International Journal of Smart Home
Vol.8, No.1 (2014), pp.113-122
http://dx.doi.org/10.14257/ijsh.2014.8.1.12

Development of Reservation Recommendation Algorithms for Charging Electric Vehicles in Smart-Grid Cities

Chan Jung Park[1], Junghoon Lee[2], Gyung Leen Park[2] and Jung Suk Hyun[3,1]

[1]*Dept. of Computer Education, Jeju National Univ., Jeju-do, Republic of Korea*
[2]*Dept. of Computer Science and Statistics, Jeju National Univ., Jeju-do, Republic of Korea*
[3]*Dept. of Management Information Systems, Jeju National Univ., Jeju-do, Republic of Korea*
{cjpark,jhlee,glpark,jshyun}@jejunu.ac.kr

Abstract

Electric vehicles (EVs) have become one of the most important means of transportation recently. So far, the research related to EVs has mainly focused on battery charging technique and battery management. Since the infrastructure for EV services in Jeju is in the inchoate stage, more various kinds of research works related to EV charging services should be performed. In this paper, we propose new reservation recommendation algorithms for charging EVs while traveling, especially in Jeju-do, to prepare for EV charging reservation systems in the near future. In our proposed algorithms, we select charging stations based on distance and route. When we recommend alternative charging stations, we provide 3 possible stations such as (i) the desired amount of battery without waiting time, (ii) the desired amount of battery with waiting time, and (iii) the limited amount of battery with waiting time. We also describe how to manage the time slots for scheduling the EVs which want to reserve charging.

Keywords: *Electric Vehicle, Battery Charge, Reservation, Recommendation, Smart Grid*

1. Introduction

After a smart grid test-bed was officially launched at Jeju-do in 2009, the concern for electric vehicles (EVs) has increased more and more recently [1, 2]. In addition, Jeju Provincial Government created the Division of Smart Grid, and the Division has developed various kinds of policies and helped the research centers and the corporations in Jeju-do. One of the Division's interests is smart transportation, and the Division tries to lay the foundations for expanded distribution of EVs. Thus, many corporations in Jeju-do have been developing various types of techniques and products for EVs since 2009. So far, the research works related to EVs have mainly focused on technologies for battery management, navigation system, and monitoring system. However, there have rarely been the research works on the services for the tourists who want to use EVs in Jeju-do while their traveling. The only one is the map for the EV charging stations in Jeju-do [3].

The necessity for charging while traveling was described in the Lee *et al.'s*, research [4]. According to the research, the total trip distance is about 195 km in Jeju-do and the average distance for one full charging is about 100 km [4]. Thus, EVs should be charged at least once while traveling. Also, a reservation system should be considered for the charging efficiency. However, unlike other reservation systems such as accommodation [5] and rent-a-car, the reservation recommendation system for charging the EVs allows the interruption while

[1] Jung Suk Hyun is the corresponding author of the paper.

ISSN: 1975-4094 IJSH

servicing. For example, when a tourist reserves a hotel room for 1 night, the tourist can use the room (the reserved object) without other people's interruptions. However, there is no reservations system which allows service suspensions for scheduling the reservation requests after reservations.

Thus, in this paper, we develop algorithms for reservation recommendation for charging EVs, especially while traveling. Also, we provide the time slot management method for reserving a specific charger. In the following section, we present how the tourists (drivers) can find the charging stations they want and how they can choose the charger they want to reserve. In section 3, we also describe data formats for our algorithms and processing steps between our reservation recommendation application and our server. Finally, we conclude our paper in section 4.

2. Background

In this section, we present the previous research works related to charging scheduling methods for EVs. In order to reserve efficiently, we firstly examine how to charge EVs efficiently in terms of waiting time, charging time, and charging cost. Most of the previous research works focused on the minimum waiting time of charge for each EV. In Qin and Zhang's research [6], they proved that the overall waiting time is minimized if all the charging demands are balanced for all charging stations [6]. In Sandford *et al.'s*, research [7], since load balancing was considered together with charging, a routing service was provided. Even though they gave availability of charging stations and reserved a time slot for charging at an available charging station, their idea is suitable for urban area. One of their important factors was traffic condition. In our paper, we consider tourist spots, the moving distance is quite long compared with urban cities.

For travel efficiency, an intelligent schedule that can minimize waiting time of charging without disrupting the travel plans is required. In the research proposed by Lee *et al.*, [4], battery charging was combined with tour schedule because most of the tourist spots in Jeju-do have EV chargers. Thus, according to tour schedule, we can easily predict which charging station will be used. They proposed an estimation model for waiting time for some specific visiting sequence. And then, they also an estimation model for finding the optimal schedule to minimize the charging waiting time. However, in their paper, there is no reservation recommendation function to select charging stations.

If we combine the algorithm for reservation recommendation with tour schedule generation, our tourists can move without disrupting the travel plans and being worried about their battery shortage. In summary, our algorithm is difference in a few respects. Firstly, our algorithm focuses the management of the time slots for reserving a specific charger. Next, the main goal of our algorithm lies in the disruption of tourists' travel but in the minimization of the waiting time of charging.

3. Reservation Recommendation Algorithms

In this section, we firstly describe how our reservation recommendation algorithms search a desirable set of charging stations based on distance and route. We assume that our reservation algorithms are on daily basis. For our reservation process, we require the current location, the remained distance to drive, the hopeful finishing charging time of an EV which wants to reserve a charger. Also, we require the desirable distance from the EV or the destination of the EV. And then, we require a preferred selection method to choose a charger. After a selection method is determined, the alternative charging stations are selected from our database. The first algorithm, Algorithm 1, is for describing the way how to find a set of chargers from our database according to distance and route. For efficiency, we use the locations of charging stations instead of those of chargers because there can be several chargers in a charging station.

Algorithm 1. To determine a set of possible charging stations for an EV from our database.

Step 1)

case 1) To search charging stations based on the shortest distance from an EV

- By using the locations (pairs of latitude and longitude) of charging stations stored in our database, we find out the alternatives within the default searching distance.

case 2) To search charging stations based on the preferred route the EV driver chooses

- By using the belonged routes of charging stations stored in our database, we find out the alternatives within the default searching distance.

Step 2) Commonly, we calculate the searching distance with the current location of the EV and the desirable distance from the EV or the destination of the EV.

Step 3) Also, we sort the alternatives with the gap between the EV and each of the alternative stations in an ascending order.

Next, after we get the alternative charging stations, we consider the chargers which belong to the selected charging stations. We assume that the number of the chargers is N. Then, in Algorithm 2, we calculate the arriving time, Ta_i, the charging time, Tc_i, and the slack time, Ts_i, for each charger i, where $1 \le i \le N$. For the Algorithm 2, we assume 5 variables have the values described in the following:

1. *default_searching_distance = 20km*
2. *default_driving_speed=60 km/hour (=1 km/minute)*
3. *the_maximum_distance_to_drive_with_one_charge = 90km*
4. *the_total_time_for_full_charging_with_normal_speed = 6 hours*
5. *the_total_time_for_full_charging_with_high_speed = 0.5 hours*

We also assume that the amount of battery consumption is constant according to a specific driving distance. And, we will get the charging time, Tc_i, for the normal speed because the calculation algorithm for the high speed is the same as for the normal speed. Then, we can define the maximum driving time with one full charge is 1 hour and a half (90/60=1.5 hours).

Algorithm 2. To calculate the arriving time, Ta_i, the charging time, Tc_i, and the slack time, Ts_i, of the ith charger, where $1 \le i \le N$.

Step 1) For each charger i,

1. The_required_distance_for_charging = RDc_i =
   A. 90 – *the remained distance to drive – the distance to get to the charger i,* when the driver wants to charge fully
   B. the input value from the driver
2. The charging time Tc_i = (90 - RDc_i) * 360 minutes /90 kms = RDc_i * 4 minutes/km

Step 2) For each charger i,

1. The_required_time_to_arrive = RTa_i
   = (the distance from the current location to the charging station i) * 4

minutes/km

2. Ta_i = now + RTa_i

Step 3) Let the hopeful finishing charging time be T_f. Then, for each charger i,

1. If $Tc_i > T_f - Ta_i$, $Ts_i = 0$
2. If $Tc_i \le T_f - Ta_i$, $Ts_i = T_f - Ta_i - Tc_i$

After we get the arriving time, Ta_i, the charging time, Tc_i, and the slack time, Ts_i, for each charging station i (where $1 \le i \le N$), we get the information about the idle time slot between Ta_i and T_f for each alternative charger. In our algorithm, we divide 24 hours into 144 time slots (6×24 slots), where we assume that the minimum charging unit for an EV is 10 minutes. Each charger has 144 time slots. Algorithm 3 describes the recommendation method for reserving an appropriate charger. After we find a charger, we can replace the charger with the charging station where the charger is located.

Algorithm 3. To list up the promising chargers.

Step 1) For each chargers, calculate the waiting time Tw_i and the actual charging time Tc_i with Ta_i and T_f by using the following Algorithm 4.

Step 2) Sort the alternative chargers in the ascending order of the Tw_i and Tc_i.

Step 3) Provide the list generated by Step 2. In the list, there are 2 different types of alternatives such as :

1. $Tw_i = \alpha$ and $Tc_i \le T_f - Ta_i$, where $\alpha \ge 0$.
2. $Tw_i = \beta$ and and $Tc_i > T_f - Ta_i$ and $Tlimited_i$ is available, where $Tlimited_i < Tc_i$ and $\beta \ge 0$.

So far, the proposed algorithms are described from users' point of view. From now on, our algorithm is described from chargers' point of view for time slot allocation management. From Algorithm 1 and Algorithm 2, a set of selectable chargers is determined. For the chargers we have to calculate the waiting time and the actual possible charging time. In order to do so, the time slot management for a charger reservation is required. As we already mentioned above, we divide 24 hours into 144 time slots. We make a formula (1) to define a time slot number for a specific time. For example, when the arriving time is 10:30 am, then the slot_number = 10 * 6 + 30/10 = 63. We also assume that all the data in the time slot table are some slot number between 0 and 143.

$$slot\_number = hour * 6 + minute/10 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots(1)$$

Algorithm 4 describes the management of the time slot allocation table for a charger reservation. Algorithm 4 and Algorithm 3 interact with each other to calculate the possible time slots efficiently for users. Our algorithm works on one-day-based reservation. At this point, we need a user ID who wants to reserve a specific charger newly. Let us a user ID be $user_i$. If there exists the intersection of time slots between any pairs of two users' requests, then one of the requests should be delayed or be served limitedly.

Algorithm 4. To manage the time slot allocation table of a charger

Step 1) Calculate the slot_number, s_i, for the arriving time of a $user_i$ by using the formula (1).

Step 2) Examine if there is any $user_j$ whose reserved starting time Tb_j and $Tb_j + Tc_j$ have the following conditions; (i) $Tb_j < s_i + Tc_i < Tb_j + Tc_j$ or

(ii) $Tb_j < s_i < Tb_j + Tc_j$ or

(iii) $s_i < Tb_j$ and $Tb_j + Tc_j < s_i + Tc_j$,

where the $user_j$s reserved the same charger as $user_i$. (* It means that some user's charging schedule intersects with $user_i$'s charging schedule. *)

1. If there is no user,
   1.1. Set the waiting time Tw_i to 0 (zero) and $full\_charging_i$ to true for $user_i$.
   1.2. Return the results ($Tb_i = s_i$, Tc_i, Tf_i, Tw_i, and $Ts_i = Tf_i - Tb_i - Tc_i$) to the $user_i$.
   1.3. Exit
2. Otherwise, go to Step 3. (* It means there are some users who already reserved the time $user_i$ wants. *)

Step 3) For each of $user_j$s who are selected in Step 2, repeat until there is no $user_j$ to examine.

1. $User_j$ satisfies the condition, (i) of Step 2.

   - If $(Tb_j + Tc_j) - s_i \leq Ts_i$, (* $user_i$ can wait until $user_j$ finishes his/her charging *)

   (1) Set $Tw_i = (Tb_j + Tc_j) - s_i$ and $full\_charging_i$ = true.
   (2) Keep the result, $Tb_i = Tb_j + Tc_j$, Tc_i, Tf_i, Tw_i, $Ts_i=Tf_i - Tb_i - Tc_i - Tw_i$ for $user_i$.
   (3) For the rest of the $user_j$s, go to Step 3.

   - If $(s_i + Tc_i) - Tb_j \leq Ts_j$ and $user_j$'s change does not affect any other users' reservations, (* $user_j$ can wait until $user_i$ finishes his/her charging *)

   (1) Set $Tw_j = (s_i + Tc_i) - Tb_i$ and $full\_charging_i$ = true.
   (2) Keep the result, $Tb_j=s_i+Tc_i$,Tc_j, Tf_j, $Ts_j = Tf_j -(s_i+Tc_i) - Tc_j$, for $user_j$.
   (3) Keep the result, $Tb_i = s_i$, Tc_i, Tf_i, Tw_i, $Ts_i=Tf_i - Tb_i - Tc_i$ for $user_i$.
   (4) For the rest of the $user_j$s, go to Step 3.

   - Otherwise, $full\_charging_i$ = false. Keep the result $Tb_i=s_i$,$Tc_i=Tc_i-(s_i+Tc_i-Tb_j)$,$Tf_i$,$Ts_i=0$ for $user_i$. And then, go to Step 3.

2. $User_j$ satisfies the conditions (ii) or (iii) of Step 2,

   - If $(Tb_j + Tc_j) - s_i \leq Ts_i$, (* $user_i$ can wait *)

   (1) Set $Tw_i = Tb_j + Tc_j - s_i$ and $full\_charging_i$ = true.
   (2) Keep the result, $Tb_i= s_i + Tw_i$, Tc_i, Tf_i, Tw_i, and $Ts_i= Tf_i-Tb_i-Tc_i-Tw_i$ for $user_i$.
   (3) For the rest of the $user_j$s, go to Step 3.

   - If $(s_i + Tc_i) - Tb_j < Ts_j$ for some $user_j$ and $user_j$'s change does not affect any other users' reservations (* $user_j$ can wait *)

   (1) Set $Tw_j = (s_i + Tc_i) - Tb_j$ and $full\_charging_i$ = true.
   (2) Keep the result, $Tb_j=Tb_j+Tw_j$,Tc_j,Tf_j,$Ts_j=Tf_j-Tb_j-Tc_j-Tw_j$, for $user_j$.
   (3) Keep the result, $Tb_i=s_i$,Tc_i,Tf_i, 0,$Ts_i=Tf_i-Tb_i-Tc_i$, for $user_i$.
   (4) For the rest of the $user_j$s, go to Step 3.

   - Otherwise,

   (1) $User_j$ satisfies the conditions (ii) of Step 2, $full\_charging_i$ = false. Keep the result $Tb_i=s_i$,$Tc_i=Tc_i-(Tb_j+Tc_j-s_i)$,$Tf_i$,$Ts_i$ for $user_i$. And then, go to Step 3.

   (2) $User_j$ satisfies the conditions (iii) of Step 2, $full\_charging_i$ = false. Keep the result

Step 4) Set $Tb_i=Tbi_L$ which means the last Tb_i among the Tb_is we get from the above procedures. Set Tc_i to the shortest Tc_i, Tci_s, among the Tc_is we get from the above procedure. Set the waiting time Tw_i to $Tbi_L - s_i$. If the original Tc_i equals to Tci_s, then $full\_charging_i$ = true, else $full\_charging_i$ = false. Return the values to $user_i$.

Finally, the user selects one among the recommended chargers which provides the best charging service. Then, the starting time for charging, the actual charging time, the slack time are store on the selected charger's time slot allocation table.

In summary, our time slot table allocation algorithm for reservation is similar to the existing reservation algorithms. For example, in the existing reservation algorithms for hotel reservation and airline reservation, since the request is accepted and arranged in the time slot allocation table, then the request cannot be aborted even though other requests have earlier deadlines. However, our algorithms are more flexible because charging schedules can be modified by schedulers without users' permission. The proposed algorithm is based on the slack time for each charging schedule, and do not permit preemption among user's requests.

4. Data Formats for the Reservations of Chargers

In this section, we describe how we design the data format for providing the services we want. Figure 1 shows the whole data structures for our algorithms. In the Figure 1 (b), the belonged route ID represents in which routes the charger is located. The route IDs can be multiple if the charger is located in a crossroads. The total number of routes in Jeju-do is 22. We assign each ID to each route. By using the route ID we can recommend charging stations near a specific route users want.

Charging station ID	Charging station name	Address	Location (altitude, longitude)	In service flag (1 : in_service, 0: not available)	Telno	The number of chargers	City No (1:Jeju, 2: Seogwipo)

(a) The data format for charging stations

Charger ID	Charging station ID	Battery Type	Speed (1:High 0: Low)	Belonged RouteID

(b) The data format for chargers

Charger ID	Charging station ID	userID	Starting time (slot number)	Deadline (slot number)	The number of charging time slots	The number of slack time slots

(c) The data format for time slot allocation management table for reservation

Figure 1. Database Design

In Figure 1 (c), the slot ID is ranged from 1 to 144. 1 means 00(hour):00(min), 2 means 00(hour):10(min), and 144 means 23(hour):50(min). If a desired charging time is 2 hours (120 minutes) and the starting time is 10:00, then the starting time slot is set to (10*6 + 1 + 0/10 = 61) and the finishing time slot is set to 61 + 120/10 (=73). For each charger, the starting time slot is unique.

Next, we describe an example how our time slot allocation table works on users' charger reservation requests in Figure 2. In the first request, since there is no reservation record, *phm* can reserve the charging station *052001-01* without waiting time.

In the second request, *khw*'s request does not conflict with any other requests because *phm*'s 60+4 = 64 < *khw*'s 70 (s_i). there is no intersection between the two requests.

In the third request, *ymb*'s request is conflict with *phm*'s request. Their situation belongs to Step 3. Thus, to *ymb*'s request can wait until *phm*'s charging is finished.

In the fourth request, *jsh*'s request is conflicted with *phm*'s, *khw*'s, and *ymb*'s requests. However, *jsh, khw*, and *phm* do not have enough slack time to delay their starting time. In case of *khw*, the number of time slot occurred by intersection is 1 (one). And *khw*'s number of slack time is 2. Thus, *khw* delayed one unit in the time slot allocation table. Anyhow, *jsh* cannot start at 66. Instead, jsh waits until *ymb* finishes the charging. Thus, *jsh* only charges 4 instead of 5.

Charger ID	Charging station ID	User ID	Starting time	Deadline	The number of charging time slots	The number of slack time slots
052001	01	phm	60	72	4	8

(a) 1st request

Charger ID	Charging station ID	User ID	Starting time slot	Deadline	The number of charging time slot	The number of slack time slots
052001	01	phm	60	72	4	8
052001	01	khw	70	84	12	2

(b) 2nd request

Charger ID	Charging station ID	User ID	Starting time slot	Deadline	The number of charging time slot	The number of slack time slots
052001	01	phm	60	72	4	8
052001	01	khw	70	84	12	2
052001	01	ymb	62	67	3	2
↓	↓	↓	↓	↓	↓	↓
052001	**01**	**ymb**	**64**	**67**	**3**	**0**

(c) 3rd request

Charger ID	Charging station ID	User ID	Starting time slot	Deadline	The number of charging time slot	The number of slack time slots
052001	01	phm	60	72	4	8
052001	01	khw	70	84	12	2
↓	↓	↓	↓	↓	↓	↓
052001	**01**	**khw**	**71**	**84**	**12**	**1**
052001	01	ymb	64	67	3	0
052001	01	jsh	66	71	5	0
↓	↓	↓	↓	↓	↓	↓
052001	**01**	**jsh**	**67**	**71**	**4**	**0**

(d) 4th request

Figure 2. An Example Scenario

5. Conclusions

In this paper, we proposed the reservation recommendation algorithms for charging EVs. We also provided two ways such as the shortest distance and the best preferred route to choose charging stations. And then, we calculated the arriving time, the charging time, the slack time, and the waiting time for each selected charging station. Finally, we recommended the chargers based on the shortest waiting time and the maximized charging time. In this paper, we assumed the one-day-based reservations and proposed the method that divided one day into 144 time slots.

In fact, unlike other reservation systems such as accommodation and rent-a-car, the charging schedules are adjustable according to their slack time. However, our algorithms are some limitations. Firstly, we do not consider the peak time for reservations. In other words, since our algorithms work well when users' requests occur evenly for 24 hours, if the users' requests occur heavily during some time period, then many of the requests cannot reserved for charging. Next, we do not consider the preemption of charging process. In fact, charging can be stopped for some reason. However, it can be resumed without any loss unlike other services such as hotel, restaurant, and airline reservations. Thus, if we consider the preemption of charging process, we can serve much more EVs than now. Also, we can use the real-time scheduling algorithms by using deadlines and slack times.

Apparently, the charging algorithms with preemption is different the basic real-time scheduling algorithms. If some users' reservation requests are already assigned some amount of time slots, they should never be aborted regardless of their deadlines. Thus, the preemption criteria and the abortion criteria should be different in our algorithms. In the near future, our reservation algorithms can be combined with the real-time scheduling algorithms for scheduling EV charging. Finally, we consider only one-day-based reservation. Thus, in the future, we should extend the possible reservation period.

Acknowledgements

This research was financially supported by the Ministry of Knowledge Economy (MKE), Korea Institute for Advancement of Technology (KIAT) through the Inter-ER Cooperation Projects.

References

[1] http://www.gsma.com/connectedliving/wp-content/uploads/2012/09/cl_jeju_09_121.pdf.
[2] http://smartgrid.jeju.go.kr/.
[3] http://www.jejuevservice.co.kr/index.php.
[4] J. Lee, H.-J. Kim and G.-L. Park, "Integration of battery charging to tour schedule generation for an EV-based rent-a-car business", Advances in Swarm Intelligence. Springer Berlin Heidelberg, **(2012)**, pp. 399-406.
[5] H. Momma, Y. Kawata, D. Hoki, S. Nagai, J. Kure and I. Takeda, "Apparatus, Method, and Computer Program for Reservation of Accommodation", U.S. Patent 0049414A1, **(2004)** March 11.
[6] H. Qin and W. Zhang, "Charging scheduling with minimal waiting in a network of electric vehicles and charging stations", Proceedings of the Eighth ACM international workshop on Vehicular inter-networking, ACM, **(2011)**, pp. 51-60.
[7] S. Bessler and J. Grønbæk, "Routing EV users towards an optimal charging plan", International Battery, Hybrid and Fuel Cell Electric Vehicle Symposium, **(2012)**.

Authors

Chan Jung Park, received B.S. at Dept. of Computer Science of Sogang University, Korea. She received M.S. and Ph.D. at KAIST and Sogang University, respectively. From 1990 to 1994, and also from 1998 to 1999, she worked for Korea Telecom as a researcher. Since 1999, she has worked for the Dept. of Computer Education at Jeju National University. Currently, she is a professor in Jeju National University. In 2010, he was a visiting scholar at University of California, Berkeley. Her research interests include creative problem solving, web app development, and data mining.

Junghoon Lee, received B.S., M.S., and Ph.D. at Dept. computer engineering, Seoul National University, Korea. From 1990 to 1992, and also in 1996, he was a senior research engineer at Lab. of optical telecommunication, Daewoo Telecom, Korea. In 1997, he joined the Dept. of computer science and statistics at Jeju National University, Korea where he is currently a professor. From 2003 to 2005 and 2009 again, he was a visiting scholar at Dept. of computer sciences, University of Texas at Austin. His research interests include real-time communication, wireless network, and smart grid.

Gyung Leen Park, received B.S. in Dept. of computer science from Chung-Ang University. He received M.S. and Ph.D. from computer science and engineering department at the University of Texas at Arlington, respectively. His research interests include scheduling in parallel and distributed systems, mobile computing, and telematics. In 1997, he was a visiting assistant professor at the University of Texas at Arlington. In 1998, he joined the Dept. of computer science and statistics at Jeju National University, Korea, where he is a professor. Currently, he is the director of the Smart Grid Research Center, Jeju National University.

Jung Suk Hyun, received Ph.D. at Sogang University, Korea. Since 2002, he has worked for the Dept. of MIS at Jeju National University. In 2011, he was a visiting scholar at University of California, Berkeley. His research interests include behavioral decision theory and creative problem solving. He received awards of the Korean Intellectual Property Office (2012), the Best Teacher Award (2011), the Award of Enhancing University Prestige of Jeju National University (2008), Best Professor Award of Achievement in Research of Jeju National University (2007).

