Design of a Multi-day Tour-and-Charging Scheduler for Electric Vehicles*

Junghoon Lee and Gyung-Leen Park**

Dept. of Computer Science and Statistics,
Jeju National University, Republic of Korea
{jhlee,glpark}@jejunu.ac.kr

Abstract. Aiming at alleviating the range anxiety problem in electric vehicles taking advantage of well-developed computational intelligence, this paper designs a multi-day tour-and-charging scheduler and measures its performance. Some tour spots have charging facilities for the vehicle battery to be charged during the tour. Our scheduler finds a multi-day visiting sequence, permitting different day-by-day start and end points. To exploit genetic algorithms for the extremely vast search space, a feasible schedule is encoded to an integer-valued vector having (n+m-1) elements, where n is the number of places to visit and m is the number of tour days. The cost function evaluates the waiting time, namely, the time amount the tourist must wait for the battery to be charged enough to reach the next place. It also integrates the time budget constraint and quantizes the tour length. The performance measurement result obtained from a prototype implementation shows that our scheme achieves 100 % schedulability until 13 places for the 2-day trip and 17 places for the 3-day trip on given parameter setting.

Keywords: Electric vehicle, multi-day trip, tour-and-charging schedule, genetic algorithm, waiting time.

1 Introduction

Energy efficiency is the most important keyword in the future power grid [1]. According to the development and penetration of electric vehicles, or EVs in short, the smart grid extends its coverage to the transport system [2]. Not just efficient power consumption, EVs have many environmental benefits over their counterparts, namely, gasoline-powered vehicles, as it is not necessary to burn fossil fuels. Hence, many modern cities are trying to accelerate the large deployment of EVs and are building city-wide charging facilities. However, in spite of recent improvement in battery technologies, it is not yet quite sure that the cost

* 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.

S. Ramanna et al. (Eds.): MIWAI 2013, LNCS 8271, pp. 108–118, 2013.

advantage in operating EVs outweighs their main drawback stemmed from range anxiety. If a user wants to drive more than the battery capacity, the EV must be charged somewhere on the route. It may take a few hours with slow chargers. Moreover, battery switch systems for EVs are rarely commercialized yet.

In the mean time, EV-based rent-a-car services are now appearing, not just in the form of a short-term hourly sharing model but also a long-term multi-day rental [3]. As their driving distance is highly likely to exceed the driving range of a fully-charged EV, drivers are facing the problem of deciding when and where to charge their EVs during their trips. Here, the visiting sequence is important, as drivers must wait and waste their time when they want to go to the next place but current battery remaining is not enough [4]. Moreover, in some places such as shopping malls and tour spots, EVs can be charged in parking areas while the drivers are taking their desired activities. How much an EV can fill electricity definitely depends on the stay time at a place. In case the number of visit places increases and charging facilities are limitedly available, this problem gets even more complex, and it is necessary to exploit sophisticated computer algorithms, mainly in artificial intelligence domains.

Generally, tourists select the set of tour places they want to visit while planning their tours. From tourist information, the stay time and the availability of chargers can be retrieved. Even if the scheduling problem is one of the most popular applications in computer sciences, it must be adapted for problem-specific constraints and scheduling goals. For example, the daily tour time, that is, the sum of driving and stay time at all places, is limited, while the waiting time must be kept as small as possible. Moreover, for multi-day trips, its search space grows too much, making it necessary to exploit suboptimal techniques such as genetic algorithms. Here, the modern well-organized communication infrastructure provides an efficient channel for such intelligent services to be delivered even to mobile users [5]. Moreover, the seamless interaction between many different objects over the Internet makes it possible for users easily specify their requirement and receive map-enriched information on diverse types of devices.

In this regard, this paper designs a multi-day tour-and-charging scheduler for EVs, aiming at reducing the inconvenience brought by their long charging time and short driving range [1]. Our research team has been developing a tour scheduler, mainly focusing on single day trips taking EVs. This effort includes waiting time estimation, genetic scheduler design, and orienteering problem modeling. The genetic scheduler will be extended for multi-day trips and its performance will be evaluated to check if it can be practically applied for an information service on EV-based rent-a-car systems. To this end, our scheme encodes each multi-day tour schedule to an integer vector, in which negative numbers are inserted to separate each day schedule. In addition, the relevant constraints are investigated and the corresponding fitness function is defined. The scheduling goal is to reduce the waiting time, for which tourist wait until their EV gets enough power to reach the next place along the route.

The rest of this paper is organized as follows: Section 2 reviews some related works. Section 3 describes the proposed scheme in detail, focusing on how to encode a schedule for multi-day tour to apply genetic operations. After Section 4 demonstrates and discusses the performance measurement results, Section 5 concludes this paper with a brief introduction of future work.

2 Related Work

As for an example of a multi-day trip scheduler, the City Trip Planner creates personalized tour routes and is currently providing services in world's most famous cities including New York [6]. Even though this service is not developed for EVs, it models tour scheduling as a team orienteering problem with time window. For a given set of user-selected spots, tour time at each spot, and tour length constraint, this service tries to maximize the sum of scores gained by visiting tour spots. The authors design a fast local search heuristic to respond to the user request within a reasonable time bound [7]. The search procedure iterates shake and insertion steps, removing and inserting some tour places, based on the estimated time window the tour spot can be placed. It automatically inserts those tour spots not selected but thought to be preferred by a tourist. It will be advantageous to insert the spots having charging facilities in the tour schedule generation for EVs.

Multi-day scheduling is quite similar to the multiple Traveling Salesman Problem (mTSP) in that each personal schedule can be mapped to a daily tour schedule [8]. In multi-day scheduling, the tourists do not have to return to the starting place. It has much more time complexity than the classic TSP, as the number of places to visit is much larger. Intuitively, to cope with this problem, a decomposition approach may cluster the places to visit into several groups [9]. In addition, it can be solved by exact solutions, heuristics, and transportations. For example, [10] first represents a visiting sequence by a chromosome having negative numbers (pseudo cities) to separate day-by-day schedules and runs evolutionary searches to achieve multi-objective goals. This encoding scheme will be exploited by this paper. In addition, mTSP is applied for vehicle routing problems which embrace multiple vehicles [11]. Here, vehicle specific constraints can be integrated in heuristic-based searches such as the Tabu scheme.

As for charging reservation over the route, [12] proposes a charging scheduling scheme with minimal waiting time on the large-scale network consisting of large number of charging stations cooperating with each other. The authors first define the waiting time as the sum of queuing time and charging time. Based on the theoretical analysis and observation that *the overall waiting time is minimized if charging demand of all charging stations is balanced*, a load distribution method is designed. Here, an EV can issue the next charging reservation request periodically. The EV-issued reservation request is forwarded to the charging stations within the driving range along the chain of the ad-hoc style communication network. Each station estimates the waiting time for the request and the result is sent backward to the issuer. The best one is selected and confirmed to both the EV and the charging station. As contrast to this model which just considers the next reservation, our scheme can build an entire tour schedule and possibly make a reservation.

In [13], authors have developed a waiting time estimation model for EV-based multi-destination tours. A genetic scheduler finds a visiting sequence having smaller waiting time just as the classical TSP reduces the tour distance, considering the charging facility availability, stay time, inter-destination distance. Moreover, to further reduce the waiting time, system-recommended tour spots, usually equipped with chargers, are added in the tour schedule [14]. This scheduler is a hybrid version of the ordinary TSP for user-selected spots and the selective TSP, or orienteering problem, for system-recommended spots. A chromosome includes all candidate recommended spots, while some of them will be deactivated with a tunable omission degree to exclude from the visit schedule. The scheduling goal is to reduce the waiting time and include as many spots as possible.

3 Tour Scheduler

3.1 Overview

Let $S = \{S_1, S_2, ..., S_n\}$ be the set of tour spots an EV driver wants to visit during the whole trip. They are all different, have their own stay time, and must be visited just once. The stay time will be affected by many factors such as user preference, weather, and the like. Considering the scheduling goal of reducing the waiting time, stay time will be set to the reasonable lower bound of the entire distribution. The stay time at S_i will be denoted by $T(S_i)$. In addition, we assume that the start and end points are given for every day, namely, $H = \{(H_i^s, H_i^e)\}$ for $1 \leq i \leq m$, where m is the number of tour days. Generally, $H_i^e = H_{i+1}^s$, as a tour team stays at a hotel and starts the next day trip from the same hotel. The start point of the first tour day, namely, H_1^s, and the end point of the last tour day, namely, H_m^e, will be pick-up and return stations of an EV rental service, respectively. Based on the road network, the distance between every pair of two spots is given for tour scheduling and $Dist(S_i, S_j)$ represents the distance between S_i and S_j.

The genetic algorithm is one of the most widely-used suboptimal search techniques, built upon the principle of natural selection and evolution [15]. Beginning from initial population consisting of given number of feasible solutions, genetic iterations improve the fitness of solutions generation by generation. Each generation is created from its parent generation by applying genetic operators such as selection, crossover, and mutation. To apply those operators, it is necessary to represent a feasible solution as an integer-valued vector. For example, the crossover operator swaps substrings from two parents, Hence, the first step of the multi-day tour-and-charging scheduler design is to encode a multi-day schedule. Basically, n user-selected tour spots are included in the schedule. Then, if the number of tour days is m, (m-1) different negative numbers are added, creating a vector of (n+m-1) discrete elements [10]. After all, a multi-day schedule can be handled just like a single-day schedule.

Figure 1 shows an example which builds a 3-day schedule for 9 places, namely, from S_1 to S_9. Hence, n is 9 and m is 3. This example also includes a chromosome for a visiting schedule, in which each daily schedule is separated by -1 and -2, resulting in 3 groups. Even if -2 comes first, it doesn't matter, as negative numbers are used just for separation of each daily schedule. Hence, the total length of a chromosome is 11. Each day has 3 places to visit, while the total trip begins and ends at rent-a-car station in the airport. The tourists stay at a hotel at the first and second days, respectively. The evaluation step calculates the predefined cost, such as the waiting time and total tour length, for each subschedule and hotel specification.

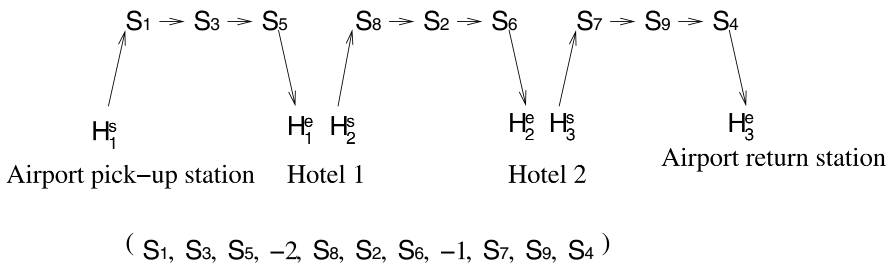


Fig. 1. Effect of genetic iterations

3.2 Fitness Function

A fitness function, or cost function evaluates the quality of a schedule. The schedules in the population are sorted by fitness values. Straightforwardly, the higher the cost, the lower the fitness. Hence, they can be used interchangeably. To estimate the waiting time, our cost function defines two primitive operations, namely, *Move* and *Stay*. Along a tour route, two operations are invoked in turn. Here, both the waiting time and the tour time of a schedule, globally defined as W and L_i ($1 \leq i \leq n$), are estimated spot by spot. First, the *Move* operation accounts for driving between two spots. If battery remaining is enough, this operation just decreases battery remaining without changing the waiting time. Otherwise, the lacking battery amount and then corresponding charging time are calculated. The waiting time increases by the charging time. The tour time also increases by the charging time in addition to the driving time.

Next, the *Stay* operation traces battery charging at a tour spot. If a tour spot has no chargers, battery remaining does not increase. Otherwise, it linearly increases according to the stay time. Here, any battery charge and discharge model can be employed [16]. However, it should not exceed the maximum capacity, B_{Max}. Hence, the visiting sequence had better avoid two or more consequent spots having chargers and long stay time. For an example of 2-day trip specified by $\{(H_1^s, H_1^e), (H_2^s, H_2^e)\}$, and $\{S_1, S_2, -1, S_3, S_4\}$, the waiting time can be calculated by sequentially invoking $Move(Dist(H_1^s, S_1))$, $Stay(T(S_1))$, $Move(Dist(S_1, S_2))$, $Stay(T(S_2))$, and $Move(Dist(S_2, H_1^e))$ for the first day while $Move(Dist(H_2^s, S_3))$, $Stay(T(S_3))$, $Move(Dist(S_3, S_4))$, $Stay(T(S_4))$, and $Move(Dist(S_4, H_2^e))$ for the second day. The total waiting time is the sum of those two. At the same time, per-day tour length, L_i, and total tour length, $\sum L_i$, are calculated for the evaluation of fitness or cost of a tour schedule.

Users can set an upper bound on the daily tour length. Hence, each daily tour time is compared with this time budget, and if it is larger than the budget, the schedule is not valid. However, it is not discarded in the population as it can contribute to the improvement of fitness by reproduction. Instead, its cost function gives the largest value to it. Next, the waiting time is most critical to the user-side convenience, and it must be kept as small as possible. Besides, we can assign the tour length of each day as evenly as possible to make room for another activities or rest. Otherwise, we can also pack the tour schedule to as small number of tour days as possible, making a large free time chunk. It is hard to decide which one is better, but either requirement can be taken into account in the cost function. For the first case, the longest tour length is counted in the cost function. After all, the cost function F for a visiting sequence, x, is defined as follows:

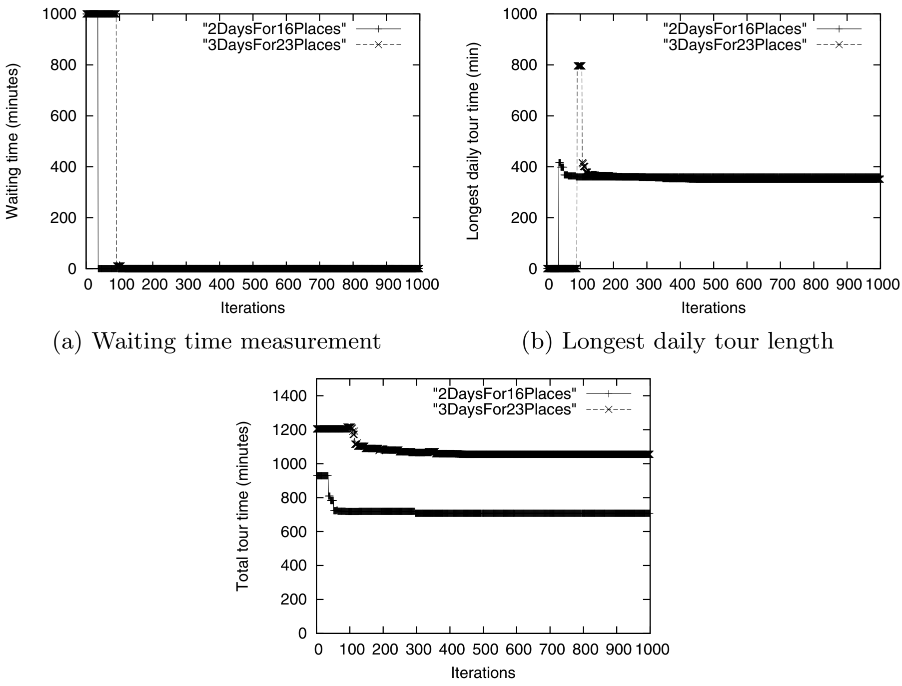
$$F(x) = \begin{cases} \infty & \text{for } \exists i \; L_i \geq T_b, \\ W \times 1000.0 + Max(L_i) & \text{for } 1 \leq \; i \; \leq \; m. \end{cases}$$

where W is the waiting time and T_b is the time budget.

4 Experiment Result

This section measures the performance of our tour scheduling scheme via a prototype implementation using Microsoft Visual Studio 2012. It employs the Roulette wheel selection method and randomly sets the initial population of chromosomes, or feasible solutions encoded by an integer vector. For better population diversity, it does not permit duplicated chromosomes in the contemporary population. As contrast, $(S_1, -1, S_2)$ and $(S_2, -1, S_1)$ are logically equivalent, having the same cost value. However, they are allowed to coexist in the population, as they can generate different offsprings. The inter-spot distance exponentially distributes with the average of 15.0 km. No two spots are separated by more than the driving range, which is the distance reachable with full battery capacity. In addition, the stay time also distributes exponentially with the average of 30 minutes, with its maximum limited to 2 hours.

The first experiment measures the effect of genetic iterations on performance metrics such as waiting time, longest daily tour time, and total tour time. As for genetic parameters, the population size is set to 32 and the number of iterations to 1,000. Its execution time is less than 1 second, regardless of the number of places to visit, for practical use. We select two sets of user-selected tour places, one for 16 places in 2 days and the other for 23 places in 3 days. As shown in Figure 2(a), the two cases begin from the cost of 1,000, which means no valid schedule is included in the initial population. Then, the 2-day case finds a feasible



(a) Waiting time measurement (b) Longest daily tour length

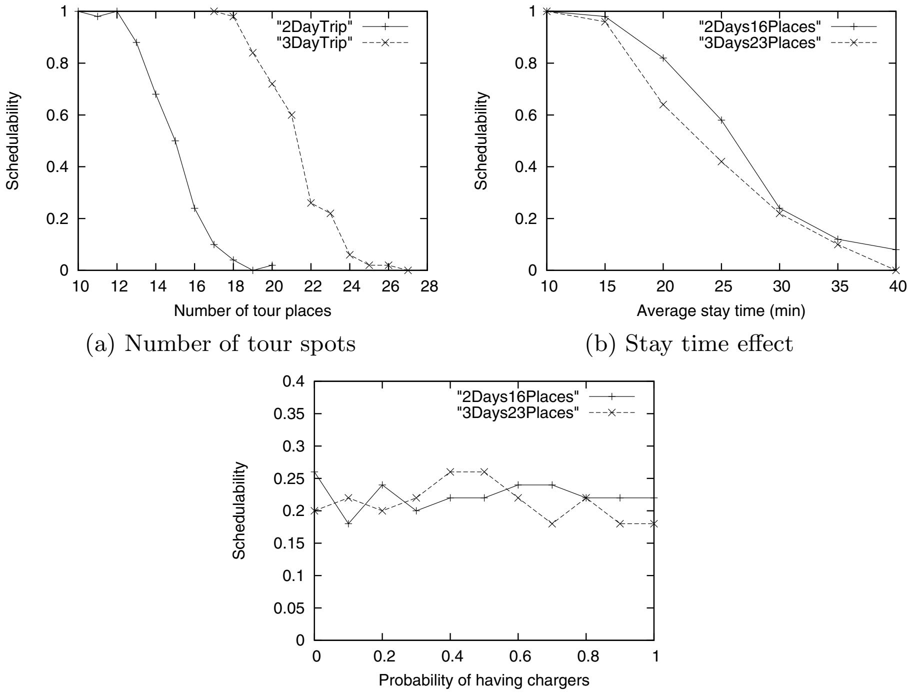
(c) Total tour length

Fig. 2. Effect of genetic iterations

solution after about 36 iterations, while the 3-day case after 106 iterations. For the 3-day case, the cost remains 10.0 temporarily, but goes down to 0 after 106 iterations. This result indicates that the time budget constraint is more critical to the cost function.

Next, Figure 2(b) plots the longest daily tour time for the two cases. Here, the value of 0 is meaningless as it takes place when there is no valid schedule. After the valid schedule is found, the daily tour time has non-zero values and then converges to a stable value. For the 2-day case, the longest daily tour becomes 416 minutes and then converges to 396 minutes. In addition, for the 3-day case it temporarily becomes 800 minutes and then converges to 351 minutes. For the case having larger number of places to visit, the longest daily tour time changes more often, or have more descending stages. Figure 2(c) plots the change in total tour length for above two cases. The total tour time in the 23-place case is definitely longer than that in the 16-place case. The initial value of the total tour time is 929 and 1,204 minutes, respectively. Then, each of them drops sharply when feasible schedules are found. After those points, the total tour length shows just minor improvement. During this iteration interval, the waiting time has already touched 0, and no more improvement is expected for it.

Now, the second experiment measures the schedulability of the proposed tour-and-charging scheduler according to the number of tour spots, average stay time, and the availability of chargers at each tour spot. The schedulability is the ratio of successfully finding a schedule having no waiting time out of 50 sets. Figure 3(a) plots the schedulability obtained by changing the number of tour spots from 10 to 20 for the 2-day trip and from 17 to 27 for the 3-day trip, respectively. For the 2-day case, up to 12 places, all selection sets are schedulable, and the schedulability begins to drop from 13 places, and touches zero on 19 places. Even 1 set is scheduled when the number of places is 20, it can be disregarded. Likewise, for the 3-day trip, all sets can be scheduled up to 17 places and no set can create a feasible schedule on 27 places.



(a) Number of tour spots

(b) Stay time effect

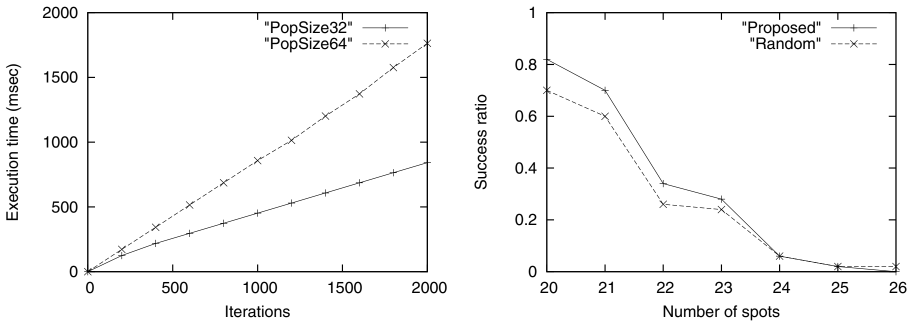
(c) probability of having chargers

Fig. 3. Schedulablity measurement

Next, Figure 3(b) shows the effect of the average stay time to schedulability. With larger stay time, EVs can get more electricity during the trip. However, it makes both the total tour length and the longest daily tour too long. Hence, according to the increase in the average stay time, the schedulability gets poorer. Particularly, when the average stay time is 40 minutes, no set can be scheduled for the 2-day trip and just 5 can be scheduled for the 3-day trip. Finally, Figure

3(c) measures the effect of the availability of chargers. At first, we expect that schedulability will be enhanced with more chargers. However, as the time budget constraint dominates in tour set scheduling, its effect is not so significant. This parameter setting makes the schedulability fluctuate between 0.17 and 0.25, no one outperforming the other.

Additionally, Figure 4(a) plots the actual execution time of the implemented scheduler on an average performance PC, which consists of 2.4 GHz Intel(R) Core(TM)2 Duo CPU and 3 GM main memory, running Windows Vista operating system. The experiment takes a tour of 24 spots for 3 days. The population size is set to 32 and 64. The effect of population size to accuracy is already measured in Figure 2, so this experiment focuses on the execution speed according to the progress of genetic loops. The execution time of a genetic algorithm mainly depends on population size and the number of iteration. As shown in Figure 4(a), even for the population size of 64, the response time is less than 1.8 second. The common genetic operator overhead makes the difference between two cases 31.4 200 iterations. However, after 2,000 iterations, the increase of the population size by 2 lead to the increase in the execution time by 2.1 times.



(a) Execution time (b) Comparison with random scheme

Fig. 4. Additional experiment

Next, we compare the proposed scheduler with a random scheduling scheme which randomly generates schedules and selects the best one for the almost same time duration needed for our scheduler. This scheme has no control policy, but is quite efficient in large search space and gives us good reference. In this experiment, 3-day tour is assumed and the population size is set to 32. Figure 4(b) plots the success ratio for both schemes and shows that the proposed scheme outperforms by 12 % for the case of 20 tour spots. According to the increase in the number of selected spots, the difference gets smaller, as both schemes can hardly find a feasible schedule.

5 Conclusions

The smart grid is extending its coverage area to the transport system for better energy efficiency, particularly making an effort to prompt the penetration of EVs into our daily lives. They are suffering from short driving range, long charging time, and insufficient charging infrastructure. Those problems can be much serious when the trip length exceeds the driving range and the tour lasts for more than one day. It can be efficiently relieved by intelligent information technologies commonly available in these days. Accordingly, in this paper, we have designed a tour-and-charging scheduler for EVs, deciding when and where to charge EVs as well as the visiting sequence having the acceptable waiting time. Suboptimal techniques are unavoidably exploited to cope with large search space resulting from the extended number of places to visit for multi-day tours.

Our scheduler is developed based on the genetic algorithm, one of the most widely used suboptimal search schemes. To exploit genetic algorithms, a feasible schedule is encoded to an integer-valued vector having (n+m-1) elements, where n is the number of places to visit and m is the number of tour days. m-1 negative number are inserted to separate each day schedule. The cost function mainly evaluates the waiting time, namely, the time amount the tourist must wait for the battery to be charged enough to reach the next station. It also integrates the time budget constraint and quantizes the daily and total tour length. The performance measurement result obtained from a prototype implementation shows that our scheme shows 100 % schedulability until 13 places for the 2-day trip and 17 places for the 3-day trip. In addition, for most cases, the reasonable answers are found in earlier stage of genetic iterations.

As future work, we are planning to apply our scheduler to the actual road network, specifically, in Jeju city, Republic of Korea. The distribution of tour places and terrain effect will give us more hints to improve or adapt the tour-and-charge scheduler proposed in this paper.

References

1. Ipakchi, A., Albuyeh, F.: Grid of the Future. IEEE Power & Energy Magazine, 52–62 (2009)
2. Goebel, C., Callaway, D.: Using ICT-Controlled Plug-in Electric Vehicles to Supply Grid Regulation in California at Different Renewable Integration Levels. IEEE Transactions on Smart Grid 4(2), 729–740 (2013)
3. Lue, A., Colorni, A., Nocerino, R., Paruscio, V.: Green Move: An Innovative Electric Vehicle-Sharing System. Procedia-Social and Behavioral Sciences 48, 2978–2987 (2012)
4. Bessler, S., Grønbæk, J.: Routing EV users towards an Optimal Charging Plan. In: International Battery, Hybrid and Fuel Cell Electric Vehicle Symposium (2012)
5. Ferreira, J., Pereira, P., Filipe, P., Afonso, J.: Recommender System for Drivers of Electric Vehicles. In: Proc. International Conference on Electronic Computer Technology, pp. 244–248 (2011)
6. Vansteenwegen, P., Souffriau, W., Berghe, G., Oudheusden, D.: The City Trip Planner: An Expert System for Tourists. Expert Systems with Applications 38, 6540–6546 (2011)

7. Vansteenwegen, P., Souffriau, W., Berghe, G., Oudheusden, D.: Iterated Local Search for the Team Orienteering Problem with Time Windows. Computers & Operations Research 36, 3281–3290 (2009)
8. Bektas, T.: The Multiple Traveling Salesman Problem: An Overview of Formulations and Solution Procedures. International Journal of Management Science 34, 209–219 (2006)
9. Lian, L., Castelain, E.: A Decomposition Approach to Solve a General Delivery Problem. Engineering Letters 18(1) (2010)
10. Shim, V., Tan, K., Tan, K.: A Hybrid Estimation of Distribution Algorithm for Solving the Multi-Objective Multiple Traveling Salesman Problem. In: IEEE World Congress on Computational Intelligence (2012)
11. Tarantilis, C., Zachariadis, E., Kiranoudis, C.: A Hybrid Metaheuristic Algorithm for the Integrated Vehicle Routing and Three-Dimensional Container-Loading Problem. IEEE Transactions on Intelligent Transportation Systems 10(2), 255–271 (2009)
12. Qin, H., Zhang, W.: Charging Scheduling with Minimal Waiting in a Network of Electric Vehicles and Charging Stations. In: ACM International Workshop on Vehicular Internetworking, pp. 51–60 (2011)
13. Lee, J., Kim, H., Park, G.: Integration of Battery Charging to Tour Schedule Generation for an EV-Based Rent-a-Car Business. In: Tan, Y., Shi, Y., Ji, Z. (eds.) ICSI 2012, Part II. LNCS, vol. 7332, pp. 399–406. Springer, Heidelberg (2012)
14. Lee, J., Park, G.: A Tour Recommendation Service for Electric Vehicles Based on a Hybrid Orienteering Model. In: ACM Symposium on Applied Computing, pp. 1652–1654 (2013)
15. Sivanandam, S., Deepa, S.: Introduction to Genetic Algorithms. Springer (2008)
16. Kim, H., Shin, K.: Scheduling of Battery Charge, Discharge, and Rest. In: IEEE Real-Time Systems Symposium, pp. 13–22 (2009)