

# Performance Analysis of an Orienteering Problem-based Trip Scheduler for Electric Vehicles

Junghoon Lee and Gyung-Leen Park

*Department of Computer Science and Statistics  
Jeju National University  
Republic of Korea  
{jhlee, glpark}@jejunu.ac.kr<sup>\*\*+</sup>*

## **Abstract**

*This paper designs a tour recommendation scheme for electric vehicles and analyzes its performance. Aiming at reducing the time waste brought by long charging time, the recommender not just decides a visiting and charging schedule for the given set of user-selected tour spots but also integrate additional places, which have chargers and provide tour activities. Genetic operators, such as reproduction, selection, and mutation, are tailored to create a combinatory schedule of essential selected and optional recommended spots based on a hybrid orienteering model. Its encoding scheme represents a visiting order by a fixed-length integer-valued vector, allowing the omission of recommendable spots. In addition, the fitness function estimates time waste by tracing the battery charge state based on the inter-destination distance and stay time along a given route. The performance measurement result obtained from a prototype implementation discovers that our recommendation service can reduce the time waste by up to 12 % for the practical range of parameter setting, compared with the legacy tour scheduling method.*

**Keywords:** *electric vehicle, tour schedule, hybrid orienteering model, genetic algorithm, waiting time*

## **1. Introduction**

From the vision of future smart transportation, electric vehicles, or EVs in short, are expected to gradually replace gasoline-powered vehicles for the sake of greenhouse gas reduction and better energy efficiency [1]. Even though EVs have many outstanding advantages in the environmental aspect, they are still too expensive for people to personally own and their charging time is too much long. Moreover, the charging infrastructure is usually under development in most smart grid cities, so the number of available charging stations will not be sufficient for the time being. In addition, even though the modern battery technology has much improved the energy storage capacity, the driving distance is quite short, leading to the *range anxiety* problem. Meanwhile, the information technology, developed in the computer science society, can alleviate this problem by intelligently selecting the time and place to charge an EV, considering a variety of information on

---

\* Gyung-Leen Park is the corresponding author.

+ This research was financially supported by the Ministry of Trade, Industry and Energy (MOTIE), Korea Institute for Advancement of Technology (KIAT) through the Inter-ER Cooperation Projects.

available charging stations, price signal changes, user preference, and generated tour plans [2, 3].

The effect of long charging time gets more serious especially when the tour distance exceeds the driving range of a fully-charged vehicle, as an EV needs to be charged en-route once or more. For residential users, their daily trip distances hardly exceed this driving range, so overnight charging can usually cover the trip length. On the contrary, if an EV visits more than one destination as in rent-a-cars or delivery vehicles, the visiting order affects total driving distance and waiting time along the route [4]. Waiting time is the time interval in which drivers cannot do anything but wait for their EVs to be charged. How to decide the visiting order is a classic computer algorithm belonging to the category of the TSP (Traveling Salesman Problem). For EVs, waiting time is more critical than the driving distance, or tour length, as it takes about 6-7 hours to completely charge an EV with slow chargers, which are still the most commonly used charging method [5]. Thus, it is necessary to accurately estimate the waiting time for a multi-destination tour. Then, a tour schedule engine traverses the search space to find either an optimal or an acceptable schedule [6].

In EV-based tours, given the set of tour spots tourists want to visit, the tour planner creates a route plan according to the user requirement [7]. Basically, a backtracking-based search can find an optimal visit sequence for the given cost criteria [8]. However, it suffers from the extensive response time when the number of destinations gets larger. As contrast, suboptimal search techniques such as genetic algorithms and simulated annealing schemes investigate just a part of the whole search space to meet the time constraint [9]. Here, if some spots can be omitted not to violate the restriction posed on tour length and time, this scheduling is equivalent to the orienteering problem, or selective TSP [10]. For the cost function which calculates the waiting time, the availability of chargers at each tour place must be considered. If a charging facility is available at a tour place, the EV can be charged during the renters take a tour. Battery remaining increases in proportion to the stay time at the place.

The long waiting time makes travelers feel so much inconvenient, if they cannot do anything while their EVs are being charged [11]. Instead, it will be much better to visit some places located near the expected path and have charging facilities, even though the places are not included in the initial user selection. The recommendable tourist places can be found through location-based query to the spatial database engine [12]. While all the places selected by tourists must be included in the final tour plan, the recommendable places don't always have to be included. After all, the route planner for EV-based tours is a combination of the legacy TSP solver for the user-selected places and the orienteering problem solver for recommendable places. In tour and charging schedule generation, we assume only slow chargers. This assumption cannot only fully utilize the current charging infrastructure mainly consisting of slow chargers but also reduce the number of fast charging operations, which may shorten the battery life.

In this regard, this paper designs a tour planner for EVs using computational intelligence and measures its performance. This service decides the visiting order, namely, tour schedule, taking into account stay time information and recommendable spots. It estimates the waiting time for a feasible schedule, represents the tour schedule by the combination of mandatory and optional places, and runs genetic algorithms in traversing the vast search space. It must be mentioned that genetic algorithms are essential to create a schedule within the reasonable time bound. Addition of recommendable places increases the number of destinations in deciding the visiting order, even though only a part of them are included in the final schedule. As they can be omitted from the schedule, the search space gets even larger. The primitive idea was designed in [13] as a short paper. This paper includes more detailed explanation on

the system development and experiment results. Our design can work client-server platforms running on ubiquitous wireless communication networks [14].

## 2. Related Work

In [15], the authors present an intelligent routing system having steps of recommendation, route generation, and final customization. Its routing engine systematically takes into account both tourists' preference and restrictions on available tour time or maximum budget to generate a personalized tour schedule. Particularly, combined with public transportation, the route generation follows the time dependent orienteering problem model [16]. Its main feature is that each distance calculation between two tour places becomes a time dependent shortest path problem, as the waiting time in taking a public bus is deeply dependent on the arrival time at a bus stop. A heuristic scheme is proposed to find a feasible route in real-time by iterative local search and perturbation phases. In this approach, the local search heuristic inserts new visits one by one to the route, while the perturbation phase removes consecutive visits from a route to avoid unnecessary waiting. Finally, a prototype is implemented for the city of San Sebastian, Spain, having about 50 POIs (Point of Interests) and 26 bus lines.

Next, TSPs with profits are a generalization of the legacy TSP [6]. Each vertex is associated with a profit and it is not necessary to visit all vertices. Its tour scheduler pursues maximizing the profit while reducing the travel cost. Those two goals may conflict, so its object function is required to harmonize them. [6] surveys and classifies the solutions for the TSP with profit. Here, different classes of applications, modeling approaches, and exact or heuristic techniques are compared. Most of all, the classification depends on the way the two objectives are addressed. The first class defines an object function combining both of them, mainly adjusting their weights. Second, the object function takes into account only profit maximization while the travel cost works as a constraint, that is, the solutions violating the cost constraint is excluded from the search space. The third class oppositely takes the profit as a constraint and its object function takes into account only on the travel cost.

For the classification described in the previous paragraph, the second class is called the orienteering problem, or interchangeably, the selective TSP. For this problem, [10] has developed a genetic algorithm where a chromosome is a sequence of visited vertices. Here, non-feasible solutions, which violate the cost constraint, are also accepted for better diversity, just with a penalty estimated by the distance from feasibility. In each chromosome, vertices are listed in the visiting order from the start and end points, while the order length is equal to the number of all given vertices, say,  $n$ . For initial population, a list of  $n$  points is generated in random order. A vertex will be removed with the omission probability, making the corresponding entry zero in the list. With this initialization, regular genetic operators such as crossovers and mutations are applied. Specifically, in each offspring, duplicated vertices will be replaced by disappearing ones. Its encoding scheme is comprehensive and robust, so our work partially employs it.

In addition, SiREV is a recommender system for EVs, aiming at timely providing right information to a mobile device carried by a driver in EVs [7]. The intelligence in the information technology minimizes the problem of range anxiety by efficiently planning daily use of an EV. The planning procedure considers diverse information on current vehicle location, public transportation, energy market, and battery remaining. Particularly, SiREV implements a standard adapter module compatible with any battery system of any manufacturer that has been duly specified. By continuously monitoring the battery status, this system recommends when and where to charge an EV. Moreover, its interface makes it possible to issue orders to start, pause, and stop charging when the EV is connected to a

charger. The mobile user interface is provided on smart phones built upon the Android architecture. It also communicates with charging station operators to reserve a specific time slot.

### 3. Tour Recommender Design

#### 3.1. Basic Idea

For a tour spot having chargers, tour time can overlap EV charging, so the tourist can save much time. The charging gain is proportional to the stay time at the place and the average stay time is generally known in tourist information services and statistics. If the battery amount is large enough to reach the next place, no waiting time is involved. However, a tour route can possibly include a subroute which is too long for an EV to reach the next spot without additional charging. This situation is more likely to take place if battery remaining is small when the EV departs a spot. Besides, a tour spot having no charging facility extends the distance between two chargeable tour spots. The EV needs to be charged either somewhere on the path or at a spot even after the tour is over. In this case, the tourists must wait without taking any tour activity, uselessly wasting their time. Instead, the tourist can avoid this time waste by visiting another place having chargers, even though they are not included in the initial selection list. For more details, refer to [13].

There can be more than one spot we can recommend, while the recommender system is required to select their subset considering the remaining battery amount, arrival time estimation at a spot, battery amount needed to reach the next spot, and the like. Sometimes, any addition of recommended spots does not possibly lead to the reduction of waiting time. It's a problem of tremendous time and space complexity. If there are  $n$  selected tour spots and  $m$  recommendable spots,  $2^m$  recommendations exist. For each subset having the cardinality of  $k$  ( $0 \leq k \leq m$ ), we have  $(n + k)!$  visiting sequences. After all, this service essentially employs suboptimal search schemes, tolerating the accuracy loss.

#### 3.2. Encoding Scheme and Object Function

Among various suboptimization techniques, genetic algorithms are known to be an efficient search strategy inspired by principles of natural selection and genetics [17]. To exploit this scheme, each schedule is first encoded into an integer-valued vector called a chromo-some. For our EV-based tour service, this vector, which represents a feasible tour schedule, consists of selected tour spots and recommended ones. The genetic scheduler begins with initial population having a predefined number of schedules. Without any restriction, many invalid schedules would appear in population in TSP style problems, as will be described in Subsection 3.3. Our scheduler accepts only valid schedules. Even though this limitation may narrow the search range and stick to a local minimum, it helps the genetic iteration to converge quickly. Otherwise, the search procedure may spend too much time in processing invalid chromosomes, failing in obtaining a reasonable quality solution.

Selected								Recommended				
2	3	6	1	5	4	0	7	8	-	10	-	12
4	7	0	1	5	6	3	2	-	9	10	-	-
⋮												
5	4	6	3	7	0	1	2	8	-	-	11	12
(a) Step 1												
4	3	10	1	6	-	0	7	8	5	-	2	12
-	7	6	-	5	9	3	2	-	1	0	10	4
⋮												
-	4	11	12	-	5	1	2	8	6	7	3	0
(b) Step 2												

**Figure 1. Main Idea**

Figure 1 shows how to encode a schedule and how to build the initial population. In this figure, there are 8 selected spots and 5 recommendable spots, respectively, thus the final vector length is 13. The selected places numbered from 0 to 7 must appear in each vector, while the recommendable places numbered from 8 to 12 are included in a vector with the given probability. The omitted spots are denoted by -1, which is marked by in Figure 1. Then, the actual visiting sequence can be obtained by removing -1 in the schedule. Hence,  $(A, B, -, C)$  and  $(A, -, B, C)$  are equivalent and regarded as duplicated ones. Step 1 generates a random sequence for the selected set and appends recommendable spots according to the omission probability. Any subset of recommendable spots can be included in this vector, ranging from none of them to all of them. Then, numbers are exchanged in Step 2. This step continues until the completion of the initial population. Duplicated schedules will be removed. To check whether a new vector already exists in the population, the sequence generator removes -1 in the schedule first and compares with the others included in the population.

For each feasible solution, it is necessary to evaluate its fitness. As this paper focuses on the waiting time for EV charging, we need to follow the sequence to find out where and how much charging is required. In our tour schedule model, an EV is fully charged when tourists start their trip, as it is charged overnight in most cases. Additionally, we assume that average stay time is known in priori. Then, the cost function is defined as in [8]. For waiting time formulation at a tour spot,  $V_i$ , let  $B_{in}^i$  denote the distance credit, when the EV arrives at  $V_i$ .  $B_{av}^i$  denotes currently available battery.  $W_i$  is the waiting time at  $V_i$ , and  $B_{out}^i$  is battery remaining on its departure. Then,

$$\begin{aligned}
 B_{av}^i &= \min(B_{max}, B_{in}^i + T(V_i)) \\
 W_i &= -\min(0, B_{av}^i - D(V_i, V_{i+1}))/C_r \\
 B_{out}^i &= \max(0, B_{av}^i - D(V_i, V_{i+1}))
 \end{aligned} \tag{1}$$

,where  $B_{max}$  is the maximum battery capacity,  $T(V_i)$  is the stay time at  $V_i$ , and  $D(V_i, V_{i+1})$  denotes the road network distance between  $V_i$  and  $V_{i+1}$ .

Here,  $B_{in}^i$ ,  $B_{out}^i$ ,  $B_{av}^i$ , and  $B_{max}$  are converted to distance reachable with respective battery remaining.  $W_i$  will be either 0 or  $B_{av}^i - D(V_i, V_{i+1})/C_r$ , where  $C_r$  is the charging ratio, that is, the time amount needed for an EV to get power to drive the unit distance, namely, 1 km.  $W_i$  is positive only when  $B_{av}^i - D(V_i, V_{i+1})$  is larger than  $B_{av}^i$ . Additionally, for a charging station which has no tour activity,  $T(V_i)$  will be 0, so charging at this station makes  $W_i$  get longer.

For more details, refer to [8]. After all, the total waiting time or the final cost,  $W$ , is the sum of all  $W_i$ 's. Namely,

$$W = \sum W_i \quad (2)$$

### 3.3. Genetic Search Scheme Design

From the initial population, each evolutionary step creates a next-generation population consisting of better schedules, or visiting sequences. It mainly selects the best solution in a population by the predefined fitness function and mates them to form the next generation. Out of existing and newly created schedules, only the fittest survive in the population, so the fitness value of the population gets better generation by generation. The genetic loop iterates selection by fitness, reproduction of population, and mutation within a single chromosome. Selection is a method that picks parents by the fitness function. Our implementation employs the Roulette Wheel selection, which gives more chances to chromosomes having better fitness values for mating. The initial population is set randomly as described in Figure 1. Here, the fitness function estimates the waiting time for a schedule while the schedule having small waiting time has a higher fitness value.

Reproduction, or crossover, is the process taking two parents and producing a child with the hope that the child will be a better solution. This operation randomly selects a pair of two crossover points and swaps the substrings from each parent. Reproduction may generate the same gene in a chromosome. It means that a single tour spot is visited more than once, so this schedule is not valid. To cope with such a problem, we implement a modified reproduction. After generating a new chromosome from two parents, the crossover procedure invalidates duplicated elements from the route, identifying the missed ones at the same time. In addition, for recommendable spots, a duplicated spot sometimes has no matched disappearing spot. Then, a new recommendable spot replaces the invalidated one.

Finally, mutation randomly exchanges elements within a chromosome to prevent the search procedure to be trapped in a local minimum. The search procedure usually sets the mandatory rate of mutation. However, our implementation observes that not a few identical chromosomes are generated in a population, when the number of user-selected tour spots is less than 15. Hence, every time a sequence is generated, the procedure checks if it is already included in the population. If so, the procedure attempts to conduct mutation until finding a new chromosome completely different from the existing ones. Here again, just like the creation of the initial population, omitted points must be excluded in investigating if two sequences are equivalent. For mutation, there is no difference between selected and recommendable tour places.

## 4. Performance Measurement

A prototype of the proposed recommendation service is implemented using Visual C++ 6.0 to assess its performance. The experiment makes it run on the platform stuffed with Intel Core2 Duo CPU, 3.0 GB memory, and Windows Vista operating system. In our implementation, the number of chromosomes in each population is set to 64, while the number of iterations is set to 1,000. Such a parameter selection is decided to generate a tour schedule within 1 second. However, the number of iterations can be increased if we integrate a kind of speed-up techniques like fast partition in the population. The distance between each spot distributes exponentially with a given average, which is tunable but fixed to 20.0 km. The stay time also distributes exponentially with the maximum of 3 hours for selected tour spots

and 1 hour for recommendable spots. In generating a new chromosome, the omission probability, namely, the probability that a recommendable spot will be included in the chromosome is set to 0.5. For each parameter setting, 20 sets are generated, and each waiting time is averaged.

The first experiment measures the waiting time according to the number of selected points ranging from 5 to 15. Each selected spot has chargers with the probability of 0.8. Figure 3 plots the waiting time for 4 cases of 0, 5, 10, and 15 recommended spots. In this experiment, we set the locations of recommendable spots randomly. Some of them can hardly contribute to the reduction of waiting time. Figure 2(1) shows that the waiting time inevitably increases according to the increase of the number of selected spots, as the total tour length also gets larger. The curve of no recommendation, represented by a solid line, is the reference waiting time. With more recommendable places, the waiting time is largely reduced. However, in some cases, the increased length of chromosomes leads to a worse schedule, as the genetic loop is more likely to fail to converge. Anyway, with 10 recommended places, the waiting time is reduced by 12 % for 8 selected places. Actually, small reduction of waiting time for the cases of less than 7 destinations leads to significant but just arithmetic improvement by 43 %. This result is a little bit exaggerated.

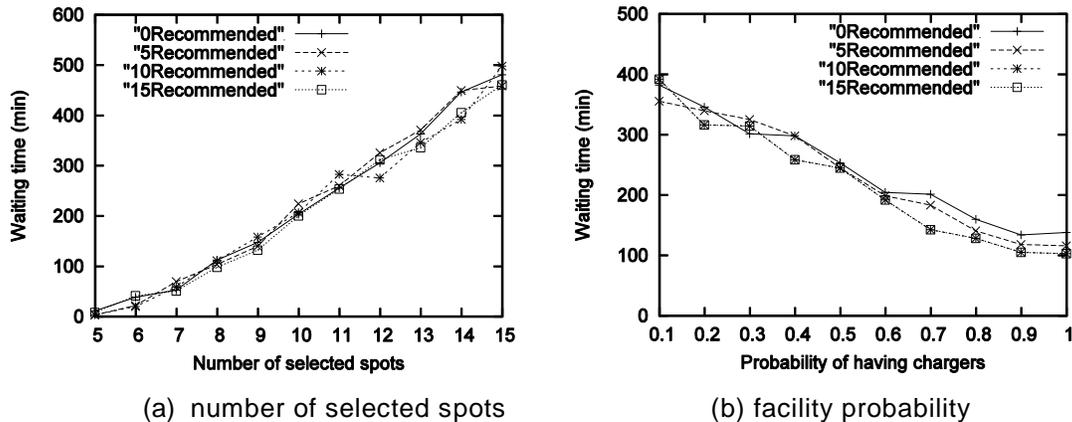


Figure 1. Waiting Time Analysis

The next experiment measures the effect of charging facilities in the selected tour spots to the waiting time, changing it from 0.1 to 1.0. Here, the number of selected spots is set to 9. As shown in Figure 2(b), the waiting time decreases according to the increase of the facility probability. Here again, 4 curves of 0, 5, 10, and 15 recommendations are plotted. With no recommendation, namely, with no integration of additional tour places, the waiting time goes from 354 min for the facility probability of 0.2 down to 125 min for the facility probability of 1.0. When tour spots have charging facilities with the probability of 0.7, the performance improvement reaches 29 %. When this probability gets higher, we can expect further reduction in the waiting time, fully benefitting from the efficiency in tour schedules. Such a long waiting time, coming from insufficient charging facilities, cannot be tolerated by tourists in practice. This parameter setting is chosen just for performance comparison.

Additionally, this section investigates the tour length according to the number of selected places and the availability of chargers, their results being plotted in Figure 3. Parameters are set in the same way as in Figure 2. Addition of some spots inevitably leads to the increase in the total driving distance. However, this effect is usually absorbed by the reduction of the waiting time. As a result, the difference between two tour lengths, one for no

recommendation and the other for more than one recommendation is less than 10 %. Moreover, according to Figure 3(a), tour length is less affected by the availability of charging facilities in tour places. The difference largely remains less than 5 %. This overhead can be sufficiently rewarded by the reduction of the waiting time. In addition, it is also possible to set a maximum bound for the tour length and it will work as a constraint in tour schedule generation.

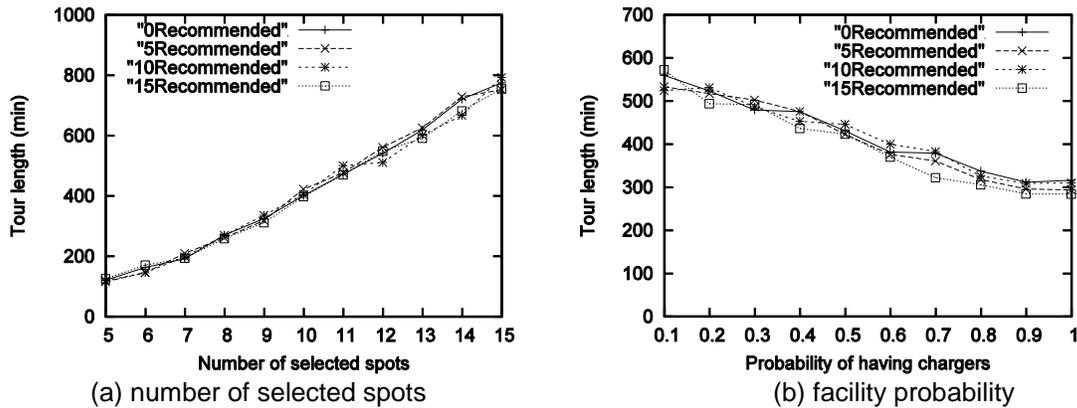


Figure 2. Tour Length Analysis

Finally, the effect of omission probability is plotted in Figure 4. If it is 0, all of recommended spots will be included in the chromosome. As contrast, if it is 1.0, none of them will be included and each chromosome consists of only user-selected spots. Our experiment measures the waiting time, changing the omission probability from 0 to 0.9, while fixing the number of recommended spots to 10 for the cases of 5, 10, and 15 selected spots, respectively. Here, a spot has chargers with the probability of 0.8. How to select the omission probability will be case-sensitive, depending on the number of selected spots, the locations of recommended spots, and the like. Figure 6 indicates that the waiting time is smallest when the omission probability is 0.5. When it is less than or equal to 0.1, the waiting time gets abnormally larger, since every newly generated chromosome has too many elements, which hardly disappear from the population. Excluding those conditions, the omission probability of 0.5 outperforms others by up to 38 %, 9.0 %, and 4.9 % for the cases of 5, 10, and 15 selected spots, respectively.

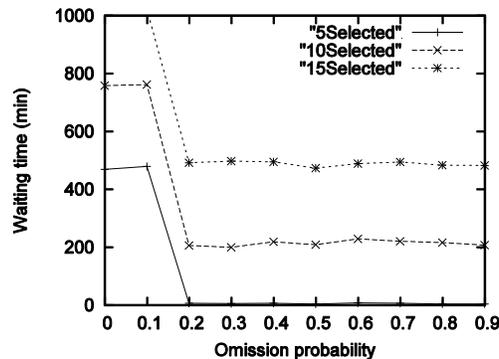


Figure 3. Effect of the Omission Probability

## 5. Conclusions

Smart transportation can achieve energy efficiency by widely deploying electric vehicles. Short driving range and long charging time are the two most critical hindrances for the penetration of EVs in our daily lives, but sophisticated computer algorithms can alleviate this problem. This paper has designed a tour scheduler which not just builds a visiting and charging schedule but also recommends additional tour places. This mechanism makes it possible to reduce time waste induced by battery charging in the EV-based tour. Genetic operations are tailored to create a final schedule combining legacy traveling salesman problem and orienteering problem solvers. Our encoding scheme represents a visiting order by a fixed-length integer-valued vector with some entries empty, while the fitness function estimates waiting time considering the overlap of touring and charging. The performance measurement result obtained from a prototype implementation discovers that our recommendation service can reduce the time waste by 12 % at maximum.

As future work, we are planning to apply the proposed scheme to the real-life tour map in Jeju City, Republic of Korea, which has established a well-known test-bed site for diverse smart grid technologies including electric vehicles. Then, the tour scheduler will integrate a reservation scheme for the charging station belonging to the route plan, also systematically taking into account the availability of fast chargers and battery swap stations.

## References

- [1] Ipakchi and F. Albuyeh, "Grid of the Future," *IEEE Power & Energy Magazine*, (2009), pp. 52–62.
- [2] E. Sortomme, M. Hindi, S. MacPherson, and S. Venkata, "Coordinated Charging of Plug-in Hybrid Electric Vehicles to Minimize Distribution System Losses," *IEEE Transactions on Smart Grid*, (2011), pp. 198–205.
- [3] C. Goebel and D. Callaway, "Using ICT-controlled plug-in electric vehicles to supply grid regulation in California at different renewable integration levels," *IEEE Transactions on Smart Grid*, vol. 4, no. 2, (2013), pp. 729–740.
- [4] S. Bessler and J. Grønbaek, "Routing EV Users towards an Optimal Charging Plan", *International Battery, Hybrid and Fuel Cell Electric Vehicle Symposium*, (2012).
- [5] C. Botsford and A. Szczepanek, "Fast Charging vs. Slow Charging: Pros and cons for the New Age of Electric Vehicles," *International Battery Hybrid Fuel Cell Electric Vehicle Symposium*, (2009).
- [6] D. Feillet, P. Dejax, and M. Gendreau, "Traveling Salesman Problems with Profits", *Transportation Science*, (2005).
- [7] J. Ferreira, P. Pereira, P. Filipe, and J. Afonso, "Recommender System for Drivers of Electric Vehicles", *Proc. International Conference on Electronic Computer Technology*, (2011), pp. 244–248.
- [8] J. Lee, H. Kim, and G. Park, "Integration of Battery Charging to Tour Schedule Generation for an EV-based Rent-a-car Business", *Lecture Notes in Computer Science*, vol. 7332, (2012), pp. 399–406.
- [9] G. Giardini and T. Kalmar-Nagy, "Genetic Algorithm for Combinational Path Planning: The Subtour Problem", *Mathematical Problems in Engineering*, (2011).
- [10] M. Tasgetiren and A. Smith, "A Genetic Algorithm for the Orienteering Problem", *Proc. Congress on Evolutionary Computing*, (2000), pp. 1190–1195.
- [11] H. Qin and W. Zhang, "Charging Scheduling with Minimal Waiting in a Network of Electric Vehicles and Charging Stations", *ACM International Workshop on Vehicular Internetworking*, (2011), pp. 51–60.
- [12] L. Backstrom, J. Kleinberg, R. Kumar, and J. Novak, "Spatial Version in Search Engine Queries", *Proc. International Conference on World Wide Web*, (2008), pp. 357–366.
- [13] J. Lee and G. Park, "A tour Recommendation Service for Electric Vehicles based on a Hybrid Orienteering Model", *ACM Symposium on Applied Computing*, (2013), pp. 1562–1564.
- [14] S. Mehar and G. Remy, "EV-planning: Electric Vehicle Itinerary Planning", *International Conference on Smart Communications in Network Technologies*, (2013).
- [15] P. Vansteenwegen, W. Souffriau, G. Berghe, and D. Oudheusden, "The City Trip Plan- ner: An Expert System for Tourists", *Expert Systems with Applications*, vol. 38, (2011), pp. 6540–6546.
- [16] A. Albiach, J. Sanchis, and D. Soler, "An Asymmetric tsp with Time Windows and with Time-dependent Travel Times and Costs: An Exact Solution Through a Graph Transformation", *European Journal of Operational Research*, vol. 183, (2008), pp. 789–802.

- [17] Y. Katsigiannis, P. Georgilakis, and E. Karapidakis, "Multiobjective Genetic Algorithm Solution to the Optimum Economic and Environmental Performance Problem of Small Autonomous Hybrid Power Systems with Renewables", *IET Renewable Power Generation*, (2010), pp. 404–419.