



FINAL REPORT

R2E: A Real-time Routing and Recharging Recommendation System for Electric Taxi Drivers

Date of report: November, 2018

Enshu, Wang, PhD, Department of Computer Science and Engineering
Chunming, Qiao, Professor, Department of Computer Science and Engineering
Lu, Su, Assistant Professor, Department of Computer Science and Engineering
Wen Dong, Assistant Professor, Department of Computer Science and Engineering

Prepared by:
Department of Computer Science and Engineering
University at Buffalo, SUNY
Davis Hall, Buffalo, New York, 14260 Address line 3

Prepared for:
Transportation Informatics Tier I University Transportation Center
204 Ketter Hall

University at Buffalo
Buffalo, NY 14260

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle R2E: A Real-time Routing and Recharging Recommendation System for Electric Taxi Drivers		5. Report Date 11.12.2018	
		6. Performing Organization Code	
7. Author(s) Enshu Wang Chunming Qiao Lu Su and Wen Dong		8. Performing Organization Report No.	
9. Performing Organization Name and Address Department of Computer Science and Engineering University at Buffalo, SUNY Davis Hall, #338, Buffalo, New York, 14260		10. Work Unit No. (TRAIS)	
		11. Contract or Grant No. DTRT13-G-UTC48	
12. Sponsoring Agency Name and Address US Department of Transportation Office of the UTC Program, RDT-30 1200 New Jersey Ave., SE Washington, DC 20590		13. Type of Report and Period Covered 09.01.2017-12.17.2018	
		14. Sponsoring Agency Code	
15. Supplementary Notes			
16. Abstract <p>Electric taxi (eTaxi) has been introduced into the public transportation systems to accelerate the greening of industrial firms all over the world. Different from traditional taxis that can refuel in minutes, eTaxis' recharging cycles can be as long as one hour. Currently, most of existing taxi recommendation systems focus on maximizing immediate reward from picking up next passengers. However, in the real world, the reward of a taxi driver is strongly correlated with the effective driving hours, especially for eTaxi drivers who are suffering from long recharging cycles. Therefore, how to maximize total reward of the eTaxi driver and when, where and how long to recharge an eTaxi have already emerged as urgent and crucial problems to be solved for the widely deployment of the eTaxi. To combat these problems, we propose a real-time routing and recharging recommendation system, called R2E, to recommend an entire route with recharging plan to maximize total reward, which considers both immediate reward from picking up next passengers and future reward from the rest of working hours. In this paper, we first formulate the serving and recharging problem of an eTaxi as a Markov Decision Process and leverage MDP to compute future reward in the rest of working hours. Then, based on immediate reward computed from historical data and future reward obtained from computational results of MDP, we propose a maximized reward routing algorithm, called MARA, to provide a fine-grained entire route including both finding next passengers and recharging plan in order to achieve maximum reward. Lastly, we evaluate our R2E on a real-world data set collected from the Shenzhen City in China. The experimental results clearly validate the effectiveness of our proposed R2E system.</p>			
17. Key Words Electric vehicle, big data study, taxi service strategy optimization.		18. Distribution Statement No restrictions. This document is available from the National Technical Information Service, Springfield, VA 22161	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 16	22. Price

R2E: A Real-time Routing and Recharging Recommendation System for Electric Taxi Drivers

Acknowledgements

Include collaborators here

Disclaimer

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the U.S. Department of Transportation's University Transportation Centers Program, in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof.

1 INTRODUCTION

In the past few years, electric vehicles have become increasingly popular, due to the advantages of low-emission, zero-pollution and high-efficiency. To advocate green commuting, several countries such as the United States and China, have already adopted electric taxi (eTaxi) into their public transportation system. However, the inherent long recharging cycle of eTaxis, limited number of charging stations and the improper charging station deployment strategies will lower eTaxis' operational efficiency [11]. Moreover, since the maximum reward is always the primary goal of eTaxi drivers, recharging strategies of eTaxis should be quite different from those of traditional taxis. That is, we need to make our best efforts to reduce loss of reward from the inherent long recharging cycle of eTaxi and consider varying location/time based passenger requests at the same time. Therefore, an intelligent eTaxi recharging recommendation system is urgently needed so as to improve their operational efficiency as well as increase their reward.

Recent efforts on recommendation services for taxi business focus on three ways. First, it is to recommend a sequence of pick-up points for taxi drivers and extract energy-efficient driving patterns [3]. The second aspect is to recommend an entire route in order to maximize probability and reward of finding next passengers [9, 19]. Last but not least, it focuses on the development of the fastest driving route [17] and the most time-efficiency recharging strategy [12] for eTaxis. Most of the existing recommendation systems are based on immediate reward from picking up next passengers. However, in the real world, the reward of taxi drivers is strongly correlated with the effective driving hours, especially for eTaxi drivers who are suffering from long recharging cycle (i.e., one hour). Thus, it's more critical for eTaxi drivers to know when, where and how long they recharge eTaxi can ensure large reward of picking up next passengers and maximize effective driving hours.

To combat these issues, we propose a new routing and recharging recommendation system for eTaxi drivers called *R2E*. The goal of *R2E* is to maximize eTaxi drivers' reward by optimizing the recommended routes with recharging strategy. In another word, *R2E* can help driver to achieve maximum potential reward by providing recommendations on when, where and how long to recharge their eTaxis instead of simply recommending a nearest charging station within minimum driving distance and/or shortest waiting time. Besides, the proposed *R2E* system can also provide a detailed route plan to show eTaxi drivers how to get to the charging station and when to get out to achieve largest potential reward.

To implement this recommendation system *R2E*, we formulate our model by leveraging Markov Decision Process (MDP), which is a general framework for optimizing sequential decision process in the presence of uncertainty. The key challenge here is how to formulate our model as a MDP and solve it efficiently with large state and action spaces. In this paper, we divide this problem into two sub-problems. First, we use one Markov state to represent the discrete time and battery as well as location of an eTaxi, and one action to represent the driver's decision either to move forward to the next location or recharge eTaxi. The probabilistic transition is determined by a random event of passenger pick-up or drop-off. This MDP model can achieve optimal decision policy and maximize future reward (i.e., from 5 hours) of eTaxi drivers, especially in the presence of limited driving range and long recharging cycle. Secondly, since MDP needs to share the same discount factor which causes total reward from the first few road segments may dominates reward from route. To avoid this, we propose a novel recursion strategy to combine immediate reward and future reward from the rest working hours on candidate routes. Based on the recursive formulation of objective function, we build a state recursion tree to represent state transition on candidate routes and implement a routing algorithm based on state recursion tree to find the optimal candidate route efficiently.

In summary, our contributions in this paper are summarized as follows:

Author's address:

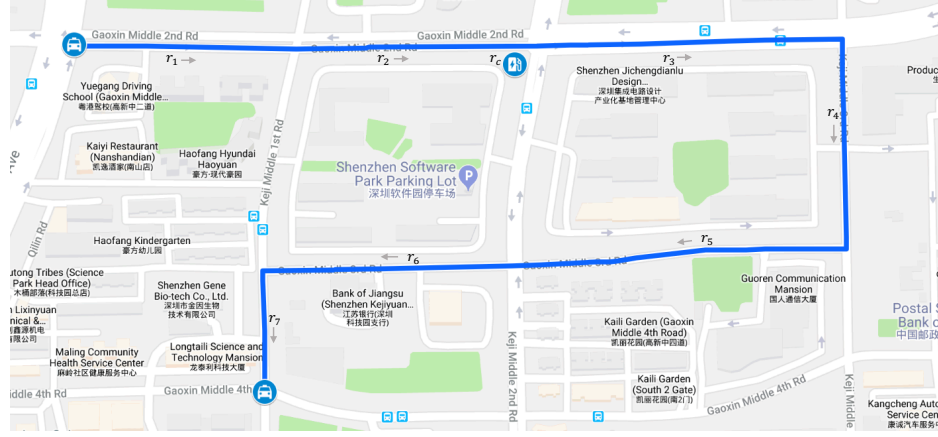


Fig. 1. An example of the route

- We formulate a MDP model to optimize eTaxi serving and recharging strategies and obtain the maximum future reward with consideration of varying location/time based passenger requests, energy cost, battery capacity and recharging duration.
- We design a novel recursion strategy to provide a detailed route with recharging strategy for an eTaxi driver on when, where and how long to recharge in order to achieve highest potential reward with consideration on both immediate reward and future reward.
- Last but not least, we evaluate the effectiveness and efficiency of our system model using a large and real-world taxi dataset in Shenzhen City.

2 PROBLEM FORMULATION

In this section, we design a real-time routing and recharging recommendation system *R2E*. Our goal is to provide a detailed route with recharging plan for eTaxi drivers. We propose a road segment based routing algorithm, called *MARA*, to compute the optimal route based on calculated reward of road segments which contains immediate reward from picking up the next passengers and future reward from the rest of working hours. Immediate reward can be calculated based on the real-world historical data. For future reward, we leverage the Markov Decision Process (MDP) to compute it. Details of each part of our *R2E* system will be discussed in the following sections.

2.1 Maximized Reward Routing Algorithm (*MARA*)

In this section, we propose a Maximized Reward Routing Algorithm (*MARA*) to compute a detailed route with recharging plan for an eTaxi driver with consideration on both immediate reward from picking up next passengers and future reward obtained from the rest of working time.

2.1.1 Preliminary. Before talking about our algorithm, we want to introduce three concepts: road segment, charging station and route.

- **Road Segments:** A long road can be divided into multiple road segments by intersections. Each road segment r is a directed segment that is associated with a direction symbol $r.dir$ (one-way or bidirectional), two terminal point $r.s$ and $r.e$, as well as several adjacent segments forming a set \mathcal{N}_r which satisfies $\forall r_i \in \mathcal{N}_r$, when $r.e = r_i.s$.

- **Charging station:** Since the location of charging station r_c is always close to the intersection, we assume that r_c is located at the intersection.
- **Route:** A route RT can be represented as a sequence of connected road segments and charging station. $RT = (r_1 \rightarrow r_2 \rightarrow \dots \rightarrow r_c \rightarrow \dots \rightarrow r_M)$, where $r_{i+1}.s = r_i.e$ ($1 \leq i \leq M$) and $r_{i+1} \in \mathcal{N}_{r_i}$. The start and end point of a route RT can be represented as $RT.s = r_1.s$ and $RT.e = r_M.e$. The position of charging station r_c in the route RT is between road segment r_i and road segment r_{i+1} , where $r_{i+1}.s = r_c = r_i.e$.

Fig.1 shows an example of the route. In this figure, the route can be represented as $RT = (r_1 \rightarrow r_2 \rightarrow r_c \rightarrow r_3 \rightarrow r_4 \rightarrow r_5 \rightarrow r_6 \rightarrow r_7)$

2.1.2 Maximized Reward Routing Algorithm (MARA). Instead of providing a sequence of pick-up points [3] or frequent non-consecutive recharging actions [15], the goal of our proposed recommendation system *R2E* is to provide a fine-grained route including both searching strategy for finding next passengers and recharging plan for eTaxi in order to achieve the maximum reward on route. This is different from traditional taxi recommendation systems, whose goal is only to maximize the immediate reward[9] and probability of picking up next passenger[19]. The expected reward of a road segment r_i in our proposed system *R2E* contains two parts: immediate reward and future reward. Immediate reward represents income from picking up next passengers or recharging cost. Meanwhile, future reward represents income that can be earned in the rest of working time after dropping off next passenger, which we have obtained from computational result of MDP and will discuss in the next section.

In general, traditional taxi recommendation systems [9, 19] consider the time and gas consumption in slotted fashion. Also, the change of the time and gas consumption on road segments is limited in one single slot and always ignored. However, our system *R2E* takes a step further to include the recharging plan for eTaxi (i.e., recharging for one hour). Therefore, the state change on time and battery can't be ignored. In this case, a single parameter r_i is not enough to describe the state transaction on route. Here, we introduce another two parameters t and b , which stand for *time* and *battery* respectively. In this case, a single state on the route can be represented as $s_i = (t, r_i, b)$ and we use \mathcal{S}_i instead of \mathcal{N}_{r_i} to describe the set of the next state transition along the route. Each vacant eTaxi can choose to recharge their eTaxi at charging station r_c or to move from the current road segment r_i to the adjacent road segment $r_j \in \mathcal{N}_{r_i}$. Formally, it can be expressed as an action $a \in A$: $A = \{ \cdot, 1, 2, \dots, \mathcal{N}_{r_i} \}$, where \cdot represents the recharging duration, \mathcal{N}_{r_i} indicate the cardinality of the set \mathcal{N}_{r_i} and the numbers index the direction to adjacent road segments. If the eTaxi driver chooses to recharge eTaxi, $\cdot > 0$ and the driver need to pay recharging fee c_t , where \cdot is recharging rate and c_t is time-varying electricity price. On the other hand, if the eTaxi driver choose to search passengers at the current road segment, the driver will obtain immediate reward of picking up the potential passengers.

Actually, the framework of this problem is similar to the MDP framework. However, the set of new states \mathcal{S} should not include states that drivers don't pick up passengers on current road segment and transitioned to the adjacent road segments. The reason is that if we include these new states at adjacent road segments when an eTaxi don't find next passengers, the immediate reward for picking up next passengers and future reward gained from the rest working time at these adjacent road segments will be discounted multiple time, due to the fact that MDP shares the same discount factor through process. And the expected reward of the route will be dominated by the first few road segments (i.e., two road segments). This conflicts with our goal. Therefore, we propose a new recursive algorithm, called Maximized Reward Routing Algorithm (*MARA*), instead of directly leveraging MDP framework to solve this problem.

In *MARA*, the set of new state \mathcal{S} only include the new state s' after dropping off passengers, which can be expressed as $s' = (t', k, b')$ and k represents the destination of passengers. However, since we assume passenger requests always happen at the road segments, there are no passenger requests happened at the charging stations. Therefore, the state at the charging station can't be transaction to the set of new state \mathcal{S} and thus it can't obtain the future reward from the new state s' . To deal with this problem, we assume that charging station r_c is located

at the road segment r_i , where $r_i.e = r_c$. Regardless of eTaxi drivers decide to recharge at the road segment r_i or search for passengers and then move to the adjacent road segments, firstly, they have to drive through the road segment and obtain immediate reward of picking up the potential passengers. Then, they decide whether to recharge eTaxi and pay recharging fee or not. Therefore, the expected immediate reward $R(s_i, a)$ at the road segment r_i can be calculated by:

$$R(s_i, a) = \sum_{s' \in S} \left(P(s_i) P_d(r_i, k) F(r_i, k) - (1 - P(s_i)) - c_t \right) \quad (1)$$

where $P(s_i)$ represents the probability of successfully picking up a passenger. And $P(s_i)$ can be influenced by the time t , the road segment r_i , as well as the battery b at state s_i . Besides, $P_d(r_i, k)$ represents probability of a passenger picked up at road segment r_i and dropped at destination k and $F(r_i, k)$ represents the corresponding fare of transporting passengers. Here, the expected reward of state s_i on route with an action a is denoted as (s_i, a) . Formally, (s_i, a) can be calculated by:

$$(s_i, a) = R(s_i, a) + \sum_{s' \in S} P_{s_i s'} V^*(s'), \quad (2)$$

The value of $V^*(s')$ can be obtained from computational result from MDP and will be discuss in the next section. Since immediate reward is always a very important concern for the drivers, we introduce $\gamma \in [0, 1]$ as the discount factor in order to adapt different requirements. When $\gamma = 0$, the problem can be reduced to maximize the immediate reward of picking up next passenger and future reward is not considered at all. eTaxi drivers will only recharge their eTaxis when battery level violates battery constraint. On the other hand, if $\gamma = 1$, future reward from the rest of working hours is fully considered. There will be cases that even though eTaxi drivers can achieve high potential immediate reward if they move to adjacent road segments and seek for passengers, they will still choose to recharge eTaxi and pay recharging cost. This is because the eTaxi drivers may achieve a much higher future reward in the rest of working hours by doing so.

Besides, each action can significantly influence the expected reward of route. In this case, we use $S_{RT} = (s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow \dots \rightarrow s_M)$ starting from s_1 to describe state transition on candidate route RT with the allowable action set. And the expected reward on route S_{RT} can be denoted as

$$G(S_{RT}, s_1, a, M) = (s_1, a) + \sum_{i=2}^M (s_i, a) \prod_{j=1}^{i-1} (1 - P(s_j)), \quad (3)$$

The length of a route denoted as M is the number of road segments in its sequence. Let P_{RT} be the probability that eTaxi driver picks up the the next passenger successfully and drive through RT . Then we have $P_{RT} = 1 - \prod_{i=1}^M (1 - P(s_i))$, where $P(s_i)$ is the pick-up probability of road segment r_i . Therefore, it is possible for us to compute an upper bound for route length M , when P_{RT} is larger than some threshold P_{th} . P_{th} can either be learned from the training data or be set by the users. .

Given current state s_1 of a eTaxi driver, a fixed route length M and a set of all candidate routes \mathbb{S}_{RT} , our MARA algorithm outputs an route with recharging plan S_{RT} , which can achieve the maximum total reward.

$$S_{RT}^* = \arg \max_{S_{RT} \in \mathbb{S}_{RT}} \{G(S_{RT}, s_1, a, M)\} \quad (4)$$

Assuming that the computational cost of finding a route $G(S_{RT}, s_1, a, M)$ is $T(M)$, there will be two possible situations:

- If there is no charging station on road segment, obviously we have $T(M) \leq NT(M-1)$, where $N = |\mathcal{N}_r|$
- If there is a charging station on road segment, similarly we have $T(M) \leq nNT(M-1)$, where n represent the number of different recharging duration.

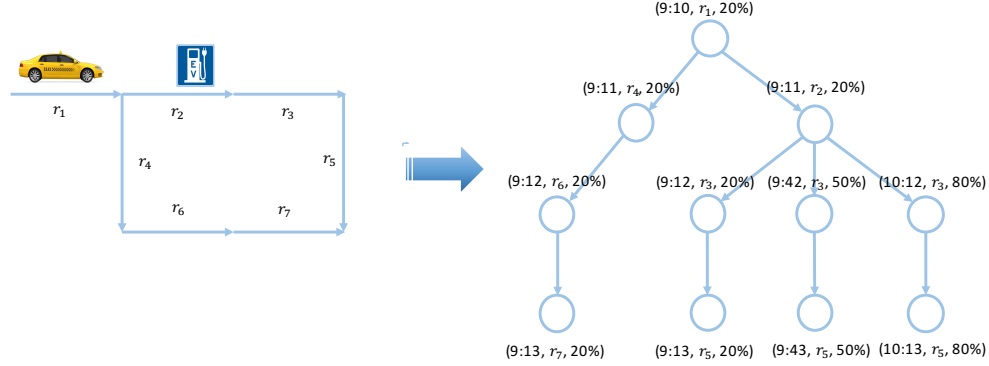


Fig. 2. An Example of State Recursion Tree

In fact, there is no candidate route in our experiment passes through two or more charging stations. After $M - 1$ times recursion, we have $T(M) \leq nN^{M-1}T(1)$. Therefore, the computational complexity of searching the optimal route is $O(nN^{M-1})$.

2.1.3 State Recursion Tree (SRT). The goal of our system *R2E* is to recommend an entire route with recharging plan that can maximize the potential reward of the route. Based on the recursive formulation of objective function, we propose a *SRT* algorithm to build up a M -depth state recursion tree, shown in Algorithm 1. That is to say, when eTaxi driver sends a request to our recommendation system, we treat the current road segment r_1 as a root and build a recursion tree to represent all candidate routes. Each node in the recursion tree will stand for a road segment. The children set of each node equals to \mathcal{N}_r , which is the reachable road segments from current road segment. If eTaxi drivers only seek for passengers on current road segment, we can build up the recursion tree as described as above.

In our system, we also ignore time and battery consumption traveling on a single road segment just like the traditional system. However, a recharging action on route will significantly affect state transition on route. That is to say, if there is a charging station r_c on a road segment r and eTaxi drivers choose to recharge for Δ duration (i.e., one hour), the recommendation system will have to take this into account in order to propose an accurate route. To address this issue, we firstly divide Δ into slots based on the predefined unit time slot for recharging duration d . Now, we obtain multiple recharging duration Δ_d , denoted by $[\lceil \Delta/d \rceil, 2\lceil \Delta/d \rceil, 3\lceil \Delta/d \rceil, \dots]$. In this case, the next potential states on route can be denoted as $s_j = (t + \Delta_d, r_j, b + \Delta_d)$, $s_j \in \mathcal{S}_i$. The time will be current time plus recommended recharging duration. The next location of eTaxi after recharging can be any adjacent road segment r_j , where $r_j \in \mathcal{N}_{r_i}$. And the battery will be equal to recharging duration multiplies recharging rate plus current battery. Based on these rules, we can calculate s_j for each $s_j \in \mathcal{S}_i$.

Figure 3 shows an example of *SRT*. In this figure, we assume that seeking time on each road segment is 1 minute. We have two choices on recharging duration: 30 mins and 60 mins. Recharging speed is 1% per minute. When an eTaxi driver passes through road segment r_2 , the eTaxi driver can choose to either directly go to the adjacent road segment r_3 or recharge his eTaxi with duration 30 mins or 60 mins.

2.1.4 A Real-time Routing and Recharging Recommendation. Based on the state recursion tree described in section 2.1.3, the routing and recharging recommendation from state s_1 can be divided into smaller tasks recursively.

ALGORITHM 1: *SRT* (s, M): State Recursion Tree

Input: state on road segment $s_1 = (t_1, r_1, b_1)$ as root, the depth M of recursion tree, maximum recharging time τ and unit time slot for recharging duration d **Output:** a M -depth state recursion tree \mathcal{R} **Initialization:** $Depth = 1$; $\mathcal{R}.le_el[i] = \emptyset, (1 \leq i \leq M)$; $\mathcal{S}_i = \emptyset$; $\mathcal{R}.le_el[1] = s_1$;**if** ($Depth = M$) **then** return \mathcal{R} **else** **for each** ($s_i \in \mathcal{R}.le_el[Depth]$) **do** **if** r_c at the road segment r_i **then** **for** $n : 0$ to $\lceil \tau/d \rceil$ **do** **if** $n \neq \lceil \tau/d \rceil$ **then** $t_j = t_i + n \lceil \tau/d \rceil, b_j = b_i + j \lceil \tau/d \rceil$ **else** $t_j = t_i + \tau, b_j = b_i + \tau$ **end** **for each** $r_j \in \mathcal{N}_{r_i}$ **do** $s = (t_j, r_j, b_j)$ and $\mathcal{S}_i \cup = \{s_j\}$ **end** **end** **else** $t_j = t_i, b' = b_i$; **for each** $r_j \in \mathcal{N}_{r_i}$ **do** $s_j = (t_j, r_j, b_j)$ and $\mathcal{S}_i \cup = \{s_j\}$ **end** **end** $\mathcal{R}.le_el[Depth + 1] \cup = \mathcal{S}_i$; **end** $Depth++ = 1$;**end**

In this case, we proposed our recursive routing algorithm $MARA(s, K)$, shown in Algorithm 2. As described in our algorithm, with input parameters $s = s_1$ and $K = M$, the optimal route starting from s_1 with length M and corresponding maximum reward will be obtained.

Different from other research work [10], which focuses on evaluating the reward of taxi service without considering response time, the goal of $R2E$ is to provide a real-time routing and recharging recommendation for eTaxi drivers. To achieve real-time efficiency, we choose to compute input parameters on outside server before hand. To do this, we first collect the final states (i.e., shifted handover) of all users on grid level. This can be learned from the training data or set by users. Next, we calculate the optimal future reward of all grid pairs between final states and potential beginning states. The benefit of using grid to represent location in state is to lower computation cost while maintaining high accuracy on future reward evaluation. Although pre-computation can be done on outside servers, computing optimal future reward of all pairs among 135,317 road segments is still a big computational cost. To improve efficiency further, we first compute immediate reward $R(s, a)$ of each road segment in the same way as previous recommendation systems for tradition taxi [9, 19]. Next, we compute future reward on each road segment based on grid computation results $V^*(s')$. Thus, for each road segment, we obtain two values: immediate reward $R(s_i, a)$ and future reward $\sum_{s' \in S} P_{s_i s'} V^*(s')$, respectively. We, then, use $R(s_i, a)$ and

ALGORITHM 2: MARA (s, K): Maximized Reward Routing Algorithm

Input: recursion tree of \mathcal{R} , discount factor γ and the depth M of state recursion tree

Output: The maximum total reward and route start from s_1 .

$Depth = M - K + 1$;

if ($Depth \neq M$) **then**

$Profit = 0$; $Route = \emptyset$;

for each ($s_i \in \mathcal{R}.le_el[Depth]$) **do**

$Profit[i] = R(s_i, a) + \sum_{s' \in \mathcal{S}} P_{ss'}^a V^*(s', a)$ and $Route[i] = s_i$;

return ($Max(Profit), Max(Route)$)

end

else

$Profit = 0$ and $Route = \emptyset$;

for each ($s_i \in \mathcal{R}.le_el[Depth]$) **do**

 ($Profit^*, Route^*$) = MARA($s, K - 1$) and $Route[i] = s_i \cup Route^*$;

$Profit[i] = R(s_i, a) + \sum_{s' \in \mathcal{S}} P_{ss'}^a V^*(s', a) + (1 - P(s_i)) \cdot Profit^*$;

return ($Max(Profit), Max(Route)$);

end

end

$\sum_{s' \in \mathcal{S}} P_{ss'} V^*(s')$ as the pre-computed result and leverage our recursive algorithm *MARA* to obtain an optimal route starting from s_1 with length M and the corresponding maximum reward.

2.2 Markov Decision Process Formulation

In this part, we will discuss how to compute the future reward from the rest of the working day. We formulate the serving and recharging problem of an eTaxi as a Markov Decision Process (MDP), which is defined by the sets of states and actions, the state transition probabilities, as well as the value function.

2.2.1 State. A MDP must guarantee the Markov property, which requires that the next state of the process only depends on the current state and the action, but not any previous state or action. A state of an eTaxi can be described with three parameters: time, location and battery.

- **Time:** we consider discrete time slots, denoted as t . Since taxi drivers usually shifted handover twice in one day [20], the working duration of eTaxi drivers can be limited to finite time horizon (i.e., 12 hours).
- **Location:** we partition the city into L non-overlapping grids and leverage \mathcal{S} to represent the location of an eTaxi.
- **Battery level:** we consider discrete battery levels, denoted as b . The feasible battery level should be within the range $[b, \bar{b}]$.

Thus, a state s can be denoted as $s = (t, \cdot, b)$.

2.2.2 Action. We denote the action set in our MDP as A . Assuming that the current grid is i and the adjacent set of grids is denoted as \mathcal{S} , each vacant eTaxi at one state can choose to recharge their eTaxi at current grid i or to move from current grid i to grid $j \in \mathcal{S}$ after seeking current grid i . Formally, it can be expressed as: $A = \{ \cdot, 1, 2, 3, 4, 5, 6, 7, 8, 9 \}$, $a \in A$ where \cdot represents the recharging duration and nine number (1–9) index the direction to next grids.

2.2.3 Value Function. The objective of each eTaxi is to maximize the expected future reward, which equals to its immediate reward at state s plus its expected future reward from new state s' transited from current state.

Parameter	Meaning
	recharging rate.
c_t	electricity price at time t .
$P(s)$	probability of successfully picking up a passenger at state s .
$P_d(\cdot, k)$	probability of a passenger picked up at grid \cdot and dropped off grid k .
$T(\cdot)$	seeking time for picking up a passenger at grid \cdot .
$T_d(\cdot, k)$	traveling time from grid \cdot to grid k .
$B(s, a)$	battery consumption for an action a at grid \cdot .
$B_d(\cdot, k)$	battery consumption for traveling from grid \cdot to grid k .
$F(\cdot, k)$	fare of transporting passengers from grid \cdot to grid k .

Table 1. Parameter Notation and Meaning

Thus, the optimal future reward for an allowable action $a = \{ \cdot, 1, 2, 3, 4, 5, 6, 7, 8, 9 \}$ at state s is denoted as $V^*(s)$. Formally, $V^*(s)$ can be expressed as follows:

$$V^*(s) = \max_{a \in A} \left(R(s, a) + \sum_{s' \in S} \left(P_{ss'}^a V^*(s') \right) \right) \quad (5)$$

where $R(s, a)$ denotes the immediate reward of action a given current state s , $P_{ss'}^a$ is transition possibility to new state $s' \in S$, given current state s and the chosen action a . $V^*(s')$ is the optimal future reward of new state s' .

2.2.4 State Transition. Assuming the current state of an eTaxi is $s = (t, \cdot, b)$, the state transition will be determined based on the parameters listed in Table 1. If an eTaxi driver choose to cruise at \cdot for time $T(\cdot)$ and then move from grid \cdot to its neighbor j , there will be two possible consequences.

- The eTaxi does not find any passenger after $T(\cdot)$ in \cdot . Then, the new state of the eTaxi transits to $s' = (t', j, b')$, where $t' = t + T(\cdot)$, $b' = b + B(s, a)$ and $j \in S$. The transition probability $P_{ss'}^a$ is $1 - P(s)$.
- The eTaxi successfully finds a passenger in grid \cdot after cruising the grid for time $T(\cdot)$ with battery consumption $B(s, a)$. Then it will use time $T_d(\cdot, k)$ to drive to one of the next grid cells as destination, denoted as k with the probability $P_d(\cdot, k)$ and battery consumption $B_d(\cdot, k)$. After finishing this trip, the time will be $t + T(\cdot) + T_d(\cdot, k)$, the next grid will be k and the battery level will be $b + B(s, a) + B_d(\cdot, k)$. Then, the eTaxi will start seeking in k again. The new state of the eTaxi is thus transitioned to $s' = (t + T(\cdot) + T_d(\cdot, k), k, b + B(s, a) + B_d(\cdot, k))$. The transition probability $P_{ss'}^a$ to grid k is $P(s)P_d(\cdot, k)$.

If an eTaxi driver chooses to recharge the eTaxi with duration \cdot in current grid \cdot , It still takes time $T(\cdot)$ to arrive at the charging station. In this process, it is possible to pick up passengers unless it runs out of energy. Thus, the state s' of the eTaxi after recharging transitions to $s' = (t + T(\cdot) + \cdot, \cdot, b + B(s, a) + \cdot)$. Moreover, the new state after dropping off passengers is the same as described above, $s' = (t + T(\cdot) + T_d(\cdot, k), k, b + B(s, a) + B_d(\cdot, k))$. Similarly, regardless of seeking in grid \cdot or recharging eTaxi in grid \cdot , the eTaxi drivers will obtain the potential immediate reward from picking up the passengers, thus, the expected immediate reward it can be expressed as:

$$R(s, a) = \sum_{s' \in S} \left(P(s)P_d(\cdot, k)F(\cdot, k) - (1 - P(s)) - c_t \right) \quad (6)$$

Next, we leverage shift handover event detection algorithm proposed in [20] to identify shift handover events and learn final states of eTaxis starting from shift handover events. In this case, final states in our proposed MDP model can be determined. Besides, we can easily determine the beginning states when users send recommendation requests. Thus, we can leverage dynamic programming algorithm by starting from the final states and then going

backward to the beginning states to solve the problem. Eventually, we will obtain the maximum future reward of the beginning states.

2.3 Learning Parameters for MDP

In this section, we discuss how to determine necessary parameters for the MDP. All parameters are learned from our historical passenger trips, seeking trips and recharging behavior.

The probability $P(s)$ of picking up a passenger at state $s = (t, i, b)$ can be estimated by dividing number of successful pickup events at state s with the total number of visits of a location by a vacant taxi. Specially, First, since the remaining battery is insufficient, certain action is unfeasible. Therefore, we consider the required battery consumption of the pickup events that can be supported by the current remaining battery b of eTaxis and counts the number of these pickup events as $N(s)$. Then, we use $N_t(i)$ to denote the number of visits in location i . Next, we leverage recharging intention identification in [12] to count the number of eTaxis' intention of taking recharging action, denoted as $N_t^r(i)$. Last but no least, we also calculate pickup counts $N_t^p(i)$ for each location. In this case, pick up probability $P(s)$ can be calculated as follows:

$$P(s) = \frac{N(s)}{N_t^p(i) + N_t(i) - N_t^r(i)} \quad (7)$$

To estimate destination probability, denoted as $P_d(i, j)$, we first calculate number of trips $N_d(i, j)$ between each pair of source i and destination j . Then, we normalize it by total number of the reachable trips $N(s)$ at source grid i .

$$P_t^d(i, j) = \frac{N_t^d(i, j)}{N(s)} \quad (8)$$

The time dependent fare $F(i, j)$ is used to represent the fare income for a trip. We take the average fare of trips in dataset between each pair of source i and destination j as the expected fare $F(i, j)$. Similarly, to estimate the traveling time $T_d(i, k)$ between each pair of source i and destination j , we use the average traveling time of trips. To estimate the energy consumption $B(s, a)$ and $B_d(i, k)$, we first calculate the average driving speed in each location at different time when taking different action. Next, we leverage energy consumption model in [1, 13] as a black-box to estimate the energy consumption for eTaxis.

To estimate these parameters on road segment, we segment GPS trajectories and map-match the GPS trajectories to road networks by leveraging algorithm in [18]. Due to the fact that people always wait for taxis at road side instead of in the middle of road as well as GPS errors on GPS trajectories, we exploit the road buffer estimation for each road segment [14].

3 EVALUATION

To validate the efficiency and effectiveness of *R2E* system, extensive experiments are performed on real world data sets collected in Shenzhen.

3.1 Data set Description

Our GPS trace dataset is collected in 2017 during the period 01/01/2017 – 06/01/2017, including over 800 eTaxis and 7000 taxis operated in Shenzhen City. This dataset contains three different types of data sources: 1) the GPS data; 2) charging infrastructure data and 3) road network data.

- **GPS Data** includes 7.56 billion taxi trajectory points with the size of 529 GB in total. Each GPS record includes the car ID, the GPS location information (i.e., longitude and latitude), time-stamp, the plate color

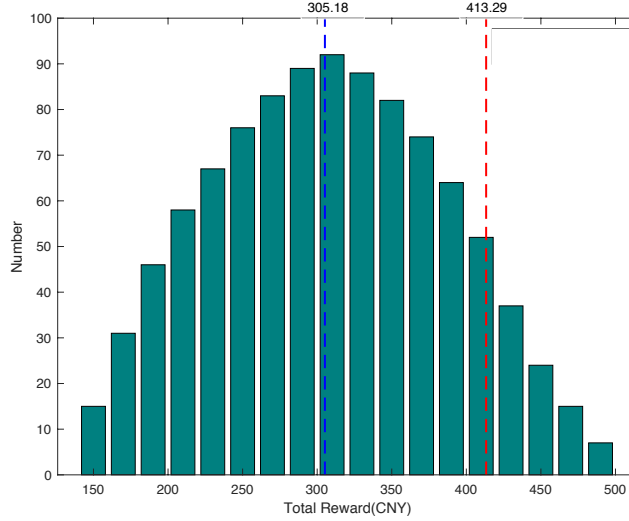


Fig. 3. Distribution of Total Reward for Day Time (5.am-5.pm).

and transaction data. The transaction data record includes pick-up/drop-off time, duration and distance taxi travelled as well as metered fare.

- **Charging Station Data** includes locations of 183 charging stations, electric price and opening dates for each station.
- **Road Network Data** includes 135,317 road segments and 105,703 road nodes in Shenzhen. Each road segment contains road ID, length and type.

3.2 Distribution of Total Reward

Our goal is to maximize total reward of an eTaxi in one day. We use shifted handover event detection proposed in [20] to identify shift handover events of eTaxis and obtain final states of eTaxi. Since most of shifted handover events happen at 5.am and 5.pm, we compare our result against real-world reward of eTaxi during day time (5.am-5.pm) and evening time (5.pm-5.am) respectively. Here, we call the data of one eTaxi of one driver with one shift handover (either 5.am-5.pm or 5.pm-5.am) a single data entry. To evaluate, we randomly select 1,000 eTaxi data entries of day time and another 1,000 data entries of evening time from our data set for evaluation and all shift handover events happen either at 5.am or 5.pm. We repeat this process 10 times and calculate the average.

The result is shown in Figure 3 and 4. The vertical axis indicates the total number of entries of given reward, while horizontal axis shows the total reward of each data entry. The blue dash line indicates the average reward of real-world rewards, while the red dash line shows our MDP result.

Figure 3 shows the average expected total reward of MDP is 35.42% higher than average total reward gained by eTaxi drivers in day time. Figure 4 shows an 47.01% increment of total reward of MDP in evening. The reason is that MDP can balance the recharging and serving cycles and capture larger-reward opportunities of hot-spots and peak-hours.

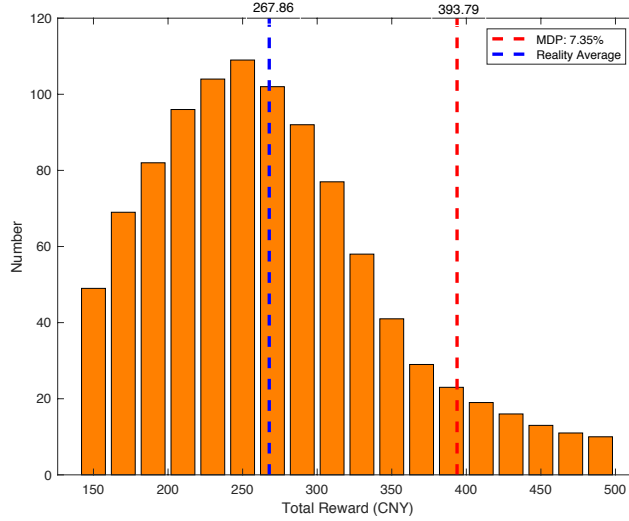


Fig. 4. Distribution of Total Reward for Evening (5.p.m-5.a.m).

3.3 Increase in Total Reward

Next, we evaluate the increment in total reward with different starting time and different initial battery level compared with real-world data. In this experiment, we randomly select 300 eTaxi data sentries for each specific setting, which include specific starting time and initial battery level (i.e., starting time: 5.am and battery level is 30%), from our data set. We compare our result with total reward gained by real-world eTaxi drivers. The detailed result is shown in Figure 5.

Here, the working duration of all selected eTaxi is between 5.am and 5.p.m, which is indicated in y-axis. X-axis shows the initial battery levels. In this experiment, we selected 30, 50, 70, 90 percent for initial battery level. From this figure, we can observe that our system can achieve a significant increase on total reward with highest increment of 142. Besides, early starting time can gain a bigger increment, which means the longer the working time is, the higher reward it can achieve. Lastly, we can observe that our system can achieve a bigger increment with lower starting battery level.

Also, we run similar experiments for evening time from 5.p.m to 5.a.m and it shows very similar results. In this case, we can conclude that our recommendation system is applicable for eTaxi drivers to increase their total reward, especially for eTaxi driver with low battery level and long remaining working time.

3.4 Comparison against Traditional Recommendation System

Moreover, we perform evaluation of total reward increment comparing against traditional taxi recommendation system. Although there exist various traditional taxi recommendation systems [2, 9, 19], the common focus of these existing traditional taxi recommendation systems is to maximize the immediate reward of picking up next passenger. However, there are no publicly available tools for us to leverage. Here, we simulate the service strategies of traditional taxi recommendation systems for evaluation.

We assume that eTaxi drivers who use traditional taxi recommendation systems can always find next passengers with maximum immediate reward within 10 road segments. If there are no passenger pick-up request within 10 road segments, it will perform a new pick-up request at the end of recommended route. These eTaxi drivers will

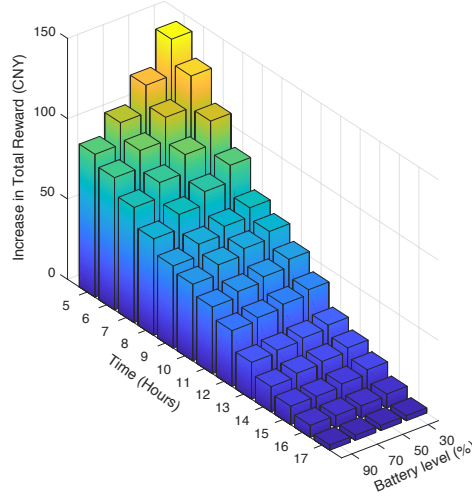


Fig. 5. Increase in Total Reward compared with Real-world eTaxi Drivers

keep serving passengers until their battery level lower than the predefined minimum battery level, which we set it to 10%. Then they will have to go to the nearest charging station for recharging to maximum battery level, which is set to 90% as recommended in [16]. The result is shown in Figure 6.

We observe that even if traditional taxi recommendation systems can pick up next passenger with maximum immediate reward, the total reward is still lower than our recommendation system. Especially, we can see when eTaxi driver started serving at 5.am with remaining 50% battery level, it achieves the biggest difference compared to our system. The reason is that eTaxi runs out of battery at 7.am and it has to do a recharging during 7.am-8.am, which is the peak-hours in morning. Besides, although we assign passengers with maximum immediate reward to eTaxi when simulating traditional taxi recommendation systems, it still may be a relative small reward compared to others in the system. Also, even if immediate reward of pick-up next passenger is large (i.e., 100 CNY), it may take you to suburbs and eTaxi drivers fail to capture large-profit opportunities at peak hours in hot-spot areas. Therefore, with consideration on future reward, our recommendation system outperforms traditional taxi recommendation system.

3.5 A case study with the different discount factors

Here, we show the examples of MARA route recommended by our approach. The recommendation request happens at time 10:35 am on Monday. At that time, the estimated remaining battery of eTaxi is 35%. We assume that the drivers' expected route length is 7. In Fig 7 we plot the optimal driving routes suggested by our recommendation system with four different discount factors, $\gamma = 1$, $\gamma = 0.6$, $\gamma = 0.2$ and $\gamma = 0$. When the driver decides the discount factor $\gamma = 1$, it means the future reward from the rest of working hour is fully consideration. As shown in Fig 7(a), the recommended route passes the charging station. The recommended recharging plan is to recharge 30 mins and the remaining battery will be up to 65%. The reason is the time from 11:00 am to 13:00 pm is the peak hour and the 65% remaining battery is enough for the eTaxi driver to serve the passenger requests during peaking hours. When $\gamma = 0.6$, it means the future reward from the rest of working hour is partially consideration. As shown in Fig 7(b), although the recommended route still passes the charging station,

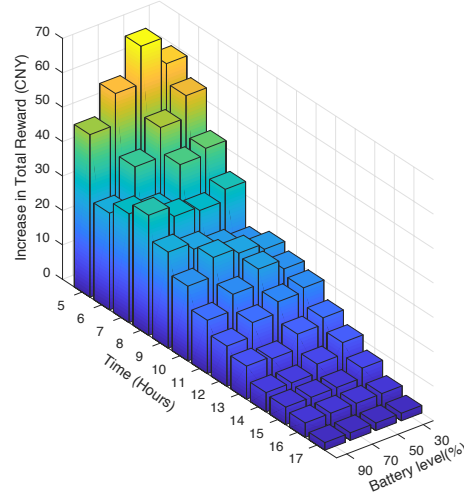


Fig. 6. Increase in Total Reward compared with Traditional Taxi Recommendation System

the recommended recharging time is reduced to 10 mins, which means eTaxi drivers need to recharge their eTaxi at peaking hours. On the other hand, the eTaxi driver also have higher immediate reward along the route. For $\gamma = 0.2$ and $\beta = 0$, as shown in Fig 7(c) and Fig 7(d), we find out that both recommended routes don't pass charging station and recommend eTaxi driver to recharge the eTaxi. It's since such small discount factors means the eTaxi drivers care more about immediate reward than future reward. The only different between them is that the aim of the recommended route shown in Fig 7(d) is to obtain the highest immediate reward. On the other hand, the recommended route shown in Fig 7(c) still consider the influence of future reward (i.e., the higher probability to hot spots).

3.6 System Running time Performance

Before showing our system performance, we want to discuss more details on the recommended route. As mentioned above, the pick-up probability on route can be computed as $1 - \prod_{i=1}^M (P(s_i))$, where M is the length of route. In fact, the average pick-up probability of one single road segment calculated based on our real-world data set is around 0.15. In order to maintain the effectiveness of our recommendation system, we should guarantee a reasonable probability (e.g., 70%) that an eTaxi will pick up next passenger on our recommended route. Besides, we should not consider the road segments which are far away from the current location because the expected reward of those segments will only perform a very small amount of contribution to the total reward. Moreover, as mentioned above, the computational complexity of our system is $O(nN^{M-1})$, which relies on the length of route M . In this case, the length of route should not be too big. Based on our experiments, the pick-up probability on recommended route can achieve 0.7 when we set $M = 7$. And when $M = 7$, the running time is around ≤ 0.012 second. As we increase to $M = 10$, the running time increases significantly. However, it can still keep less than 1 second. In this case, we believe that $M = 7$ is a suitable setting in our system since it can ensure the high pick-up probability as well as high efficiency. The overall run time performance of our system is shown in Figure 8. As a result, we can conclude that our *R2E* recommendation system can achieve high efficiency in responding users' requests.

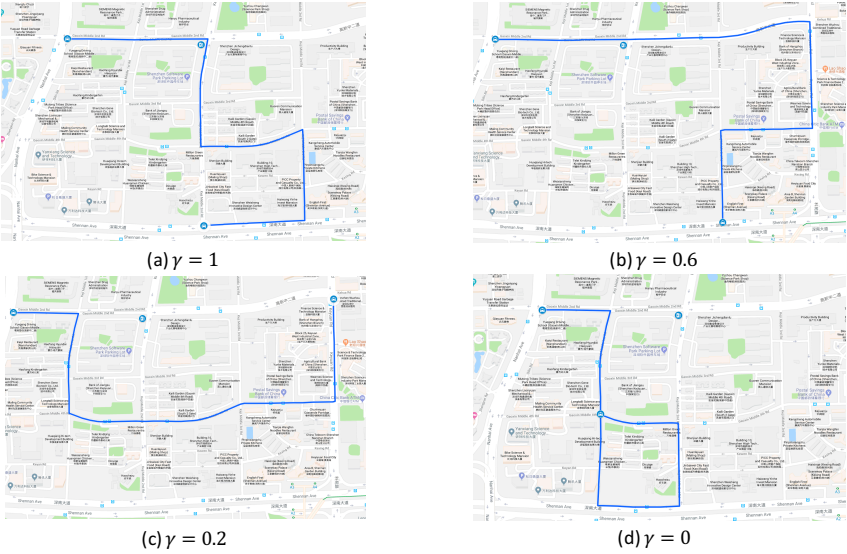


Fig. 7. A Case Study with the different discount factors

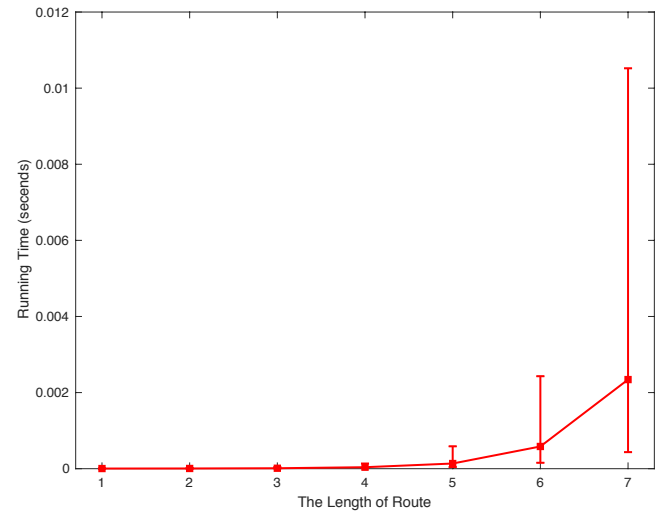


Fig. 8. Running Time

4 RELATED WORK

Recently, the abundant taxi GPS trajectories data has enable new way of doing taxi business. Many efforts have been made on developing recommendation system by leveraging taxi GPS trajectories data.

4.1 Taxi Recommendation System

Ge *et al.* [3] proposes a mobile sequential recommendation model to provide a taxi driver with a sequence of pick-up points so as to maximize reward from pick-up next passengers. Li *et al.* [5] study the passenger-finding strategies (hunting or waiting) of taxi drivers in Hangzhou and use $L1$ Norm support vector machine to select features for classifying the passenger-finding strategies in terms of drivers' performance. Powell *et al.* [8] propose an approach to suggest profit (grid-based) locations for taxi drivers by constructing a Spatio-Temporal profitability map, on which, the nearby regions of the driver are scored according to the potential reward calculated by the historical data. Instead of recommending separate pick-up locations, recent approaches is to maximize the reward of taxi drivers by providing an entire driving route. The drivers are able to find a passenger by following the recommendations. Yuan *et al.* study the strategies for improving taxi drivers' income and provide a detailed route to a parking lots with maximum potential reward or highest pick-up probability. Qu *et al.* propose a cost-effective recommendation system for taxi drivers to maximize their net profit. Different from the taxi recommendation whose goal is to maximize potential immediate reward from picking-up next passengers, our *R2E* system aims to provide an entire route with recharging plan to maximize potential total reward which contains both immediate reward from picking-up next passengers and future reward from the rest of working day.

4.2 Recharging Strategies of eTaxis

Currently, studies on recharging strategies of public transportation systems usually utilize the theory of scheduling optimization and trajectory mining. Lu *et al.* [6] propose a heuristic dispatching strategy for eTaxis to reduce waiting time at charging station by considering influence of passenger requests, remaining power of eTaxis and the availability of charging stations. Malandrino *et al.* [7] conduct their research on the interaction between charging stations and electric vehicles, and aim to discover a balanced time/price trade-off between electric vehicle drivers and power supply merchants via game theory. Yang *et al.* [15] investigate the optimal scheduling problem of recharging cycle and serving actions for eTaxis with time-varying service incomes and provide a non-consecutive recharging recommendation. In [21], by mining the taxis' trajectories, both service strategies and refuel behaviors of traditional taxis are investigated from a spatial-temporal economic view. Hou *et al.* [4] propose a transfer-allowed shared eTaxi strategy in order to decrease idle distance cost and increase the number of passengers delivered by eTaxis. Our proposed *R2E* system is different from the above methods which aim to minimize waiting time, idle distance and recharging cost. Our goal is to provide an entire route and a detailed recharging plan to tell an eTaxi driver when, where and how long to recharging the eTaxi can achieve highest potential total reward.

5 CONCLUSION

In this paper, we propose a real-time routing and recharging recommendation system, *R2E*, for eTaxi drivers to maximize their total reward by providing an entire route with recharging plan. Firstly, we calculate immediate reward from historical data. Next, we formulate the serving and recharging problem of an eTaxi as a Markov Decision Process and leverage MDP to compute future reward in the rest of working hours. In this process, we introduce $\gamma \in [0, 1]$ as the discount factor in order to adapt different requirements of future reward. Besides, due to the problem that MDP shares the same discount factor through process, We can not directly apply the similar MDP framework to solve this problem. This is because future reward gained from the rest working time at these adjacent road segments will be discounted multiple times and the first few road segments will dominate

the total reward, which conflicts with our goal. Therefore, we propose a new recursion algorithm, Maximized Reward Routing Algorithm, MARA, for solution. Lastly, we evaluate our *R2E* on a real-world data set to show the effectiveness of our proposed *R2E* system.

REFERENCES

- [1] Chi-Kin Chau, Khaled Elbassioni, and Chien-Ming Tseng. 2016. Fuel minimization of plug-in hybrid electric vehicles by optimizing drive mode selection. In *Proceedings of the Seventh International Conference on Future Energy Systems*. ACM, 13.
- [2] Ye Ding, Siyuan Liu, Jiansu Pu, and Lionel M Ni. 2013. Hunts: A trajectory recommendation system for effective and efficient hunting of taxi passengers. In *Mobile Data Management (MDM), 2013 IEEE 14th International Conference on*, Vol. 1. IEEE, 107–116.
- [3] Yong Ge, Hui Xiong, Alexander Tuzhilin, Keli Xiao, Marco Gruteser, and Michael Pazzani. 2010. An energy-efficient mobile recommender system. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 899–908.
- [4] Yunfei Hou, Weida Zhong, Lu Su, Kevin Hulme, Adel W Sadek, and Chunming Qiao. 2016. Taset: Improving the efficiency of electric taxis with transfer-allowed rideshare. *IEEE Transactions on Vehicular Technology* 65, 12 (2016), 9518–9528.
- [5] Bin Li, Daqing Zhang, Lin Sun, Chao Chen, Shijian Li, Guande Qi, and Qiang Yang. 2011. Hunting or waiting? Discovering passenger-finding strategies from a large-scale real-world taxi dataset. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on*. IEEE, 63–68.
- [6] Jun-Li Lu, Mi-Yen Yeh, Yu-Ching Hsu, Shun-Neng Yang, Chai-Hien Gan, and Ming-Syan Chen. 2012. Operating electric taxi fleets: A new dispatching strategy with charging plans. In *Electric Vehicle Conference (IEVC), 2012 IEEE International*. IEEE, 1–8.
- [7] Francesco Malandrino, Claudio Casetti, and Carla-Fabiana Chiasserini. 2015. A holistic view of ITS-enhanced charging markets. *IEEE Transactions on Intelligent Transportation Systems* 16, 4 (2015), 1736–1745.
- [8] Jason W Powell, Yan Huang, Favien Bastani, and Minhe Ji. 2011. Towards reducing taxicab cruising time using spatio-temporal profitability maps. In *International Symposium on Spatial and Temporal Databases*. Springer, 242–260.
- [9] Meng Qu, Hengshu Zhu, Junming Liu, Guannan Liu, and Hui Xiong. 2014. A cost-effective recommender system for taxi drivers. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 45–54.
- [10] Huigui Rong, Xun Zhou, Chang Yang, Zubair Shafiq, and Alex Liu. 2016. The rich and the poor: A Markov decision process approach to optimizing taxi driver revenue efficiency. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management*. ACM, 2329–2334.
- [11] NYC Taxi, Limousine Commission, et al. 2013. Take charge: A roadmap to electric New York City taxis. (2013).
- [12] Zhiyong Tian, Taeho Jung, Yi Wang, Fan Zhang, Lai Tu, Chengzhong Xu, Chen Tian, and Xiang-Yang Li. 2016. Real-time charging station recommendation system for electric-vehicle taxis. *IEEE Transactions on Intelligent Transportation Systems* 17, 11 (2016), 3098–3109.
- [13] Hewu Wang, Xiaobin Zhang, and Minggao Ouyang. 2015. Energy consumption of electric vehicles based on real-world driving patterns: A case study of Beijing. *Applied Energy* 157 (2015), 710–719.
- [14] Hui Xiong, Shashi Shekhar, Yan Huang, Vipin Kumar, Xiaobin Ma, and Jin Soung Yoc. 2004. A framework for discovering co-location patterns in data sets with extended spatial objects. In *Proceedings of the 2004 SIAM International Conference on Data Mining*. SIAM, 78–89.
- [15] Zaiyue Yang, Lihao Sun, Min Ke, Zhiguo Shi, and Jiming Chen. 2014. Optimal charging strategy for plug-in electric taxi with time-varying profits. *IEEE Transactions on Smart Grid* 5, 6 (2014), 2787–2797.
- [16] Murat Yilmaz and Philip T Krein. 2013. Review of battery charger topologies, charging power levels, and infrastructure for plug-in electric and hybrid vehicles. *IEEE Transactions on Power Electronics* 28, 5 (2013), 2151–2169.
- [17] Jing Yuan, Yu Zheng, Chengyang Zhang, Wenlei Xie, Xing Xie, Guangzhong Sun, and Yan Huang. 2010. T-drive: driving directions based on taxi trajectories. In *Proceedings of the 18th SIGSPATIAL International conference on advances in geographic information systems*. ACM, 99–108.
- [18] Jing Yuan, Yu Zheng, Chengyang Zhang, Xing Xie, and Guang-Zhong Sun. 2010. An interactive-voting based map matching algorithm. In *Proceedings of the 2010 Eleventh International Conference on Mobile Data Management*. IEEE Computer Society, 43–52.
- [19] Jing Yuan, Yu Zheng, Liuhang Zhang, Xing Xie, and Guangzhong Sun. 2011. Where to find my next passenger. In *Proceedings of the 13th international conference on Ubiquitous computing*. ACM, 109–118.
- [20] Daqing Zhang, Lin Sun, Bin Li, Chao Chen, Gang Pan, Shijian Li, and Zhaohui Wu. 2015. Understanding taxi service strategies from taxi GPS traces. *IEEE Transactions on Intelligent Transportation Systems* 16, 1 (2015), 123–135.
- [21] Fuzheng Zhang, Nicholas Jing Yuan, David Wilkie, Yu Zheng, and Xing Xie. 2015. Sensing the pulse of urban refueling behavior: A perspective from taxi mobility. *ACM Transactions on Intelligent Systems and Technology (TIST)* 6, 3 (2015), 37.