A Preemption-based Reservation Management Algorithm for Charging Electrical Vehicles

Hyun Mi Park [1], Moon Bo Yang [1], Hyunwoo Kang [1], Chan Jung Park [1], Seong Baeg Kim [1], Junghoon Lee [2], Gyung Leen Park [2]

[1.] Department of Computer Education, Jeju National University, Jeju-si, Jeju-do 690-756, Republic of Korea
[2.] Department of Computer Science and Statistics, Jeju National University, Jeju-si, Jeju-do 690-756, Republic of Korea
Phm2468@hanmail.net, didansqh@gmail.com, hyunwoo3447@naver.com, cjpark@jejunu.ac.kr, sbkim@jejunu.ac.kr, jhlee@jejunu.ac.kr, glpark@jejunu.ac.kr

Abstract: The number of charging stations is smaller than the number of electric vehicles (EV) now. Efficient scheduling algorithms for charging EVs are necessary in smart grid cities like Jeju Island in South Korea. Thus, there have been a lot of research works on charging scheduling methods so far. However, we have rarely found the system that can reserve chargers for EVs as well as manage the chargers for efficient charging. In this paper, we propose a preemption-based reservation algorithm for charging EVs. Especially, our algorithm considers the chargers at tourist spots in Jeju-do. In order to achieve our research objectives, we implemented an Android-based App for the tourists. With our App, the tourists can easily reserve the charging stations by recommending the better chargers in terms of the waiting time and the amount of charging. Finally, we evaluate the performance of the reservation allocation algorithm with execution time and the number of missing requests.
[Park HM, Yang MB, Kang H, Park CJ, Kim, SB, Lee J, Park GL. **A Preemption-based Reservation Management Algorithm for Charging Electric Vehicles**. *Life Sci J* 2014;11(7):694-699] (ISSN:1097-8135). http://www.lifesciencesite.com. 100

Keywords: Electric vehicle; charging; scheduling; smart grid; reservation; renewable energy

1. Introduction

Since Jeju-do became a smart grid city in 2009, many researchers in Korea have performed their research about the technologies related to smart grid. Among the research works, the technologies for electric vehicles (EVs) have attracted the researchers' attention especially in Jeju-do because Jeju Special Self-Governing Province has devoted huge amount of energy to EV proliferation that is one of the main themes in the smart grid city, Jeju-do, since 2012. Among the components of an EV, battery charging is the key component for operating the EV. Thus, various types of battery charging mechanisms have also attracted people's attention.

This paper focuses on a reservation system for charging EVs. There have been a lot of research works related to this topic (Gerding et al., 2011), (Clement et al., 2010), (Stein et al., 2012). Unlike the previous works, our paper developed an App for EV drivers to reserve a specific charger by recommending a set of the most desirable charging stations. Since the circumference of Jeju-do is approximately 253kms. Thus, if a tourist wants to drive the whole distance in a day with an EV, s/he has to charge his or her EV more than once. Also, we provided a web-based reservation management system for the drivers' reservations. As shown in Figure 1, we can find a few of applications for charging EVs. However, most of them are quite simple because they only provide the locations of all the EV charging stations in a designated city and the brief information about the EV charting stations.

In addition, unlike the existing reservation methods such as hotels and flights, our reservation algorithm permits preemption during service period and partial service. This means that for a restaurant, hotel, and rent-a-car reservation system, a customer's service does not allow any interruption and partial service. However, the property of electricity such as discontinuity allows service interruption and partial service in the aspect of the amount of the requested charging.

Figure 1. Electric vehicle charging information system provided by Korea Environment Corporation (https://evcis.or.kr)

Thus, we proposed a new algorithm for calculating the number of charging amount and the availability of each charger located in a certain distance from a driver. The proposed algorithm allows the preemption of charging. And then, we developed a reservation management system that can add, cancel, and update reservations of the customers. The functions of our system are shown in Figure 2. Finally, we evaluate our algorithms performance in terms of the execution time and the number of missing requests for a certain amount of charging requests.

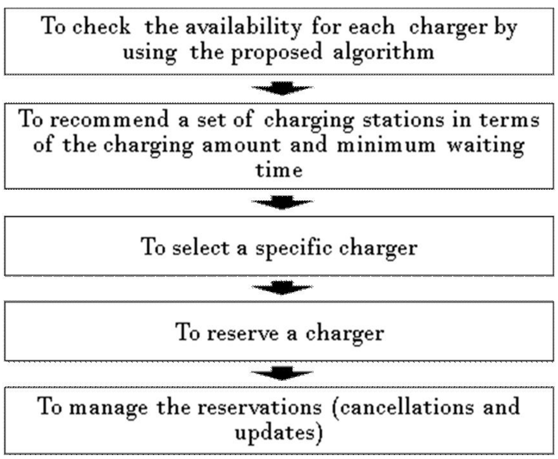


Figure 2. The functions and the sequence of our reservation management system

Our paper is composed as follows. In the following section 2, we present the assumptions of the paper and describe the flows how our system works with a sequence diagram. Also, we describe the reservation allocation algorithm briefly. In section 3, we show the implemented results of our App and the results of performance analysis of our allocation algorithm. Finally, we conclude our paper in section 4.

2. Assumptions and the Proposed Algorithm

In this section, we firstly describe our algorithms' assumptions, and then explain our proposed algorithms. We assume that if there is no previous reserved request, then a newly arrived charging request start at its starting time immediately.

For our system, we also assumed that

- our reservation algorithms work well within one day.
- the default searching distance = 20kms
- the driving speed=60 km/hour
- the maximum distance to drive with one full charging = 90kms
- the total time for full charging with normal speed = 6 hours
- the total time for full charging with high speed = 0.5 hours

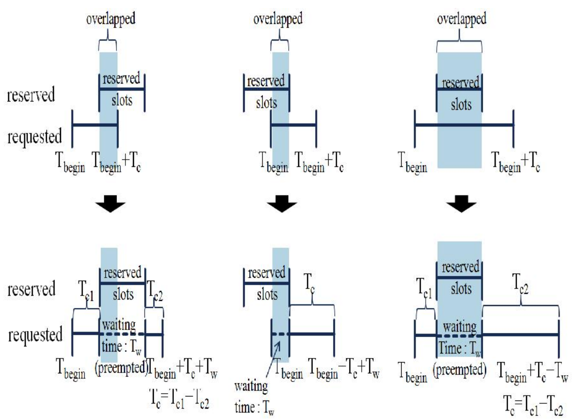
Next, we describe our reservation allocation algorithm in Algorithm 1. For the reservation, we divide 24 hours into 144 time slots, which means that we assume that the minimum charging time for an EV is 10 minutes. Then, we can calculate a specific time into the corresponding time slot number. The information about the idle time slot is maintained by the database for each charger.

Step1) To find a set of reserved requests' allocation information, which are over-lapped with the newly arrived request for charging.
Step2) If there is no reserved user's request, then announce that this charger i can be allocated to the newly arrived user's request, full charging is possible, and the waiting time is zero.
Step 3) If there exist some overlapped reservat-ion request, calculate the number of the idle slots between the arrival time and the finishing time of the newly arrived user's request for each charger.
Step 4) Also, it calculates the waiting time of the newly arrived user's request.
Step 5) To calculate the possible amount of the newly arrived user's charging request by using the number of available time slots. At this point, we divide the new request's charging time into several time periods.
Step 6) To return the results to the user.

Algorithm 1. Reservation allocation algorithm

For the recommendation process, let N_t be the total number of the idle time slots between the arrival time and the deadline of a driver's request. And, let N_r be the requested number of time slots for charging of the driver. Let T_f be the deadline of the driver's request. Let T_{ai}, T_{ci}, and T_{wi} be the arrival time, the charging time, and the waiting time of the *i*th charger respectively. The following process will occur for the alternative chargers that are located within a default searching distance. For the *i*th charger, if $N_t \geq N_r$, then the driver can charge fully. However, if $N_t < N_r$, then we can calculate the amount of charging by using T_{ai}, T_{wi}, and T_f. We can get the waiting time of the *i*th charger from Algorithm 1. After we get the waiting times and the amount of charging for all the alternative chargers, we recommend the chargers based on the combination value of the biggest charging amount and the shortest distance time to the driver. In other words, unlike the existing recommendation methods of navigation systems, in our algorithm we rely not only on the shortest distance but also other factors such as the biggest charging amount and the smallest waiting time at the same time.

Figure 3 describes the 3 possible situations in detail when two charging requests have some overlapped period in their service time. Figure 3 (a) represents that a newly arrive request requires an earlier starting time. But it requires a certain amount of the reserved time slots at the end of its charging time. In this case, the new request should be preempted during the period of the reserved request's charging time. If the deadline of the new request is long enough to charge, then it can restart charging after the reserved request finishes. Figure 3 (b) shows that a newly arrived request wants to charge before the reserved request finishes. In this case, the new request waits until the reserved request finishes charging if the deadline of the new request is long enough to wait. Figure 3 (c) is the combination of Figure 3 (a) and Figure 3 (b). According to the deadline of the new request, the amount of charging is determined.



(a) case 1 (b) case 2 (c) case 3

Figure 3. Three overlapped cases between a new charging request and the reserved ones

In Figure 4, we explain our algorithm with an example use. In Figure 4 (a), the user, *phm*, requests a charging from 60(10:00 am) for 40 minutes. The deadline is 72(12:00 pm) and its slack time is 80 minutes (8 time slots). Since there is no previous reserved request, *phm* can charge the full amount without any waiting time. We assume that *phm* selects this charger, 052001. In Figure 4 (b), the user, *khw*, requests a charging from 70(11:40 am) for 2 hours. Since *phm*'s charging finishes at 64 (10:40 am), there is no conflict between two requests. In Figure 4 (c), the next driver, *ymb*, requests a charging from 62(10:20am) for 30 minutes. At this time, due to the *phm*'s reserved request (charging time : 60 ~ 64), *ymb* should wait for 20 minutes until *phm* finishes (Figure 3 (b) case).

Finally, the user, *cjp*, requests a charging from 56(09:20am) for 80 minutes. At this time, *cjp* can start charging at 56. However, due to *phm*'s request, the new request temporarily stop at 60. After 64, since *ymb*'s request reserved from 64 to 67, *cjp* should wait until 67. The *cjp*'s request resumes the charging at 67, but it should be finished before 70 due to *khw*'s request. As a result, *cjp* waits for 70 minutes and charges for 70 minutes (partial charging) if *cjp* selects the charger 052001.

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

(a) First request

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

(b) Second request

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

(c) Third request

Charger ID	User ID	Starting time slot	Deadline	The number of charging time slot	The number of slack time slots
052001	phm	60	72	4	8
052001	khw	70	84	12	2
052001	ymb	64	67	3	0
052001	cjp	56	72	8	8
↓	↓	↓	↓	↓	↓
052001	cjp	56	60	4	0
052001	cjp	67	70	3	0

(d) Fourth request

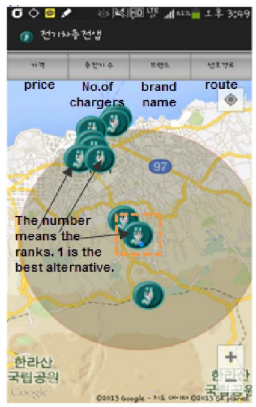
Figure 4. An example use of our time slot allocation algorithm

3. Implementation Results and Performance Evaluation

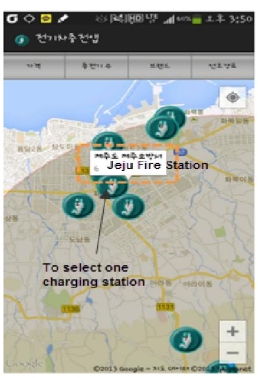
In this section, we firstly describe the component of our proposed system. Next, we also show the figures how our App works for EV reservations. Finally, we present the result of the performance evaluation in terms of the number of processed requests and the number of missed requests.

Our system is divided into two parts. The first part is the guidance and the reservation App of EV charging. The second part is the server for managing reservation information for charging stations. The first part consists of 3 sub-parts: (i) to get drivers' information , (ii) to show the map that contains the recommending charging stations obtained from Algorithm 1, and (iii) to request, to retrieve, or to cancel reservations. When we show the map, drivers can choose three alternatives such as the map based on distance, the map based on the brand name of battery, and the map based on the charging cost. The second part consists of 3 parts: (i) to select the set of alternative, (ii) to recommend the chargers selected in Algorithm 1, and (iii) to manage reservations.

As shown in Figure 5, when a driver inputs the remained amount of battery, and the charging amount he or she wants (Figure 5 (a)), the App shows the locations of the best charging stations the driver can charge as much as and as cheap as the driver wants within the default distance (Figure 5 (b)). When the driver clicks one of the charging stations (Figure 5 (c)), he or she can get the information about the charging station such as distance, charging starting time, possible charging time, and so on (Figure 5 (d)).



(a) To search charging stations (b) Alternatives



(c)To select one (d) The result information of (c)

Figure 5. Electric vehicle charging guidance and reservation App

The driver can reserve one charging station by clicking the button showed in the dashed box. Then, the driver can see the list of reservation status and withdraw the reservation he or she did. Besides, these figures, drivers can retrieve their reservation information as well as cancel their reservation requests. Our development environment is as follows:

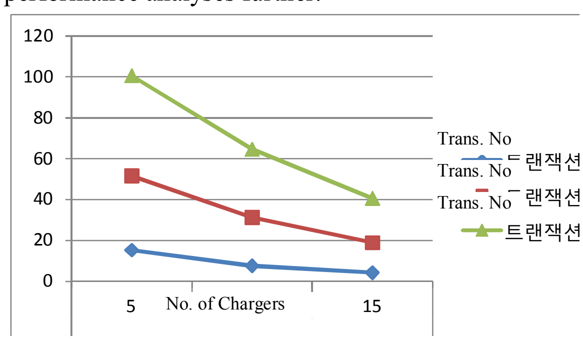
- Programming language : Java, JSP
- Web server : Apache Tomcat
- App : Android
- Toos : Eclipse

Next, we evaluate the performance of our reservation allocation algorithm. Table 1 shows the parameters of the simulation environment.

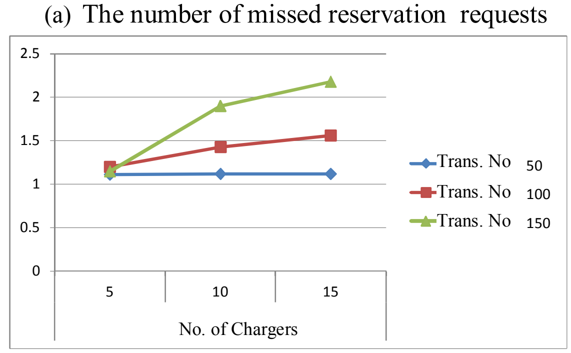
Table 1. Environment for performance test

Parameters	Value
OS	Windows 7
Processor	Intel® Core™i5-2500 CPU 3.3GHz
Ram	4GB
No. of Transactions	50 ~ 150
No. of ports per charger	1
No. of charging stations	5 ~ 15
No. of time slots	144

Figure 6 (a) shows the number of missed reservation requests and Figure 6 (b) show the total execution time for all transactions. When the number of chargers increases, the number of missed requests rapidly decreases. Next, due to the number of missed requests decreases by increasing the number of chargers, the total execution time increases. And, the time complexity of our algorithm is O(n). This experiment was performed in (Yang et al., 2014). However, based on the experiment, we expanded our performance analyses further.



(a) The number of missed reservation requests

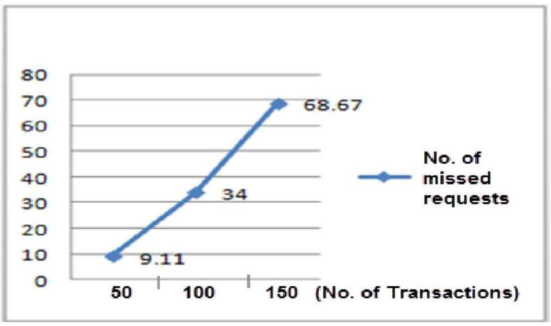


(b) The total execution time for all transactions

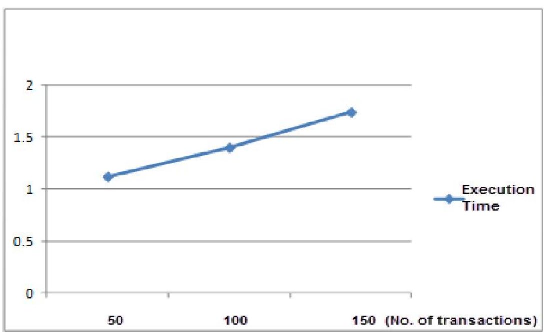
Figure 6. Performance Evaluation I

Next, we should consider the peak time people usually want to charge. Also, we should develop more efficient charging time allocation algorithm for the later reservation requests. Currently, the later reservation requests have higher possibility to be preempted than the earlier requests.

Figure 7 shows the average number of the missed requests (transactions) and the average execution time of a given number of requests (transactions). We repeated the experiment three times for each case. Figure 7 (a) represents when the number of requests increases, the number of missed requests also increases linearly. Figure 7 (b) represents when the number of requests increases, the execution time also increases linearly.



(a)The average no. of missed requests



(b)The total execution time

Figure 7. Performance Evaluation II

4. Discussions

There have been a lot of research works related to EV charging scheduling methods so far (Qin and Zhang, 2011)(Gan et al., 2011)(Lee et al., 2011). In addition, many of the research works have not considered tourist places but big cities. Also, we rarely found the applications for EV drivers. Many of them are for charging stations. The existing apps for EVs are quite simple.

In this paper, we developed an Android-based App to reserve EV chargers for drivers. Unlike the previous services, our App did not only consider the distance from an EV to a specific charger but also give the alternatives the EV driver can choose the best charging station with the best amount of battery and the cheapest charging cost with preemptions. Also, our system maintained the information all chargers at Jeju-do in fact and it can show the status of chargers drivers want to check.

However, we must extend our algorithm in many respects in the near future. Firstly, we should consider the multiple charging ports per chargers. Secondly, we use the Euclidian distance when we calculate the distance from the current location of a driver to the destination the driver wants to get. Thirdly, we excluded the various types of parameters that can affect the battery consumption in real environment such as temperature, altitude, road shape and so on. We should develop more sophisticated algorithm to calculate the distance and the time that an EV can get to a specific charging station. Next, our reservation time allocation algorithm is very straightforward, and works well within one-day reservation systems. In our algorithm, we did not consider the delay of the reserved requests. We only consider the preemption of the newly arrived request. Also, we should consider many aspects in the near future.

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.

Corresponding Author:
Processor Chan Jung Park
Department of Computer Education
Jeju National University
Jeju-si, Jeju-do, 690-756, Republic of Korea
E-mail: cjpark@jejunu.ac.kr

References

1. Clement-Nyns K, Haesen E, Driesen, J. The impact of charging plug-in hybrid electric vehicles on a residential distribution grid. IEEE Transactions on Power Systems, 2010; 25(1): 371-80.
2. Gan L, Topcu U, Low S. Optimal decentralized protocol for electric vehicle charging. Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC), 2011; 5798-5804.
3. Gerding EH, Robu V, Stein S, Parkes DC, Rogers A, Jennings NR. Online mechanism design for electric vehicle charging. Proceedings of the 10th International Conference on Auto- nomous

Agents and Multiagent Systems-Volume 2, 2011; 811-18.
4. Lee J, Park GL, Kim HJ, Jeon H. Fast scheduling policy for electric vehicle charging stations in smart transportation. Proceedings of the 2011 ACM Symposium on Research in Applied Computation. ACM, 2011; 110-112.
5. Qin H, Zhang W. 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; 51-60.
6. Stein S, Gerding E, Robu V, Jennings NR. A model-based online mechanism with pre-commitment and its application to electric vehicle charging. Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 2, 2012; 669-76.
7. Yang MB, Kang H, Park HM, Park CJ, Kim SB, Lee J, Park GL. Development of a Preemption-based Reservation Management System for Electric Vehicle Charging. Proceedings of the 2014 FTRA International Conference on Advanced Computing and Services. FTRA, 2014; 39-41.
8. https://evcis.or.kr.

3/10/2014