Online Energy-optimal Routing for Electric Vehicles with Combinatorial Multi-arm Semi-Bandit

Xiaowei Chen[1], Jiawei Xue[1], Xinwu Qian[1], Juan Suarez[1], Satish V. Ukkusuri[1,†]

***Abstract*— We have seen a rapid growth in the adoption of electric vehicle in today's commercial mobility service market, from carrying passengers to the delivery of goods. One cardinal issue concerning the operation of EVs is the optimal routing of EV fleets with limited battery capacity. In this study, we investigate the energy-optimal online routing problem for the fleet of EVs, which focuses on identifying real-time minimum electricity consumption paths (MECP) for multiple OD pairs with limited information. We develop a multi-OD combinatorial multi-arm semi-bandit model (MCMAB) that uses the fleet of EVs as sensors in the transportation network and promotes the utilization of common information shared by different OD pairs. We further enrich the model with the path elimination policy to obtain MECP of high confidence while significantly reducing the number of learning iterations and the number of explorations needed. We demonstrate the effectiveness of the MCMAB and the efficiency of the path elimination policy with comprehensive numerical experiments in Manhattan, NYC. The results show that the proposed online routing algorithms can achieve near-optimal MECPs efficiently, and the quality of the solutions is significantly better than using the shortest travel time paths as approximate MECPs.**

I. Introduction

The electric vehicle (EV) has undergone tremendous growth in the past few years, and it has been shown that electrified urban transportation systems can effectively decrease greenhouse gas emissions and reduce travel costs [1]. To promote wider adoption of EVs, key challenges such as improving battery performance, eco-routing and optimal deployment of charging facilities have attracted significant attention and existing studies have exclusively focused on commuting vehicles [2], [3], [4], [5], [6]. Yet there is a missing linkage between the use of EVs and the large-scale deployment of vehicles for commercial services such as urban on-demand mobility service and urban logistics. For commercial uses, one bottleneck is to efficiently route the EV fleet in real-time, which will significantly affect the time in service and hence determine the operation efficiency. This motivates us to investigate the online energy-optimal routing problem for a fleet of EVs, which has significant implications for applications such as routing electric buses in urban traffic and routing electric robots for food delivery.

There are three major challenges for the energy-optimal EV routing problem. First, the traffic condition is largely stochastic and so is the battery consumption of EVs. Two EVs traversing the same link at the same time are likely to consume different amounts of electricity due to stop-and-go traffic and carrying capacity. Second, the EV operators may have access to only limited local information, e.g. the historical energy consumption of their own fleet which covers part of the network. While there is an exponential number of paths in an urban setting, this issue makes it difficult to identify the energy-optimal path with partial observations effectively. On the other hand, different routes for the same OD and even for different ODs may share common segments and this observation provides the opportunity to mitigate the information limitation. This leads to the third challenge on how to effectively use the data from the EV fleet to identify energy-optimal routing policies.

Previous studies suggested several solutions to the EV routing problem. One possible approach is to use the shortest path algorithm such as the Dijkstra algorithm [7] and K-shortest path algorithm [8], and represent link cost with energy consumption or approximate energy consumption with link travel time. However, these algorithms are not suitable for EV routing due to the possible existence of cycles (EV may incur negative link energy consumption) and travel time may not be a good estimator of the actual electricity usage. On the other hand, some researchers also proposed the stochastic routing problem and considered that the link travel time follows certain distributions [9], [10]. However, it is not realistic to accurately describe the actual electricity consumption with only limited information from the EV fleet, letting alone the non-stationary traffic condition that varies dramatically over time.

We notice that the EV routing problem can be analogous to the multi-arm bandit problem (MAB), where the decision-makers (EV operator) need to choose the best action (energy-optimal route) among a set of candidate actions (set of candidate routes) with unknown reward (stochastic energy consumption). And the goal is to find the most profitable action through efficient explorations with minimum data [11]. For EV routing problem, each EV reveals an observation of the unknown rewards after completing a trip, and the aggregate information from the fleet of EVs which make multiple trips among different ODs constitutes an online routing problem to determine the optimal path with a balance between exploring new actions and exploiting previous actions. Since each route action consists of the choice of several links, and that different routes may share common link segments, the offline EV routing problem is a combinatorial optimization problem and its online version falls under the category of the combinatorial multi-arm bandit problem (CMAB) [12]. Several methods have been proposed to solve the MAB, such

[1] Lyles School of Civil Engineering, Purdue University, West Lafayette, IN, USA.

[†] Corresponding author, sukkusur@purdue.edu

as ϵ-Greedy [13], Upper Confidence Bounds (UCB) [14] and Thompson Sampling [15], and these methods were later extended to address the CMAB problem [16], [17], [18]. These algorithms, together with the analytical results on upper bound and lower bound of the regret (the reward gap between selected action and global optimal action), build the foundation for our EV routing methodology development.

In this paper, we propose to solve the online energy-optimal EV routing problem based on combinatorial multi-arm semi-bandit. Here, semi-bandit means that only the energy-consumption of each link along the traversed path by EVs will be revealed and we assume no knowledge of other unvisited routes [17]. We consider that, during a short period of time, the link-level energy consumption follows an unknown distribution which will be estimated from real-time trip and energy-consumption information from the fleet of EVs. And the fleet of EVs is scheduled to serve a predefined set of OD pairs and the operator needs to identify the energy-optimal route of each OD pair. While there exists a large number of routes across all the ODs, we further devise the route elimination algorithm to guide the exploration of the bandit model and to enhance the sample efficiency in light of the limited information in real-time. To summarize, our proposed model is a multi-OD CMAB problem (MCMAB) and this study has three major contributions:

(1) We develop a MCMAB framework to effectively utilize the shared information among different routes and different OD pairs, which makes the best use of the limited traffic and energy consumption information obtained by the fleet of EVs.
(2) We devise the UCB based solution algorithm for the MCMAB and the corresponding oracle for obtaining the energy-optimal route, which is further embedded with the route elimination algorithm to improve sample efficiency.
(3) We demonstrate the effectiveness of the MCMAB and the route elimination scheme through a real-world case study, where we simulate the energy consumption of a fleet of EVs in the Manhattan network and conduct comprehensive experiments to evaluate the performances of the model.

The rest of the paper is organized as follows. The notations of parameters and variables used in this paper are shown in Table I. Section II discusses the setting of the online energy-optimal EV routing problem setting and presents the multi-arm bandit routing algorithm as well as the action elimination policy. Section III describes the background information of our case study and discusses the major results obtained from the numerical experiments. Finally, we conclude the study with major findings and future directions in section IV.

TABLE I
TABLE OF NOTATION

Parameters	**Descriptions**
$G(V,E)$	The road network where a node in V represents an intersection and an link in E represents a road segment
$u,v;o,d$	Two distinct intersections in $G(V,E)$
$\mathcal{P}(u,v);\mathcal{P}(o,d)$	All simple paths from node u to node v; from node o to node d
p	A path in $\mathcal{P}$
$\|p\|$	The number of links in path p
$od(p)$	The origin and destination of path p
T_{od}	The times that od has appeared
e	A road segment in p
Variables	**Descriptions**
o_t, d_t	The origin and destination node of the trip at time t
$C_e(t)$	The energy consumption for an EV passing road segment e at time t
$C(o,d)$	The energy consumption for an EV traveling from o to d
$r_e(t)$	Reward of link e at time t, which is the opposite number of $C_e(t)$
p^*	The best route with largest expected reward
R	Cumulative regret from time 1 to time $T-1$
$\hat{Q}_T(p)$	The average reward for path p from time 1 to time $T-1$
$\hat{U}_T(p)$	The UCB (upper confidence bound) for path p from time 1 to time $T-1$
p_T^{UCB}	The selected path using the UCB algorithm at time T
$\mathcal{P}_T(o_T,d_T)$	The potential paths from o_T to d_T at the beginning of time T
$m_T(o_T,d_T)$	The medium of all expected reward $\{r_p(T)\}$

II. SEMI-BANDIT ONLINE MULTI-OD ROUTING PROBLEM

A. Single OD Routing Model

Consider the road network $G(V,E)$ where V is the node set and E is the link set. Here, $G(V,E)$ is a primal model where a node represents an intersection and an link represents a road segment. For each pair of nodes in the network $u,v \in V$, $u \neq v$, all simple paths (simple path J refers the path where nodes along the path are all distinct) from node u to node v form the candidate path set $\mathcal{P}(u,v)$. For each path $p \in \mathcal{P}$, $p \subseteq \{0,1\}^{|E|}$. We denote the e-th entry of p to be p_e, $e = 1,2,...,|E|$. $p_e = 1$ if and only if the link e belongs to path p. Each link e is associated with the link travel time and the link electricity consumption, and both of which follow certain unknown distributions. We denote the random variable for the electricity consumption of an EV traversing link e as C_e. At time T, we assume the knowledge of electricity consumption $C_e(t)$ realized in previous time steps, where $t = 1,2,...,T-1$ respectively.

The decision-making process for the online routing problem starts from time step $t = 1$ and proceeds with $t = 2,3,4...$ up to the predefined terminal step. For every time step t, the vehicle with the origin o and destination d selects one certain route $p(t) \in \mathcal{P}(o,d)$ and arrives at the destination d. At the end of the trip, the energy consumption $C_e(t)$ of each link $e \in p(t)$ that traveled by this vehicle is revealed, which indicates that we are able to retrieve the energy consumption information for only the links that we have visited so far. In this case, the energy consumption for each link e along the path $p(t)$ is realized, and this corresponds to a $semi-bandit$ feedback process, which is distinct from a typical $bandit$ feedback process where only end-to-end delay is returned as in the traditional routing problems.

The objective of the routing problem is to minimize the

energy consumption $C(o,d)$ by choosing an appropriate route between o and d, we denote the reward of the bandit as $r(o,d) = -C(o,d)$. For our bandit problem, we set the objective as to maximize the cumulative reward, which translates into minimizing the cumulative regret over time (gap of the reward between the selected route $p(t)$ and the best route p^*, the best route is defined as the oracle minimum energy consumption path for the OD pair):

$$\mathbb{E}R = \mathbb{E}\sum_{t=1}^{T-1}(\sum_{e\in p^*} r_e(t) - \sum_{e\in p(t)} r_e(t)) \tag{1}$$

One trivial approach to solve equation 1 is to consider each path being independent of the other and cast it as a conventional MAB problem where each path is treated as a single arm. Then the UCB algorithm can be directly implemented to sample and update the reward by pulling each arm, and eventually identify the optimal arm to minimize the cumulative regret in the long run. But this can be extremely inefficient as it ignores the existence of common link segments among candidate paths. Instead, we would like to take advantage of the combinatorial nature between links and paths and treat each link as the arm. Then each path p can be considered as a super arm whose reward is the sum of rewards of all links $e \in p$. Each time we pull the super arm, we can update the reward information at the link level, which contributes to broadcasting the information to other paths of common link segments. In this regard, we introduce the combinatorial UCB algorithm (CUCB) [19] to estimates the upper confidence bound $U^T(p)$ of each action p with high probability, which follows:

$$\sum_{e\in p} r_e(T) \leq \hat{Q}^T(p) + \hat{U}^T(p) \tag{2}$$

Here, $\hat{Q}_T(p)$ is the average reward of path p in previous time steps:

$$\hat{Q}_T(p) = \sum_{e\in p} \frac{\sum_{t=1}^{T-1} \mathbb{I}[e \in p(t)] r_e(t)}{\sum_{t=1}^{T-1} \mathbb{I}[e \in p(t)]} \tag{3}$$

where $\mathbb{I}[e \in p(t)]$ is an indicator variable and is equal to 1 if and only if $e \in p(t)$, and 0 otherwise. Using the "Hoeffding's Inequality" [20], we set the confidence region $\hat{U}^T(p)$ to be:

$$\hat{U}_T(p) = \sum_{e\in p} \frac{\sqrt{\frac{3ln(T_{od(p)})}{2\sum_{t=1}^{T-1}\mathbb{I}[e\in p(t)]}}}{|p|} \tag{4}$$

Here, we introduce the path length $|p|$ to balance the difference of paths with large and small number of links. The CUCB algorithm proceeds by iteratively selecting a path following a given oracle algorithm, revealing the electricity consumption on each link, and update the observed average electricity consumption on each link. And we define the oracle algorithm as to select the path of the highest UCB (lowest potential energy consumption) value out of all the candidate paths, which can be written as:

$$p_T^{UCB} = argmax_{p\in\mathcal{P}(o,d)}(\hat{Q}_T(p) + \hat{U}_T(p)) \tag{5}$$

We summarize the pseudo-code for the CUCB algorithm in Algorithm 1.

Algorithm 1 CUCB for the CMAB problem

Require: $G(V,E)$; $o, d, \mathcal{P}(o,d)$; $\{p(t), \{r_e(t)\}_{e\in p(t)}\}, t = 1, 2, ..., T-1$
Ensure: $p(T)$;
1: calculate $\hat{Q}_T(p)$ using equation (3)
2: calculate $\hat{U}_T(p)$ using equation (4)
3: select p_T^{UCB} using equation (5)
4: **return** p_T^{UCB}

Note that p_T^{UCB} belongs to a predefined set $\mathcal{P}(o,d)$ and all paths in $\mathcal{P}(o,d)$ contain a link at most once, so we do not need to deal with the "negative cycle" issue.

B. Multi-OD Routing Model

For the case of routing a fleet of EVs, the operator needs to identify the optimal routes across all the possible OD pairs. Instead of performing CUCB for individual OD independently, there may also exist overlapping links among different ODs, which is similar to the link overlapping case among different paths of each OD pair. This motivates us to extend Algorithm 1 to account for the combinatorial nature of the multi-OD routing problem. At time step t, each candidate od pair is associated with a candidate path set $\mathcal{P}(o_t, d_t)$ and will generate the observations of all links along a chosen path. We can, therefore,introduce a straightforward modification to Algorithm 1 by updating the link information (electricity consumption, visited counts) from all paths of all the OD pairs simultaneously. And the process can be summarized in Algorithm 2.

Algorithm 2 CUCB for multiple ODs (MCMAB)

Require: $G(V,E)$; $\{o_t, d_t, p(t), \{r_e(t)\}_{e\in p(t)}\}, t = 1, 2, ..., T-1$; $\mathcal{P}(o_T, d_T)$ (the potential simple paths from o_T to node d_T);
Ensure: $p(T)$
1: **for** any p in $\mathcal{P}(o_T, d_T)$ **do**
2: calculate $\hat{Q}_T(p)$ using equation (3)
3: calculate $\hat{U}_T(p)$ using equation (4)
4: calculate $r_p(T)$ using $r_p(T) = \hat{Q}_T(p) + \hat{U}_T(p)$
5: **end for**
6: select $p_T^{UCB} = argmax_{p\in\mathcal{P}_T(o_T,d_T)} r_p(T)$
7: **return** p_T^{UCB}

C. Path elimination

While there may exist a large number of candidate paths for each OD pair, only a few of them will have close-to-optimal performances. Since UCB is an algorithm that balances between exploration and exploitation, significant resources will be wasted on exploring bad paths if we maintain a large candidate path set. In such cases, it is more efficient to delete the path with bad performance as the exploration process continues, which helps to speed up the

algorithm convergence and identify better-performed paths. By eliminating sub-optimal actions through the iteration process, it helps to reduce the size of the path sets to be searched in the future iteration and focus on the exploration among the optimal paths.

In this study, we develop the path elimination policy based on the Path Elimination method introduced in [21]. The proposed elimination policy requires the input of a minimum number of actions to keep (δ_1), the elimination probability (α), and the size of each elimination (δ_2). α increases from 0 to 1 over the iteration, which ensures a small chance of elimination in early stages but allows for aggressive elimination with more exploration conducted. The elimination algorithm takes the observed mean path travel time at time T as an input, and removes the worst δ_2 candidate paths from the path set based on the probability α. The path elimination algorithm is summarized in Algorithm 3.

Algorithm 3 Path elimination algorithm

Require: $G(V,E)$; $\{o_t, d_t, p(t), \{r_e(t)\}_{e\in p(t)}\}, t = 1,2,...,T-1$; $\mathcal{P}_T(o_T,d_T)$; δ_1, δ_2.
Ensure: $p(T)$; $\mathcal{P}_{T+1}(o_T,d_T)$

```
for any p in P_T(o_T, d_T) do
    calculate Q̂_T(p) using equation (3)
    calculate Û_T(p) using equation (4)
    calculate r_p(T) using r_p(T) = Q̂_T(p) + Û_T(p)
end for
select p_T^UCB = argmax_{p∈P_T}(o_T, d_T) r_p(T)
P_{T+1}(o_T, d_T) = P_t(o_T, d_T)
generate a random number r in [0,1]
α = e^{a*(T/b−1)}
if |P_t(o_T, d_T)| > δ_1 and r < α then
    Find the δ_2 percentile value m_T(o_T, d_T) of {r_p(T)}_{p∈P_T(o_T,d_T)}.
    for p ∈ P_T(o_T, d_T) do
        if r_p(T) < m_T(o_T, d_T) then
            P_{T+1}(o_T, d_T) = P_{T+1}(o_T, d_T)/{p}
        end if
    end for
end if
return p_T^UCB; P_{T+1}(o_T, d_T)
```

III. Numerical Experiments

A. Experiment setting

We choose Manhattan in New York City as the case study area to evaluate the performances of our models for routing a fleet of EVs across 10 selected OD pairs, as shown in Figure 1. The Manhattan road network is extracted from New York State open data and only link segments whose speed limit is above 35 mph are kept. And we measure the average speed for link energy consumption calculation using speed data from Google Map API and open data from Uber Movement. The processed Manhattan network has 1,455 nodes, 2,475 links, with a mean velocity of 20.4 mph.

The 10 OD pairs are selected with a small chance of possible overlapping link segments. For each OD pair, we use the k-shortest path algorithm to generate the candidate path set based on the path travel time, where we set k=50. We dispatch up to 1000 vehicles on each OD pair, which leads to 10^3 iterations for each OD and MCMAB based algorithms, while 10^4 iterations for performing the CMAB algorithm, to better compare the results of different algorithms, in the figures, we extend the results from 10^3 iterations for multi-OD into 10^4 iterations for single-OD.

In addition to the models discussed in this study, we also introduce several benchmark algorithms for comparison purposes, and are summarized as follows:

1) Shortest path (oracle): return the energy consumption of the path with the shortest travel time at the current step.

2) Shortest path (one step lag): current energy consumption of the shortest travel time path of previous steps.

3) Shortest path (10 steps average): the current energy consumption of the shortest travel time path based on the average performances from $t-10$ to $t-1$.

4) CMAB: choose the path independently for each OD pair using Algorithm 1.

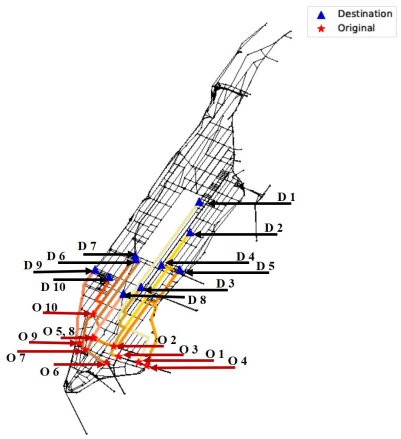

Fig. 1. Manhattan Network

B. Energy calculation

For calculating EV energy consumption, we develop a physical model based on the results in [22]. The model introduces vehicle specific power(VSP) to describe the vehicle dynamics and second-by-second battery usage. In our implementation, we simulate the dynamics of an EV traversing each link based on the speed distribution to generate samples of energy consumption, and we then use a B-spline interpolation to construct the empirical distribution. This method first produces a second-by-second profile of speed and acceleration of an EV on a given path. We then feed this profile into the physical model, which outputs the energy consumption distribution of the EVs. Note that our CUCB algorithms are still valid when the link energy consumption is calculated via other energy models or is directly obtained from the real-time consumption data of EVs.

C. Results

Figure 2 shows the cumulative regret across 6 different methods. The overall regret for MCMAB with path elimination is significantly smaller than others, which supports the fast convergence and the superior performance in finding the optimal path of the method. The contributing factor of this lowest overall energy consumption is its smaller candidate path set. This helps the algorithm to focus on a few paths whose performances are all close to the MECP and the algorithm can focus on further distinguishing the performance differences among the remaining high-quality paths. The results also demonstrate the superior performance of the MCMAB than that of the CMAB, and this highlights the importance of utilizing the combinatorial nature and sharing information among different paths from OD pairs. On the other hand, one may identify MECPs by using the shortest travel time path as an approximation. And our results also demonstrate the potential bias of such approaches, where oftentimes the shortest travel time path does not necessarily be the MECP. Specifically, the oracle shortest path approach yields the best performances in all three shortest path methods, and the resulting cumulative regret is higher than the worst-performing bandit algorithm (CMAB). And its regret at the final step is 175% higher than the best performing MCMAB with path elimination. Moreover, in practice, we do not have access to oracle shortest path information, and the other two feasible shortest path approaches based on historical information may result in another 70% increase in total regret. These results clearly state the importance of an online routing model for the energy-optimal routing problem and prove the superiority of our proposed solutions in terms of both sample efficiency and solution quality.

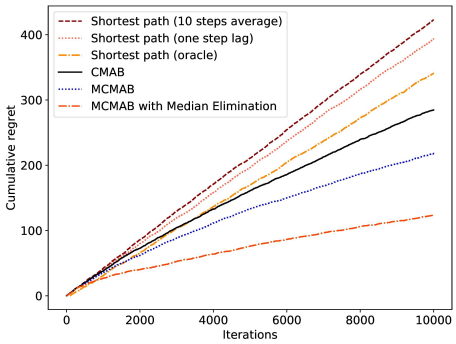

Fig. 2. Cumulative regret

Figure 3 can quantify the gaps of the shortest path (oracle) with the three bandit algorithms. In the first 500 iterations, CMAB is found to perform worse than the oracle shortest due to the excessive explorations spent on the large initial candidate path set. After that, the CMAB starts to differentiate good paths from the bad ones and select paths that are more energy-efficient than the shortest travel time path. On the other hand, the other two bandit algorithms are found to perform consistently better than the oracle shortest path. This again illustrates the importance of information sharing in the MCMAB model. While the two MCMAB algorithms have almost identical performances in the first 200 iterations, the path elimination starts to show effectiveness in the following iterations, and the increasingly large gap suggests the consistent correctness of the path removal operations. At the end of the 10000 iterations, we report that MCMAB with path elimination helps to save a total of 298.9 kWh as compared to the oracle shortest path policy, and the savings for CMAB and MCMAB are 138.2 kWh and 204.8 kWh respectively. And if we average over the final 1000 iterations, MCMAB with path elimination finds path (0.010 kWh) that is 66.7% more efficient than the shortest travel time path (0.033 kWh), representing a saving of 23 kWh per thousand trips. In this regard, the MCMAB algorithm can help to achieve immense energy savings for large-scale EV fleet in real-world applications.

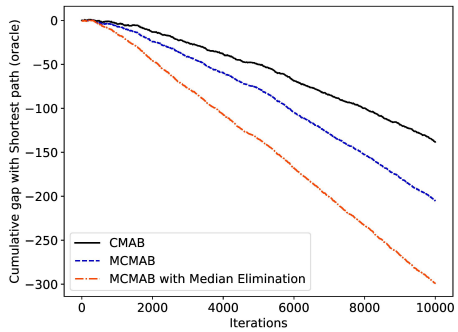

Fig. 3. Gap compared with Shortest path (oracle)

We also visualize the detailed algorithm performances on each OD, and the results can be found in Figure 4. These results are clear evidence that the shortest travel path is unlikely the MECP and not even a good approximation for certain ODs. In some of the cases, the shortest path is found to perform better than the bandit algorithms, nevertheless, the cumulative gap is in general very small (smaller than 10). Moreover, for cases where bandit algorithms are superior, we find the differences between bandit algorithms and shortest paths are much higher. These results state that the bandit algorithms are able to identify consistently good routes across different ODs.

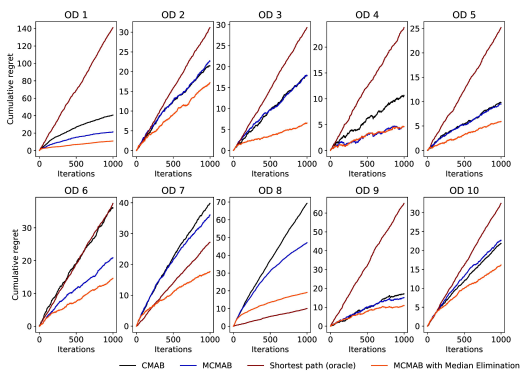

Fig. 4. Per OD cumulative regret

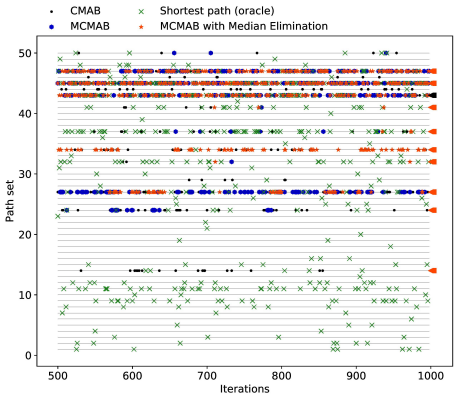

Fig. 5. The distribution of visited paths of the routing algorithms (OD 9)

Finally, in Figure 5, we present the visit distribution of the algorithms on selecting different candidate paths in the last 500 iterations, where each mark denotes one path selection at a iteration. And we mark the top 10 good-performance paths (counted from the MECPs at each iteration) with orange arrows on the right side, and the oracle MECP with black arrow. It is observed that the CMAB visited diverse paths, which is indicative of the ongoing exploration process due to the large candidate path set. On the contrary, both MCMAB based algorithms are found to land their path selections much more frequently within the top 10 paths, and the path elimination process helps the MCMAB to focus almost exclusively on the good-performance paths. For the oracle shortest path, many of its selections are found to lie outside the top 10 paths, and this observation is consistent with the cumulative regret as shown in Figure 4.

IV. Conclusion

In this study, we develop the MCMAB model and the path elimination algorithm to find MECP for the online routing of a commercial EV fleet. Our results show that the proposed algorithms can yield near-optimal MECP with limited information, and the performances are significantly better than the shortest travel time-based approaches. The proposed algorithms may therefore contribute to more efficient EV battery management, and will help EV operators to prepare EVs for large-scale deployment and commercial usage. As for future extensions, we plan to introduce the correlation matrix to capture the link electricity consumption dependencies to improve the learning efficiency, and we will derive theoretical proofs on the upper bound of the long-term expected reward regret for the MCMAB models.

Acknowledgement

This work is partly funded by the U.S. Department of Energy, Office of Energy Efficiency and Renewable Energy, under Award Number DE-EE0008524. The authors are solely responsible for the findings in this paper.

References

[1] Jia Ying Yong, Vigna K Ramachandaramurthy, Kang Miao Tan, and Nadarajah Mithulananthan. A review on the state-of-the-art technologies of electric vehicle, its impacts and prospects. *Renewable and Sustainable Energy Reviews*, 49:365–385, 2015.

[2] Chwei-Sen Wang, Oskar H Stielau, and Grant A Covic. Design considerations for a contactless electric vehicle battery charger. *IEEE Transactions on industrial electronics*, 52(5):1308–1314, 2005.

[3] Fang He, Di Wu, Yafeng Yin, and Yongpei Guan. Optimal deployment of public charging stations for plug-in hybrid electric vehicles. *Transportation Research Part B: Methodological*, 47:87–101, 2013.

[4] Ran Bi, Jiajian Xiao, Dominik Pelzer, David Ciechanowicz, David Eckhoff, and Alois Knoll. A simulation-based heuristic for city-scale electric vehicle charging station placement. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–7. IEEE, 2017.

[5] Xinwu Qian, Jiawei Xue, Stanislav Sobolevsky, Chao Yang, and Satish Ukkusuri. Stationary spatial charging demand distribution for commercial electric vehicles in urban area. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 220–225. IEEE, 2019.

[6] Rui Chen, Xinwu Qian, Lixin Miao, and Satish V Ukkusuri. Optimal charging facility location and capacity for electric vehicles considering route choice and charging time equilibrium. *Computers & Operations Research*, 113:104776, 2020.

[7] Edsger W Dijkstra et al. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[8] Jin Y Yen. Finding the k shortest loopless paths in a network. *management Science*, 17(11):712–716, 1971.

[9] Jeffrey P Kharoufeh and Natarajan Gautam. Deriving link travel-time distributions via stochastic speed processes. *Transportation Science*, 38(1):97–106, 2004.

[10] Zonggen Yi and Peter H Bauer. Optimal stochastic eco-routing solutions for electric vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 19(12):3807–3817, 2018.

[11] Jean-Yves Audibert, Rémi Munos, and Csaba Szepesvári. Exploration–exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410(19):1876–1902, 2009.

[12] Wei Chen, Yajun Wang, and Yang Yuan. Combinatorial multi-armed bandit: General framework and applications. In *International Conference on Machine Learning*, pages 151–159, 2013.

[13] John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In *Advances in neural information processing systems*, pages 817–824, 2008.

[14] Peter Auer and Ronald Ortner. Ucb revisited: Improved regret bounds for the stochastic multi-armed bandit problem. *Periodica Mathematica Hungarica*, 61(1-2):55–65, 2010.

[15] Shipra Agrawal and Navin Goyal. Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on learning theory*, pages 39–1, 2012.

[16] Nicolo Cesa-Bianchi and Gábor Lugosi. Combinatorial bandits. *Journal of Computer and System Sciences*, 78(5):1404–1422, 2012.

[17] Mohammad Sadegh Talebi, Zhenhua Zou, Richard Combes, Alexandre Proutiere, and Mikael Johansson. Stochastic online shortest path routing: The value of feedback. *IEEE Transactions on Automatic Control*, 63(4):915–930, 2017.

[18] Wei Chen, Liwei Wang, Haoyu Zhao, and Kai Zheng. Combinatorial semi-bandit in the non-stationary environment. *arXiv preprint arXiv:2002.03580*, 2020.

[19] Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvari. Tight regret bounds for stochastic combinatorial semi-bandits. In *Artificial Intelligence and Statistics*, pages 535–543, 2015.

[20] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. In *The Collected Works of Wassily Hoeffding*, pages 409–426. Springer, 1994.

[21] Eyal Even-Dar, Shie Mannor, and Yishay Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of machine learning research*, 7(Jun):1079–1105, 2006.

[22] Ricardo Maia, Marco Silva, Rui Araújo, and Urbano Nunes. Electric vehicle simulator for energy consumption studies in electric mobility systems. In *2011 IEEE Forum on Integrated and Sustainable Transportation Systems*, pages 227–232. IEEE, 2011.