passenger requests to EVs and for scheduling EVs for charging, and consequently, four algorithms using their combination for the overall problem addressed that tries to maximize the number of requests served. Detailed simulation results are presented on a real-world dataset to show that the algorithms perform quite well.

The rest of the paper is organized as follows. Section 2 provides a brief overview of related works in the area. Section 3 presents the formal definition of the problem. Section 4 presents the proposed algorithms. Detailed simulation results are shown in Section 5. Finally, Section 6 concludes the paper.

2 RELATED WORKS

The problem of charging EV fleets have been extensively investigated, both in the area of planning the charging infrastructure (the number and location of charging stations, size of EV fleet, charging station equipment, etc.) (Häll et al., 2018; Zhang et al., 2020; Schiffer and Walther, 2018; Guo et al., 2021; Shehadeh et al., 2021; Wu et al., 2021), location-routing optimization (Schiffer and Walther, 2017; Hua et al., 2019; Stumpe et al., 2021; Ma and Xie, 2021), and operational planning such as making decisions on vehicle routes, charging time and place, amount of charge etc. (Chen et al., 2016; Zalesak and Samaranayake, 2021; Wang et al., 2018; Lin et al., 2021; Shi et al., 2020; Guo and Xu, 2022; Lin et al., 2018; Kullman et al., 2022).

In the area of EV fleet charging while serving passenger requests, which is the focus of this paper, Lu et al. (Lu et al., 2012) introduce a dispatching policy designed to optimize electric taxi operations by factoring in taxi demand, the state of charge of electric taxis, and the presence of battery charging/switching stations. The primary goal is to minimize recharging waiting times, ultimately increasing the number of working hours for taxi drivers. Chen et al. (Chen et al., 2018) develop a mathematical model to address the optimal routing and charging of EV fleets within a road network, considering factors such as EV charging rates, charging costs, and state-of-charge requirements. The primary objective is to minimize a weighted combination of route distances, travel times, and charging expenses while ensuring that all passenger requests are met. Zalesak et al. (Zalesak and Samaranayake, 2021) focus on an online electric vehicle ride-sharing system, where real-time customer requests arrive with specific entry times, origins, and destinations. The primary objective was to minimize the cost of accommodated trips while penalizing unserved requests. Yu et al. (Yu et al., 2021) tackle the dynamic optimization problem of maximizing total profit in a vehicle dispatching system by considering various factors such as customer requests, EV charging rates, and penalties for dispatching time and delays. The primary objective is to determine optimal vehicle dispatching, relocation, and recharging decisions to maximize revenue while minimizing penalties. As observed earlier, these works are not scalable to large number of EVs and requests, and most of these works solve the offline version of the problem.

3 PROBLEM FORMULATION

We consider a city with a set of fixed charging stations located at specific locations in the city, where a ridesharing service operator operates a fleet of EVs to service passenger requests. The total duration of operation (for example, from 6 am to 8 pm in a day etc.) is broken up into $\mathbb{T}$ time instants. The road network in the city is modeled as a directed graph $G = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ denotes the set of vertices or nodes, and $\mathcal{A}$ is the set of edges (roads between nodes). The nodes can be charging stations, pickup or dropoff locations, or other geographical locations of relevance. The distance between two nodes i and j is denoted by $d(i, j)$. Note that due to the nature of roads, it may happen that $d(i, j) \neq d(j, i)$.

Let $\mathcal{V}$ denote the set of all EVs belonging to the fleet. All EVs are assumed to be identical, and are centrally controlled by the ridesharing operator. Each EV can serve only one passenger request at a time. An EV $v \in \mathcal{V}$ can be represented as a tuple $\langle s_v, loc_{v,t}, soc_{v,t}, u, b\rangle$, where $s_v \in \mathcal{N}$ denotes the start location of the EV, $loc_{v,t}$ denotes the location of the EV v at time t, $soc_{v,t}$ denotes the state of charge (SOC) of the EV v at time t (battery charge remaining as a percentage of the total battery capacity), u denotes the speed of the EV (assumed to be constant when the EV moves), and b denotes the charge consumption per unit distance (assumed to be constant). Thus the time taken to travel from node i to j is $t_{i,j} = d(i, j)/u$. Also, the battery consumption while travelling from node i to j is given by $\beta_{i,j} = b \cdot d(i, j)$. The battery capacity of all vehicles is denoted by Q (in kWh).

Let $\mathcal{S}$ be the set of all fixed charging stations (FCS). A charging station $s \in \mathcal{S}$ can be represented as a tuple $\langle loc_s, c_s, \alpha_s, q_{s,t}\rangle$, where $loc_s \in \mathcal{N}$ denotes the location of the charging station s, c_s denotes the capacity of the charging station s (the maximum number of vehicles that can be charged at the charging station at the same time), α_s denotes the charging rate of the charging station, and $q_{s,t}$ denotes the number of vehicles in the queue for charging at the charging station s at time t. Every charging station follows the *First Come, First Serve* (FCFS) policy for servicing charging requests.

Let $\mathcal{R}$ be the set of all passenger requests that are received over the time period under consideration. Each request $r \in \mathcal{R}$ can be represented as a tuple $\langle e_r, pick_r, drop_r \rangle$, where e_r denotes the time the request is raised, $pick_r \in \mathcal{N}$ denotes the pickup location, and $drop_r \in \mathcal{N}$ denotes the drop-off location. The time when an EV arrives at the pickup location for request r is denoted as a_r. We also impose a QoS constraint on serving passenger requests in the form of a maximum waiting time parameter, t_w^{max}, within which a passenger needs to be picked up after making a request. Note that given an arbitrary set of passenger requests and a number of EVs, it may not be possible to serve all requests. If it is not possible to assign a vehicle to a request satisfying this QoS requirement, then that request is rejected.

The output variables for the problem can be represented by the following matrices, which cover the state of all EVs over all time instants.

1. $\mathcal{V} \times \mathbb{T}$ matrix *Avail*, where $Avail[v,t] = 1$, if vehicle v is available for serving a passenger request at time t, 0 otherwise.
2. $\mathcal{V} \times \mathbb{T}$ matrix *MovingToFCS*, where $MovingToFCS[v,t] = 1$, if vehicle v is moving towards a charging station at time t, 0 otherwise.
3. $\mathcal{V} \times \mathbb{T}$ matrix *WaitFCS*, where $WaitFCS[v,t] = 1$, if vehicle v is waiting for its turn, in the queue, at a charging station at time t, 0 otherwise.
4. $\mathcal{V} \times \mathbb{T}$ matrix *MovingToPickup*, where $MovingToPickup[v,t] = 1$, if vehicle v is moving towards a pickup location of a passenger at time t, 0 otherwise.
5. $\mathcal{V} \times \mathcal{S} \times \mathbb{T}$ matrix *Charging*, where $Charging[v,s,t] = 1$, if vehicle v is being charged at charging station s at time t, 0 otherwise.
6. $\mathcal{V} \times \mathcal{R} \times \mathbb{N}$ matrix *Req*, where $Req[v,r,t] = 1$, if vehicle v is assigned to passenger request r at time t, 0 otherwise. The time that a vehicle v is assigned to a request r includes all time instants starting from the time the decision of assigning v to r is made till the passenger is dropped off at the dropoff locaton of r; thus it includes the time when v is moving towards the pickup location of the r, and the actual journey time from the pickup location to the dropoff location.

The primary objective of the problem is to maximize the number of passenger requests served. Let $I_{v,r,t}$ denote an indicator variable that takes a value 1 if $Req[v,r,t] = 1$ and $Req[v,r,t-1] = 0$, 0 otherwise. The number of requests served is then given by

$$Num\_Served = \sum_{v \in \mathcal{V}.r \in \mathcal{R},t \in \mathcal{T}} I_{v,r,t}$$

Hence, the objective is to maximize *Num_Served* subject to the following constraints.

1. If a passenger request is served, it must be served by exactly one EV.
   $\forall r \in \mathcal{R}, \forall t_1,t_2 \in \mathbb{T}, \forall v_1,v_2 \in \mathcal{V}, (Req[v_1,r,t_1] = Req[v_2,r,t_2] = 1) \implies (v_1 = v_2)$
2. A request can be assigned to a vehicle only if it is free.
   $\forall r \in \mathcal{R}, \forall v \in \mathcal{V}, \forall t \in \mathbb{T}, (Req[v,r,t] = 1 \wedge Req[v,r,t-1] = 0) \implies (Avail[v,t-1] = 1)$
3. Any vehicle can serve only one request at a time.
   $\forall v \in \mathcal{V}, \forall r_1,r_2 \in \mathcal{R}, \forall t \in \mathbb{T}, (Req[v,r_1,t] = Req[v,r_2,t] = 1) \implies (r_1 \neq r_2)$
4. Once a request is assigned to a vehicle, it stays assigned to the same vehicle till the vehicle reaches the request's dropoff location.
   $\forall r \in \mathcal{R}, (\exists v \in \mathcal{V}, t \in \mathbb{T}, Req[v,r,t] = 1 \wedge Req[v,r,t-1] = 0) \implies ((\forall t' \in [t,t_d], Req[v,r,t'] = 1)$ and $Req[v,r,t_d+1] = 0)$ where t_d is the dropoff time of the corresponding request.
5. If a request r is accepted, then the passenger should not have to wait for more than t_w^{max} time, i.e., the pickup location should be visited by the vehicle before $(e_r + t_w^{max})$.
   $\forall r \in \mathcal{R}, (\exists v \in \mathcal{V}\ \exists t \in \mathbb{T}, (Req[v,r,t] = 1 \wedge Req[v,r,t-1] = 0) \implies (\exists t' \in [t, e_r + t_w^{max}], loc_{v,t'} = pick_r)$.
6. A request is accepted by a vehicle only if it has enough charge left to reach the nearest charging station after dropping the passenger.
   $\forall v \in \mathcal{V}, \forall r \in \mathcal{R}, (\exists t \in T, (Req[v,r,t] = 1 \wedge Req[v,r,t-1] = 0) \implies (soc_{v,t} \geq b \times (d(loc_{v,t}, pick_r) + d(pick_r, drop_r) + d(drop_r, loc_{s_{min}})))$, where $s_{min} = \arg\min_{s \in \mathcal{S}} d(drop_r, loc_s)$ (the charging station nearest to the dropoff location).
7. At any time instant, an EV can be in exactly one of the following states: available (idle), moving to an FCS, waiting at an FCS, being charged at an

FCS, or assigned to a passenger request.

$$\forall v \in \mathcal{V}, \forall t \in \mathbb{N}, Avail[v,t] + MovingToFCS[v,t] + WaitFCS[v,t] + \sum_{s \in \mathcal{S}} Charging[v,s,t] + \sum_{r \in \mathcal{R}} Req[v,r,t] = 1$$

8. Each charging station can charge a maximum of c_s EVs simultaneously.
$\forall s \in \mathcal{S}, \forall t \in \mathbb{T}, \sum_{v \in \mathcal{V}} Charging[v,s,t] \leq c_s$
9. An EV will not be charged while it is serving a passenger request.
$\forall v \in \mathcal{V}, \forall r \in \mathcal{R}, \forall t \in \mathbb{T}, \forall s \in \mathcal{S}, (Req[v,r,t] \wedge Charging[v,s,t]) = 0$
10. When an EV starts charging at a charging station, it stops charging only when its SOC reaches 100%.
$\forall v \in \mathcal{V}, \forall s \in \mathcal{S}, \forall t \in \mathbb{T}, (Charging[v,s,t] = 1 \wedge Charging[v,s,t-1] = 0) \implies ((\forall t' \in [t,t_d], Charging[v,s,t_d] = 1)$ and $Charging[v,s,t_d+1] = 0)$ where $t_d = t + (100 - soc_{v,t-1}) * Q/\alpha_s$

The constraints on the system can be categorized as passenger request constraints (Constraints 1 to 6), constraints on movement of EVs (Constraint 7), and battery/charging constraints (Constraints 8 to 10).

While maximizing the number of requests served is the primary objective, it is also important to have lower average waiting time of all requests served (indicating how fast an EV arrived at the pickup location after a passenger request is made) and lower average distance travelled by an EV during which it does not serve any passenger requests (a measure of wasted travel).

4 ALGORITHMS

In this section, four heuristic algorithms are presented for assigning EVs to passenger requests and scheduling the EVs for charging. We first present two policies for the assignment of EVs to passenger requests (Sec. 4.1), *Nearest Feasible EV Assignment* and *Dropoff Demand Based EV Assignment*. We then present two policies for charge scheduling of the EVs (Sec 4.2): *Waiting Time Based Charging Policy* and *SOC Comparison Based Charging Policy*. Based on these policies, the following four algorithms for assigning EVs to passenger requests and scheduling the EVs for charging are proposed, simply by taking all combination of the policies.

- **[C1: Nearest, Waiting Time]** - Nearest Feasible EV Assignment, Waiting Time Based Charging Policy.
- **[C2: Nearest, SOC Comparison]** - Nearest Feasible EV Assignment , SOC Comparison Based Charging Policy.
- **[C3: Dropoff Demand Based, Waiting Time]** - Dropoff Demand Based EV Assignment, Waiting Time Based Charging Policy.
- **[C4: Dropoff Demand Based, SOC Comparison]** - Dropoff Demand Based EV Assignment, SOC Comparison Based Charging Policy.

Sec 4.3 also proposes an optimization based on routing idle EVs to specific locations in advance that may prove helpful in some specific scenarios. This optimization can be used with any of the above mentioned four algorithms.

4.1 Assignment of EVs to Passenger Requests

Passenger requests arrive in the system in an online manner. For each request received, a *feasible* set of EVs is determined first, and then an EV is chosen from this set for serving the request based on some policy. An EV is said to be feasible for a request if all of the following conditions are satisfied:

- The EV is currently available, i.e., it is not assigned to any other request and has not been dispatched for charging.
- The time taken for the EV to reach the pickup location of the request is less than or equal to the maximum waiting time of the request.
- After dropping the passenger at the dropoff location of the request, the EV would have sufficient charge left to reach the nearest charging station.

After determining the set of feasible EVs for a request, we consider two alternative policies to assign an EV from the set of feasible EVs to the request.

Nearest Feasible EV Assignment. In this policy, the EV in the feasible set that is closest to the pickup location of the request is assigned to serve the request. The intuition behind this is that it minimizes the wait time for the passenger, and the EV can complete this trip faster (compared to any other EV), and hence can become ready again for serving further requests sooner.

Dropoff Demand Based EV Assignment. In this policy, an EV is assigned from the feasible set based on an estimate of the future demand of

EVs at the drop location. If the estimated future demand is deemed to be high at the drop location, then we assign the feasible EV with the highest SOC to the current request. Otherwise, we assign the feasible EV with the lowest SOC to the request. The intuition behind dispatching the EV with the highest SOC is that if the demand at the drop location is high, then after completing this request, the EV can be assigned another request closer to the drop location. On the other hand, we send the EV with the lowest SOC if the demand is low, so that after completing the current request, the EV can go for charging if needed.

The demand at a location varies with different parameters such as the time of the day, day of the week etc. To estimate the future demand at a location,, we use a simple estimate based on the past request data seen so far. To calculate the demand at a location n, the set of all requests seen so far ($totalReq$) is considered. From all such requests, the number of requests ($numRadiusReq$) with their pickup locations within a fixed radius ($DEMAND\_RADIUS$) of the location n, are counted. The demand at location n is then estimated as the ratio $numRadiusReq/totalReq$. If this ratio is higher than a threshold, the demand is said to be high; otherwise the demand is taken as low. It is important to note that while we have used a simple scheme to estimate the demand, any other method that gives a demand estimate for a place and time using past data can be easily plugged into our proposed algorithms.

4.2 Charge Scheduling Policies

The charge scheduling policy determines when an EV should go for charging. The EV always goes to the closest charging station for charging, and gets charged till its SOC reaches 100%. The following two charging policies are proposed.

4.2.1 Waiting Time Based Charging Policy

In this policy, an EV is dispatched for charging if the following conditions are satisfied:

- The SOC of the EV is below a threshold $HIGH\_SOC$, and at least one of the following two conditions hold:
  - The SOC of the EV is less than a threshold (MIN_SOC).
  - The EV has been idle (not servicing any request) for more than a threshold amount of time ($CHARGE\_MAX\_WAIT$).

The intuition behind this policy is straightforward. An EV with a low charge should go for charging as otherwise it cannot serve any request anyway. In addition, even if an EV has sufficient charge, it can still go for charging if the demand is low (high idle time), thereby utilizing the idle time to get more charge to be able to serve future requests better. The parameter $HIGH\_SOC$ is kept high so that an EV with high charge does not go for charging even if it is idle for a long time.

4.2.2 SOC Comparison Based Charging Policy

In this policy, an EV v is dispatched for charging if the following conditions hold:

- The SOC of v is below a threshold $HIGH\_SOC$, and at least one of the following two conditions hold:
  - The SOC of v is less than a threshold (MIN_SOC).
  - v has been idle (not servicing any request) for more than a threshold amount of time ($CHARGE\_MIN\_WAIT$), and of all EVs within a fixed radius ($CHARGE\_RADIUS$) of the current location of v, at least a fraction ($HIGHER\_SOC\_RATIO$) has SOC higher than v.

The intuition here is similar to the earlier policy, except that it is also ensured that when an EV is sent for charging, there are some EVs in its neighborhood that can serve a probable future request with pickup location around the EV being sent for charging. The wait time parameter in this policy should typically be less than that considered in the previous policy, as the additional condition ensures that there are sufficient number of EVs available for serving requests even if this EV goes to charge after waiting a smaller amount of time.

4.3 Routing Idle EVs

We also propose an additional optimization, which proves beneficial in some scenarios. It can be used with any of the proposed algorithms. Essentially, some idle EVs can be routed to locations where the future demand is anticipated to be high, so that when a passenger request does come, there is a higher chance of finding a free EV close by to service the request. Again, since the future demand cannot be known exactly in advance, it is approximated using the past demand. The optimization consists of the following steps.

- *Finding the locations with the highest predicted demand*: For each location n, a small time window (ROUTING_WINDOW) in the recent past is considered, and all requests that have arrived in this window are considered. Out of these requests, we count the number of requests that could be reached by an EV before the maximum wait time, had an EV been standing at location n. This gives the number of eligible requests in a neighbourhood. This, divided by the current number of EVs within a reachable distance of location n, gives the demand estimate.

- *Finding an idle EV to route to the location with the highest predicted demand*: All EVs that are idle, that have their SOC above a threshold, and the number of available EVs in their neighbourhood (within a radius NEARBY_RADIUS) above a threshold (MIN_NEARBY_EVS) are considered to be candidate EVs for routing. Out of these, the EV closest to the location with the highest predicted demand is chosen and sent to that location. Note that in this case, EVs can move from one location to another without being assigned to a passenger request and even if it is not dispatched for charging.

5 SIMULATION RESULTS

The proposed algorithms are evaluated by simulating them on a real-world dataset. We consider the map of Manhattan in New York. The area considered spans around 41 km^2. Request patterns are obtained from the New York City Taxi Trip Dataset (Donovan and Work, 2016). This dataset contains information about multiple yellow taxi trips on each day from 2010 to 2013. From this dataset, the day of 15 January, 2013, is chosen for sampling requests. Only requests that are made between 7 am and 7 pm are considered, which gives a total of 266704 requests on this day. The dataset gives the pickup and dropoff coordinates (latitude and longitude), the pickup and dropoff times, the time taken to complete the trip, and the trip distance. These pickup and dropoff coordinates serve as nodes on the map. Some random coordinates on the map are also chosen to serve as nodes for placing charging stations and choosing initial position of the EVs. For a pickup-dropoff node pair, the distance between them, and the time it would take to reach one node from the other, is known directly from the dataset. For any other pair of nodes, the distance between them is taken to be the geodesic distance between them (the minimum distance between the nodes on the surface of the earth). The time to travel from one node to the other is calculated by dividing the geodesic distance by the constant speed of the EV.

The values of the parameters related to charging stations and EVs are shown in Table 1. For all EV parameters, we consider values similar to an average real-world EV, for ex. Nissan Leaf (Nissan, 2023).

Table 1: Parameter values for experimental setup.

Charging Stations	
Capacity	1
Charging Rate	40 kW per hour
EVs	
Initial SOC	100%
Battery Capacity	40 kWh
Speed	25 km/h
Range with full charge	240 km
Other Constants	
DEMAND_RADIUS	1 km
DEMAND_THRESHOLD	0.1
HIGH_SOC	80%
MIN_SOC	20%
CHARGE_MAX_WAIT	1 hr
CHARGE_MIN_WAIT	10 min
CHARGE_RADIUS	5 km
HIGHER_SOC_RATIO	0.5

The number of EVs are varied from 5 to 40. EVs have random starting points on the map. The number of charging stations is kept constant at 10, whose locations are randomly chosen on the map of Manhattan. A fixed number of requests are sampled from the entire set of requests on the specified day between 7 am and 7 pm. In particular, the number of requests sampled are taken as 500 and 1000. The maximum wait time is kept fixed at 15 minutes for each request.

Three datasets for evaluation are constructed first from the total number of requests in the original dataset. For the first dataset, requests are sampled uniformly from the total pool of requests. For the second dataset, requests are sampled to ensure a proper mix of various trip distances, and for the third dataset, requests are sampled so that they form a specific pattern. The exact details of how the requests are chosen are mentioned before presenting the results for the corresponding scenarios. As mentioned, for each dataset, two values are used for the number of requests sampled, 500 and 1000. The simulation results are shown on these three datasets separately, which we refer to as Scenario 1, Scenario 2, and Scenario 3. It may also be noted that the results with the optimization of routing idle EVs included is shown for Scenario 3 only, as for the first two scenarios, it does not provide much benefit due to the absence of any specific request/demand pattern. All four algorithms are run on each of the three scenarios. All results reported are the average over 10 runs.

The following metrics are measured to evaluate the performance of the proposed algorithm: the number of requests served, average waiting time per served request, average extra distance travelled per EV, and average time spent charging per EV. The first three metrics have been mentioned earlier to be objectives of interest. We also measure the average time spent charging per EV, as this parameter is important for maintaining EV availability and reducing downtime.

5.1 An Upper Bound for the Number of Requests Served

In order to get an estimate of how well the proposed algorithms perform, we try to calculate a loose upper bound on the number of requests that can be served by all the EVs for comparison. We first calculate the maximum total distance all EVs could travel had they been continuously moving. The trips (requests) are then sorted in non-decreasing order of their trip distance (distance between pickup and dropoff). The requests are served in this order until the maximum total distance is exceeded or all requests have been served. This calculation does not consider the waiting time for a request (basically, the maximum waiting time allowed is taken as infinite), ignores the actual ordering of the requests, and also does not consider the distances involved in going to charging stations. However, it takes into account an important component of the total travel distance – moving from the dropoff location of the last request to the pickup location of the next request. This is approximated using the median dropoff-pickup distance of all dropoff-pickup pairs. Experimentally, too, the actual value comes to be very close to this. Since the requests are considered in non-decreasing order of trip distance, we call this the Shortest Trip First (STF) bound.

To see why the STF bound is an upper bound for the number of requests served, we note that the bound firstly removes all restrictions on serving a request, such as the restriction on maximum waiting time, so all requests are eligible to be served. Secondly, the actual request arrival times are ignored, thereby eliminating the times an EV has to wait for requests to arrive, and they can continuously serve requests. Finally, the idea that serving requests in increasing order of their trip times is always at least as good as any other ordering of the requests is fairly common in the realm of greedy algorithms, and can be proved using a simple exchange argument.

In all of the three scenarios next, in the plot for the percentage of requests served, we also show the STF bound value in gray color.

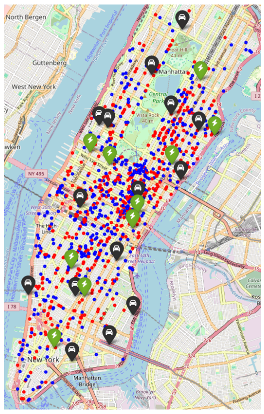

Figure 1: Request Pattern for Scenario 1.

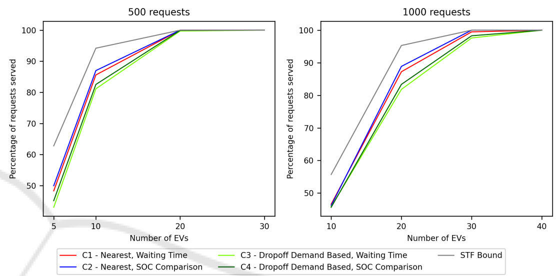

Figure 2: Percentage of requests served.

5.2 Scenario 1 - Uniform Sampling

In this case, 500 and 1000 requests are sampled uniformly from the entire set of requests on the specified day, between 7 a.m. and 7 p.m. Sampling requests uniformly maintains the same proportion of requests according to the distance between the pickup and dropoff location, as in the original dataset. It is seen that around 88% of the requests chosen have a trip distance of less than 5 km, 10% of the requests have a trip distance in the range of 5 to 10 km, and the rest 2% have a trip distance of more than 10 km.

The request pattern is shown on the map in Fig. 1. Red dots represent pickup locations, and blue dots represent dropoff locations. The green icons with the lightning signs denote charging stations, and the black icons with the cars show the initial position of the EVs.

Fig. 2 shows the variation in the percentage of requests served with the number of EVs. It is observed that in both cases, the number of requests served increases with the number of EVs as expected. There is no significant difference in the number of requests served between the four algorithms, though it can be seen that Algorithm C2 has the highest number of re-

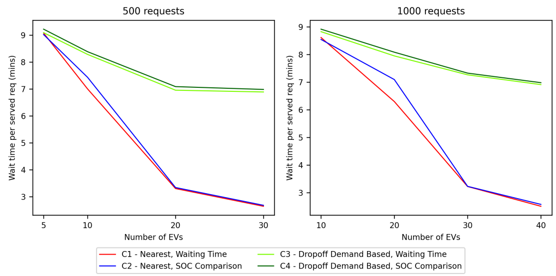

Figure 3: Waiting time per served request.

Figure 4: Extra distance travelled per EV.

quests served, followed by C1, C4, and C3 in that order. This shows that the algorithms with SOC comparison based charging policy perform better, compared to their waiting time based counterparts. This is likely because when there are sufficient number of EVs for serving requests, idle EVs who have other EVs in its neighborhood with higher SOC can go for charging and be better prepared for serving other requests in future. It is also seen that the demand based EV assignment policy performs slightly worse because in this dataset used, the demand at all locations is more or less uniform, and there is no specific pattern in the requests. In comparison to the STF bound, it is seen that with 500 requests, the best algorithm falls short of the STF bound by around 12% with 5 EVs and by around 7% with 10 EVs. With 1000 requests, the best algorithm falls short of the STF bound by around 9% with 10 EVs and by around 7% with 20 EVs. Considering the number of assumptions and relaxations made while calculating the bound, it can be argued that the proposed algorithms perform considerably well.

Fig. 3 shows the variation in the average waiting time per served request with the number of EVs. It is seen that for both the algorithms with the nearest EV assignment strategy, the waiting time decreases rapidly with an increase in the number of EVs. With more EVs available, the average distance between a vehicle and a passenger requesting a ride is likely to decrease, which would lead to a decrease in the waiting time of the passenger. Also, the waiting time of the algorithms with the demand based EV assignment policy is higher, compared to the algorithms with the nearest EV assignment scheme. This arises for the obvious reason that in the nearest EV scheme, the nearest EV is sent for a passenger request, and so, obviously the waiting time would have been more if any other EV was sent.

Fig. 4 shows the variation in the extra distance travelled per EV with the number of EVs. It can be seen that the extra distance travelled by EVs is higher for the algorithms with the demand based EV assignment scheme, compared to the nearest EV assignment scheme. This is understandable as the EV being sent to a pickup location has to travel more in this case, compared to the case if the nearest EV was sent. Also, the extra distance travelled in the case of the algorithms with the SOC comparison based charging policy is slightly higher than those with the waiting time based charging policy. This is because, in the former scheme, the EVs go for charging more frequently as it has less strict conditions for an EV to go for charging. Thus, the distance travelled while reaching the charging station adds to the extra distance travelled in this case.

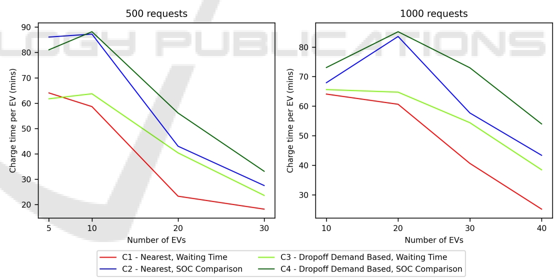

Figure 5: Time spent charging per EV.

Fig. 5 shows the variation in the time spent charging per EV with the number of EVs. We can observe that the average charging time per EV for the waiting time based charging policy decreases with an increase in the number of EVs. With more EVs, the load on each EV decreases, thus the SOC of EVs decreases slowly, and the EVs do not need to charge as frequently as they would have to had the load on each EV been high. However, for the SOC comparison based charging policy, there is a small spike when the number of EVs increases from 5 to 10 in case of 500 requests and from 10 to 20 in case of 1000 requests. This is because now, for each EV, there are possibly more EVs with a higher SOC in its neighborhood,

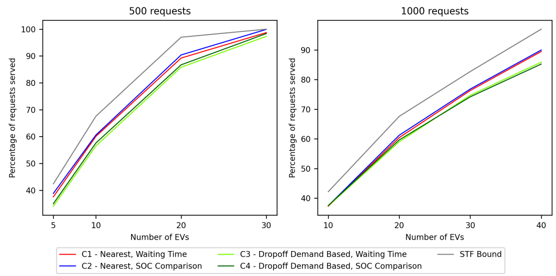

Figure 6: Percentage of requests served.

thus leading to the charging conditions being fulfilled more often, leading to higher charge time. It then decreases again because the number of EVs becomes a more dominant factor compared to the increased charging time. Also, with more requests, the charging time increases, as with more load on the system, each EV serves more requests and loses charge faster, hence needing to charge more frequently. Another important observation is that the time spent charging is higher for algorithms with the SOC comparison based charging policy. With a higher number of EVs, more often than not, EVs have other EVs near them, and the waiting time in the SOC comparison policy is lower than that in the waiting time policy. So, the conditions to decide whether to go for charging become less strict in the SOC comparison based policy, leading to a higher time spent charging by the EVs.

5.3 Scenario 2 - Sampling Based on Trip Distance

In this case, we sample the requests such that 1/3rd of the requests have their trip distance (distance between pickup and dropoff) less than 5 km, another 1/3rd of the requests have their trip distance between 5 and 10 km, and the remaining 1/3rd of the requests have their trip distance greater than 10 km. Thus the trip distances are more widely distributed, and also allows for evaluating the performance when there are more number of longer trips.

Fig. 6 shows the variation in the percentage of requests served with the number of EVs. As expected, the percentage of requests served increases with an increase in the number of EVs for all the algorithms. The algorithms with the SOC comparison based charging policy perform slightly better, as in this case there are more trips with longer distances, causing them to run out of charge and go for charging more often. This makes them better prepared for accepting future requests with long trip distances. For both 500 and 1000 requests, the best algorithm falls short of the STF bound by 3-7% only depending on the number of EVs, showing that the proposed algorithms perform quite well.

Fig. 7, Fig. 8, and Fig. 9 show the variation in average waiting time per served request, average extra distance travelled per EV, and average time spent charging per EV respectively with the number of EVs. The trends observed here are similar to that of the previous scenario, for similar reasons as explained earlier.

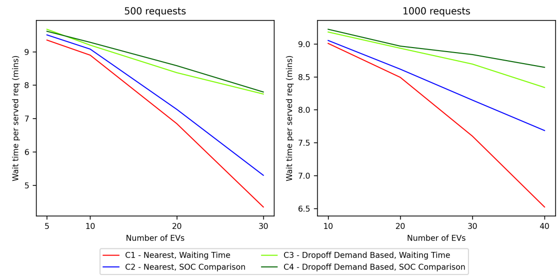

Figure 7: Waiting time per served request.

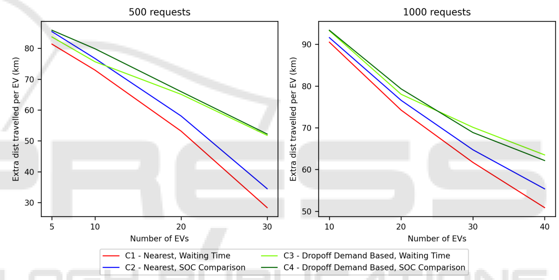

Figure 8: Extra distance travelled per EV.

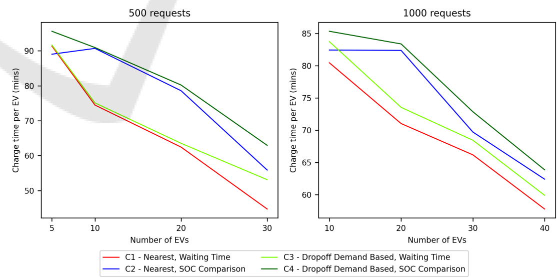

Figure 9: Time spent charging per EV.

5.4 Scenario 3 - Sampling Based on Pickup Location

In this scenario, we choose a 6 km^2 area at the centre of Manhattan, and sample requests such that 70% of requests have their pickup location inside this designated area, and the rest 30% have their pickups outside this designated area. Often, in the real world, it happens that for particular times of a day, there is one area where traffic is concentrated (for example, the downtown area of a city), and most people request rides starting from that area. The request sampling attempts to model this scenario where demand is concentrated more in certain areas.

For this scenario, we also perform additional simulations with the optimization of routing idle EVs enabled. However, due to space constraint, we report results with the optimization enabled only for the best performing strategy (C4 - Dropoff Demand Based and SOC Comparison).

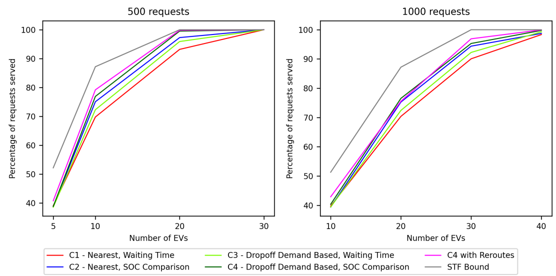

Figure 10: Percentage of requests served.

Fig. 10 shows the variation in the percentage of requests served with the number of EVs. It can be seen that in this case, C4 performs the best, followed by C2, C3, and C1 in that order. We also observe that C4, with the optimization of routing idle EVs to areas of high demand, gives a 2-3% boost, compared to C4 without it. This is expected as the demand is mostly concentrated in one region in this case, and hence routing EVs back to the area of the city where there is higher demand helps in serving more requests. It is seen that the strategies with the demand based EV assignment perform better than their nearest EV counterparts, because there is a very distinct demand pattern in the dataset in this case. Also, the SOC comparison based charging policy performs better as compared to the waiting time based policy as more often than not, after dropping off a passenger, the EV will not get another request near that dropoff location, because the pickups are mostly concentrated in one area. So, they will need to travel some extra distance to go back to the location with more requests to serve the next request. Since with the SOC comparison based policy, EVs go for charging more often, fewer requests are rejected because of the EVs having insufficient SOC to reach a pickup location that is far, thus leading to more requests being served. With 500 and 1000 requests, the best algorithm falls short of the STF bound by 11-14% and 5-11% respectively, depending on the number of EVs. This again shows that the proposed algorithms perform quite well in this scenario also.

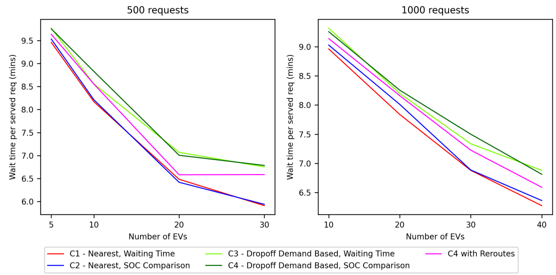

Figure 11: Waiting time per served request.

Fig. 11 shows the variation in the waiting time per served request with the number of EVs. It is again observed that the waiting time decreases rapidly with an increase in the number of EVs. Also, comparing between C4 with and without the optimization enabled, we see that the average waiting time for served requests decreases on routing the idle EVs. Routing idle EVs to areas with high demand allows requests to find an EV both fast and close to the pickup location, thus leading to a reduction in the waiting time.

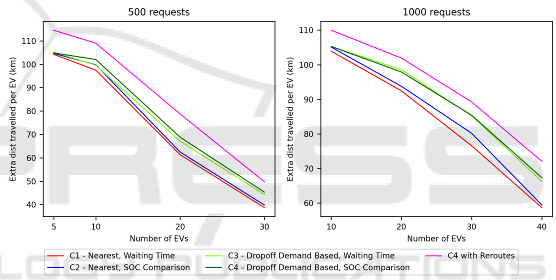

Figure 12: Extra distance travelled per EV.

Fig. 12 shows the variation in the extra distance travelled per EV with the number of EVs. Similar to the previous two scenarios, the extra distance travelled by EVs is higher for the algorithms with the demand based EV assignment scheme, compared to the nearest EV assignment scheme, as EVs have to travel more to reach the pickup location in this case. Also, the extra distance travelled by EVs increases on routing the idle EVs. This is understandable as now the EVs are moving from their location to locations with higher demand, thus contributing to the extra distance travelled.

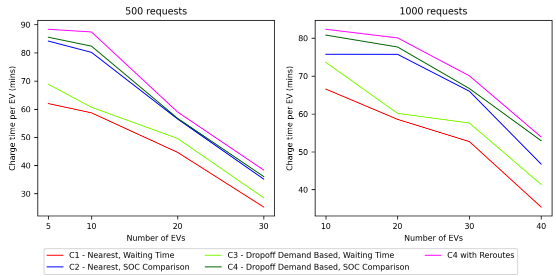

Figure 13: Time spent charging per EV.

Fig. 13 shows the variation in the time spent charging per EV with the number of EVs. Again, it can be seen that the average charging time per EV decreases with an increase in the number of EVs as expected. Also, time spent charging is higher when we route the idle EVs because now they have to travel more, leading to them running out of charge faster and more frequently, thus leading to more frequent charging.

6 CONCLUSION

In this paper, we have addressed the problem of management of an EV fleet for ridesharing to satisfy passenger requests. We presented a set of four algorithms for the problem and presented detailed simulation results on a real world dataset to show that the algorithms perform well. The work can be further extended by considering other objective functions such as operator profit, passenger's ride cost etc., and the use of a set of mobile charging stations owned by the fleet operator for EV charging to reduce operator cost.

REFERENCES

Chen, T., Zhang, B., Pourbabak, H., Kavousi-Fard, A., and Su, W. (2018). Optimal routing and charging of an electric vehicle fleet for high-efficiency dynamic transit systems. *IEEE Transactions on Smart Grid*, 9(4):3563–3572.

Chen, T. D., Kockelman, K. M., and Hanna, J. P. (2016). Operations of a shared, autonomous, electric vehicle fleet: Implications of vehicle & charging infrastructure decisions. *Transportation Research Part A: Policy and Practice*, 94:243–254.

Donovan, B. and Work, D. (2016). New york city taxi trip data (2010-2013). https://doi.org/10.13012/J8PN93H8.

Guo, G. and Xu, Y. (2022). A deep reinforcement learning approach to ride-sharing vehicle dispatching in autonomous mobility-on-demand systems. *IEEE Intelligent Transportation Systems Magazine*, 14(1):128–140.

Guo, Z., Hao, M., Yu, B., and Yao, B. (2021). Robust minimum fleet problem for autonomous and human-driven vehicles in on-demand ride services considering mixed operation zones. *Transportation Research Part C: Emerging Technologies*, 132:103390.

Hua, Y., Zhao, D., Wang, X., and Li, X. (2019). Joint infrastructure planning and fleet management for one-way electric car sharing under time-varying uncertain demand. *Transportation Research Part B: Methodological*, 128:185–206.

Häll, C., Ceder, A., Ekström, J., and Quttineh, N.-H. (2018). Adjustments of public transit operations planning process for the use of electric buses. *Journal of Intelligent Transportation Systems*, 23:1–15.

Kullman, N. D., Cousineau, M., Goodson, J. C., and Mendoza, J. E. (2022). Dynamic ride-hailing with electric vehicles. *Transportation Science*, 56(3):775–794.

Lin, B., Ghaddar, B., and Nathwani, J. (2021). Electric vehicle routing with charging/discharging under time-variant electricity prices. *Transportation Research Part C: Emerging Technologies*, 130:103285.

Lin, K., Zhao, R., Xu, Z., and Zhou, J. (2018). Efficient large-scale fleet management via multi-agent deep reinforcement learning. In *24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, page 1774–1783.

Lu, J.-L., Yeh, M.-Y., Hsu, Y.-C., Yang, S.-N., Gan, C.-H., and Chen, M.-S. (2012). Operating electric taxi fleets: A new dispatching strategy with charging plans. In *2012 IEEE International Electric Vehicle Conference*, pages 1–8.

Ma, T.-Y. and Fang, Y. (2022). Survey of charging management and infrastructure planning for electrified demand-responsive transport systems: Methodologies and recent developments. *European Transport Research Review*, 14.

Ma, T.-Y. and Xie, S. (2021). Optimal fast charging station locations for electric ridesharing with vehicle-charging station assignment. *Transportation Research Part D: Transport and Environment*, 90:102682.

Mor, A. and Speranza, M. (2020). Vehicle routing problems over time: a survey. *4OR*, 18.

Nissan (2023). Nissan leaf features. https://www.nissanusa.com/vehicles/electric-cars/leaf/features.html.

Schiffer, M. and Walther, G. (2017). The electric location routing problem with time windows and partial recharging. *European Journal of Operational Research*, 260(3):995–1013.

Schiffer, M. and Walther, G. (2018). Strategic planning of electric logistics fleet networks: A robust location-routing approach. *Omega*, 80:31–42.

Shehadeh, K. S., Wang, H., and Zhang, P. (2021). Fleet sizing and allocation for on-demand last-mile transportation systems. *Transportation Research Part C: Emerging Technologies*, 132:103387.

Shi, J., Gao, Y., Wang, W., Yu, N., and Ioannou, P. A. (2020). Operating electric vehicle fleet for ride-hailing services with reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, 21(11):4822–4834.

Stumpe, M., Rößler, D., Schryen, G., and Kliewer, N. (2021). Study on sensitivity of electric bus systems under simultaneous optimization of charging infrastructure and vehicle schedules. *EURO Journal on Transportation and Logistics*, 10:100049.

Wang, Y., Bi, J., Guan, W., and Zhao, X. (2018). Optimising route choices for the travelling and charging of battery electric vehicles by considering multiple objectives. *Transportation Research Part D: Transport and Environment*, 64:246–261.

Wu, X., Feng, Q., Bai, C., Lai, C. S., Jia, Y., and Lai, L. L. (2021). A novel fast-charging stations locational planning model for electric bus transit system. *Energy*, 224:120106.

Yu, G., Liu, A., Zhang, J., and Sun, H. (2021). Optimal operations planning of electric autonomous vehicles via asynchronous learning in ride-hailing systems. *Omega*, 103:102448.

Zalesak, M. and Samaranayake, S. (2021). Real time operation of high-capacity electric vehicle ridesharing fleets. *Transportation Research Part C: Emerging Technologies*, 133:103413.

Zhang, H., Sheppard, C. J. R., Lipman, T. E., and Moura, S. J. (2020). Joint fleet sizing and charging system planning for autonomous electric vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 21(11):4725–4738.